OpenStack

API Complete Reference

(August 2, 2016)

openstack™

docs.openstack.org

OpenStack API Complete Reference

Copyright © 2010-2016 OpenStack Foundation All rights reserved.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Table of Contents

1. Block Storage API v2 (CURRENT)

Manages volumes and snapshots for use with the Block Storage API, also known as cinder services.

Method	URI	Description
API versions		
GET	`/`	Lists information for all Block Storage API versions.
GET	`/v2`	Shows details for Block Storage API v2.
API extensions (extensions)		
GET	`/v2/{tenant_id}/extensions`	Lists Block Storage API extensions.
Limits (limits)		
GET	`/v2/{tenant_id}/limits`	Shows absolute limits for a tenant.
Volumes (volumes)		
POST	`/v2/{tenant_id}/volumes`	Creates a volume.
GET	`/v2/{tenant_id}/volumes{?sort,limit,marker}`	Lists summary information for all Block Storage volumes that the tenant can access.
GET	`/v2/{tenant_id}/volumes/detail{?sort,limit,marker}`	Lists all Block Storage volumes, with details, that the tenant can access.
GET	`/v2/{tenant_id}/volumes/{volume_id}`	Shows details for a volume.
PUT	`/v2/{tenant_id}/volumes/{volume_id}`	Updates a volume.
DELETE	`/v2/{tenant_id}/volumes/{volume_id}`	Deletes a volume.
POST	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Creates metadata for a volume.
GET	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Shows metadata for a volume.
PUT	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Updates metadata for a volume.
Volume type access (volumes)		
POST	`/v2/{tenant_id}/types/{volume_type}/action`	Adds private volume type access to a project.
POST	`/v2/{tenant_id}/types/{volume_type}/action`	Removes private volume type access from a project.
GET	`/v2/{tenant_id}/types/{volume_type}/os-volume-type-access`	Lists project IDs that have access to private volume type.
Volume actions (volumes, action)		
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Extends the size of a volume to a requested size, in gibibytes (GiB). Specify the `os-extend` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Resets the status, attach status, and migration status for a volume. Specify the `os-reset_status` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Sets the image metadata for a volume. Specify the `os-set_image_metadata` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Removes image metadata, by key, from a volume. Specify the `os-unset_image_metadata` action in the request body and the `key` for the metadata key and value pair that you want to remove.

Method	URI	Description
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Attaches a volume to a server. Specify the os-attach action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Removes a volume from Block Storage management without removing the back-end storage object that is associated with it. Specify the os-unmanage action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Forces a volume to detach. Specify the os-force_detach action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Promotes a replicated volume. Specify the os-promote-replica action in the request body.
POST	/v2/{tenant_id}/volumes/{volume_id}/action	Re-enables replication of a volume. Specify the volume-replica-reenable action in the request body.
Backups (backups)		
POST	/v2/{tenant_id}/backups	Creates a Block Storage backup from a volume.
GET	/v2/{tenant_id}/backups{?sort_key,sort_dir,limit,marker}	Lists Block Storage backups to which the tenant has access.
GET	/v2/{tenant_id}/backups/detail{?sort_key,sort_dir,limit,marker}	Lists Block Storage backups, with details, to which the tenant has access.
GET	/v2/{tenant_id}/backups/{backup_id}	Shows details for a backup.
DELETE	/v2/{tenant_id}/backups/{backup_id}	Deletes a backup.
POST	/v2/{tenant_id}/backups/{backup_id}/restore	Restores a Block Storage backup to an existing or new Block Storage volume.
Backup actions (backups, action)		
POST	/v2/{tenant_id}/backups/{backup_id}/action	Force-deletes a backup. Specify the os-force_delete action in the request body.
Capabilities for storage back ends (capabilities)		
GET	/v2/{tenant_id}/capabilities/{hostname}	Shows capabilities for a storage back end.
Quota sets extension (os-quota-sets)		
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}{?usage}	Shows quotas for a tenant.
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}	Updates quotas for a tenant.
DELETE	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}	Deletes quotas for a tenant so the quotas revert to default values.
GET	/v2/{tenant_id}/os-quota-sets/defaults	Gets default quotas for a tenant.
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Enables an admin user to show quotas for a tenant and user.
PUT	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Updates quotas for a tenant and user.
DELETE	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}	Deletes quotas for a user so that the quotas revert to default values.
GET	/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}	Shows details for quotas for a tenant and user.
Quality of service (QoS) specifications (qos-specs)		
POST	/v2/{tenant_id}/qos-specs	Creates a QoS specification.
GET	/v2/{tenant_id}/qos-specs{?sort_key,sort_dir,limit,marker}	Lists quality of service (QoS) specifications.

Method	URI	Description
GET	/v2/{tenant_id}/qos-specs/{qos_id}	Shows details for a QoS specification.
PUT	/v2/{tenant_id}/qos-specs/{qos_id}	Sets keys in a QoS specification.
DELETE	/v2/{tenant_id}/qos-specs/{qos_id}	Deletes a QoS specification.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associate	Associates a QoS specification with a volume type.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate	Disassociates a QoS specification from a volume type.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all	Disassociates a QoS specification from all associations.
GET	/v2/{tenant_id}/qos-specs/{qos_id}/associations	Lists all associations for a QoS specification.
Volume types (types)		
GET	/v2/{tenant_id}/types{?sort_key,sort_dir,limit,marker}	Lists volume types.
POST	/v2/{tenant_id}/types	Creates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates a volume type.
PUT	/v2/{tenant_id}/types/{volume_type_id}	Updates the extra specifications that are assigned to a volume type.
GET	/v2/{tenant_id}/types/{volume_type_id}	Shows details for a volume type.
DELETE	/v2/{tenant_id}/types/{volume_type_id}	Deletes a volume type.
Volume snapshots (snapshots)		
POST	/v2/{tenant_id}/snapshots	Creates a volume snapshot, which is a point-in-time, complete copy of a volume. You can create a volume from a snapshot.
GET	/v2/{tenant_id}/snapshots{?sort_key,sort_dir,limit,marker}	Lists all Block Storage snapshots, with summary information, that the tenant can access.
GET	/v2/{tenant_id}/snapshots/detail	Lists all Block Storage snapshots, with details, that the tenant can access.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a snapshot.
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a snapshot.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Shows metadata for a snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}/metadata	Updates metadata for a snapshot.
Volume manage extension (os-volume-manage)		
POST	/v2/{tenant_id}/os-volume-manage	Creates a Block Storage volume by using existing storage rather than allocating new storage.
Volume image metadata extension (os-vol-image-meta)		
GET	/v2/{tenant_id}/os-vol-image-meta	Shows image metadata for a volume.
Back-end storage pools		
GET	/v2/{tenant_id}/scheduler-stats/get_pools{?detail}	Lists all back-end storage pools.
Volume transfer		
POST	/v2/{tenant_id}/os-volume-transfer	Creates a volume transfer.

Method	URI	Description
GET	/v2/{tenant_id}/os-volume-transfer	Lists volume transfers.
GET	/v2/{tenant_id}/os-volume-transfer/detail	Lists volume transfers, with details.
GET	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Shows details for a volume transfer.
DELETE	/v2/{tenant_id}/os-volume-transfer/{transfer_id}	Deletes a volume transfer.
POST	/v2/{tenant_id}/os-volume-transfer/{transfer_id}/accept	Accepts a volume transfer.
Consistency groups		
GET	/v2/{tenant_id}/consistencygroups{?sort_key,sort_dir,limit,marker}	Lists consistency groups.
POST	/v2/{tenant_id}/consistencygroups	Creates a consistency group.
GET	/v2/{tenant_id}/consistency-groups/detail{?sort_key,sort_dir,limit,marker}	Lists consistency groups with details.
POST	/v2/{tenant_id}/consistency-groups/create_from_src	Creates a consistency group from source.
GET	/v2/{tenant_id}/consistency-groups/{consistencygroup_id}	Shows details for a consistency group.
POST	/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/delete	Deletes a consistency group.
PUT	/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/update	Updates a consistency group.
Consistency group snapshots		
GET	/v2/{tenant_id}/cgsnapshots	Lists all consistency group snapshots.
POST	/v2/{tenant_id}/cgsnapshots	Creates a consistency group snapshot.
GET	/v2/{tenant_id}/cgsnapshots/detail	Lists all consistency group snapshots with details.
GET	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Shows details for a consistency group snapshot.
DELETE	/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}	Deletes a consistency group snapshot.

1.1. API versions

Method	URI	Description
GET	/	Lists information for all Block Storage API versions.
GET	/v2	Shows details for Block Storage API v2.

1.1.1. List API versions

Method	URI	Description
GET	/	Lists information for all Block Storage API versions.

Normal response codes: 200300

1.1.1.1. Request

This operation does not accept a request body.

1.1.1.2. Response

Example 1.1. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "DEPRECATED",
            "updated": "2014-06-28T12:20:21Z",
            "links": [
                {
                    "href": "http://docs.openstack.org/",
                    "type": "text/html",
                    "rel": "describedby"
                },
                {
                    "href": "http://10.0.2.15:8776/v1/",
                    "rel": "self"
                }
            ],
            "min_version": "",
            "version": "",
            "media-types": [
                {
                    "base": "application/json",
                    "type": "application/vnd.openstack.volume+json;version=1"
                }
            ],
            "id": "v1.0"
        },
        {
            "status": "SUPPORTED",
            "updated": "2014-06-28T12:20:21Z",
            "links": [
                {
                    "href": "http://docs.openstack.org/",
                    "type": "text/html",
                    "rel": "describedby"
                },
                {
                    "href": "http://10.0.2.15:8776/v2/",
                    "rel": "self"
                }
            ],
            "min_version": "",
```

```
            "version": "",
            "media-types": [
                {
                    "base": "application/json",
                    "type": "application/vnd.openstack.volume+json;version=1"
                }
            ],
            "id": "v2.0"
        },
        {
            "status": "CURRENT",
            "updated": "2016-02-08T12:20:21Z",
            "links": [
                {
                    "href": "http://docs.openstack.org/",
                    "type": "text/html",
                    "rel": "describedby"
                },
                {
                    "href": "http://10.0.2.15:8776/v3/",
                    "rel": "self"
                }
            ],
            "min_version": "3.0",
            "version": "{Current_Max_Version}",
            "media-types": [
                {
                    "base": "application/json",
                    "type": "application/vnd.openstack.volume+json;version=1"
                }
            ],
            "id": "v3.0"
        }
    ]
}
```

Example 1.2. List API versions: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<choices xmlns="http://docs.openstack.org/common/api/v1.0" xmlns:atom="http://
www.w3.org/2005/Atom">
    <version status="DEPRECATED" id="v1.0">
        <media-types>
            <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
            <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
        </media-types>
        <atom:link href="http://23.253.248.171:8776/v1/.xml" rel="self" />
    </version>
    <version status="SUPPORTED" id="v2.0">
        <media-types>
            <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
            <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
        </media-types>
        <atom:link href="http://23.253.248.171:8776/v2/.xml" rel="self" />
    </version>
    <version status="CURRENT" id="v3.0">
```

```
        <media-types>
            <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
            <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
        </media-types>
        <atom:link href="http://23.253.248.171:8776/v3/.xml" rel="self" />
    </version>
</choices>
```

This operation does not return a response body.

1.1.2. Show API v2 details

Method	URI	Description
GET	/v2	Shows details for Block Storage API v2.

Normal response codes: 200203

1.1.2.1. Request

This operation does not accept a request body.

1.1.2.2. Response

Example 1.3. Show API v2 details: JSON response

```
{
    "choices": [
        {
            "status": "SUPPORTED",
            "media-types": [
                {
                    "base": "application/xml",
                    "type": "application/vnd.openstack.volume+xml;version=1"
                },
                {
                    "base": "application/json",
                    "type": "application/vnd.openstack.volume+json;version=1"
                }
            ],
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.248.171:8776/v1/v2.json",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "CURRENT",
            "media-types": [
                {
                    "base": "application/xml",
                    "type": "application/vnd.openstack.volume+xml;version=1"
                },
                {
                    "base": "application/json",
                    "type": "application/vnd.openstack.volume+json;version=1"
                }
            ],
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.248.171:8776/v2/v2.json",
                    "rel": "self"
                }
            ]
```

```
        }
    ]
}
```

This table shows the body parameters for the show api v2 details response:

Name	Type	Description
`location`	AnyURI *(Required)*	Full URL to a service or server.

Example 1.4. Show API v2 details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<choices xmlns="http://docs.openstack.org/common/api/v1.0" xmlns:atom="http://
www.w3.org/2005/Atom">
    <version status="SUPPORTED" id="v1.0">
        <media-types>
            <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
            <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
        </media-types>
        <atom:link href="http://23.253.248.171:8776/v1/v2.xml" rel="self" />
    </version>
    <version status="CURRENT" id="v2.0">
        <media-types>
            <media-type base="application/xml" type="application/vnd.
openstack.volume+xml;version=1" />
            <media-type base="application/json" type="application/vnd.
openstack.volume+json;version=1" />
        </media-types>
        <atom:link href="http://23.253.248.171:8776/v2/v2.xml" rel="self" />
    </version>
</choices>
```

This operation does not return a response body.

1.2. API extensions (extensions)

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists Block Storage API extensions.

1.2.1. List API extensions

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists Block Storage API extensions.

Normal response codes: 200300

1.2.1.1. Request

This table shows the URI parameters for the list api extensions request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.2.1.2. Response

Example 1.5. List API extensions: JSON response

```
{
    "extensions": [
        {
            "updated": "2013-04-18T00:00:00+00:00",
            "name": "SchedulerHints",
            "links": [],
            "namespace": "http://docs.openstack.org/block-service/ext/
scheduler-hints/api/v2",
            "alias": "OS-SCH-HNT",
            "description": "Pass arbitrary key/value pairs to the scheduler."
        },
        {
            "updated": "2011-06-29T00:00:00+00:00",
            "name": "Hosts",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/hosts/api/v1.
1",
            "alias": "os-hosts",
            "description": "Admin-only host administration."
        },
        {
            "updated": "2011-11-03T00:00:00+00:00",
            "name": "VolumeTenantAttribute",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/
volume_tenant_attribute/api/v1",
            "alias": "os-vol-tenant-attr",
            "description": "Expose the internal project_id as an attribute of
 a volume."
        },
        {
            "updated": "2011-08-08T00:00:00+00:00",
            "name": "Quotas",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/quotas-sets/
api/v1.1",
            "alias": "os-quota-sets",
```

```
            "description": "Quota management support."
        },
        {
            "updated": "2011-08-24T00:00:00+00:00",
            "name": "TypesManage",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/types-manage/
api/v1",
            "alias": "os-types-manage",
            "description": "Types manage support."
        },
        {
            "updated": "2013-07-10T00:00:00+00:00",
            "name": "VolumeEncryptionMetadata",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/os-volume-
encryption-metadata/api/v1",
            "alias": "os-volume-encryption-metadata",
            "description": "Volume encryption metadata retrieval support."
        },
        {
            "updated": "2012-12-12T00:00:00+00:00",
            "name": "Backups",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/backups/api/
v1",
            "alias": "backups",
            "description": "Backups support."
        },
        {
            "updated": "2013-07-16T00:00:00+00:00",
            "name": "SnapshotActions",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/snapshot-
actions/api/v1.1",
            "alias": "os-snapshot-actions",
            "description": "Enable snapshot manager actions."
        },
        {
            "updated": "2012-05-31T00:00:00+00:00",
            "name": "VolumeActions",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/volume-actions/
api/v1.1",
            "alias": "os-volume-actions",
            "description": "Enable volume actions\n    "
        },
        {
            "updated": "2013-10-03T00:00:00+00:00",
            "name": "UsedLimits",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/used-limits/
api/v1.1",
            "alias": "os-used-limits",
            "description": "Provide data on limited resources that are being
 used."
        },
        {
            "updated": "2012-05-31T00:00:00+00:00",
            "name": "VolumeUnmanage",
```

```
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/volume-
unmanage/api/v1.1",
            "alias": "os-volume-unmanage",
            "description": "Enable volume unmanage operation."
        },
        {
            "updated": "2011-11-03T00:00:00+00:00",
            "name": "VolumeHostAttribute",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/
volume_host_attribute/api/v1",
            "alias": "os-vol-host-attr",
            "description": "Expose host as an attribute of a volume."
        },
        {
            "updated": "2013-07-01T00:00:00+00:00",
            "name": "VolumeTypeEncryption",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/volume-type-
encryption/api/v1",
            "alias": "encryption",
            "description": "Encryption support for volume types."
        },
        {
            "updated": "2013-06-27T00:00:00+00:00",
            "name": "AvailabilityZones",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/os-
availability-zone/api/v1",
            "alias": "os-availability-zone",
            "description": "Describe Availability Zones."
        },
        {
            "updated": "2013-08-02T00:00:00+00:00",
            "name": "Qos_specs_manage",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/qos-specs/api/
v1",
            "alias": "qos-specs",
            "description": "QoS specs support."
        },
        {
            "updated": "2011-08-24T00:00:00+00:00",
            "name": "TypesExtraSpecs",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/types-extra-
specs/api/v1",
            "alias": "os-types-extra-specs",
            "description": "Type extra specs support."
        },
        {
            "updated": "2013-08-08T00:00:00+00:00",
            "name": "VolumeMigStatusAttribute",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/
volume_mig_status_attribute/api/v1",
            "alias": "os-vol-mig-status-attr",
            "description": "Expose migration_status as an attribute of a
 volume."
```

```
        },
        {
            "updated": "2012-08-13T00:00:00+00:00",
            "name": "CreateVolumeExtension",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/image-create/
api/v1",
            "alias": "os-image-create",
            "description": "Allow creating a volume from an image in the
 Create Volume v1 API."
        },
        {
            "updated": "2014-01-10T00:00:00-00:00",
            "name": "ExtendedServices",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/
extended_services/api/v2",
            "alias": "os-extended-services",
            "description": "Extended services support."
        },
        {
            "updated": "2012-06-19T00:00:00+00:00",
            "name": "ExtendedSnapshotAttributes",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/
extended_snapshot_attributes/api/v1",
            "alias": "os-extended-snapshot-attributes",
            "description": "Extended SnapshotAttributes support."
        },
        {
            "updated": "2012-12-07T00:00:00+00:00",
            "name": "VolumeImageMetadata",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/
volume_image_metadata/api/v1",
            "alias": "os-vol-image-meta",
            "description": "Show image metadata associated with the volume."
        },
        {
            "updated": "2012-03-12T00:00:00+00:00",
            "name": "QuotaClasses",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/quota-classes-
sets/api/v1.1",
            "alias": "os-quota-class-sets",
            "description": "Quota classes management support."
        },
        {
            "updated": "2013-05-29T00:00:00+00:00",
            "name": "VolumeTransfer",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/volume-
transfer/api/v1.1",
            "alias": "os-volume-transfer",
            "description": "Volume transfer management support."
        },
        {
            "updated": "2014-02-10T00:00:00+00:00",
            "name": "VolumeManage",
            "links": [],
```

```
            "namespace": "http://docs.openstack.org/volume/ext/os-volume-
manage/api/v1",
            "alias": "os-volume-manage",
            "description": "Allows existing backend storage to be 'managed' by
 Cinder."
        },
        {
            "updated": "2012-08-25T00:00:00+00:00",
            "name": "AdminActions",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/admin-actions/
api/v1.1",
            "alias": "os-admin-actions",
            "description": "Enable admin actions."
        },
        {
            "updated": "2012-10-28T00:00:00-00:00",
            "name": "Services",
            "links": [],
            "namespace": "http://docs.openstack.org/volume/ext/services/api/
v2",
            "alias": "os-services",
            "description": "Services support."
        }
    ]
}
```

This table shows the body parameters for the list api extensions response:

Name	Type	Description
`name`	String *(Required)*	The name of the extension. For example, "Fox In Socks."
`description`	String *(Required)*	The extension description.
`namespace`	String *(Required)*	Link associated to the extension.
`alias`	String *(Required)*	The alias for the extension. For example, "FOXNSOX", "os-availability-zone", "os-extended-quotas", "os-share-unmanage" or "os-used-limits."
`links`	List *(Required)*	List of links related to the extension.
`updated`	DateTime *(Required)*	The date and time stamp when the extension was last updated.

Example 1.6. List API extensions: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<extensions xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/common/api/v1.0">
    <extension alias="OS-SCH-HNT" updated="2013-04-18T00:00:00+00:00"
        namespace="http://docs.openstack.org/block-service/ext/scheduler-
hints/api/v2"
        name="SchedulerHints">
        <description>Pass arbitrary key/value pairs to the
            scheduler.</description>
```

```
    </extension>
    <extension alias="os-hosts" updated="2011-06-29T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/hosts/api/v1.1"
        name="Hosts">
        <description>Admin-only host administration.</description>
    </extension>
    <extension alias="os-vol-tenant-attr"
        updated="2011-11-03T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/
volume_tenant_attribute/api/v1"
        name="VolumeTenantAttribute">
        <description>Expose the internal project_id as an attribute of
            a volume.</description>
    </extension>
    <extension alias="os-quota-sets"
        updated="2011-08-08T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/quotas-sets/api/v1.1"
        name="Quotas">
        <description>Quota management support.</description>
    </extension>
    <extension alias="os-types-manage"
        updated="2011-08-24T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/types-manage/api/v1"
        name="TypesManage">
        <description>Types manage support.</description>
    </extension>
    <extension alias="os-volume-encryption-metadata"
        updated="2013-07-10T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/os-volume-encryption-
metadata/api/v1"
        name="VolumeEncryptionMetadata">
        <description>Volume encryption metadata retrieval
            support.</description>
    </extension>
    <extension alias="backups" updated="2012-12-12T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/backups/api/v1"
        name="Backups">
        <description>Backups support.</description>
    </extension>
    <extension alias="os-snapshot-actions"
        updated="2013-07-16T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/snapshot-actions/api/
v1.1"
        name="SnapshotActions">
        <description>Enable snapshot manager actions.</description>
    </extension>
    <extension alias="os-volume-actions"
        updated="2012-05-31T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume-actions/api/v1.
1"
        name="VolumeActions">
        <description>Enable volume actions </description>
    </extension>
    <extension alias="os-used-limits"
        updated="2013-10-03T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/used-limits/api/v1.1"
        name="UsedLimits">
        <description>Provide data on limited resources that are being
            used.</description>
    </extension>
```

```
    <extension alias="os-volume-unmanage"
        updated="2012-05-31T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume-unmanage/api/
v1.1"
        name="VolumeUnmanage">
        <description>Enable volume unmanage operation.</description>
    </extension>
    <extension alias="os-vol-host-attr"
        updated="2011-11-03T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume_host_attribute/
api/v1"
        name="VolumeHostAttribute">
        <description>Expose host as an attribute of a
            volume.</description>
    </extension>
    <extension alias="encryption" updated="2013-07-01T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume-type-
encryption/api/v1"
        name="VolumeTypeEncryption">
        <description>Encryption support for volume
            types.</description>
    </extension>
    <extension alias="os-availability-zone"
        updated="2013-06-27T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/os-availability-zone/
api/v1"
        name="AvailabilityZones">
        <description>Describe Availability Zones.</description>
    </extension>
    <extension alias="qos-specs" updated="2013-08-02T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/qos-specs/api/v1"
        name="Qos_specs_manage">
        <description>QoS specs support.</description>
    </extension>
    <extension alias="os-types-extra-specs"
        updated="2011-08-24T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/types-extra-specs/api/
v1"
        name="TypesExtraSpecs">
        <description>Type extra specs support.</description>
    </extension>
    <extension alias="os-vol-mig-status-attr"
        updated="2013-08-08T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/
volume_mig_status_attribute/api/v1"
        name="VolumeMigStatusAttribute">
        <description>Expose migration_status as an attribute of a
            volume.</description>
    </extension>
    <extension alias="os-image-create"
        updated="2012-08-13T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/image-create/api/v1"
        name="CreateVolumeExtension">
        <description>Allow creating a volume from an image in the
            Create Volume v1 API.</description>
    </extension>
    <extension alias="os-extended-services"
        updated="2014-01-10T00:00:00-00:00"
        namespace="http://docs.openstack.org/volume/ext/extended_services/api/
v2"
```

```
        name="ExtendedServices">
        <description>Extended services support.</description>
    </extension>
    <extension alias="os-extended-snapshot-attributes"
        updated="2012-06-19T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/
extended_snapshot_attributes/api/v1"
        name="ExtendedSnapshotAttributes">
        <description>Extended SnapshotAttributes
            support.</description>
    </extension>
    <extension alias="os-vol-image-meta"
        updated="2012-12-07T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume_image_metadata/
api/v1"
        name="VolumeImageMetadata">
        <description>Show image metadata associated with the
            volume.</description>
    </extension>
    <extension alias="os-quota-class-sets"
        updated="2012-03-12T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/quota-classes-sets/
api/v1.1"
        name="QuotaClasses">
        <description>Quota classes management support.</description>
    </extension>
    <extension alias="os-volume-transfer"
        updated="2013-05-29T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/volume-transfer/api/
v1.1"
        name="VolumeTransfer">
        <description>Volume transfer management support.</description>
    </extension>
    <extension alias="os-volume-manage"
        updated="2014-02-10T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/os-volume-manage/api/
v1"
        name="VolumeManage">
        <description>Allows existing back end storage to be 'managed'
            by cinder.</description>
    </extension>
    <extension alias="os-admin-actions"
        updated="2012-08-25T00:00:00+00:00"
        namespace="http://docs.openstack.org/volume/ext/admin-actions/api/v1.
1"
        name="AdminActions">
        <description>Enable admin actions.</description>
    </extension>
    <extension alias="os-services" updated="2012-10-28T00:00:00-00:00"
        namespace="http://docs.openstack.org/volume/ext/services/api/v2"
        name="Services">
        <description>Services support.</description>
    </extension>
</extensions>
```

This operation does not return a response body.

1.3. Limits (limits)

Shows absolute limits for a tenant.

An absolute limit value of `-1` indicates that the absolute limit for the item is infinite.

Method	URI	Description
GET	`/v2/{tenant_id}/limits`	Shows absolute limits for a tenant.

1.3.1. Show absolute limits

Method	URI	Description
GET	`/v2/{tenant_id}/limits`	Shows absolute limits for a tenant.

An absolute limit value of `-1` indicates that the absolute limit for the item is infinite.

Normal response codes: 200203

1.3.1.1. Request

This table shows the URI parameters for the show absolute limits request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.3.1.2. Response

Example 1.7. Show absolute limits: JSON response

```
{
    "limits": {
        "rate": [],
        "absolute": {
            "totalSnapshotsUsed": 0,
            "maxTotalBackups": 10,
            "maxTotalVolumeGigabytes": 1000,
            "maxTotalSnapshots": 10,
            "maxTotalBackupGigabytes": 1000,
            "totalBackupGigabytesUsed": 0,
            "maxTotalVolumes": 10,
            "totalVolumesUsed": 0,
            "totalBackupsUsed": 0,
            "totalGigabytesUsed": 0
        }
    }
}
```

Example 1.8. Show absolute limits: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<limits xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/common/api/v1.0">
    <rates/>
    <absolute>
        <limit name="totalSnapshotsUsed" value="0"/>
        <limit name="maxTotalBackups" value="10"/>
        <limit name="maxTotalVolumeGigabytes" value="1000"/>
        <limit name="maxTotalSnapshots" value="10"/>
        <limit name="maxTotalBackupGigabytes" value="1000"/>
        <limit name="totalBackupGigabytesUsed" value="0"/>
        <limit name="maxTotalVolumes" value="10"/>
        <limit name="totalVolumesUsed" value="0"/>
```

```
        <limit name="totalBackupsUsed" value="0"/>
        <limit name="totalGigabytesUsed" value="0"/>
    </absolute>
</limits>
```

This operation does not return a response body.

1.4. Volumes (volumes)

A volume is a detachable block storage device similar to a USB hard drive. You can attach a volume to one instance at a time.

The `snapshot_id` and `source_volid` parameters specify the ID of the snapshot or volume from which this volume originates. If the volume was not created from a snapshot or source volume, these values are null.

When you create, list, update, or delete volumes, the possible status values are:

Table 1.1. Volume statuses

Status	Description
creating	The volume is being created.
available	The volume is ready to attach to an instance.
attaching	The volume is attaching to an instance.
in-use	The volume is attached to an instance.
deleting	The volume is being deleted.
error	A volume creation error occurred.
error_deleting	A volume deletion error occurred.
backing-up	The volume is being backed up.
restoring-backup	A backup is being restored to the volume.
error_restoring	A backup restoration error occurred.
error_extending	An error occurred while attempting to extend a volume.

Method	URI	Description
POST	`/v2/{tenant_id}/volumes`	Creates a volume.
GET	`/v2/{tenant_id}/volumes{?sort,limit,marker}`	Lists summary information for all Block Storage volumes that the tenant can access.
GET	`/v2/{tenant_id}/volumes/detail{?sort,limit,marker}`	Lists all Block Storage volumes, with details, that the tenant can access.
GET	`/v2/{tenant_id}/volumes/{volume_id}`	Shows details for a volume.
PUT	`/v2/{tenant_id}/volumes/{volume_id}`	Updates a volume.
DELETE	`/v2/{tenant_id}/volumes/{volume_id}`	Deletes a volume.
POST	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Creates metadata for a volume.
GET	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Shows metadata for a volume.
PUT	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Updates metadata for a volume.

1.4.1. Create volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes`	Creates a volume.

To create a bootable volume, include the UUID of the image from which you want to create the volume in the `imageRef` attribute in the request body.

Preconditions

- You must have enough volume storage quota remaining to create a volume of size requested.

Asynchronous Postconditions

- With correct permissions, you can see the volume status as `available` through API calls.
- With correct access, you can see the created volume in the storage system that OpenStack Block Storage manages.

Troubleshooting

- If volume status remains `creating` or shows another error status, the request failed. Ensure you meet the preconditions then investigate the storage back end.
- Volume is not created in the storage system that OpenStack Block Storage manages.
- The storage node needs enough free storage space to match the size of the volume creation request.

Normal response codes: 202

1.4.1.1. Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.9. Create volume: JSON request

```
{
    "volume": {
        "size": 10,
        "availability_zone": null,
        "source_volid": null,
        "description": null,
        "multiattach ": false,
        "snapshot_id": null,
        "name": null,
        "imageRef": null,
        "volume_type": null,
        "metadata": {},
        "source_replica": null,
        "consistencygroup_id": null
    }
}
```

This table shows the body parameters for the create volume request:

Name	Type	Description
volume	Dict *(Required)*	A volume object.
size	Int *(Required)*	The size of the volume, in gibibytes (GiB).
availability_zone	String *(Optional)*	The availability zone.
source_volid	UUID *(Optional)*	The UUID of the source volume. The API creates a new volume with the same size as the source volume.
description	String *(Optional)*	The volume description.
multiattach	Boolean *(Optional)*	To enable this volume to attach to more than one server, set this value to true. Default is false.
snapshot_id	UUID *(Optional)*	To create a volume from an existing snapshot, specify the UUID of the volume snapshot. The volume is created in same availability zone and with same size as the snapshot.
name	String *(Optional)*	The volume name.
imageRef	UUID *(Optional)*	The UUID of the image from which you want to create the volume. Required to create a bootable volume.
volume_type	String *(Optional)*	The volume type. To create an environment with multiple-storage back ends, you must specify a volume type. Block Storage volume back ends are spawned as children to cinder-volume, and they are keyed from a unique queue. They are named cinder- volume.HOST.BACKEND. For example, cinder-volume.ubuntu.lvmdriver. When a volume is created, the scheduler chooses an appropriate back end to handle the request based on the volume type. Default is None. For information about how to use volume types to create multiple-storage back ends, see Configure multiple-storage back ends.
metadata	Dict *(Optional)*	One or more metadata key and value pairs that are associated with the volume.
source_replica	UUID *(Optional)*	The UUID of the primary volume to clone.
consistencygroup_id	UUID *(Optional)*	The UUID of the consistency group.
scheduler_hints	Dict *(Optional)*	The dictionary of data to send to the scheduler.

Example 1.10. Create volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    name="vol-001" description="Another volume."
```

```
    size="2"/>
```

This operation does not accept a request body.

1.4.1.2. Response

Example 1.11. Create volume: JSON response

```
{
    "volume": {
        "status": "creating",
        "migration_status": null,
        "user_id": "0eea4eabcf184061a3b6db1e0daaf010",
        "attachments": [],
        "links": [
            {
                "href": "http://23.253.248.171:8776/v2/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
                "rel": "self"
            },
            {
                "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "nova",
        "bootable": "false",
        "encrypted": false,
        "created_at": "2015-11-29T03:01:44.000000",
        "description": null,
        "updated_at": null,
        "volume_type": "lvmdriver-1",
        "name": "test-volume-attachments",
        "replication_status": "disabled",
        "consistencygroup_id": null,
        "source_volid": null,
        "snapshot_id": null,
        "multiattach": false,
        "metadata": {},
        "id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
        "size": 2
    }
}
```

This table shows the body parameters for the create volume response:

Name	Type	Description
`volume`	Dict *(Required)*	A `volume` object.
`status`	String *(Required)*	The volume status.
`migration_status`	String *(Required)*	The volume migration status.
`user_id`	UUID	The UUID of the user.

Name	Type	Description
	(Required)	
`attachments`	List *(Required)*	Instance attachment information. If this volume is attached to a server instance, the attachments list includes the UUID of the attached server, an attachment UUID, the name of the attached host, if any, the volume UUID, the device, and the device UUID. Otherwise, this list is empty.
`links`	List *(Required)*	The volume links.
`availability_zone`	String *(Required)*	The availability zone.
`bootable`	Boolean *(Required)*	Enables or disables the bootable attribute. You can boot an instance from a bootable volume.
`encrypted`	Boolean *(Required)*	If true, this volume is encrypted.
`created_at`	DateTime *(Required)*	The date and time when the resource was created. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` For example, `2015-08-27T09:49:58-05:00`. The `±hh:mm` value, if included, is the time zone as an offset from UTC.
`description`	String *(Required)*	The volume description.
`updated_at`	DateTime *(Required)*	The date and time when the resource was updated. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` For example, `2015-08-27T09:49:58-05:00`. The `±hh:mm` value, if included, is the time zone as an offset from UTC. In the previous example, the offset value is `-05:00`. If the `updated_at` date and time stamp is not set, its value is `null`.
`volume_type`	String *(Required)*	The volume type. In an environment with multiple-storage back ends, the scheduler determines where to send the volume based on the volume type. For information about how to use volume types to create multiple-storage back ends, see Configure multiple-storage back ends.
`name`	String *(Required)*	The volume name.
`replication_status`	String *(Required)*	The volume replication status.
`consistencygroup_id`	UUID *(Required)*	The UUID of the consistency group.
`source_volid`	UUID *(Required)*	The UUID of the source volume.

Name	Type	Description
`snapshot_id`	UUID *(Required)*	The UUID of the source volume snapshot. The API creates a new volume snapshot with the same size as the source volume snapshot.
`multiattach`	Boolean *(Required)*	If true, this volume can attach to more than one instance.
`metadata`	Dict *(Required)*	One or more metadata key and value pairs that are associated with the volume.
`id`	UUID *(Required)*	The UUID of the volume.
`size`	Int *(Required)*	The size of the volume, in gibibytes (GiB).

Example 1.12. Create volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/volume/api/v1" status="creating"
    name="vol-001" availability_zone="nova" bootable="false"
    created_at="2014-02-21 20:18:33.122452"
    description="Another volume." volume_type="None"
    snapshot_id="None" source_volid="None"
    id="83960a54-8dad-4fd8-bc41-33c71e098e04" size="2">
    <attachments/>
    <metadata/>
</volume>
```

This operation does not return a response body.

1.4.2. List volumes

Method	URI	Description
GET	`/v2/{tenant_id}/volumes{?sort,limit,marker}`	Lists summary information for all Block Storage volumes that the tenant can access.

Normal response codes: 200

1.4.2.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list volumes request:

Name	Type	Description
`sort`	String *(Optional)*	Comma-separated list of sort keys and optional sort directions in the form of <key>[:<direction>]. A valid direction is `asc` (ascending) or `desc` (descending).
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.4.2.2. Response

Example 1.13. List volumes: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumes xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
    <volume name="vol-004" id="45baf976-c20a-4894-a7c3-c94b7376bf55">
        <attachments/>
        <metadata/>
    </volume>
    <volume name="vol-003" id="5aa119a8-d25b-45a7-8d1b-88e127885635">
        <attachments/>
        <metadata/>
    </volume>
</volumes>
```

Example 1.14. List volumes: JSON response

```
{
    "volumes": [
        {
```

```
            "id": "45baf976-c20a-4894-a7c3-c94b7376bf55",
            "links": [
                {
                    "href": "http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
                    "rel": "bookmark"
                }
            ],
            "name": "vol-004"
        },
        {
            "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
            "links": [
                {
                    "href": "http://localhost:8776/
v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                    "rel": "bookmark"
                }
            ],
            "name": "vol-003"
        }
    ]
}
```

This operation does not return a response body.

1.4.3. List volumes with details

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/detail{?sort,limit,marker}`	Lists all Block Storage volumes, with details, that the tenant can access.

Normal response codes: 200

1.4.3.1. Request

This table shows the URI parameters for the list volumes with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list volumes with details request:

Name	Type	Description
`sort`	String *(Optional)*	Comma-separated list of sort keys and optional sort directions in the form of <key>[:<direction>]. A valid direction is `asc` (ascending) or `desc` (descending).
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.4.3.2. Response

Example 1.15. List volumes with details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumes
    xmlns:os-vol-image-meta="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Image_Metadata.html"
    xmlns:os-vol-tenant-attr="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Tenant_Attribute.html"
    xmlns:os-vol-host-attr="http://docs.openstack.org/openstack-block-storage/
2.0/content/Volume_Host_Attribute.html"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
    <volume status="available" name="vol-004" availability_zone="nova"
        created_at="2013-02-25 06:36:28" description="Another volume."
        volume_type="None" source_volid="None" snapshot_id="None"
        id="45baf976-c20a-4894-a7c3-c94b7376bf55" size="1"
        os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
        os-vol-host-attr:host="ip-10-168-107-25">
        <attachments/>
        <metadata>
            <meta key="contents">junk</meta>
        </metadata>
    </volume>
```

```
    <volume status="available" name="vol-003" availability_zone="nova"
        created_at="2013-02-25 02:40:21"
        description="This is yet, another volume." volume_type="None"
        source_volid="None" snapshot_id="None"
        id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
        os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
        os-vol-host-attr:host="ip-10-168-107-25">
        <attachments/>
        <metadata>
            <meta key="contents">not junk</meta>
        </metadata>
    </volume>
</volumes>
```

Example 1.16. List volumes with details: JSON response

```
{
    "volumes": [
        {
            "migration_status": null,
            "attachments": [
                {
                    "server_id": "f4fda93b-06e0-4743-8117-bc8bcecd651b",
                    "attachment_id": "3b4db356-253d-4fab-bfa0-e3626c0b8405",
                    "host_name": null,
                    "volume_id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
                    "device": "/dev/vdb",
                    "id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38"
                }
            ],
            "links": [
                {
                    "href": "http://23.253.248.171:8776/v2/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
                    "rel": "self"
                },
                {
                    "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/6edbc2f4-1507-44f8-ac0d-
eed1d2608d38",
                    "rel": "bookmark"
                }
            ],
            "availability_zone": "nova",
            "os-vol-host-attr:host": "difleming@lvmdriver-1#lvmdriver-1",
            "encrypted": false,
            "os-volume-replication:extended_status": null,
            "replication_status": "disabled",
            "snapshot_id": null,
            "id": "6edbc2f4-1507-44f8-ac0d-eed1d2608d38",
            "size": 2,
            "user_id": "32779452fcd34ae1a53a797ac8a1e064",
            "os-vol-tenant-attr:tenant_id":
 "bab7d5c60cd041a0a36f7c4b6e1dd978",
            "os-vol-mig-status-attr:migstat": null,
            "metadata": {
                "readonly": false,
                "attached_mode": "rw"
            },
```

```
            "status": "in-use",
            "description": null,
            "multiattach": true,
            "os-volume-replication:driver_data": null,
            "source_volid": null,
            "consistencygroup_id": null,
            "os-vol-mig-status-attr:name_id": null,
            "name": "test-volume-attachments",
            "bootable": "false",
            "created_at": "2015-11-29T03:01:44.000000",
            "volume_type": "lvmdriver-1"
        },
        {
            "migration_status": null,
            "attachments": [],
            "links": [
                {
                    "href": "http://23.253.248.171:8776/
v2/bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/173f7b48-
c4c1-4e70-9acc-086b39073506",
                    "rel": "self"
                },
                {
                    "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/volumes/173f7b48-
c4c1-4e70-9acc-086b39073506",
                    "rel": "bookmark"
                }
            ],
            "availability_zone": "nova",
            "os-vol-host-attr:host": "difleming@lvmdriver-1#lvmdriver-1",
            "encrypted": false,
            "os-volume-replication:extended_status": null,
            "replication_status": "disabled",
            "snapshot_id": null,
            "id": "173f7b48-c4c1-4e70-9acc-086b39073506",
            "size": 1,
            "user_id": "32779452fcd34ae1a53a797ac8a1e064",
            "os-vol-tenant-attr:tenant_id":
 "bab7d5c60cd041a0a36f7c4b6e1dd978",
            "os-vol-mig-status-attr:migstat": null,
            "metadata": {},
            "status": "available",
            "volume_image_metadata": {
                "kernel_id": "8a55f5f1-78f7-4477-8168-977d8519342c",
                "checksum": "eb9139e4942121f22bbc2afc0400b2a4",
                "min_ram": "0",
                "ramdisk_id": "5f6bdf8a-92db-4988-865b-60bdd808d9ef",
                "disk_format": "ami",
                "image_name": "cirros-0.3.4-x86_64-uec",
                "image_id": "b48c53e1-9a96-4a5a-a630-2e74ec54ddcc",
                "container_format": "ami",
                "min_disk": "0",
                "size": "25165824"
            },
            "description": "",
            "multiattach": false,
            "os-volume-replication:driver_data": null,
            "source_volid": null,
            "consistencygroup_id": null,
```

```
            "os-vol-mig-status-attr:name_id": null,
            "name": "test-volume",
            "bootable": "true",
            "created_at": "2015-11-29T02:25:18.000000",
            "volume_type": "lvmdriver-1"
        }
    ]
}
```

This operation does not return a response body.

1.4.4. Show volume details

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/{volume_id}`	Shows details for a volume.

Preconditions

- The volume must exist.

Normal response codes: 200

1.4.4.1. Request

This table shows the URI parameters for the show volume details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

This operation does not accept a request body.

1.4.4.2. Response

Example 1.17. Show volume details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume
    xmlns:os-vol-image-meta="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Image_Metadata.html"
    xmlns:os-vol-tenant-attr="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Tenant_Attribute.html"
    xmlns:os-vol-host-attr="http://docs.openstack.org/openstack-block-storage/
2.0/content/Volume_Host_Attribute.html"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content"
    status="available" name="vol-003" availability_zone="nova"
    bootable="false" created_at="2013-02-25 02:40:21"
    description="This is yet, another volume." volume_type="None"
    source_volid="None" snapshot_id="None"
    id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
    os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
    os-vol-host-attr:host="ip-10-168-107-25">
    <attachments/>
    <metadata>
        <meta key="contents">not junk</meta>
    </metadata>
</volume>
```

Example 1.18. Show volume details: JSON response

```
{
    "volume": {
        "status": "available",
```

```
        "attachments": [],
        "links": [
            {
                "href": "http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                "rel": "self"
            },
            {
                "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "nova",
        "bootable": "false",
        "os-vol-host-attr:host": "ip-10-168-107-25",
        "source_volid": null,
        "snapshot_id": null,
        "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "description": "Super volume.",
        "name": "vol-002",
        "created_at": "2013-02-25T02:40:21.000000",
        "volume_type": "None",
        "os-vol-tenant-attr:tenant_id": "0c2eba2c5af04d3f9e9d0d410b371fde",
        "size": 1,
        "os-volume-replication:driver_data": null,
        "os-volume-replication:extended_status": null,
        "metadata": {
            "contents": "not junk"
        }
    }
}
```


This operation does not return a response body.

1.4.5. Update volume

Method	URI	Description
PUT	`/v2/{tenant_id}/volumes/{volume_id}`	Updates a volume.

Normal response codes: 200

1.4.5.1. Request

This table shows the URI parameters for the update volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

This table shows the body parameters for the update volume request:

Name	Type	Description
`volume`	Dict *(Required)*	A `volume` object.
`name`	String *(Optional)*	The volume name.
`description`	String *(Optional)*	The volume description.
`metadata`	Dict *(Optional)*	One or more metadata key and value pairs that are associated with the volume.

Example 1.19. Update volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    name="vol-003" description="This is yet, another volume."/>
```

Example 1.20. Update volume: JSON request

```
{
    "volume": {
        "name": "vol-003",
        "description": "This is yet, another volume."
    }
}
```

This operation does not accept a request body.

1.4.5.2. Response

Example 1.21. Update volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
```

```
<volume xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content"
    status="available" name="vol-003" availability_zone="nova"
    created_at="2013-02-25 02:40:21"
    description="This is yet, another volume." volume_type="None"
    source_volid="None" snapshot_id="None"
    id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1">
    <attachments/>
    <metadata>
        <meta key="contents">not junk</meta>
    </metadata>
</volume>
```

Example 1.22. Update volume: JSON response

```
{
    "volume": {
        "status": "available",
        "migration_status": null,
        "user_id": "0eea4eabcf184061a3b6db1e0daaf010",
        "attachments": [],
        "links": [
            {
                "href": "http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                "rel": "self"
            },
            {
                "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "nova",
        "bootable": "false",
        "encrypted": false,
        "created_at": "2015-11-29T03:01:44.000000",
        "description": "This is yet, another volume.",
        "updated_at": null,
        "volume_type": "lvmdriver-1",
        "name": "vol-003",
        "replication_status": "disabled",
        "consistencygroup_id": null,
        "source_volid": null,
        "snapshot_id": null,
        "multiattach": false,
        "metadata": {
            "contents": "not junk"
        },
        "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "size": 1
    }
}
```

This operation does not return a response body.

1.4.6. Delete volume

Method	URI	Description
DELETE	`/v2/{tenant_id}/volumes/{volume_id}`	Deletes a volume.

Preconditions

- Volume status must be `available`, `in-use`, `error`, or `error_restoring`.
- You cannot already have a snapshot of the volume.
- You cannot delete a volume that is in a migration.

Asynchronous Postconditions

- The volume is deleted in volume index.
- The volume managed by OpenStack Block Storage is deleted in storage node.

Troubleshooting

- If volume status remains in `deleting` or becomes `error_deleting` the request failed. Ensure you meet the preconditions then investigate the storage back end.
- The volume managed by OpenStack Block Storage is not deleted from the storage system.

Normal response codes: 202

1.4.6.1. Request

This table shows the URI parameters for the delete volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

This operation does not accept a request body.

1.4.7. Create volume metadata

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Creates metadata for a volume.

Normal response codes: 202

1.4.7.1. Request

This table shows the URI parameters for the create volume metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.23. Create volume metadata: JSON request

```
{
    "metadata": {
        "name": "metadata0"
    }
}
```

1.4.7.2. Response

Example 1.24. Create volume metadata: JSON response

```
{
    "metadata": {
        "name": "metadata0"
    }
}
```

1.4.8. Show volume metadata

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Shows metadata for a volume.

Normal response codes: 200

1.4.8.1. Request

This table shows the URI parameters for the show volume metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

This operation does not accept a request body.

1.4.8.2. Response

Example 1.25. Show volume metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1"/>
```

Example 1.26. Show volume metadata: JSON response

```
{
    "metadata": {}
}
```

This operation does not return a response body.

1.4.9. Update volume metadata

Method	URI	Description
PUT	`/v2/{tenant_id}/volumes/{volume_id}/metadata`	Updates metadata for a volume.

Replaces metadata items that match keys. Does not modify items that are not in the request.

Normal response codes: 200

1.4.9.1. Request

This table shows the URI parameters for the update volume metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.27. Update volume metadata: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata>
    <meta key="key">v2</meta>
</metadata>
```

Example 1.28. Update volume metadata: JSON request

```
{
    "metadata": {
        "name": "metadata1"
    }
}
```

This operation does not accept a request body.

1.4.9.2. Response

Example 1.29. Update volume metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="key">v2</meta>
</metadata>
```

Example 1.30. Update volume metadata: JSON response

```
{
    "metadata": {
        "name": "metadata1"
    }
}
```

This operation does not return a response body.

1.5. Volume type access (volumes)

Private volume type access to project.

By default, volumes types are public. To create a private volume type, set the `is_public` boolean field to `false` at volume type creation time. To control access to a private volume type, user needs to add a project to or remove a project from the volume type. Private volume types without projects are only accessible by users with the administrative role and context.

Method	URI	Description
POST	`/v2/{tenant_id}/types/{volume_type}/action`	Adds private volume type access to a project.
POST	`/v2/{tenant_id}/types/{volume_type}/action`	Removes private volume type access from a project.
GET	`/v2/{tenant_id}/types/{volume_type}/os-volume-type-access`	Lists project IDs that have access to private volume type.

1.5.1. Add private volume type access

Method	URI	Description
POST	`/v2/{tenant_id}/types/{volume_type}/action`	Adds private volume type access to a project.

Normal response codes: 202

1.5.1.1. Request

This table shows the URI parameters for the add private volume type access request:

Name	Type	Description
`{tenant_id}`	UUID	The ID of the project. Volume Type access to be added to this project ID.
`{volume_type}`	UUID	The ID of Volume Type to be accessed by project.

Example 1.31. Add private volume type access: JSON request

```
{
    "addProjectAccess": {
        "project": "f270b245cb11498ca4031deb7e141cfa"
    }
}
```

Example 1.32. Add private volume type access: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<addProjectAccess id="bb4f8f7f-fc38-4807-bd78-5710792205e1">
  <project>"f270b245cb11498ca4031deb7e141cfa"</project>
</addProjectAccess>
```

This operation does not accept a request body.

1.5.2. Remove private volume type access

Method	URI	Description
POST	`/v2/{tenant_id}/types/{volume_type}/action`	Removes private volume type access from a project.

Normal response codes: 202

1.5.2.1. Request

This table shows the URI parameters for the remove private volume type access request:

Name	Type	Description
`{tenant_id}`	UUID	The ID of the project. Volume Type access to be added to this project ID.
`{volume_type}`	UUID	The ID of Volume Type to be accessed by project.

Example 1.33. Remove private volume type access: JSON request

```
{
    "removeProjectAccess": {
        "project": "f270b245cb11498ca4031deb7e141cfa"
    }
}
```

Example 1.34. Remove private volume type access: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<removeProjectAccess id="bb4f8f7f-fc38-4807-bd78-5710792205e1">
  <project>"f270b245cb11498ca4031deb7e141cfa"</project>
</removeProjectAccess>
```

This operation does not accept a request body.

1.5.3. List private volume type access details

Method	URI	Description
GET	`/v2/{tenant_id}/types/{volume_type}/os-volume-type-access`	Lists project IDs that have access to private volume type.

Normal response codes: 200

1.5.3.1. Request

This table shows the URI parameters for the list private volume type access details request:

Name	Type	Description
`{tenant_id}`	UUID	The ID of the project. Volume Type access to be added to this project ID.
`{volume_type}`	UUID	The ID of Volume Type to be accessed by project.

1.5.3.2. Response

Example 1.35. List private volume type access details: JSON response

```
{
    "volume_type_access": {
        "volume_type_id": "3c67e124-39ad-4ace-a507-8bb7bf510c26",
        "project_id": "f270b245cb11498ca4031deb7e141cfa"
    }
}
```

1.6. Volume actions (volumes, action)

Extends the size of, resets statuses for, sets image metadata for, and removes image metadata from a volume. Attaches a volume to a server, detaches a volume from a server, and removes a volume from Block Storage management without actually removing the back-end storage object associated with it.

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Extends the size of a volume to a requested size, in gibibytes (GiB). Specify the `os-extend` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Resets the status, attach status, and migration status for a volume. Specify the `os-reset_status` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Sets the image metadata for a volume. Specify the `os-set_image_metadata` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Removes image metadata, by key, from a volume. Specify the `os-unset_image_metadata` action in the request body and the `key` for the metadata key and value pair that you want to remove.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Attaches a volume to a server. Specify the `os-attach` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Removes a volume from Block Storage management without removing the back-end storage object that is associat-

Method	URI	Description
		ed with it. Specify the `os-unmanage` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Forces a volume to detach. Specify the `os-force_detach` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Promotes a replicated volume. Specify the `os-promote-replica` action in the request body.
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Re-enables replication of a volume. Specify the `volume-replica-reenable` action in the request body.

1.6.1. Extend volume size

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Extends the size of a volume to a requested size, in gibibytes (GiB). Specify the `os-extend` action in the request body.

Preconditions

- Volume status must be `available`.
- Sufficient amount of storage must exist to extend the volume.
- The user quota must have sufficient volume storage.

Troubleshooting

- An `error_extending` volume status indicates that the request failed. Ensure that you meet the preconditions and retry the request. If the request fails again, investigate the storage back end.

Normal response codes: 202

1.6.1.1. Request

This table shows the URI parameters for the extend volume size request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.36. Extend volume size: JSON request

```
{
    "os-extend": {
        "new_size": 3
    }
}
```

1.6.2. Reset volume statuses

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Resets the status, attach status, and migration status for a volume. Specify the `os-reset_status` action in the request body.

Normal response codes: 202

1.6.2.1. Request

This table shows the URI parameters for the reset volume statuses request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.37. Reset volume statuses: JSON request

```
{
    "os-reset_status": {
        "status": "available",
        "attach_status": "detached",
        "migration_status": "migrating"
    }
}
```

1.6.3. Set image metadata for volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Sets the image metadata for a volume. Specify the `os-set_image_metadata` action in the request body.

Normal response codes: 202

1.6.3.1. Request

This table shows the URI parameters for the set image metadata for volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.38. Set image metadata for volume: JSON request

```
{
    "os-set_image_metadata": {
        "metadata": {
            "image_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
            "image_name": "image",
            "kernel_id": "155d900f-4e14-4e4c-a73d-069cbf4541e6",
            "ramdisk_id": "somedisk"
        }
    }
}
```

1.6.4. Remove image metadata from volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Removes image metadata, by key, from a volume. Specify the `os-unset_image_metadata` action in the request body and the `key` for the metadata key and value pair that you want to remove.

Normal response codes: 202

1.6.4.1. Request

This table shows the URI parameters for the remove image metadata from volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.39. Remove image metadata from volume: JSON request

```
{
    "os-unset_image_metadata": {
        "key": "ramdisk_id"
    }
}
```

1.6.5. Attach volume to server

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Attaches a volume to a server. Specify the `os-attach` action in the request body.

Preconditions

- Volume status must be `available`.
- You should set `instance_uuid` or `host_name`.

Normal response codes: 202

1.6.5.1. Request

This table shows the URI parameters for the attach volume to server request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.40. Attach volume to server: JSON request

```
{
    "os-attach": {
        "instance_uuid": "95D9EF50-507D-11E5-B970-0800200C9A66",
        "mountpoint": "/dev/vdc"
    }
}
```

1.6.6. Unmanage volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Removes a volume from Block Storage management without removing the back-end storage object that is associated with it. Specify the `os-unmanage` action in the request body.

Preconditions

- Volume status must be `available`.

Normal response codes: 202

1.6.6.1. Request

This table shows the URI parameters for the unmanage volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.41. Unmanage volume: JSON request

```
{
    "os-unmanage": {}
}
```

1.6.7. Force detach volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Forces a volume to detach. Specify the `os-force_detach` action in the request body.

Rolls back an unsuccessful detach operation after you disconnect the volume.

Normal response codes: 202

1.6.7.1. Request

This table shows the URI parameters for the force detach volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.42. Force detach volume: JSON request

```
{
    "os-force_detach": {
        "attachment_id": "d8777f54-84cf-4809-a679-468ffed56cf1",
        "connector": {
            "initiator": "iqn.2012-07.org.fake:01"
        }
    }
}
```

1.6.8. Promote replicated volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Promotes a replicated volume. Specify the `os-promote-replica` action in the request body.

Normal response codes: 202

1.6.8.1. Request

This table shows the URI parameters for the promote replicated volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.43. Promote replicated volume: JSON request

```
{
    "os-promote-replica": {}
}
```

1.6.9. Reenable volume replication

Method	URI	Description
POST	`/v2/{tenant_id}/volumes/{volume_id}/action`	Re-enables replication of a volume. Specify the `volume-replica-reenable` action in the request body.

Normal response codes: 202

1.6.9.1. Request

This table shows the URI parameters for the reenable volume replication request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

Example 1.44. Reenable volume replication: JSON request

```
{
    "os-reenable-replica": {}
}
```

1.7. Backups (backups)

A backup is a full copy of a volume stored in an external service. The service can be configured. The only supported service is Object Storage. A backup can subsequently be restored from the external service to either the same volume that the backup was originally taken from or to a new volume. Backup and restore operations can only be carried out on volumes that are in an unattached and available state.

When you create, list, or delete backups, these status values are possible:

Table 1.2. Backup statuses

Status	Description
creating	The backup is being created.
available	The backup is ready to restore to a volume.
deleting	The backup is being deleted.
error	A backup error occurred.
restoring	The backup is being restored to a volume.
error_restoring	A backup restoration error occurred.

If an error occurs, you can find more information about the error in the `fail_reason` field for the backup.

Method	URI	Description
POST	`/v2/{tenant_id}/backups`	Creates a Block Storage backup from a volume.
GET	`/v2/{tenant_id}/backups{?sort_key,sort_dir,limit,marker}`	Lists Block Storage backups to which the tenant has access.

Method	URI	Description
GET	`/v2/{tenant_id}/backups/detail{?sort_key,sort_dir,limit,marker}`	Lists Block Storage backups, with details, to which the tenant has access.
GET	`/v2/{tenant_id}/backups/{backup_id}`	Shows details for a backup.
DELETE	`/v2/{tenant_id}/backups/{backup_id}`	Deletes a backup.
POST	`/v2/{tenant_id}/backups/{backup_id}/restore`	Restores a Block Storage backup to an existing or new Block Storage volume.

1.7.1. Create backup

Method	URI	Description
POST	`/v2/{tenant_id}/backups`	Creates a Block Storage backup from a volume.

Normal response codes: 202

1.7.1.1. Request

This table shows the URI parameters for the create backup request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.45. Create backup: JSON request

```
{
    "backup": {
        "container": null,
        "description": null,
        "name": "backup001",
        "volume_id": "64f5d2fb-d836-4063-b7e2-544d5c1ff607",
        "incremental": true
    }
}
```

1.7.1.2. Response

Example 1.46. Create backup: JSON response

```
{
    "backup": {
        "id": "deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
        "links": [
            {
                "href": "http://localhost:8776/v2/
c95fc3e4afe248a49a28828f286a7b38/backups/deac8b8c-35c9-4c71-
acaa-889c2d5d5c8e",
                "rel": "self"
            },
            {
                "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/deac8b8c-35c9-4c71-
acaa-889c2d5d5c8e",
                "rel": "bookmark"
            }
        ],
        "name": "backup001"
    }
}
```

1.7.2. List backups

Method	URI	Description
GET	`/v2/{tenant_id}/backups{?sort_key, sort_dir,limit,marker}`	Lists Block Storage backups to which the tenant has access.

Normal response codes: 200

1.7.2.1. Request

This table shows the URI parameters for the list backups request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list backups request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.7.2.2. Response

Example 1.47. List backups: JSON response

```
{
    "backups": [
        {
            "id": "2ef47aee-8844-490c-804d-2a8efe561c65",
            "links": [
                {
                    "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
```

```
                    "rel": "bookmark"
                }
            ],
            "name": "backup001"
        },
        {
            "id": "4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
            "links": [
                {
                    "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
                    "rel": "bookmark"
                }
            ],
            "name": "backup002"
        }
    ]
}
```

1.7.3. List backups with details

Method	URI	Description
GET	`/v2/{tenant_id}/backups/detail{?sort_key,sort_dir,limit,marker}`	Lists Block Storage backups, with details, to which the tenant has access.

Normal response codes: 200

1.7.3.1. Request

This table shows the URI parameters for the list backups with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list backups with details request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.7.3.2. Response

Example 1.48. List backups with details: JSON response

```
{
    "backups": [
        {
            "availability_zone": "az1",
            "container": "volumebackups",
            "created_at": "2013-04-02T10:35:27.000000",
            "description": null,
            "fail_reason": null,
            "id": "2ef47aee-8844-490c-804d-2a8efe561c65",
            "links": [
                {
                    "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
                    "rel": "self"
```

```
                },
                {
                    "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
                    "rel": "bookmark"
                }
            ],
            "name": "backup001",
            "object_count": 22,
            "size": 1,
            "status": "available",
            "volume_id": "e5185058-943a-4cb4-96d9-72c184c337d6",
            "is_incremental": true,
            "has_dependent_backups": false
        },
        {
            "availability_zone": "az1",
            "container": "volumebackups",
            "created_at": "2013-04-02T10:21:48.000000",
            "description": null,
            "fail_reason": null,
            "id": "4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
            "links": [
                {
                    "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
                    "rel": "bookmark"
                }
            ],
            "name": "backup002",
            "object_count": 22,
            "size": 1,
            "status": "available",
            "volume_id": "e5185058-943a-4cb4-96d9-72c184c337d6",
            "is_incremental": true,
            "has_dependent_backups": false
        }
    ]
}
```

1.7.4. Show backup details

Method	URI	Description
GET	`/v2/{tenant_id}/backups/{backup_id}`	Shows details for a backup.

Normal response codes: 200

1.7.4.1. Request

This table shows the URI parameters for the show backup details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{backup_id}`	UUID	The UUID for a backup.

This operation does not accept a request body.

1.7.4.2. Response

Example 1.49. Show backup details: JSON response

```
{
    "backup": {
        "availability_zone": "az1",
        "container": "volumebackups",
        "created_at": "2013-04-02T10:35:27.000000",
        "description": null,
        "fail_reason": null,
        "id": "2ef47aee-8844-490c-804d-2a8efe561c65",
        "links": [
            {
                "href": "http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
                "rel": "self"
            },
            {
                "href": "http://localhost:8776/
c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
                "rel": "bookmark"
            }
        ],
        "name": "backup001",
        "object_count": 22,
        "size": 1,
        "status": "available",
        "volume_id": "e5185058-943a-4cb4-96d9-72c184c337d6",
        "is_incremental": true,
        "has_dependent_backups": false
    }
}
```

1.7.5. Delete backup

Method	URI	Description
DELETE	`/v2/{tenant_id}/backups/{backup_id}`	Deletes a backup.

Normal response codes: 204

1.7.5.1. Request

This table shows the URI parameters for the delete backup request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{backup_id}`	UUID	The UUID for a backup.

This operation does not accept a request body.

1.7.6. Restore backup

Method	URI	Description
POST	`/v2/{tenant_id}/backups/{backup_id}/restore`	Restores a Block Storage backup to an existing or new Block Storage volume.

You must specify either the UUID or name of the volume. If you specify both the UUID and name, the UUID takes priority.

Normal response codes: 202

1.7.6.1. Request

This table shows the URI parameters for the restore backup request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{backup_id}`	UUID	The UUID for a backup.

Example 1.50. Restore backup: JSON request

```
{
    "restore": {
        "name": "vol-01",
        "volume_id": "64f5d2fb-d836-4063-b7e2-544d5c1ff607"
    }
}
```

1.7.6.2. Response

Example 1.51. Restore backup: JSON response

```
{
    "restore": {
        "backup_id": "2ef47aee-8844-490c-804d-2a8efe561c65",
        "volume_id": "795114e8-7489-40be-a978-83797f2c1dd3"
    }
}
```

1.8. Backup actions (backups, action)

Force-deletes a backup.

Method	URI	Description
POST	`/v2/{tenant_id}/backups/{backup_id}/action`	Force-deletes a backup. Specify the `os-force_delete` action in the request body.

1.8.1. Force-delete backup

Method	URI	Description
POST	`/v2/{tenant_id}/backups/{backup_id}/action`	Force-deletes a backup. Specify the `os-force_delete` action in the request body.

This operations deletes the backup and any backup data.

The backup driver returns the `405` status code if it does not support this operation.

Normal response codes: 202

Error response codes: 404, 405

1.8.1.1. Request

This table shows the URI parameters for the force-delete backup request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{backup_id}`	UUID	The UUID for a backup.

Example 1.52. Force-delete backup: JSON request

```
{
    "os-force_delete": {}
}
```

1.9. Capabilities for storage back ends (capabilities)

Shows capabilities for a storage back end.

Method	URI	Description
GET	`/v2/{tenant_id}/capabilities/{hostname}`	Shows capabilities for a storage back end.

1.9.1. Show back-end capabilities

Method	URI	Description
GET	`/v2/{tenant_id}/capabilities/{hostname}`	Shows capabilities for a storage back end.

Normal response codes: 200

1.9.1.1. Request

This table shows the URI parameters for the show back-end capabilities request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{hostname}`	String	The name of the host that hosts the storage back end.

1.9.1.2. Response

Example 1.53. Show back-end capabilities: JSON response

```
{
    "namespace": "OS::Storage::Capabilities::fake",
    "vendor_name": "OpenStack",
    "volume_backend_name": "lvm",
    "pool_name": "pool",
    "driver_version": "2.0.0",
    "storage_protocol": "iSCSI",
    "display_name": "Capabilities of Cinder LVM driver",
    "description": "These are volume type options provided by Cinder LVM
 driver, blah, blah.",
    "visibility": "public",
    "properties": {
        "compression": {
            "title": "Compression",
            "description": "Enables compression.",
            "type": "boolean"
        },
        "qos": {
            "title": "QoS",
            "description": "Enables QoS.",
            "type": "boolean"
        },
        "replication": {
            "title": "Replication",
            "description": "Enables replication.",
            "type": "boolean"
        },
        "thin_provisioning": {
            "title": "Thin Provisioning",
            "description": "Sets thin provisioning.",
            "type": "boolean"
        }
    }
}
```

This operation does not return a response body.

1.10. Quota sets extension (os-quota-sets)

Administrators only, depending on policy settings.

Shows, updates, and deletes quotas for a tenant.

Method	URI	Description
GET	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}{?usage}`	Shows quotas for a tenant.
PUT	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}`	Updates quotas for a tenant.
DELETE	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}`	Deletes quotas for a tenant so the quotas revert to default values.
GET	`/v2/{tenant_id}/os-quota-sets/defaults`	Gets default quotas for a tenant.
GET	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Enables an admin user to show quotas for a tenant and user.
PUT	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Updates quotas for a tenant and user.
DELETE	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Deletes quotas for a user so that the quotas revert to default values.
GET	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}`	Shows details for quotas for a tenant and user.

1.10.1. Show quotas

Method	URI	Description
GET	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}{?usage}`	Shows quotas for a tenant.

Normal response codes: 200

1.10.1.1. Request

This table shows the URI parameters for the show quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.

This table shows the query parameters for the show quotas request:

Name	Type	Description
`usage`	Boolean *(Optional)*	Set to `usage=true` to show quota usage. Default is `false`.

This operation does not accept a request body.

1.10.1.2. Response

Example 1.54. Show quotas response: JSON

```
{
    "quota_set": {
        "gigabytes": 5,
        "snapshots": 10,
        "volumes": 20
    }
}
```

This table shows the body parameters for the show quotas response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.

Name	Type	Description
injected_file_content_bytes	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int *(Required)*	The number of bytes that are allowed for each injected file path.
injected_files	Int *(Required)*	The number of injected files that are allowed for each tenant.
instances	Int *(Required)*	The number of instances that are allowed for each tenant.
key_pairs	Int *(Required)*	The number of key pairs that are allowed for each user.
metadata_items	Int *(Required)*	The number of metadata items that are allowed for each instance.
ram	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int *(Optional)*	The number of rules that are allowed for each security group.
security_groups	Int *(Required)*	The number of security groups that are allowed for each tenant.
in_use	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
reserved	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.55. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.2. Update quotas

Method	URI	Description
PUT	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}`	Updates quotas for a tenant.

Normal response codes: 200

1.10.2.1. Request

This table shows the URI parameters for the update quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.

Example 1.56. Update quotas response: JSON

```
{
    "quota_set": {
        "snapshots": 45
    }
}
```

This table shows the body parameters for the update quotas request:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	(Required)	
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.57. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <snapshots>45</snapshots>
</quota_set>
```

This operation does not accept a request body.

1.10.2.2. Response

Example 1.58. Update quota response: JSON

```
{
    "quota_set": {
        "snapshots": 45
    }
}
```

This table shows the body parameters for the update quotas response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int	The number of bytes of content that are allowed for each injected file.

Name	Type	Description
	(Required)	
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.59. Update quota response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.3. Delete quotas

Method	URI	Description
DELETE	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}`	Deletes quotas for a tenant so the quotas revert to default values.

Normal response codes: 200

1.10.3.1. Request

This table shows the URI parameters for the delete quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.

This operation does not accept a request body.

1.10.4. Get default quotas

Method	URI	Description
GET	`/v2/{tenant_id}/os-quota-sets/defaults`	Gets default quotas for a tenant.

Normal response codes: 200

1.10.4.1. Request

This table shows the URI parameters for the get default quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.10.4.2. Response

Example 1.60. Get default quotas response: JSON

```
{
    "quota_set": {
        "gigabytes": 5,
        "snapshots": 10,
        "volumes": 20
    }
}
```

This table shows the body parameters for the get default quotas response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int	The number of injected files that are allowed for each tenant.

Name	Type	Description
	(Required)	
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.61. Get default quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.5. Show quotas for user

Method	URI	Description
GET	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Enables an admin user to show quotas for a tenant and user.

Normal response codes: 200

1.10.5.1. Request

This table shows the URI parameters for the show quotas for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

This operation does not accept a request body.

1.10.5.2. Response

Example 1.62. Show quotas for user response: JSON

```
{
    "quota_set": {
        "snapshots": 45
    }
}
```

This table shows the body parameters for the show quotas for user response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int	The number of injected files that are allowed for each tenant.

Name	Type	Description
	(Required)	
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.63. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.6. Update quotas for user

Method	URI	Description
PUT	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Updates quotas for a tenant and user.

Normal response codes: 200

1.10.6.1. Request

This table shows the URI parameters for the update quotas for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

Example 1.64. Update quotas for user request: JSON

```
{
    "quota_set": {
        "snapshots": 45
    }
}
```

This table shows the body parameters for the update quotas for user request:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	(Required)	
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.65. Update quotas for user request: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <snapshots>45</snapshots>
</quota_set>
```

This operation does not accept a request body.

1.10.6.2. Response

Example 1.66. Update quotas for user response: JSON

```
{
    "quota_set": {
        "snapshots": 45
    }
}
```

This table shows the body parameters for the update quotas for user response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int	The number of bytes of content that are allowed for each injected file.

Name	Type	Description
	(Required)	
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 1.67. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <gigabytes>5</gigabytes>
  <snapshots>10</snapshots>
  <volumes>20</volumes>
</quota_set>
```

This operation does not return a response body.

1.10.7. Delete quotas for user

Method	URI	Description
DELETE	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Deletes quotas for a user so that the quotas revert to default values.

Normal response codes: 200

1.10.7.1. Request

This table shows the URI parameters for the delete quotas for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

This operation does not accept a request body.

1.10.8. Show quota details for user

Method	URI	Description
GET	`/v2/{admin_tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}`	Shows details for quotas for a tenant and user.

Normal response codes: 200

1.10.8.1. Request

This table shows the URI parameters for the show quota details for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

This operation does not accept a request body.

1.10.8.2. Response

Example 1.68. Show quota details for user response: JSON

```
{
    "quota_set": {
        "snapshots": 45
    }
}
```

1.11. Quality of service (QoS) specifications (qos-specs)

Administrators only, depending on policy settings.

Creates, lists, shows details for, associates, disassociates, sets keys, unsets keys, and deletes quality of service (QoS) specifications.

Method	URI	Description
POST	`/v2/{tenant_id}/qos-specs`	Creates a QoS specification.
GET	`/v2/{tenant_id}/qos-specs{?sort_key,sort_dir,limit,marker}`	Lists quality of service (QoS) specifications.
GET	`/v2/{tenant_id}/qos-specs/{qos_id}`	Shows details for a QoS specification.
PUT	`/v2/{tenant_id}/qos-specs/{qos_id}`	Sets keys in a QoS specification.
DELETE	`/v2/{tenant_id}/qos-specs/{qos_id}`	Deletes a QoS specification.
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/associate`	Associates a QoS specification with a volume type.
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/disassociate`	Disassociates a QoS specification from a volume type.

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all`	Disassociates a QoS specification from all associations.
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/associations`	Lists all associations for a QoS specification.

1.11.1. Create QoS specification

Method	URI	Description
POST	`/v2/{tenant_id}/qos-specs`	Creates a QoS specification.

Specify one or more key and value pairs in the request body.

Normal response codes: 202

1.11.1.1. Request

This table shows the URI parameters for the create qos specification request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.69. Create QoS specification: JSON request

```
{
    "qos_specs": {
        "availability": "100",
        "name": "reliability-spec",
        "numberOfFailures": "0"
    }
}
```

This table shows the body parameters for the create qos specification request:

Name	Type	Description
`qos_specs`	Dict *(Required)*	A `qos_specs` object.
`name`	String *(Required)*	The name of the QoS specification.
`consumer`	String *(Optional)*	The consumer type.

Example 1.70. Create QoS specification: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<qos_specs name="performance-spec" delay="0" throughput="100" />
```

This operation does not accept a request body.

1.11.1.2. Response

Example 1.71. Create QoS specification: JSON response

```
{
    "qos_specs": {
        "specs": {
            "numberOfFailures": "0",
```

```
            "availability": "100"
        },
        "consumer": "back-end",
        "name": "reliability-spec",
        "id": "599ef437-1c99-42ec-9fc6-239d0519fef1"
    },
    "links": [
        {
            "href": "http://23.253.248.171:8776/
v2/bab7d5c60cd041a0a36f7c4b6e1dd978/qos_specs/
599ef437-1c99-42ec-9fc6-239d0519fef1",
            "rel": "self"
        },
        {
            "href": "http://23.253.248.171:8776/
bab7d5c60cd041a0a36f7c4b6e1dd978/qos_specs/
599ef437-1c99-42ec-9fc6-239d0519fef1",
            "rel": "bookmark"
        }
    ]
}
```

This table shows the body parameters for the create qos specification response:

Name	Type	Description
`qos_specs`	Dict *(Required)*	A `qos_specs` object.
`specs`	Dict *(Required)*	A `specs` object.
`consumer`	String *(Required)*	The consumer type.
`name`	String *(Required)*	The name of the QoS specification.
`id`	UUID *(Required)*	The generated ID for the QoS specification.
`links`	List *(Required)*	The QoS specification links.

Example 1.72. Create QoS specification: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<qos_specs>
    <qos_spec consumer="back-end" id="e1f84cf3-6b1e-4f25-8130-3244f69ec7c1"
 name="performance-spec">
        <specs>
            <delay>0</delay>
            <throughput>100</throughput>
        </specs>
    </qos_spec>
</qos_specs>
```

This operation does not return a response body.

1.11.2. List QoS specs

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs{?sort_key,sort_dir,limit,marker}`	Lists quality of service (QoS) specifications.

Normal response codes: 200300

1.11.2.1. Request

This table shows the URI parameters for the list qos specs request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list qos specs request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an image attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.11.2.2. Response

Example 1.73. List QoS specs: JSON response

```
{
    "qos_specs": [
        {
            "specs": {
                "availability": "100",
                "numberOfFailures": "0"
            },
            "consumer": "back-end",
            "name": "reliability-spec",
            "id": "0388d6c6-d5d4-42a3-b289-95205c50dd15"
        },
        {
            "specs": {
                "delay": "0",
                "throughput": "100"
```

```
            },
            "consumer": "back-end",
            "name": "performance-spec",
            "id": "ecfc6e2e-7117-44a4-8eec-f84d04f531a8"
        }
    ]
}
```

This table shows the body parameters for the list qos specs response:

Name	Type	Description
`qos_specs`	Dict *(Required)*	A `qos_specs` object.
`specs`	Dict *(Required)*	Specification key and value pairs.
`consumer`	String *(Required)*	The consumer type.
`name`	String *(Required)*	The name of the QoS specification.
`id`	UUID *(Required)*	The generated ID for the QoS specification.

Example 1.74. List QoS specs: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
    <qos_spec consumer="back-end"
        id="0388d6c6-d5d4-42a3-b289-95205c50dd15"
        name="reliability-spec">
        <specs>
            <availability>100</availability>
            <numberOfFailures>0</numberOfFailures>
        </specs>
    </qos_spec>
    <qos_spec consumer="back-end"
        id="ecfc6e2e-7117-44a4-8eec-f84d04f531a8"
        name="performance-spec">
        <specs>
            <delay>0</delay>
            <throughput>100</throughput>
        </specs>
    </qos_spec>
</qos_specs>
```

This operation does not return a response body.

1.11.3. Show QoS specification details

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs/{qos_id}`	Shows details for a QoS specification.

Normal response codes: 200

Error response codes: computeFault (400, 500, ...), serviceUnavailable (503), badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), itemNotFound (404)

1.11.3.1. Request

This table shows the URI parameters for the show qos specification details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.

This operation does not accept a request body.

1.11.3.2. Response

Example 1.75. Show QoS specification details: JSON response

```
{
    "qos_specs": {
        "specs": {
            "availability": "100",
            "numberOfFailures": "0"
        },
        "consumer": "back-end",
        "name": "reliability-spec",
        "id": "0388d6c6-d5d4-42a3-b289-95205c50dd15"
    },
    "links": [
        {
            "href": "http://23.253.228.211:8776/v2/
e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-
b289-95205c50dd15",
            "rel": "self"
        },
        {
            "href": "http://23.253.228.211:8776/
e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-
b289-95205c50dd15",
            "rel": "bookmark"
        }
    ]
}
```

This table shows the body parameters for the show qos specification details response:

Name	Type	Description
`qos_specs`	Dict	A `qos_specs` object.

Name	Type	Description
	(Required)	
`specs`	Dict *(Required)*	Specification key and value pairs.
`consumer`	String *(Required)*	The consumer type.
`name`	String *(Required)*	The name of the QoS specification.
`id`	UUID *(Required)*	The generated ID for the QoS specification.
`links`	List *(Required)*	The QoS specification links.

Example 1.76. Show QoS specification details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
    <qos_spec consumer="back-end"
        id="0388d6c6-d5d4-42a3-b289-95205c50dd15"
        name="reliability-spec">
        <specs>
            <availability>100</availability>
            <numberOfFailures>0</numberOfFailures>
        </specs>
    </qos_spec>
</qos_specs>
```

This operation does not return a response body.

1.11.4. Set keys in QoS specification

Method	URI	Description
PUT	`/v2/{tenant_id}/qos-specs/{qos_id}`	Sets keys in a QoS specification.

Normal response codes: 200

1.11.4.1. Request

This table shows the URI parameters for the set keys in qos specification request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.

Example 1.77. Set keys in QoS specification: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<qos_specs delay="2" throughput="100"/>
```

Example 1.78. Set keys in QoS specification: JSON request

```
{
    "qos_specs": {
        "delay": "1"
    }
}
```

This operation does not accept a request body.

1.11.4.2. Response

Example 1.79. Set keys in QoS specification: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
    <qos_spec/>
</qos_specs>
```

Example 1.80. Set keys in QoS specification: JSON response

```
{
    "qos_specs": {
        "delay": "1"
    }
}
```

This operation does not return a response body.

1.11.5. Delete QoS specification

Method	URI	Description
DELETE	`/v2/{tenant_id}/qos-specs/{qos_id}`	Deletes a QoS specification.

Normal response codes: 202

1.11.5.1. Request

This table shows the URI parameters for the delete qos specification request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.
`{force}`	Boolean	To delete a QoS specification even if it is in-use, set to `true`. Default is `false`.

This operation does not accept a request body.

1.11.6. Associate QoS specification with volume type

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/associate`	Associates a QoS specification with a volume type.

Normal response codes: 202

1.11.6.1. Request

This table shows the URI parameters for the associate qos specification with volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.11.7. Disassociate QoS specification from volume type

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/disassociate`	Disassociates a QoS specification from a volume type.

Normal response codes: 202

1.11.7.1. Request

This table shows the URI parameters for the disassociate qos specification from volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.11.8. Disassociate QoS specification from all associations

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/disassociate_all`	Disassociates a QoS specification from all associations.

Normal response codes: 202

1.11.8.1. Request

This table shows the URI parameters for the disassociate qos specification from all associations request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.

This operation does not accept a request body.

1.11.9. Get all associations for QoS specification

Method	URI	Description
GET	`/v2/{tenant_id}/qos-specs/{qos_id}/associations`	Lists all associations for a QoS specification.

Normal response codes: 200

1.11.9.1. Request

This table shows the URI parameters for the get all associations for qos specification request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{qos_id}`	String	The ID of the QoS specification.

This operation does not accept a request body.

1.11.9.2. Response

Example 1.81. Get all associations for QoS specification: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<qos_specs>
    <qos_spec consumer="back-end"
        id="0388d6c6-d5d4-42a3-b289-95205c50dd15"
        name="reliability-spec">
        <specs>
            <availability>100</availability>
            <numberOfFailures>0</numberOfFailures>
        </specs>
    </qos_spec>
</qos_specs>
```

Example 1.82. Get all associations for QoS specification: JSON response

```
{
    "qos_specs": {
        "specs": {
            "availability": "100",
            "numberOfFailures": "0"
        },
        "consumer": "back-end",
        "name": "reliability-spec",
        "id": "0388d6c6-d5d4-42a3-b289-95205c50dd15"
    },
    "links": [
        {
            "href": "http://23.253.228.211:8776/v2/
e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-
b289-95205c50dd15",
            "rel": "self"
        },
```

```
        {
            "href": "http://23.253.228.211:8776/
e1cf63117ae74309a5bcc2002a23be8b/qos_specs/0388d6c6-d5d4-42a3-
b289-95205c50dd15",
            "rel": "bookmark"
        }
    ]
}
```

This operation does not return a response body.

1.12. Volume types (types)

Method	URI	Description
GET	`/v2/{tenant_id}/types{?sort_key, sort_dir,limit,marker}`	Lists volume types.
POST	`/v2/{tenant_id}/types`	Creates a volume type.
PUT	`/v2/{tenant_id}/types/{volume_type_id}`	Updates a volume type.
PUT	`/v2/{tenant_id}/types/{volume_type_id}`	Updates the extra specifications that are assigned to a volume type.
GET	`/v2/{tenant_id}/types/{volume_type_id}`	Shows details for a volume type.
DELETE	`/v2/{tenant_id}/types/{volume_type_id}`	Deletes a volume type.

1.12.1. List volume types

Method	URI	Description
GET	`/v2/{tenant_id}/types{?sort_key, sort_dir,limit,marker}`	Lists volume types.

Normal response codes: 200

1.12.1.1. Request

This table shows the URI parameters for the list volume types request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list volume types request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.12.1.2. Response

Example 1.83. List volume types: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_types
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content">
    <volume_type id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="SSD">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
    </volume_type>
    <volume_type id="8eb69a46-df97-4e41-9586-9a40a7533803" name="SATA"
    />
</volume_types>
```

Example 1.84. List volume types: JSON response

```
{
```

```
    "volume_types": [
        {
            "extra_specs": {
                "capabilities": "gpu"
            },
            "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
            "name": "SSD"
        },
        {
            "extra_specs": {},
            "id": "8eb69a46-df97-4e41-9586-9a40a7533803",
            "name": "SATA"
        }
    ]
}
```

This operation does not return a response body.

1.12.2. Create volume type

Method	URI	Description
POST	`/v2/{tenant_id}/types`	Creates a volume type.

To create an environment with multiple-storage back ends, you must specify a volume type. Block Storage volume back ends are spawned as children to `cinder-volume`, and they are keyed from a unique queue. They are named `cinder-volume.HOST.BACKEND`. For example, `cinder-volume.ubuntu.lvmdriver`. When a volume is created, the scheduler chooses an appropriate back end to handle the request based on the volume type.

For information about how to use volume types to create multiple-storage back ends, see Configure multiple-storage back ends.

Normal response codes: 200

1.12.2.1. Request

This table shows the URI parameters for the create volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.85. Create volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1" xmlns:os-volume-
type-access="http://docs.openstack.org/openstack-block-storage/2.0/ext/os-
volume-type-access/api/v2.0" name="vol-type-001" description="volume type
 0001" os-volume-type-access:is_public="true">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
</volume_type>
```

Example 1.86. Create volume type: JSON request

```
{
    "volume_type": {
        "name": "vol-type-001",
        "description": "volume type 0001",
        "os-volume-type-access:is_public": true,
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not accept a request body.

1.12.2.2. Response

Example 1.87. Create volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<volume_type
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
    description="volume type 001" is_public="true">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 1.88. Create volume type: JSON response

```
{
    "volume_type": {
        "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
        "name": "vol-type-001",
        "description": "volume type 001",
        "is_public": "true",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

1.12.3. Update volume type

Method	URI	Description
PUT	`/v2/{tenant_id}/types/{volume_type_id}`	Updates a volume type.

To create an environment with multiple-storage back ends, you must specify a volume type. The API spawns Block Storage volume back ends as children to `cinder-volume`, and keys them from a unique queue. The API names the back ends `cinder-volume.HOST.BACKEND`. For example, `cinder-volume.ubuntu.lvmdriver`. When you create a volume, the scheduler chooses an appropriate back end for the volume type to handle the request.

For information about how to use volume types to create multiple-storage back ends, see Configure multiple-storage back ends.

Normal response codes: 200

1.12.3.1. Request

This table shows the URI parameters for the update volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

Example 1.89. Update volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1" xmlns:os-volume-
type-access="http://docs.openstack.org/openstack-block-storage/2.0/ext/os-
volume-type-access/api/v2.0" name="vol-type-001" description="volume type
 0001" is_public="true">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
</volume_type>
```

Example 1.90. Update volume type: JSON request

```
{
    "volume_type": {
        "name": "vol-type-001",
        "description": "volume type 0001",
        "is_public": true,
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not accept a request body.

1.12.3.2. Response

Example 1.91. Update volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
    description="volume type 001" is_public="true">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 1.92. Update volume type: JSON response

```
{
    "volume_type": {
        "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
        "name": "vol-type-001",
        "description": "volume type 001",
        "is_public": "true",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

1.12.4. Update extra specs for a volume type

Method	URI	Description
PUT	`/v2/{tenant_id}/types/{volume_type_id}`	Updates the extra specifications that are assigned to a volume type.

Normal response codes: 200

1.12.4.1. Request

This table shows the URI parameters for the update extra specs for a volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

Example 1.93. Update extra specs for a volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1" xmlns:os-volume-
type-access="http://docs.openstack.org/openstack-block-storage/2.0/ext/os-
volume-type-access/api/v2.0" name="vol-type-001" description="volume type
 0001" is_public="true">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
</volume_type>
```

Example 1.94. Update extra specs for a volume type: JSON request

```
{
    "volume_type": {
        "name": "vol-type-001",
        "description": "volume type 0001",
        "is_public": true,
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not accept a request body.

1.12.4.2. Response

Example 1.95. Update extra specs for a volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
    description="volume type 001" is_public="true">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
```

```
    </extra_specs>
</volume_type>
```

Example 1.96. Update extra specs for a volume type: JSON response

```
{
    "volume_type": {
        "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
        "name": "vol-type-001",
        "description": "volume type 001",
        "is_public": "true",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

1.12.5. Show volume type details

Method	URI	Description
GET	`/v2/{tenant_id}/types/{volume_type_id}`	Shows details for a volume type.

Normal response codes: 200

1.12.5.1. Request

This table shows the URI parameters for the show volume type details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.12.5.2. Response

Example 1.97. Show volume type details: JSON response

```
{
    "volume_type": {
        "id": "6685584b-1eac-4da6-b5c3-555430cf68ff",
        "name": "vol-type-001",
        "description": "volume type 001",
        "is_public": "true",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This table shows the body parameters for the show volume type details response:

Name	Type	Description
`volume_type`	Dict *(Required)*	The `volume_type` object.
`name`	String *(Required)*	The name of the volume type.
`extra_specs`	Dict *(Required)*	A set of key and value pairs that contains the specifications for a volume type.
`description`	String *(Optional)*	The description of the volume type.
`is_public`	Boolean *(Optional)*	Indicates whether the volume type is public or not. Default is `true`.

Example 1.98. Show volume type details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<volume_type
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="vol-type-001"
    description="volume type 001" is_public="true">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

This operation does not return a response body.

1.12.6. Delete volume type

Method	URI	Description
DELETE	`/v2/{tenant_id}/types/{volume_type_id}`	Deletes a volume type.

Normal response codes: 202

1.12.6.1. Request

This table shows the URI parameters for the delete volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

1.13. Volume snapshots (snapshots)

A snapshot is a point-in-time copy of the data that a volume contains.

When you create, list, or delete snapshots, these status values are possible:

Table 1.3. Snapshot statuses

Status	Description
creating	The snapshot is being created.
available	The snapshot is ready to use.
deleting	The snapshot is being deleted.
error	A snapshot creation error occurred.
error_deleting	A snapshot deletion error occurred.

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots`	Creates a volume snapshot, which is a point-in-time, complete copy of a volume. You can create a volume from a snapshot.
GET	`/v2/{tenant_id}/snapshots{?sort_key,sort_dir,limit,marker}`	Lists all Block Storage snapshots, with summary information, that the tenant can access.
GET	`/v2/{tenant_id}/snapshots/detail`	Lists all Block Storage snapshots, with details, that the tenant can access.
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Updates a snapshot.
DELETE	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Deletes a snapshot.
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}/metadata`	Shows metadata for a snapshot.

Method	URI	Description
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}/metadata`	Updates metadata for a snapshot.

1.13.1. Create snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots`	Creates a volume snapshot, which is a point-in-time, complete copy of a volume. You can create a volume from a snapshot.

Normal response codes: 202

1.13.1.1. Request

This table shows the URI parameters for the create snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the body parameters for the create snapshot request:

Name	Type	Description
`snapshot`	String *(Required)*	A partial representation of a snapshot used in the creation process.
`name`	String *(Optional)*	The name of the snapshot. Default is `None`.
`description`	String *(Optional)*	A description for the snapshot. Default is `None`.
`volume_id`	UUID *(Required)*	To create a snapshot from an existing volume, specify the UUID of the existing volume.
`force`	Boolean *(Optional)*	Indicates whether to snapshot, even if the volume is attached. Default is `false`.

Example 1.99. Create snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    name="snap-001" description="Daily backup"
    volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" force="true"/>
```

Example 1.100. Create snapshot: JSON request

```
{
    "snapshot": {
        "name": "snap-001",
        "description": "Daily backup",
        "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "force": true
    }
}
```

This operation does not accept a request body.

1.13.1.2. Response

Example 1.101. Create snapshot: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot status="creating" description="Daily backup"
    created_at="2013-02-25T03:56:53.081642"
    volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
    id="ffa9bc5e-1172-4021-acaf-cdcd78a9584d" name="snap-001">
    <metadata/>
</snapshot>
```

Example 1.102. Create snapshot: JSON response

```
{
    "snapshot": {
        "status": "creating",
        "description": "Daily backup",
        "created_at": "2013-02-25T03:56:53.081642",
        "metadata": {},
        "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "size": 1,
        "id": "ffa9bc5e-1172-4021-acaf-cdcd78a9584d",
        "name": "snap-001"
    }
}
```

This operation does not return a response body.

1.13.2. List snapshots

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots{?sort_key,sort_dir,limit,marker}`	Lists all Block Storage snapshots, with summary information, that the tenant can access.

Normal response codes: 200

1.13.2.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list snapshots request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.13.2.2. Response

Example 1.103. List snapshots: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots>
    <snapshot status="available"
        description="volume snapshot"
        created_at="2015-11-29 02:25:51+00:00"
        volume_id="173f7b48-c4c1-4e70-9acc-086b39073506"
        size="1" id="b1323cda-8e4b-41c1-afc5-2fc791809c8c"
        name="test-volume-snapshot">
        <metadata>
            <meta key="name">test</meta>
        </metadata>
    </snapshot>
</snapshots>
```

Example 1.104. List snapshots: JSON response

```
{
    "snapshots": [
        {
            "status": "available",
            "metadata": {
                "name": "test"
            },
            "name": "test-volume-snapshot",
            "volume_id": "173f7b48-c4c1-4e70-9acc-086b39073506",
            "created_at": "2015-11-29T02:25:51.000000",
            "size": 1,
            "id": "b1323cda-8e4b-41c1-afc5-2fc791809c8c",
            "description": "volume snapshot"
        }
    ]
}
```

This operation does not return a response body.

1.13.3. List snapshots with details

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/detail`	Lists all Block Storage snapshots, with details, that the tenant can access.

Normal response codes: 200

1.13.3.1. Request

This table shows the URI parameters for the list snapshots with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.13.3.2. Response

Example 1.105. List snapshots with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots
    xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/volume/
ext/extended_snapshot_attributes/api/v1">
    <snapshot status="available"
        description="volume snapshot"
        created_at="2015-11-29 02:25:51+00:00"
        volume_id="173f7b48-c4c1-4e70-9acc-086b39073506"
        size="1" id="b1323cda-8e4b-41c1-afc5-2fc791809c8c"
        name="test-volume-snapshot"
        os-extended-snapshot-attributes:project_id=
"bab7d5c60cd041a0a36f7c4b6e1dd978"
        os-extended-snapshot-attributes:progress="100%">
        <metadata>
            <meta key="name">test</meta>
        </metadata>
    </snapshot>
</snapshots>
```

Example 1.106. List snapshots with details: JSON response

```
{
    "snapshots": [
        {
            "status": "available",
            "metadata": {
                "name": "test"
            },
            "os-extended-snapshot-attributes:progress": "100%",
            "name": "test-volume-snapshot",
            "volume_id": "173f7b48-c4c1-4e70-9acc-086b39073506",
            "os-extended-snapshot-attributes:project_id":
 "bab7d5c60cd041a0a36f7c4b6e1dd978",
            "created_at": "2015-11-29T02:25:51.000000",
```

```
            "size": 1,
            "id": "b1323cda-8e4b-41c1-afc5-2fc791809c8c",
            "description": "volume snapshot"
        }
    ]
}
```

This operation does not return a response body.

1.13.4. Show snapshot details

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.

Normal response codes: 200

1.13.4.1. Request

This table shows the URI parameters for the show snapshot details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.4.2. Response

Example 1.107. Show snapshot details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/openstack-
block-storage/2.0/content/Extended_Snapshot_Attributes.html"
  status="available" description="Very important"
  created_at="2013-02-25 04:13:17"
  volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
  id="2bb856e1-b3d8-4432-a858-09e4ce939389" name="snap-001"
  os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
  os-extended-snapshot-attributes:progress="100%">
  <metadata/>
</snapshot>
```

Example 1.108. Show snapshot details: JSON response

```
{
    "snapshot": {
        "status": "available",
        "os-extended-snapshot-attributes:progress": "100%",
        "description": "Daily backup",
        "created_at": "2013-02-25T04:13:17.000000",
        "metadata": {},
        "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "os-extended-snapshot-attributes:project_id":
 "0c2eba2c5af04d3f9e9d0d410b371fde",
        "size": 1,
        "id": "2bb856e1-b3d8-4432-a858-09e4ce939389",
        "name": "snap-001"
    }
}
```

This operation does not return a response body.

1.13.5. Update snapshot

Method	URI	Description
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Updates a snapshot.

Normal response codes: 200

1.13.5.1. Request

This table shows the URI parameters for the update snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 1.109. Update snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    name="snap-002" description="This is yet, another snapshot."/>
```

Example 1.110. Update snapshot: JSON request

```
{
    "snapshot": {
        "name": "snap-002",
        "description": "This is yet, another snapshot."
    }
}
```

This operation does not accept a request body.

1.13.5.2. Response

Example 1.111. Update snapshot: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/openstack-
block-storage/2.0/content/Extended_Snapshot_Attributes.html"
    status="available"
    description="This is yet, another snapshot"
    created_at="2013-02-20T08:11:34.000000"
    volume_id="2402b902-0b7a-458c-9c07-7435a826f794"
    size="1"
    id="4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2"
    name="snap-002"
    os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
    os-extended-snapshot-attributes:progress="100%">
    <metadata/>
</snapshot>
```

Example 1.112. Update snapshot: JSON response

```
{
    "snapshot": {
        "created_at": "2013-02-20T08:11:34.000000",
        "description": "This is yet, another snapshot",
        "name": "snap-002",
        "id": "4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2",
        "size": 1,
        "status": "available",
        "volume_id": "2402b902-0b7a-458c-9c07-7435a826f794"
    }
}
```

This operation does not return a response body.

1.13.6. Delete snapshot

Method	URI	Description
DELETE	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Deletes a snapshot.

Normal response codes: 202

1.13.6.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.7. Show snapshot metadata

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}/metadata`	Shows metadata for a snapshot.

Normal response codes: 200

1.13.7.1. Request

This table shows the URI parameters for the show snapshot metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

1.13.7.2. Response

Example 1.113. Show snapshot metadata: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="name">test</meta>
</metadata>
```

Example 1.114. Show snapshot metadata: JSON response

```
{
    "metadata": {
        "name": "test"
    }
}
```

This operation does not return a response body.

1.13.8. Update snapshot metadata

Method	URI	Description
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}/metadata`	Updates metadata for a snapshot.

Replaces metadata items that match keys. Does not modify items that are not in the request.

Normal response codes: 200

1.13.8.1. Request

This table shows the URI parameters for the update snapshot metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 1.115. Update snapshot metadata: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata>
    <meta key="key">v2</meta>
</metadata>
```

Example 1.116. Update snapshot metadata: JSON request

```
{
    "metadata": {
        "key": "v2"
    }
}
```

This operation does not accept a request body.

1.13.8.2. Response

Example 1.117. Update snapshot metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="key">v2</meta>
</metadata>
```

Example 1.118. Update snapshot metadata: JSON response

```
{
    "metadata": {
        "key": "v2"
    }
}
```

This operation does not return a response body.

1.14. Volume manage extension (os-volume-manage)

Creates volumes by using existing storage instead of allocating new storage.

Method	URI	Description
POST	`/v2/{tenant_id}/os-volume-manage`	Creates a Block Storage volume by using existing storage rather than allocating new storage.

1.14.1. Manage existing volume

Method	URI	Description
POST	`/v2/{tenant_id}/os-volume-manage`	Creates a Block Storage volume by using existing storage rather than allocating new storage.

The caller must specify a reference to an existing storage volume in the *ref* parameter in the request. Although each storage driver might interpret this reference differently, the driver should accept a reference structure that contains either a *source-volume- id* or *source-volume-name* element, if possible.

The API chooses the size of the volume by rounding up the size of the existing storage volume to the next gibibyte (GiB).

Normal response codes: 202

1.14.1.1. Request

This table shows the URI parameters for the manage existing volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.119. Show manage existing request: JSON

```
{
    "volume": {
        "host": "geraint-VirtualBox",
        "ref": {
            "source-volume-name": "existingLV",
            "source-volume-id": "1234"
        },
        "name": "New Volume",
        "availability_zone": "az2",
        "description": "Volume imported from existingLV",
        "volume_type": null,
        "bootable": true,
        "metadata": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

1.14.1.2. Response

Example 1.120. Manage existing volume: JSON response

The response is the same as the volume create response.

```
{
    "volume": {
        "status": "creating",
        "user_id": "eae1472b5fc5496998a3d06550929e7e",
```

```
        "attachments": [],
        "links": [
            {
                "href": "http://10.0.2.15:8776/v2/
87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-
c781-4cd4-847d-5f2ec8cbd91c",
                "rel": "self"
            },
            {
                "href": "http://10.0.2.15:8776/
87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-
c781-4cd4-847d-5f2ec8cbd91c",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "az2",
        "bootable": "false",
        "encrypted": "false",
        "created_at": "2014-07-18T00:12:54.000000",
        "description": "Volume imported from existingLV",
        "os-vol-tenant-attr:tenant_id": "87c8522052ca4eed98bc672b4c1a3ddb",
        "volume_type": null,
        "name": "New Volume",
        "source_volid": null,
        "snapshot_id": null,
        "metadata": {
            "key2": "value2",
            "key1": "value1"
        },
        "id": "23cf872b-c781-4cd4-847d-5f2ec8cbd91c",
        "size": 0
    }
}
```


1.15. Volume image metadata extension (os-vol-image-meta)

Shows image metadata that is associated with a volume.

Method	URI	Description
GET	`/v2/{tenant_id}/os-vol-image-meta`	Shows image metadata for a volume.

1.15.1. Show image metadata for volume

Method	URI	Description
GET	`/v2/{tenant_id}/os-vol-image-meta`	Shows image metadata for a volume.

When the request is made, the caller must specify a reference to an existing storage volume in the `ref` element. Each storage driver may interpret the existing storage volume reference differently but should accept a reference structure containing either a `source-volume-id` or `source-volume-name` element, if possible.

Normal response codes: 202

1.15.1.1. Request

This table shows the URI parameters for the show image metadata for volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.121. Show manage existing request: JSON

```
{
    "volume": {
        "host": "geraint-VirtualBox",
        "ref": {
            "source-volume-name": "existingLV",
            "source-volume-id": "1234"
        },
        "name": "New Volume",
        "availability_zone": "az2",
        "description": "Volume imported from existingLV",
        "volume_type": null,
        "bootable": true,
        "metadata": {
            "key1": "value1",
            "key2": "value2"
        }
    }
}
```

1.15.1.2. Response

Example 1.122. Show image metadata for volume: JSON response

The response is the same as for a volume create API call. The size of the volume is chosen by inspecting the size of the existing storage volume and rounding up to the next gibibyte (GiB).

```
{
    "volume": {
        "status": "creating",
        "user_id": "eae1472b5fc5496998a3d06550929e7e",
        "attachments": [],
        "links": [
```

```
            {
                "href": "http://10.0.2.15:8776/v2/
87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-
c781-4cd4-847d-5f2ec8cbd91c",
                "rel": "self"
            },
            {
                "href": "http://10.0.2.15:8776/
87c8522052ca4eed98bc672b4c1a3ddb/volumes/23cf872b-
c781-4cd4-847d-5f2ec8cbd91c",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "az2",
        "bootable": "false",
        "encrypted": "false",
        "created_at": "2014-07-18T00:12:54.000000",
        "description": "Volume imported from existingLV",
        "os-vol-tenant-attr:tenant_id": "87c8522052ca4eed98bc672b4c1a3ddb",
        "volume_type": null,
        "name": "New Volume",
        "source_volid": null,
        "snapshot_id": null,
        "metadata": {
            "key2": "value2",
            "key1": "value1"
        },
        "id": "23cf872b-c781-4cd4-847d-5f2ec8cbd91c",
        "size": 0
    }
}
```


1.16. Back-end storage pools

Administrator only. Lists all back-end storage pools that are known to the scheduler service.

Method	URI	Description
GET	`/v2/{tenant_id}/scheduler-stats/get_pools{?detail}`	Lists all back-end storage pools.

1.16.1. List back-end storage pools

Method	URI	Description
GET	`/v2/{tenant_id}/scheduler-stats/get_pools{?detail}`	Lists all back-end storage pools.

Normal response codes: 200

1.16.1.1. Request

This table shows the URI parameters for the list back-end storage pools request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list back-end storage pools request:

Name	Type	Description
`detail`	Boolean *(Optional)*	Indicates whether to show pool details or only pool names in the response. Set to `true` to show pool details. Set to `false` to show only pool names. Default is `false`.

This operation does not accept a request body.

1.16.1.2. Response

Example 1.123. List back-end storage pools: JSON response

```
{
    "pools": [
        {
            "name": "pool1",
            "capabilities": {
                "updated": "2014-10-28T00:00:00-00:00",
                "total_capacity": 1024,
                "free_capacity": 100,
                "volume_backend_name": "pool1",
                "reserved_percentage": 0,
                "driver_version": "1.0.0",
                "storage_protocol": "iSCSI",
                "QoS_support": false
            }
        },
        {
            "name": "pool2",
            "capabilities": {
                "updated": "2014-10-28T00:00:00-00:00",
                "total_capacity": 512,
                "free_capacity": 200,
                "volume_backend_name": "pool2",
                "reserved_percentage": 0,
                "driver_version": "1.0.1",
                "storage_protocol": "iSER",
```

```
                "QoS_support": true
            }
        }
    ]
}
```

1.17. Volume transfer

Transfers a volume from one user to another user.

Method	URI	Description
POST	`/v2/{tenant_id}/os-volume-transfer`	Creates a volume transfer.
GET	`/v2/{tenant_id}/os-volume-transfer`	Lists volume transfers.
GET	`/v2/{tenant_id}/os-volume-transfer/detail`	Lists volume transfers, with details.
GET	`/v2/{tenant_id}/os-volume-transfer/{transfer_id}`	Shows details for a volume transfer.
DELETE	`/v2/{tenant_id}/os-volume-transfer/{transfer_id}`	Deletes a volume transfer.
POST	`/v2/{tenant_id}/os-volume-transfer/{transfer_id}/accept`	Accepts a volume transfer.

1.17.1. Create volume transfer

Method	URI	Description
POST	`/v2/{tenant_id}/os-volume-transfer`	Creates a volume transfer.

Normal response codes: 202

1.17.1.1. Request

This table shows the URI parameters for the create volume transfer request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.124. Create volume transfer: JSON request

```
{
    "transfer": {
        "volume_id": "c86b9af4-151d-4ead-b62c-5fb967af0e37",
        "name": "first volume"
    }
}
```

1.17.1.2. Response

Example 1.125. Create volume transfer: JSON response

```
{
    "transfer": {
        "id": "1a7059f5-8ed7-45b7-8d05-2811e5d09f24",
        "created_at": "2015-02-25T03:56:53.081642",
        "name": "first volume",
        "volume_id": "c86b9af4-151d-4ead-b62c-5fb967af0e37",
        "auth_key": "9266c59563c84664",
        "links": [
            {
                "href": "http://localhost/v2/firstproject/volumes/3",
                "rel": "self"
            },
            {
                "href": "http://localhost/firstproject/volumes/3",
                "rel": "bookmark"
            }
        ]
    }
}
```

1.17.2. List volume transfers

Method	URI	Description
GET	`/v2/{tenant_id}/os-volume-transfer`	Lists volume transfers.

Normal response codes: 200

1.17.2.1. Request

This table shows the URI parameters for the list volume transfers request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.17.2.2. Response

Example 1.126. List volume transfers: JSON response

```
{
    "transfers": [
        {
            "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
            "name": "first volume transfer",
            "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
            "links": [
                {
                    "href": "http://localhost/v2/firstproject/volumes/1",
                    "rel": "self"
                },
                {
                    "href": "http://localhost/firstproject/volumes/1",
                    "rel": "bookmark"
                }
            ]
        },
        {
            "id": "f26c0dee-d20d-4e80-8dee-a8d91b9742a1",
            "name": "second volume transfer",
            "volume_id": "673db275-379f-41af-8371-e1652132b4c1",
            "links": [
                {
                    "href": "http://localhost/v2/firstproject/volumes/2",
                    "rel": "self"
                },
                {
                    "href": "http://localhost/firstproject/volumes/2",
                    "rel": "bookmark"
                }
            ]
        }
    ]
}
```

1.17.3. List volume transfers, with details

Method	URI	Description
GET	`/v2/{tenant_id}/os-volume-transfer/detail`	Lists volume transfers, with details.

Normal response codes: 200

1.17.3.1. Request

This table shows the URI parameters for the list volume transfers, with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.17.3.2. Response

Example 1.127. List volume transfers, with details: JSON response

```
{
    "transfers": [
        {
            "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
            "created_at": "2015-02-25T03:56:53.081642",
            "name": "first volume transfer",
            "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
            "links": [
                {
                    "href": "http://localhost/v2/firstproject/volumes/1",
                    "rel": "self"
                },
                {
                    "href": "http://localhost/firstproject/volumes/1",
                    "rel": "bookmark"
                }
            ]
        },
        {
            "id": "f26c0dee-d20d-4e80-8dee-a8d91b9742a1",
            "created_at": "2015-03-25T03:56:53.081642",
            "name": "second volume transfer",
            "volume_id": "673db275-379f-41af-8371-e1652132b4c1",
            "links": [
                {
                    "href": "http://localhost/v2/firstproject/volumes/2",
                    "rel": "self"
                },
                {
                    "href": "http://localhost/firstproject/volumes/2",
                    "rel": "bookmark"
                }
            ]
        }
```

```
    ]
}
```

1.17.4. Show volume transfer details

Method	URI	Description
GET	`/v2/{tenant_id}/os-volume-transfer/{transfer_id}`	Shows details for a volume transfer.

Normal response codes: 200

1.17.4.1. Request

This table shows the URI parameters for the show volume transfer details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{transfer_id}`	UUID	The unique identifier for a volume transfer.

This operation does not accept a request body.

1.17.4.2. Response

Example 1.128. Show volume transfer details: JSON response

```
{
    "transfer": {
        "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
        "created_at": "2015-02-25T03:56:53.081642",
        "name": "first volume transfer",
        "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
        "links": [
            {
                "href": "http://localhost/v2/firstproject/volumes/1",
                "rel": "self"
            },
            {
                "href": "http://localhost/firstproject/volumes/1",
                "rel": "bookmark"
            }
        ]
    }
}
```

1.17.5. Delete volume transfer

Method	URI	Description
DELETE	`/v2/{tenant_id}/os-volume-transfer/{transfer_id}`	Deletes a volume transfer.

Normal response codes: 202

1.17.5.1. Request

This table shows the URI parameters for the delete volume transfer request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{transfer_id}`	UUID	The unique identifier for a volume transfer.

This operation does not accept a request body.

1.17.6. Accept volume transfer

Method	URI	Description
POST	`/v2/{tenant_id}/os-volume-transfer/{transfer_id}/accept`	Accepts a volume transfer.

Normal response codes: 202

1.17.6.1. Request

This table shows the URI parameters for the accept volume transfer request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{transfer_id}`	UUID	The unique identifier for a volume transfer.

Example 1.129. Accept volume transfer: JSON request

```
{
    "accept": {
        "auth_key": "9266c59563c84664"
    }
}
```

1.17.6.2. Response

Example 1.130. Accept volume transfer: JSON response

```
{
    "transfer": {
        "id": "cac5c677-73a9-4288-bb9c-b2ebfb547377",
        "name": "first volume transfer",
        "volume_id": "894623a6-e901-4312-aa06-4275e6321cce",
        "links": [
            {
                "href": "http://localhost/v2/firstproject/volumes/1",
                "rel": "self"
            },
            {
                "href": "http://localhost/firstproject/volumes/1",
                "rel": "bookmark"
            }
        ]
    }
}
```

1.18. Consistency groups

Consistency groups enable you to create snapshots at the exact same point in time from multiple volumes. For example, a database might place its tables, logs, and configuration on separate volumes. To restore this database from a previous point in time, it makes sense to restore the logs, tables, and configuration together from the exact same point in time.

Use the `policy.json` file to grant permissions for these actions to limit roles.

Method	URI	Description
GET	`/v2/{tenant_id}/consistencygroups{?sort_key,sort_dir,limit,marker}`	Lists consistency groups.
POST	`/v2/{tenant_id}/consistencygroups`	Creates a consistency group.
GET	`/v2/{tenant_id}/consistency-groups/detail{?sort_key,sort_dir,limit,marker}`	Lists consistency groups with details.
POST	`/v2/{tenant_id}/consistency-groups/create_from_src`	Creates a consistency group from source.
GET	`/v2/{tenant_id}/consistency-groups/{consistencygroup_id}`	Shows details for a consistency group.
POST	`/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/delete`	Deletes a consistency group.
PUT	`/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/update`	Updates a consistency group.

1.18.1. List consistency groups

Method	URI	Description
GET	`/v2/{tenant_id}/consistencygroups{?sort_key,sort_dir,limit,marker}`	Lists consistency groups.

Normal response codes: 200

1.18.1.1. Request

This table shows the URI parameters for the list consistency groups request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list consistency groups request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an image attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.18.1.2. Response

Example 1.131. List consistency groups: JSON response

```
{
    "consistencygroups": [
        {
            "id": "6f519a48-3183-46cf-a32f-41815f813986",
            "name": "my-cg1"
        },
        {
            "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
            "name": "my-cg2"
        }
    ]
}
```

1.18.2. Create consistency group

Method	URI	Description
POST	`/v2/{tenant_id}/consistencygroups`	Creates a consistency group.

Normal response codes: 202

1.18.2.1. Request

This table shows the URI parameters for the create consistency group request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.132. Create consistency group: JSON request

```
{
    "consistencygroup": {
        "name": "firstcg",
        "description": "first consistency group",
        "volume_types": [
            "type1",
            "type2"
        ],
        "user_id": "6f519a48-3183-46cf-a32f-41815f814546",
        "project_id": "6f519a48-3183-46cf-a32f-41815f815555",
        "availability_zone": "az0",
        "status": "creating"
    }
}
```

1.18.2.2. Response

Example 1.133. Create consistency group: JSON response

```
{
    "consistencygroup": {
        "id": "6f519a48-3183-46cf-a32f-41815f816666",
        "name": "firstcg"
    }
}
```

1.18.3. List consistency groups with details

Method	URI	Description
GET	`/v2/{tenant_id}/consistency-groups/detail{?sort_key,sort_dir,limit,marker}`	Lists consistency groups with details.

Normal response codes: 200

1.18.3.1. Request

This table shows the URI parameters for the list consistency groups with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list consistency groups with details request:

Name	Type	Description
`sort_key`	String *(Optional)*	Sorts by an image attribute. A valid value is `name`, `status`, `container_format`, `disk_format`, `size`, `id`, `created_at`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` attribute value.
`sort_dir`	String *(Optional)*	Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, default is `desc`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

1.18.3.2. Response

Example 1.134. List consistency groups with details: JSON response

```
{
    "consistencygroups": [
        {
            "id": "6f519a48-3183-46cf-a32f-41815f813986",
            "status": "available",
            "availability_zone": "az1",
            "created_at": "2015-09-16T09:28:52.000000",
            "name": "my-cg1",
            "description": "my first consistency group",
            "volume_types": [
                "123456"
            ]
        },
        {
```

```
            "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
            "status": "error",
            "availability_zone": "az2",
            "created_at": "2015-09-16T09:31:15.000000",
            "name": "my-cg2",
            "description": "Edited description",
            "volume_types": [
                "234567"
            ]
        }
    ]
}
```

This table shows the body parameters for the list consistency groups with details response:

Name	Type	Description
`id`	UUID *(Required)*	The UUID of the consistency group.
`status`	String *(Required)*	The consistency group status. A valid value is `creating`, `available`, `error`, `deleting`, `updating`, or `invalid`.
`availability_zone`	String *(Required)*	The availability zone name.
`created_at`	DateTime *(Required)*	The date and time when the resource was created. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` For example, `2015-08-27T09:49:58-05:00`. The `±hh:mm` value, if included, is the time zone as an offset from UTC.
`name`	String *(Required)*	The consistency group name.
`description`	String *(Required)*	The consistency group description.
`volume_types`	List *(Required)*	The list of volume type IDs.

Example 1.135. List consistency groups with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<consistencygroups xmlns:consistencygroups="http://docs.openstack.org/volume/
ext/consistencygroups/api/v1">
    <consistencygroup
        id="6f519a48-3183-46cf-a32f-41815f813986"
        status="available"
        availability_zone="az1"
        created_at="2015-09-16T09:28:52.000000"
        name="my-cg1"
        description="my first consistency group">
        <volume_types>
            <volume_type>"123456"</volume_type>
        </volume_types>
    </consistencygroup>
    <consistencygroup
```

```
        id="aed36625-a6d7-4681-ba59-c7ba3d18c148"
        status="error"
        availability_zone="az2"
        created_at="2015-09-16T09:31:15.000000"
        name="my-cg2"
        description="Edited description">
        <volume_types>
            <volume_type>"234567"</volume_type>
        </volume_types>
    </consistencygroup>
</consistencygroups>
```

This operation does not return a response body.

1.18.4. Create consistency group from source

Method	URI	Description
POST	`/v2/{tenant_id}/consistency-groups/create_from_src`	Creates a consistency group from source.

Normal response codes: 202

1.18.4.1. Request

This table shows the URI parameters for the create consistency group from source request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.136. Create consistency group from source: JSON request

```
{
    "consistencygroup-from-src": {
        "name": "firstcg",
        "description": "first consistency group",
        "cgsnapshot_id": "6f519a48-3183-46cf-a32f-41815f813986",
        "source_cgid": "6f519a48-3183-46cf-a32f-41815f814546",
        "user_id": "6f519a48-3183-46cf-a32f-41815f815555",
        "project_id": "6f519a48-3183-46cf-a32f-41815f814444",
        "status": "creating"
    }
}
```

1.18.4.2. Response

Example 1.137. Create consistency group from source: JSON response

```
{
    "consistencygroup": {
        "id": "6f519a48-3183-46cf-a32f-41815f816666",
        "name": "firstcg"
    }
}
```

1.18.5. Show consistency group details

Method	URI	Description
GET	`/v2/{tenant_id}/consistency-groups/{consistencygroup_id}`	Shows details for a consistency group.

Normal response codes: 200

1.18.5.1. Request

This table shows the URI parameters for the show consistency group details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{consistencygroup_id}`	String	The ID of the consistency group.

This operation does not accept a request body.

1.18.5.2. Response

Example 1.138. Show consistency group details: JSON response

```
{
    "consistencygroup": {
        "id": "6f519a48-3183-46cf-a32f-41815f813986",
        "status": "available",
        "availability_zone": "az1",
        "created_at": "2015-09-16T09:28:52.000000",
        "name": "my-cg1",
        "description": "my first consistency group",
        "volume_types": [
            "123456"
        ]
    }
}
```

Example 1.139. Show consistency group details: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<consistencygroups xmlns:consistencygroups="http://docs.openstack.org/volume/
ext/consistencygroups/api/v1">
    <consistencygroup
        id="6f519a48-3183-46cf-a32f-41815f813986"
        status="available"
        availability_zone="az1"
        created_at="2015-09-16T09:28:52.000000"
        name="my-cg1"
        description="my first consistency group">
        <volume_types>
            <volume_type>"123456"</volume_type>
        </volume_types>
    </consistencygroup>
</consistencygroups>
```

This operation does not return a response body.

1.18.6. Delete consistency group

Method	URI	Description
POST	`/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/delete`	Deletes a consistency group.

Normal response codes: 202

1.18.6.1. Request

This table shows the URI parameters for the delete consistency group request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{consistencygroup_id}`	String	The ID of the consistency group.

Example 1.140. Delete consistency group: JSON request

```
{
    "consistencygroup": {
        "force": false
    }
}
```

1.18.7. Update consistency group

Method	URI	Description
PUT	`/v2/{tenant_id}/consistency-groups/{consistencygroup_id}/update`	Updates a consistency group.

Normal response codes: 202

1.18.7.1. Request

This table shows the URI parameters for the update consistency group request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{consistencygroup_id}`	String	The ID of the consistency group.

This table shows the body parameters for the update consistency group request:

Name	Type	Description
`name`	String *(Optional)*	The consistency group name.
`description`	String *(Optional)*	The consistency group description.
`add_volumes`	String *(Optional)*	One or more volume UUIDs, separated by commas, to add to the volume consistency group.
`remove_volumes`	String *(Optional)*	One or more volume UUIDs, separated by commas, to remove from the volume consistency group.

Example 1.141. Update consistency group: XML request

```
{
    "consistencygroup": {
        "name": "my_cg",
        "description": "My consistency group",
        "add_volumes": "volume-uuid-1,volume-uuid-2",
        "remove_volumes": "volume-uuid-8,volume-uuid-9"
    }
}
```

This operation does not accept a request body.

1.19. Consistency group snapshots

Lists all, lists all with details, shows details for, creates, and deletes consistency group snapshots.

Method	URI	Description
GET	`/v2/{tenant_id}/cgsnapshots`	Lists all consistency group snapshots.

Method	URI	Description
POST	`/v2/{tenant_id}/cgsnapshots`	Creates a consistency group snapshot.
GET	`/v2/{tenant_id}/cgsnapshots/detail`	Lists all consistency group snapshots with details.
GET	`/v2/{tenant_id}/cgsnap-shots/{cgsnapshot_id}`	Shows details for a consistency group snapshot.
DELETE	`/v2/{tenant_id}/cgsnap-shots/{cgsnapshot_id}`	Deletes a consistency group snapshot.

1.19.1. List consistency group snapshots

Method	URI	Description
GET	`/v2/{tenant_id}/cgsnapshots`	Lists all consistency group snapshots.

Normal response codes: 200

1.19.1.1. Request

This table shows the URI parameters for the list consistency group snapshots request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.19.1.2. Response

Example 1.142. List consistency group snapshots: JSON response

```
{
    "cgsnapshots": [
        {
            "id": "6f519a48-3183-46cf-a32f-41815f813986",
            "name": "my-cg1"
        },
        {
            "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
            "name": "my-cg2"
        }
    ]
}
```

1.19.2. Create consistency group snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/cgsnapshots`	Creates a consistency group snapshot.

Normal response codes: 202

1.19.2.1. Request

This table shows the URI parameters for the create consistency group snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 1.143. Create consistency group snapshot: JSON request

```
{
    "cgsnapshot": {
        "consistencygroup_id": "6f519a48-3183-46cf-a32f-41815f814546",
        "name": "firstcg",
        "description": "first consistency group",
        "user_id": "6f519a48-3183-46cf-a32f-41815f814444",
        "project_id": "6f519a48-3183-46cf-a32f-41815f815555",
        "status": "creating"
    }
}
```

This operation does not accept a request body.

1.19.2.2. Response

Example 1.144. Create consistency group snapshot: JSON response

```
{
    "cgsnapshot": {
        "id": "6f519a48-3183-46cf-a32f-41815f816666",
        "name": "firstcg"
    }
}
```

This operation does not return a response body.

1.19.3. List consistency group snapshots with details

Method	URI	Description
GET	`/v2/{tenant_id}/cgsnapshots/detail`	Lists all consistency group snapshots with details.

Normal response codes: 200

1.19.3.1. Request

This table shows the URI parameters for the list consistency group snapshots with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

1.19.3.2. Response

Example 1.145. List consistency group snapshots with details: JSON response

```
{
    "cgsnapshots": [
        {
            "id": "6f519a48-3183-46cf-a32f-41815f813986",
            "consistencygroup_id": "6f519a48-3183-46cf-a32f-41815f814444",
            "status": "available",
            "created_at": "2015-09-16T09:28:52.000000",
            "name": "my-cg1",
            "description": "my first consistency group"
        },
        {
            "id": "aed36625-a6d7-4681-ba59-c7ba3d18c148",
            "consistencygroup_id": "aed36625-a6d7-4681-ba59-c7ba3d18dddd",
            "status": "error",
            "created_at": "2015-09-16T09:31:15.000000",
            "name": "my-cg2",
            "description": "Edited description"
        }
    ]
}
```

This operation does not return a response body.

1.19.4. Show consistency group snapshot details

Method	URI	Description
GET	`/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}`	Shows details for a consistency group snapshot.

Normal response codes: 200

1.19.4.1. Request

This table shows the URI parameters for the show consistency group snapshot details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{cgsnapshot_id}`	String	The ID of the consistency group snapshot.

This operation does not accept a request body.

1.19.4.2. Response

Example 1.146. Show consistency group snapshot details: JSON response

```
{
    "cgsnapshot": {
        "id": "6f519a48-3183-46cf-a32f-41815f813986",
        "consistencygroup_id": "6f519a48-3183-46cf-a32f-41815f814444",
        "status": "available",
        "created_at": "2015-09-16T09:28:52.000000",
        "name": "my-cg1",
        "description": "my first consistency group"
    }
}
```

This operation does not return a response body.

1.19.5. Delete consistency group snapshot

Method	URI	Description
DELETE	`/v2/{tenant_id}/cgsnapshots/{cgsnapshot_id}`	Deletes a consistency group snapshot.

Normal response codes: 202

1.19.5.1. Request

This table shows the URI parameters for the delete consistency group snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{cgsnapshot_id}`	String	The ID of the consistency group snapshot.

This operation does not accept a request body.

2. Block Storage API v1 (DEPRECATED)

Block Storage API v1 is deprecated in Kilo.

Manages volumes and snapshots for use with the Block Storage API, also known as cinder services.

When you create, list, or delete volumes, the following status values are possible:

- CREATING. The volume is being created.
- AVAILABLE. The volume is ready to attach to an instance.
- ATTACHING. The volume is attaching to an instance.
- IN-USE. The volume is attached to an instance.
- DELETING. The volume is being deleted.
- ERROR. An error occurred.
- ERROR_DELETING. A volume deletion error occurred.

Method	URI	Description
API versions		
GET	`/`	Lists information about all Block Storage API versions.
GET	`/v1`	Shows Block Storage API v1 details.
Volumes		
POST	`/v1/{tenant_id}/volumes`	Creates a volume.
GET	`/v1/{tenant_id}/volumes`	Lists all volumes.
GET	`/v1/{tenant_id}/volumes/detail`	Lists all volumes, with details.
GET	`/v1/{tenant_id}/volumes/{volume_id}`	Shows details for a volume.
DELETE	`/v1/{tenant_id}/volumes/{volume_id}`	Deletes a volume.
Volume types		
GET	`/v1/{tenant_id}/types`	Lists volume types.
POST	`/v1/{tenant_id}/types`	Creates a volume type.
PUT	`/v1/{tenant_id}/types/{volume_type_id}`	Updates a volume type.
PUT	`/v1/{tenant_id}/types/{volume_type_id}`	Updates the extra specifications for a volume type.
GET	`/v1/{tenant_id}/types/{volume_type_id}`	Shows details for a volume type.
DELETE	`/v1/{tenant_id}/types/{volume_type_id}`	Deletes a volume type.
Snapshots		
POST	`/v1/{tenant_id}/snapshots`	Creates a snapshot.
GET	`/v1/{tenant_id}/snapshots`	Lists all snapshots.
GET	`/v1/{tenant_id}/snapshots/detail`	Lists all snapshots, with details.
GET	`/v1/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.
DELETE	`/v1/{tenant_id}/snapshots/{snapshot_id}`	Deletes a snapshot.
GET	`/v1/{tenant_id}/snapshots/{snapshot_id}/metadata`	Shows metadata for a snapshot.

Method	URI	Description
PUT	`/v1/{tenant_id}/snapshots/{snapshot_id}/metadata`	Updates metadata for a snapshot.
Quota sets extension (os-quota-sets)		
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}{?usage}`	Shows quotas for a tenant.
PUT	`/v1/{tenant_id}/os-quota-sets/{tenant_id}`	Updates quotas for a tenant.
DELETE	`/v1/{tenant_id}/os-quota-sets/{tenant_id}`	Deletes quotas for a tenant so the quotas revert to default values.
GET	`/v1/{tenant_id}/os-quota-sets/defaults`	Shows default quotas for a tenant.
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Enables an admin user to show quotas for a tenant and user.
POST	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Updates quotas for a tenant and user.
DELETE	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Deletes quotas for a user so that the quotas revert to default values.
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}`	Shows details for quotas for a tenant and user.

2.1. API versions

Lists information about API versions.

Method	URI	Description
GET	`/`	Lists information about all Block Storage API versions.
GET	`/v1`	Shows Block Storage API v1 details.

2.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Block Storage API versions.

Normal response codes: 200300

2.1.1.1. Request

This operation does not accept a request body.

2.1.1.2. Response

Example 2.1. List API versions: JSON response

```
{
    "versions": [
        {
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.211.234:8776/v1/",
                    "rel": "self"
                }
            ],
            "status": "DEPRECATED",
            "updated": "2014-06-28T12:20:21Z"
        },
        {
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.211.234:8776/v2/",
                    "rel": "self"
                }
            ],
            "status": "CURRENT",
            "updated": "2012-11-21T11:33:21Z"
        }
    ]
}
```

2.1.2. Show API v1 details

Method	URI	Description
GET	/v1	Shows Block Storage API v1 details.

Normal response codes: 200203

2.1.2.1. Request

This operation does not accept a request body.

2.1.2.2. Response

Example 2.2. Show API v1 details: JSON response

```
{
    "version": {
        "id": "v1.0",
        "links": [
            {
                "href": "http://23.253.211.234:8776/v1/",
                "rel": "self"
            },
            {
                "href": "http://docs.openstack.org/",
                "rel": "describedby",
                "type": "text/html"
            }
        ],
        "media-types": [
            {
                "base": "application/xml",
                "type": "application/vnd.openstack.volume+xml;version=1"
            },
            {
                "base": "application/json",
                "type": "application/vnd.openstack.volume+json;version=1"
            }
        ],
        "status": "DEPRECATED",
        "updated": "2014-06-28T12:20:21Z"
    }
}
```

2.2. Volumes

The `snapshot_id` and `source_volid` parameters specify the ID of the snapshot or volume from which the volume originates. If the volume was not created from a snapshot or source volume, these values are null.

Method	URI	Description
POST	/v1/{tenant_id}/volumes	Creates a volume.
GET	/v1/{tenant_id}/volumes	Lists all volumes.

Method	URI	Description
GET	`/v1/{tenant_id}/volumes/detail`	Lists all volumes, with details.
GET	`/v1/{tenant_id}/vol-umes/{volume_id}`	Shows details for a volume.
DELETE	`/v1/{tenant_id}/vol-umes/{volume_id}`	Deletes a volume.

2.2.1. Create volume

Method	URI	Description
POST	`/v1/{tenant_id}/volumes`	Creates a volume.

Normal response codes: 201

2.2.1.1. Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 2.3. Create volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/volume/api/v1"
        display_name="vol-001"
        display_description="Another volume."
        size="30"
        volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
        availability_zone="us-east1">
    <metadata>
        <meta key="contents">junk</meta>
    </metadata>
</volume>
```

Example 2.4. Create volume: JSON request

```
{
    "volume": {
        "display_name": "vol-001",
        "display_description": "Another volume.",
        "size": 30,
        "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "metadata": {
            "contents": "junk"
        },
        "availability_zone": "us-east1"
    }
}
```

This operation does not accept a request body.

2.2.1.2. Response

Example 2.5. Create volume: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/volume/api/v1"
        id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
        display_name="vol-001"
        display_description="Another volume."
        status="active"
```

```
        size="30"
        volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
        availability_zone="us-east1"
        bootable="false"
        created_at="2012-02-14T20:53:07Z">
    <metadata>
        <meta key="contents">junk</meta>
    </metadata>
</volume>
```

Example 2.6. Create volume: JSON response

```
{
    "volume": {
        "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "display_name": "vol-001",
        "display_description": "Another volume.",
        "status": "active",
        "size": 30,
        "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "metadata": {
            "contents": "junk"
        },
        "availability_zone": "us-east1",
        "bootable": "false",
        "snapshot_id": null,
        "attachments": [
            {
                "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
                "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
                "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
                "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
                "host_name": "mitaka",
                "device": "/"
            }
        ],
        "created_at": "2012-02-14T20:53:07Z"
    }
}
```

This operation does not return a response body.

2.2.2. List volumes

Method	URI	Description
GET	/v1/{tenant_id}/volumes	Lists all volumes.

Normal response codes: 200

2.2.2.1. Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.2.2.2. Response

Example 2.7. List volumes: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volumes xmlns="http://docs.openstack.org/volume/api/v1">
    <volume xmlns="http://docs.openstack.org/volume/api/v1"
            id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
            display_name="vol-001"
            display_description="Another volume."
            status="active"
            size="30"
            volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
            availability_zone="us-east1"
            created_at="2012-02-14T20:53:07Z">
        <metadata>
            <meta key="contents">junk</meta>
        </metadata>
    </volume>
    <volume xmlns="http://docs.openstack.org/volume/api/v1"
            id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
            display_name="vol-002"
            display_description="Yet another volume."
            status="active"
            size="25"
            volume_type="96c3bda7-c82a-4f50-be73-ca7621794835"
            availability_zone="us-east2"
            created_at="2012-03-15T19:10:03Z" />
</volumes>
```

Example 2.8. List volumes: JSON response

```
{
    "volumes": [
        {
            "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
            "display_name": "vol-001",
            "display_description": "Another volume.",
            "status": "active",
```

```
            "size": 30,
            "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
            "metadata": {
                "contents": "junk"
            },
            "availability_zone": "us-east1",
            "snapshot_id": null,
            "attachments": [
                {
                    "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
                    "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
                    "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
                    "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
                    "host_name": "mitaka",
                    "device": "/"
                }
            ],
            "created_at": "2012-02-14T20:53:07Z"
        },
        {
            "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
            "display_name": "vol-002",
            "display_description": "Yet another volume.",
            "status": "active",
            "size": 25,
            "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
            "metadata": {},
            "availability_zone": "us-east2",
            "snapshot_id": null,
            "attachments": [],
            "created_at": "2012-03-15T19:10:03Z"
        }
    ]
}
```


This operation does not return a response body.

2.2.3. List volumes, with details

Method	URI	Description
GET	`/v1/{tenant_id}/volumes/detail`	Lists all volumes, with details.

Normal response codes: 200

2.2.3.1. Request

This table shows the URI parameters for the list volumes, with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.2.3.2. Response

Example 2.9. List volumes, with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volumes xmlns="http://docs.openstack.org/volume/api/v1">
    <volume xmlns="http://docs.openstack.org/volume/api/v1"
            id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
            display_name="vol-001"
            display_description="Another volume."
            status="active"
            size="30"
            volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
            availability_zone="us-east1"
            created_at="2012-02-14T20:53:07Z">
        <metadata>
            <meta key="contents">junk</meta>
        </metadata>
    </volume>
    <volume xmlns="http://docs.openstack.org/volume/api/v1"
            id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
            display_name="vol-002"
            display_description="Yet another volume."
            status="active"
            size="25"
            volume_type="96c3bda7-c82a-4f50-be73-ca7621794835"
            availability_zone="us-east2"
            created_at="2012-03-15T19:10:03Z" />
</volumes>
```

Example 2.10. List volumes, with details: JSON response

```
{
    "volumes": [
        {
            "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
            "display_name": "vol-001",
            "display_description": "Another volume.",
            "status": "active",
```

```
            "size": 30,
            "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
            "metadata": {
                "contents": "junk"
            },
            "availability_zone": "us-east1",
            "snapshot_id": null,
            "attachments": [
                {
                    "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
                    "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
                    "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
                    "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
                    "host_name": "mitaka",
                    "device": "/"
                }
            ],
            "created_at": "2012-02-14T20:53:07Z"
        },
        {
            "id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
            "display_name": "vol-002",
            "display_description": "Yet another volume.",
            "status": "active",
            "size": 25,
            "volume_type": "96c3bda7-c82a-4f50-be73-ca7621794835",
            "metadata": {},
            "availability_zone": "us-east2",
            "snapshot_id": null,
            "attachments": [],
            "created_at": "2012-03-15T19:10:03Z"
        }
    ]
}
```


This operation does not return a response body.

2.2.4. Show volume details

Method	URI	Description
GET	`/v1/{tenant_id}/volumes/{volume_id}`	Shows details for a volume.

Normal response codes: 200

2.2.4.1. Request

This table shows the URI parameters for the show volume details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

This operation does not accept a request body.

2.2.4.2. Response

Example 2.11. Show volume details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume xmlns="http://docs.openstack.org/volume/api/v1"
        id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
        display_name="vol-001"
        display_description="Another volume."
        status="active"
        size="30"
        volume_type="289da7f8-6440-407c-9fb4-7db01ec49164"
        availability_zone="us-east1"
        bootable="false"
        created_at="2012-02-14T20:53:07Z">
    <metadata>
        <meta key="contents">junk</meta>
    </metadata>
</volume>
```

Example 2.12. Show volume details: JSON response

```
{
    "volume": {
        "id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "display_name": "vol-001",
        "display_description": "Another volume.",
        "status": "active",
        "size": 30,
        "volume_type": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "metadata": {
            "contents": "junk"
        },
        "availability_zone": "us-east1",
        "bootable": "false",
        "snapshot_id": null,
        "attachments": [
```

```
            {
                "attachment_id": "03987cd1-0ad5-40d1-9b2a-7cc48295d4fa",
                "id": "47e9ecc5-4045-4ee3-9a4b-d859d546a0cf",
                "volume_id": "6c80f8ac-e3e2-480c-8e6e-f1db92fe4bfe",
                "server_id": "d1c4788b-9435-42e2-9b81-29f3be1cd01f",
                "host_name": "mitaka",
                "device": "/"
            }
        ],
        "created_at": "2012-02-14T20:53:07Z"
    }
}
```

This operation does not return a response body.

2.2.5. Delete volume

Method	URI	Description
DELETE	`/v1/{tenant_id}/volumes/{volume_id}`	Deletes a volume.

Normal response codes: 202

2.2.5.1. Request

This table shows the URI parameters for the delete volume request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_id}`	UUID	The UUID of the volume.

This operation does not accept a request body.

2.3. Volume types

Lists, creates, updates, shows information for, and deletes volume types.

Method	URI	Description
GET	`/v1/{tenant_id}/types`	Lists volume types.
POST	`/v1/{tenant_id}/types`	Creates a volume type.
PUT	`/v1/{tenant_id}/types/{volume_type_id}`	Updates a volume type.
PUT	`/v1/{tenant_id}/types/{volume_type_id}`	Updates the extra specifications for a volume type.
GET	`/v1/{tenant_id}/types/{volume_type_id}`	Shows details for a volume type.
DELETE	`/v1/{tenant_id}/types/{volume_type_id}`	Deletes a volume type.

2.3.1. List volume types

Method	URI	Description
GET	/v1/{tenant_id}/types	Lists volume types.

Normal response codes: 200

2.3.1.1. Request

This table shows the URI parameters for the list volume types request:

Name	Type	Description
{tenant_id}	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.3.1.2. Response

Example 2.13. List volume types: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_types xmlns="http://docs.openstack.org/volume/api/v1">
    <volume_type id="289da7f8-6440-407c-9fb4-7db01ec49164"
                 name="vol-type-001">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
    </volume_type>
    <volume_type id="96c3bda7-c82a-4f50-be73-ca7621794835"
                 name="vol-type-002" />
</volume_types>
```

Example 2.14. List volume types: JSON response

```
{
    "volume_types": [
        {
            "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
            "name": "vol-type-001",
            "extra_specs": {
                "capabilities": "gpu"
            }
        },
        {
            "id": "96c3bda7-c82a-4f50-be73-ca7621794835",
            "name": "vol-type-002",
            "extra_specs": {}
        }
    ]
}
```

This operation does not return a response body.

2.3.2. Create volume type

Method	URI	Description
POST	`/v1/{tenant_id}/types`	Creates a volume type.

Normal response codes: 200

2.3.2.1. Request

This table shows the URI parameters for the create volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 2.15. Create volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
        name="vol-type-001">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
</volume_type>
```

Example 2.16. Create volume type: JSON request

```
{
    "volume_type": {
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not accept a request body.

2.3.2.2. Response

Example 2.17. Create volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
             id="289da7f8-6440-407c-9fb4-7db01ec49164"
             name="vol-type-001">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 2.18. Create volume type: JSON response

```
{
    "volume_type": {
```

```
        "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

2.3.3. Update volume type

Method	URI	Description
PUT	`/v1/{tenant_id}/types/{volume_type_id}`	Updates a volume type.

Normal response codes: 200

2.3.3.1. Request

This table shows the URI parameters for the update volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

Example 2.19. Update volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
        name="vol-type-001">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
</volume_type>
```

Example 2.20. Update volume type: JSON request

```
{
    "volume_type": {
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not accept a request body.

2.3.3.2. Response

Example 2.21. Update volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
             id="289da7f8-6440-407c-9fb4-7db01ec49164"
             name="vol-type-001">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 2.22. Update volume type: JSON response

```
{
```

```
    "volume_type": {
        "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

2.3.4. Update extra specs for a volume type

Method	URI	Description
PUT	`/v1/{tenant_id}/types/{volume_type_id}`	Updates the extra specifications for a volume type.

Normal response codes: 200

2.3.4.1. Request

This table shows the URI parameters for the update extra specs for a volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

Example 2.23. Update extra specs for a volume type: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
        name="vol-type-001">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
</volume_type>
```

Example 2.24. Update extra specs for a volume type: JSON request

```
{
    "volume_type": {
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not accept a request body.

2.3.4.2. Response

Example 2.25. Update extra specs for a volume type: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
             id="289da7f8-6440-407c-9fb4-7db01ec49164"
             name="vol-type-001">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 2.26. Update extra specs for a volume type: JSON response

```
{
```

```
    "volume_type": {
        "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

2.3.5. Show volume type details

Method	URI	Description
GET	`/v1/{tenant_id}/types/{volume_type_id}`	Shows details for a volume type.

Normal response codes: 200

2.3.5.1. Request

This table shows the URI parameters for the show volume type details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

2.3.5.2. Response

Example 2.27. Show volume type details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type xmlns="http://docs.openstack.org/volume/api/v1"
             id="289da7f8-6440-407c-9fb4-7db01ec49164"
             name="vol-type-001">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

Example 2.28. Show volume type details: JSON response

```
{
    "volume_type": {
        "id": "289da7f8-6440-407c-9fb4-7db01ec49164",
        "name": "vol-type-001",
        "extra_specs": {
            "capabilities": "gpu"
        }
    }
}
```

This operation does not return a response body.

2.3.6. Delete volume type

Method	URI	Description
DELETE	`/v1/{tenant_id}/types/{volume_type_id}`	Deletes a volume type.

Normal response codes: 202

2.3.6.1. Request

This table shows the URI parameters for the delete volume type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{volume_type_id}`	UUID	The UUID for an existing volume type.

This operation does not accept a request body.

2.4. Snapshots

Creates, lists, shows information for, and deletes snapshots. Shows and updates snapshot metadata.

Method	URI	Description
POST	`/v1/{tenant_id}/snapshots`	Creates a snapshot.
GET	`/v1/{tenant_id}/snapshots`	Lists all snapshots.
GET	`/v1/{tenant_id}/snapshots/detail`	Lists all snapshots, with details.
GET	`/v1/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.
DELETE	`/v1/{tenant_id}/snapshots/{snapshot_id}`	Deletes a snapshot.
GET	`/v1/{tenant_id}/snapshots/{snapshot_id}/metadata`	Shows metadata for a snapshot.
PUT	`/v1/{tenant_id}/snapshots/{snapshot_id}/metadata`	Updates metadata for a snapshot.

2.4.1. Create snapshot

Method	URI	Description
POST	`/v1/{tenant_id}/snapshots`	Creates a snapshot.

Normal response codes: 201

2.4.1.1. Request

This table shows the URI parameters for the create snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 2.29. Create snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot xmlns="http://docs.openstack.org/volume/api/v1"
          name="snap-001" display_name="snap-001"
          display_description="Daily backup"
          volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
          force="true"/>
```

Example 2.30. Create snapshot: JSON request

```
{
    "snapshot": {
        "display_name": "snap-001",
        "display_description": "Daily backup",
        "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "force": true
    }
}
```

This operation does not accept a request body.

2.4.1.2. Response

Example 2.31. Create snapshot: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot xmlns="http://docs.openstack.org/volume/api/v1"
          id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
          display_name="snap-001"
          display_description="Daily backup"
          volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
          status="available"
          size="30"
          created_at="2012-02-29T03:50:07Z" />
```

Example 2.32. Create snapshot: JSON response

```
{
    "snapshot": {
```

```
        "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
        "display_name": "snap-001",
        "display_description": "Daily backup",
        "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "status": "available",
        "size": 30,
        "created_at": "2012-02-29T03:50:07Z"
    }
}
```

This operation does not return a response body.

2.4.2. List snapshots

Method	URI	Description
GET	`/v1/{tenant_id}/snapshots`	Lists all snapshots.

Normal response codes: 200

2.4.2.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.4.2.2. Response

Example 2.33. List snapshots: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots xmlns="http://docs.openstack.org/volume/api/v1">
    <snapshot id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
              display_name="snap-001"
              display_description="Daily backup"
              volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
              status="available"
              size="30"
              created_at="2012-02-29T03:50:07Z">
        <metadata>
              <meta key="contents">junk</meta>
        </metadata>
    </snapshot>
    <snapshot id="e479997c-650b-40a4-9dfe-77655818b0d2"
              display_name="snap-002"
              display_description="Weekly backup"
              volume_id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
              status="available"
              size="25"
              created_at="2012-03-19T01:52:47Z" />
</snapshots>
```

Example 2.34. List snapshots: JSON response

```
{
    "snapshots": [
        {
            "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
            "display_name": "snap-001",
            "display_description": "Daily backup",
            "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
            "status": "available",
            "size": 30,
            "created_at": "2012-02-29T03:50:07Z",
            "metadata": {
```

```
                "contents": "junk"
            }
        },
        {
            "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
            "display_name": "snap-002",
            "display_description": "Weekly backup",
            "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
            "status": "available",
            "size": 25,
            "created_at": "2012-03-19T01:52:47Z",
            "metadata": {}
        }
    ]
}
```

This operation does not return a response body.

2.4.3. List snapshots with details

Method	URI	Description
GET	`/v1/{tenant_id}/snapshots/detail`	Lists all snapshots, with details.

Normal response codes: 200

2.4.3.1. Request

This table shows the URI parameters for the list snapshots with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.4.3.2. Response

Example 2.35. List snapshots with details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshots xmlns="http://docs.openstack.org/volume/api/v1">
    <snapshot id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
              display_name="snap-001"
              display_description="Daily backup"
              volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
              status="available"
              size="30"
              created_at="2012-02-29T03:50:07Z">
        <metadata>
              <meta key="contents">junk</meta>
        </metadata>
    </snapshot>
    <snapshot id="e479997c-650b-40a4-9dfe-77655818b0d2"
              display_name="snap-002"
              display_description="Weekly backup"
              volume_id="76b8950a-8594-4e5b-8dce-0dfa9c696358"
              status="available"
              size="25"
              created_at="2012-03-19T01:52:47Z" />
</snapshots>
```

Example 2.36. List snapshots with details: JSON response

```
{
    "snapshots": [
        {
            "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
            "display_name": "snap-001",
            "display_description": "Daily backup",
            "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
            "status": "available",
            "size": 30,
            "created_at": "2012-02-29T03:50:07Z",
            "metadata": {
```

```
                "contents": "junk"
            }
        },
        {
            "id": "e479997c-650b-40a4-9dfe-77655818b0d2",
            "display_name": "snap-002",
            "display_description": "Weekly backup",
            "volume_id": "76b8950a-8594-4e5b-8dce-0dfa9c696358",
            "status": "available",
            "size": 25,
            "created_at": "2012-03-19T01:52:47Z",
            "metadata": {}
        }
    ]
}
```

This operation does not return a response body.

2.4.4. Show snapshot details

Method	URI	Description
GET	`/v1/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.

Normal response codes: 200

2.4.4.1. Request

This table shows the URI parameters for the show snapshot details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

2.4.4.2. Response

Example 2.37. Show snapshot details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot xmlns="http://docs.openstack.org/volume/api/v1"
          id="3fbbcccf-d058-4502-8844-6feeffdf4cb5"
          display_name="snap-001"
          display_description="Daily backup"
          volume_id="521752a6-acf6-4b2d-bc7a-119f9148cd8c"
          status="available"
          size="30"
          created_at="2012-02-29T03:50:07Z" />
```

Example 2.38. Show snapshot details: JSON response

```
{
    "snapshot": {
        "id": "3fbbcccf-d058-4502-8844-6feeffdf4cb5",
        "display_name": "snap-001",
        "display_description": "Daily backup",
        "volume_id": "521752a6-acf6-4b2d-bc7a-119f9148cd8c",
        "status": "available",
        "size": 30,
        "created_at": "2012-02-29T03:50:07Z"
    }
}
```

This operation does not return a response body.

2.4.5. Delete snapshot

Method	URI	Description
DELETE	`/v1/{tenant_id}/snapshots/{snapshot_id}`	Deletes a snapshot.

Normal response codes: 202

2.4.5.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

2.4.6. Show snapshot metadata

Method	URI	Description
GET	`/v1/{tenant_id}/snapshots/{snapshot_id}/metadata`	Shows metadata for a snapshot.

Normal response codes: 200

2.4.6.1. Request

This table shows the URI parameters for the show snapshot metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

2.4.6.2. Response

Example 2.39. Show snapshot metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
    xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/volume/
ext/extended_snapshot_attributes/api/v1"
    status="available" description="None"
    created_at="2014-05-06 17:59:52"
    volume_id="ebd80b99-bc3d-4154-9d28-5583baa80580" size="10"
    id="dfcd17fe-3b64-44ba-b95f-1c9c7109ef95" name="my-snapshot"
    os-extended-snapshot-attributes:project_id=
"7e0105e19cd2466193729ef78b604f79"
    os-extended-snapshot-attributes:progress="0%">
    <metadata>
        <meta key="key">v1</meta>
    </metadata>
</snapshot>
```

Example 2.40. Show snapshot metadata: JSON response

```
{
    "snapshot": {
        "status": "available",
        "os-extended-snapshot-attributes:progress": "0%",
        "description": null,
        "created_at": "2014-05-06T17:59:52.000000",
        "metadata": {
            "key": "v1"
        },
        "volume_id": "ebd80b99-bc3d-4154-9d28-5583baa80580",
        "os-extended-snapshot-attributes:project_id":
 "7e0105e19cd2466193729ef78b604f79",
        "size": 10,
        "id": "dfcd17fe-3b64-44ba-b95f-1c9c7109ef95",
        "name": "my-snapshot"
```

```
    }
}
```

This operation does not return a response body.

2.4.7. Update snapshot metadata

Method	URI	Description
PUT	`/v1/{tenant_id}/snapshots/{snapshot_id}/metadata`	Updates metadata for a snapshot.

Normal response codes: 200

2.4.7.1. Request

This table shows the URI parameters for the update snapshot metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 2.41. Update snapshot metadata: XML request

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata>
    <meta key="key">v1</meta>
</metadata>
```

Example 2.42. Update snapshot metadata: JSON request

```
{
    "metadata": {
        "key": "v1"
    }
}
```

This operation does not accept a request body.

2.4.7.2. Response

Example 2.43. Update snapshot metadata: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<metadata xmlns="http://docs.openstack.org/compute/api/v1.1">
    <meta key="key">v1</meta>
</metadata>
```

Example 2.44. Update snapshot metadata: JSON response

```
{
    "metadata": {
        "key": "v1"
    }
}
```

This operation does not return a response body.

2.5. Quota sets extension (os-quota-sets)

Administrators only, depending on policy settings.

Shows, updates, and deletes quotas for a tenant.

Method	URI	Description
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}{?usage}`	Shows quotas for a tenant.
PUT	`/v1/{tenant_id}/os-quota-sets/{tenant_id}`	Updates quotas for a tenant.
DELETE	`/v1/{tenant_id}/os-quota-sets/{tenant_id}`	Deletes quotas for a tenant so the quotas revert to default values.
GET	`/v1/{tenant_id}/os-quota-sets/defaults`	Shows default quotas for a tenant.
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Enables an admin user to show quotas for a tenant and user.
POST	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Updates quotas for a tenant and user.
DELETE	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Deletes quotas for a user so that the quotas revert to default values.
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}`	Shows details for quotas for a tenant and user.

2.5.1. Show quotas

Method	URI	Description
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}{?usage}`	Shows quotas for a tenant.

Normal response codes: 200

2.5.1.1. Request

This table shows the URI parameters for the show quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.

This table shows the query parameters for the show quotas request:

Name	Type	Description
`usage`	Boolean *(Optional)*	Set to `usage=true` to show quota usage. Default is `false`.

This operation does not accept a request body.

2.5.1.2. Response

Example 2.45. Show quotas response: JSON

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "id": "fake_tenant",
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "injected_files": 5,
        "instances": 10,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}
```

This table shows the body parameters for the show quotas response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int	The number of instance cores that are allowed for each tenant.

Name	Type	Description
	(Required)	
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 2.46. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
```

```
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.2. Update quotas

Method	URI	Description
PUT	`/v1/{tenant_id}/os-quota-sets/{tenant_id}`	Updates quotas for a tenant.

Normal response codes: 200

2.5.2.1. Request

This table shows the URI parameters for the update quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.

Example 2.47. Update quotas response: JSON

```
{
    "quota_set": {
        "security_groups": 45
    }
}
```

This table shows the body parameters for the update quotas request:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Optional)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Optional)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Optional)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Optional)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Optional)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Optional)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Optional)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Optional)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	(Optional)	
`metadata_items`	Int *(Optional)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Optional)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Optional)*	The number of security groups that are allowed for each tenant.

Example 2.48. Show quotas response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <security_groups>45</security_groups>
</quota_set>
```

This operation does not accept a request body.

2.5.2.2. Response

Example 2.49. Update quota response: JSON

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "injected_files": 5,
        "instances": 10,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 45
    }
}
```

This table shows the body parameters for the update quotas response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int	The number of floating IP addresses that are allowed for each tenant.

Name	Type	Description
	(Required)	
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 2.50. Update quota response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set>
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>45</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.3. Delete quotas

Method	URI	Description
DELETE	`/v1/{tenant_id}/os-quota-sets/{tenant_id}`	Deletes quotas for a tenant so the quotas revert to default values.

Normal response codes: 200

2.5.3.1. Request

This table shows the URI parameters for the delete quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.

This operation does not accept a request body.

2.5.4. Show default quotas

Method	URI	Description
GET	`/v1/{tenant_id}/os-quota-sets/defaults`	Shows default quotas for a tenant.

Normal response codes: 200

2.5.4.1. Request

This table shows the URI parameters for the show default quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

2.5.4.2. Response

Example 2.51. Show default quotas: JSON response

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "id": "fake_tenant",
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "injected_files": 5,
        "instances": 10,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}
```

This table shows the body parameters for the show default quotas response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int	The ID for the quota set.

Name	Type	Description
	(Required)	
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 2.52. Show default quotas: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.5. Show quotas for user

Method	URI	Description
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Enables an admin user to show quotas for a tenant and user.

Normal response codes: 200

2.5.5.1. Request

This table shows the URI parameters for the show quotas for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

This operation does not accept a request body.

2.5.5.2. Response

Example 2.53. Show quotas for user response: JSON

```
{
    "quota_set": {
        "cores": 20,
        "fixed_ips": -1,
        "floating_ips": 10,
        "id": "fake_tenant",
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "injected_files": 5,
        "instances": 10,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}
```

This table shows the body parameters for the show quotas for user response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int	The number of floating IP addresses that are allowed for each tenant.

Name	Type	Description
	(Required)	
`id`	Int *(Required)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Required)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Required)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Required)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int *(Required)*	The number of key pairs that are allowed for each user.
`metadata_items`	Int *(Required)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Required)*	The number of security groups that are allowed for each tenant.
`in_use`	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
`reserved`	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 2.54. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <cores>20</cores>
  <fixed_ips>-1</fixed_ips>
  <floating_ips>10</floating_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>10</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.6. Update quotas for user

Method	URI	Description
POST	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Updates quotas for a tenant and user.

Normal response codes: 200

2.5.6.1. Request

This table shows the URI parameters for the update quotas for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

Example 2.55. Update quotas for user request: JSON

```
{
    "quota_set": {
        "force": true,
        "instances": 9
    }
}
```

This table shows the body parameters for the update quotas for user request:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Optional)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Optional)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.
`floating_ips`	Int *(Optional)*	The number of floating IP addresses that are allowed for each tenant.
`id`	Int *(Optional)*	The ID for the quota set.
`injected_file_content_bytes`	Int *(Optional)*	The number of bytes of content that are allowed for each injected file.
`injected_file_path_bytes`	Int *(Optional)*	The number of bytes that are allowed for each injected file path.
`injected_files`	Int *(Optional)*	The number of injected files that are allowed for each tenant.
`instances`	Int *(Optional)*	The number of instances that are allowed for each tenant.
`key_pairs`	Int	The number of key pairs that are allowed for each user.

Name	Type	Description
	(Optional)	
`metadata_items`	Int *(Optional)*	The number of metadata items that are allowed for each instance.
`ram`	Int *(Optional)*	The amount of instance RAM in megabytes that are allowed for each tenant.
`security_group_rules`	Int *(Optional)*	The number of rules that are allowed for each security group.
`security_groups`	Int *(Optional)*	The number of security groups that are allowed for each tenant.

Example 2.56. Update quotas for user request: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set id="fake_tenant">
  <force>true</force>
  <instances>9</instances>
</quota_set>
```

This operation does not accept a request body.

2.5.6.2. Response

Example 2.57. Update quotas for user response: JSON

```
{
    "quota_set": {
        "cores": 20,
        "floating_ips": 10,
        "fixed_ips": -1,
        "injected_file_content_bytes": 10240,
        "injected_file_path_bytes": 255,
        "injected_files": 5,
        "instances": 9,
        "key_pairs": 100,
        "metadata_items": 128,
        "ram": 51200,
        "security_group_rules": 20,
        "security_groups": 10
    }
}
```

This table shows the body parameters for the update quotas for user response:

Name	Type	Description
`quota_set`	Dict *(Required)*	A `quota_set` object.
`cores`	Int *(Required)*	The number of instance cores that are allowed for each tenant.
`fixed_ips`	Int *(Required)*	The number of fixed IP addresses that are allowed for each tenant. Must be equal to or greater than the number of allowed instances.

Name	Type	Description
floating_ips	Int *(Required)*	The number of floating IP addresses that are allowed for each tenant.
id	Int *(Required)*	The ID for the quota set.
injected_file_content_bytes	Int *(Required)*	The number of bytes of content that are allowed for each injected file.
injected_file_path_bytes	Int *(Required)*	The number of bytes that are allowed for each injected file path.
injected_files	Int *(Required)*	The number of injected files that are allowed for each tenant.
instances	Int *(Required)*	The number of instances that are allowed for each tenant.
key_pairs	Int *(Required)*	The number of key pairs that are allowed for each user.
metadata_items	Int *(Required)*	The number of metadata items that are allowed for each instance.
ram	Int *(Required)*	The amount of instance RAM in megabytes that are allowed for each tenant.
security_group_rules	Int *(Optional)*	The number of rules that are allowed for each security group.
security_groups	Int *(Required)*	The number of security groups that are allowed for each tenant.
in_use	String *(Optional)*	The in use data size. Visible only if you set the `usage=true` query parameter.
reserved	Int *(Optional)*	Reserved volume size. Visible only if you set the `usage=true` query parameter.

Example 2.58. Show quotas for user response: XML

```
<?xml version='1.0' encoding='UTF-8'?>
<quota_set>
  <cores>20</cores>
  <floating_ips>10</floating_ips>
  <fixed_ips>-1</fixed_ips>
  <injected_file_content_bytes>10240</injected_file_content_bytes>
  <injected_file_path_bytes>255</injected_file_path_bytes>
  <injected_files>5</injected_files>
  <instances>9</instances>
  <key_pairs>100</key_pairs>
  <metadata_items>128</metadata_items>
  <ram>51200</ram>
  <security_group_rules>20</security_group_rules>
  <security_groups>10</security_groups>
</quota_set>
```

This operation does not return a response body.

2.5.7. Delete quotas for user

Method	URI	Description
DELETE	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/{user_id}`	Deletes quotas for a user so that the quotas revert to default values.

Normal response codes: 200

2.5.7.1. Request

This table shows the URI parameters for the delete quotas for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

This operation does not accept a request body.

2.5.8. Show quota details for user

Method	URI	Description
GET	`/v1/{tenant_id}/os-quota-sets/{tenant_id}/detail/{user_id}`	Shows details for quotas for a tenant and user.

Normal response codes: 200

2.5.8.1. Request

This table shows the URI parameters for the show quota details for user request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{admin_tenant_id}`	UUID	The UUID of the administrative tenant.
`{user_id}`	String	The user ID. Specify in the URI as `user_id={user_id}`.

This operation does not accept a request body.

2.5.8.2. Response

Example 2.59. Show quota details for user response: JSON

```
{
    "quota_set": {
        "cores": {
            "in_use": 0,
            "limit": 20,
            "reserved": 0
        },
        "fixed_ips": {
            "in_use": 0,
            "limit": -1,
            "reserved": 0
        },
        "floating_ips": {
            "in_use": 0,
            "limit": 10,
            "reserved": 0
        },
        "injected_files": {
            "in_use": 0,
            "limit": 5,
            "reserved": 0
        },
        "instances": {
            "in_use": 0,
            "limit": 10,
            "reserved": 0
        },
        "key_pairs": {
            "in_use": 0,
            "limit": 100,
            "reserved": 0
        },
```

```
        "metadata_items": {
            "in_use": 0,
            "limit": 128,
            "reserved": 0
        },
        "ram": {
            "in_use": 0,
            "limit": 51200,
            "reserved": 0
        },
        "security_groups": {
            "in_use": 0,
            "limit": 10,
            "reserved": 0
        },
        "injected_file_content_bytes": {
            "in_use": 0,
            "limit": 10240,
            "reserved": 0
        },
        "injected_file_path_bytes": {
            "in_use": 0,
            "limit": 255,
            "reserved": 0
        },
        "security_group_rules": {
            "in_use": 0,
            "limit": 20,
            "reserved": 0
        }
    }
}
```

3. Clustering API v1 (CURRENT)

Clustering service for OpenStack.

For more information, see Senlin.

Method	URI	Description
API versions		
GET	`/`	Lists information for all Clustering API versions.
Actions		
GET	`/v1/actions{?limit,marker,sort,global_project,name,target,action}`	Lists all actions.
GET	`/v1/actions/{action_id}`	Shows details for an action.
Build information		
GET	`/v1/build-info`	Shows build information for a Senlin deployment.
Clusters		
GET	`/v1/clusters{?limit,marker,sort,global_project,name,status}`	Lists clusters.
POST	`/v1/clusters`	Creates a cluster.
GET	`/v1/clusters/{cluster_id}`	Shows details for a cluster.
PATCH	`/v1/clusters/{cluster_id}`	Updates a cluster.
DELETE	`/v1/clusters/{cluster_id}`	Deletes a cluster.
POST	`/v1/clusters/{cluster_id}/actions`	Triggers an action on a cluster.
Cluster policies		
GET	`/v1/clusters/{cluster_id}/policies{?sort,enabled}`	Lists all policies for a cluster.
GET	`/v1/clusters/{cluster_id}/policies/{policy_id}`	Shows details for a policy for a cluster.
Events		
GET	`/v1/events{?limit,marker,sort,global_project,obj_id,obj_type,obj_name,cluster_id,action}`	Lists events.
GET	`/v1/events/{event_id}`	Shows details for an event.
Nodes		
GET	`/v1/nodes{?limit,marker,sort,global_project,cluster_id,name,status}`	Lists all nodes.
POST	`/v1/nodes`	Creates a node.
GET	`/v1/nodes/{node_id}{?show_details}`	Shows data for a node.
PATCH	`/v1/nodes/{node_id}`	Updates a node.
DELETE	`/v1/nodes/{node_id}`	Deletes a node.
POST	`/v1/nodes/{node_id}/actions`	Triggers an action on a node.
Policies		
GET	`/v1/policies{?limit,marker,sort,global_project,name,type}`	Lists all policies.
POST	`/v1/policies`	Creates a policy.
GET	`/v1/policies/{policy_id}`	Shows details for a policy.
PATCH	`/v1/policies/{policy_id}`	Updates a policy.
DELETE	`/v1/policies/{policy_id}`	Deletes a policy.

Method	URI	Description
Policy types		
GET	`/v1/policy-types`	Lists all supported policy types.
GET	`/v1/policy-types/{policy_type}`	Shows details for a policy type.
Profiles		
GET	`/v1/profiles{?limit,marker,sort,global_project,name,type,metadata}`	Lists all profiles.
POST	`/v1/profiles`	Creates a profile.
GET	`/v1/profiles/{profile_id}`	Shows details for a profile.
PATCH	`/v1/profiles/{profile_id}`	Updates a profile.
DELETE	`/v1/profiles/{profile_id}`	Deletes a profile.
Profile types		
GET	`/v1/profile-types`	Lists supported profile types.
GET	`/v1/profile-types/{profile_type}`	Shows details for a profile type.
Receivers		
GET	`/v1/receivers{?limit,marker,sort,global_project,name,type,user,cluster_id,action}`	Lists all receivers.
POST	`/v1/receivers`	Creates a receiver.
GET	`/v1/receivers/{receiver_id}`	Shows details for a receiver.
DELETE	`/v1/receivers/{receiver_id}`	Deletes a receiver.
Webhooks		
POST	`/v1/webhooks/{webhook_id}/trigger{?V,params}`	Triggers a webhook receiver.

3.1. API versions

Lists information for all Clustering API versions.

Method	URI	Description
GET	/	Lists information for all Clustering API versions.

3.1.1. List versions

Method	URI	Description
GET	/	Lists information for all Clustering API versions.

Normal response codes: 200

3.1.1.1. Request

This operation does not accept a request body.

3.1.1.2. Response

Example 3.1. List versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://192.168.12.34:8778/v1/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

3.2. Actions

Lists all actions and shows details for an action.

Method	URI	Description
GET	`/v1/actions{?limit,marker,sort,global_project,name,target,action}`	Lists all actions.
GET	`/v1/actions/{action_id}`	Shows details for an action.

3.2.1. List actions

Method	URI	Description
GET	`/v1/actions{?limit,marker,sort,global_project,name,target,action}`	Lists all actions.

Normal response codes: 200

3.2.1.1. Request

This table shows the query parameters for the list actions request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`name`	String *(Optional)*	Filters the response by an action name. Use this filter multiple times to filter by multiple names.
`target`	String *(Optional)*	Filters the response by the targeted object ID that is associated with an action. An object can be a cluster, a node, and so on. Use this filter multiple times to filter by multiple targets.
`action`	String *(Optional)*	Filters the response by an action name. Use this filter multiple times to filter by multiple names.

This operation does not accept a request body.

3.2.1.2. Response

Example 3.2. List actions: JSON response

An `actions` list that contains all records that meet the criteria. Each record provides detailed data about an action.


```
{
    "actions": [
        {
            "action": "NODE_CREATE",
            "cause": "RPC Request",
            "created_at": "2015-12-04T04:54:41",
            "depended_by": [],
            "depends_on": [],
            "end_time": 1425550000.0,
            "id": "2366d440-c73e-4961-9254-6d1c3af7c167",
            "inputs": {},
            "interval": -1,
            "name": "node_create_0df0931b",
            "outputs": {},
            "owner": null,
            "start_time": 1425550000.0,
            "status": "SUCCEEDED",
            "status_reason": "Action completed successfully.",
            "target": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
            "timeout": 3600,
            "updated_at": null
        },
        {
            "action": "NODE_DELETE",
            "cause": "RPC Request",
            "created_at": "2015-11-04T05:21:41",
            "depended_by": [],
            "depends_on": [],
            "end_time": 1425550000.0,
            "id": "edce3528-864f-41fb-8759-f4707925cc09",
            "inputs": {},
            "interval": -1,
            "name": "node_delete_f0de9b9c",
            "outputs": {},
            "owner": null,
            "start_time": 1425550000.0,
            "status": "SUCCEEDED",
            "status_reason": "Action completed successfully.",
            "target": "f0de9b9c-6d48-4a46-af21-2ca8607777fe",
            "timeout": 3600,
            "updated_at": null
        }
    ]
}
```

3.2.2. Show action details

Method	URI	Description
GET	`/v1/actions/{action_id}`	Shows details for an action.

Normal response codes: 200

3.2.2.1. Request

This table shows the URI parameters for the show action details request:

Name	Type	Description
`{action_id}`	UUID	The UUID of the action.

This operation does not accept a request body.

3.2.2.2. Response

Example 3.3. Show action details: JSON response

An `action` map. Contains fields such as `id`, `action`, and so on.

```
{
    "action": "CLUSTER_DELETE",
    "cause": "RPC Request",
    "context": {},
    "created_at": "2015-06-27T05:09:43",
    "depended_by": [],
    "depends_on": [],
    "end_time": 1423570000.0,
    "id": "ffbb9175-d510-4bc1-b676-c6aba2a4ca81",
    "inputs": {},
    "interval": -1,
    "name": "cluster_delete_fcc9b635",
    "outputs": {},
    "owner": null,
    "start_time": 1423570000.0,
    "status": "FAILED",
    "status_reason": "Cluster action FAILED",
    "target": "fcc9b635-52e3-490b-99f2-87b1640e4e89",
    "timeout": 3600,
    "updated_at": null
}
```

3.3. Build information

Shows build information for a Senlin deployment.

Method	URI	Description
GET	`/v1/build-info`	Shows build information for a Senlin deployment.

3.3.1. Show build information

Method	URI	Description
GET	`/v1/build-info`	Shows build information for a Senlin deployment.

Normal response codes: 200

3.3.1.1. Request

This operation does not accept a request body.

3.3.1.2. Response

Example 3.4. Show build information: JSON response

```
{
    "build_info": {
        "api": {
            "revision": "1.0"
        },
        "engine": {
            "revision": "2.0"
        }
    }
}
```

3.4. Clusters

Lists all clusters and creates, shows information for, updates, deletes, and triggers an action on a cluster.

Method	URI	Description
GET	`/v1/clusters{?limit,marker,sort,global_project,name,status}`	Lists clusters.
POST	`/v1/clusters`	Creates a cluster.
GET	`/v1/clusters/{cluster_id}`	Shows details for a cluster.
PATCH	`/v1/clusters/{cluster_id}`	Updates a cluster.
DELETE	`/v1/clusters/{cluster_id}`	Deletes a cluster.
POST	`/v1/clusters/{cluster_id}/actions`	Triggers an action on a cluster.

3.4.1. List clusters

Method	URI	Description
GET	`/v1/clusters{?limit,marker,sort,global_project,name,status}`	Lists clusters.

Normal response codes: 200

3.4.1.1. Request

This table shows the query parameters for the list clusters request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`name`	String *(Optional)*	Filters the response by a cluster name. Use this filter multiple times to filter by multiple names.
`status`	String *(Optional)*	Filters the response by a cluster status. Use this filter multiple times to filter by multiple statuses.

This operation does not accept a request body.

3.4.1.2. Response

Example 3.5. List clusters: JSON response

```
{
    "clusters": [
```

```
        {
            "created_at": "2015-02-10T14:26:14",
            "data": {},
            "desired_capacity": 4,
            "domain": null,
            "id": "7d85f602-a948-4a30-afd4-e84f47471c15",
            "init_at": "2015-02-10T14:26:11",
            "max_size": -1,
            "metadata": {},
            "min_size": 0,
            "name": "cluster1",
            "nodes": [
                "b07c57c8-7ab2-47bf-bdf8-e894c0c601b9",
                "ecc23d3e-bb68-48f8-8260-c9cf6bcb6e61",
                "da1e9c87-e584-4626-a120-022da5062dac"
            ],
            "policies": [],
            "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
            "profile_name": "mystack",
            "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
            "status": "ACTIVE",
            "status_reason": "Cluster scale-in succeeded",
            "timeout": 3600,
            "updated_at": null,
            "user": "5e5bf8027826429c96af157f68dc9072"
        }
    ]
}
```

3.4.2. Create cluster

Method	URI	Description
POST	`/v1/clusters`	Creates a cluster.

Normal response codes: 202

3.4.2.1. Request

Example 3.6. Create cluster: JSON request

```
{
    "cluster": {
        "desired_capacity": 0,
        "max_size": -1,
        "metadata": {},
        "min_size": 0,
        "name": "test_cluster",
        "profile_id": "mystack",
        "timeout": null
    }
}
```

3.4.2.2. Response

Example 3.7. Create cluster: JSON response

```
{
    "cluster": {
        "created_at": null,
        "data": {},
        "desired_capacity": 4,
        "domain": null,
        "id": "45edadcb-c73b-4920-87e1-518b2f29f54b",
        "init_at": "2015-02-10T14:16:10",
        "max_size": -1,
        "metadata": {},
        "min_size": 0,
        "name": "test_cluster",
        "nodes": [],
        "policies": [],
        "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
        "profile_name": "mystack",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "status": "INIT",
        "status_reason": "Initializing",
        "timeout": 3600,
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.4.3. Show cluster details

Method	URI	Description
GET	`/v1/clusters/{cluster_id}`	Shows details for a cluster.

Normal response codes: 200

3.4.3.1. Request

This table shows the URI parameters for the show cluster details request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster.

This operation does not accept a request body.

3.4.3.2. Response

Example 3.8. Show cluster details: JSON response

```
{
    "cluster": {
        "created_at": "2015-02-11T15:13:20",
        "data": {},
        "desired_capacity": 0,
        "domain": null,
        "id": "45edadcb-c73b-4920-87e1-518b2f29f54b",
        "init_at": "2015-02-10T14:26:10",
        "max_size": -1,
        "metadata": {},
        "min_size": 0,
        "name": "test_cluster",
        "nodes": [],
        "policies": [],
        "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
        "profile_name": "mystack",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "status": "ACTIVE",
        "status_reason": "Creation succeeded",
        "timeout": 3600,
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.4.4. Update cluster

Method	URI	Description
PATCH	`/v1/clusters/{cluster_id}`	Updates a cluster.

Normal response codes: 202

3.4.4.1. Request

This table shows the URI parameters for the update cluster request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster.

Example 3.9. Update cluster: JSON request

```
{
    "cluster": {
        "metadata": null,
        "name": null,
        "profile_id": null,
        "timeout": "30"
    }
}
```

3.4.4.2. Response

Example 3.10. Update cluster: JSON response

```
{
    "cluster": {
        "created_at": "2015-02-11T15:13:20",
        "data": {},
        "desired_capacity": 0,
        "domain": null,
        "id": "45edadcb-c73b-4920-87e1-518b2f29f54b",
        "init_at": "2015-02-10T14:26:10",
        "max_size": -1,
        "metadata": {},
        "min_size": 0,
        "name": "test_cluster",
        "nodes": [],
        "policies": [],
        "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
        "profile_name": "mystack",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "status": "INIT",
        "status_reason": "Initializing",
        "timeout": 3600,
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.4.5. Delete cluster

Method	URI	Description
DELETE	`/v1/clusters/{cluster_id}`	Deletes a cluster.

Normal response codes: 202

3.4.5.1. Request

This table shows the URI parameters for the delete cluster request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster.

This operation does not accept a request body.

3.4.5.2. Response

The `Location` response header contains the action that the request triggered.

This operation does not return a response body.

3.4.6. Trigger cluster action

Method	URI	Description
POST	`/v1/clusters/{cluster_id}/actions`	Triggers an action on a cluster.

Normal response codes: 202

3.4.6.1. Request

This table shows the URI parameters for the trigger cluster action request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster.

3.4.6.2. Response

The `Location` response header contains the action that the request initiates.

This operation does not return a response body.

3.5. Cluster policies

Lists all cluster policies and shows information for a cluster policy.

Method	URI	Description
GET	`/v1/clusters/{cluster_id}/policies{?sort,enabled}`	Lists all policies for a cluster.
GET	`/v1/clusters/{cluster_id}/policies/{policy_id}`	Shows details for a policy for a cluster.

3.5.1. List policies

Method	URI	Description
GET	`/v1/clusters/{cluster_id}/policies{?sort,enabled}`	Lists all policies for a cluster.

Normal response codes: 200

3.5.1.1. Request

This table shows the URI parameters for the list policies request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster.

This table shows the query parameters for the list policies request:

Name	Type	Description

`sort`	String
	
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`enabled`	Boolean *(Optional)*	Filters the response by a policy enabled status on the cluster.

This operation does not accept a request body.

3.5.1.2. Response

Example 3.11. List policies: JSON response

A `cluster_policies` list contains records of cluster-policy bindings. Each record contains cluster-related fields, the policy, and the binding itself.

```
{
    "cluster_policies": [
        {
            "cluster_id": "7d85f602-a948-4a30-afd4-e84f47471c15",
            "cluster_name": "cluster4",
            "enabled": true,
            "id": "06be3a1f-b238-4a96-a737-ceec5714087e",
            "policy_id": "714fe676-a08f-4196-b7af-61d52eeded15",
            "policy_name": "dp01",
```

```
            "policy_type": "senlin.policy.deletion-1.0"
        },
        {
            "cluster_id": "7d85f602-a948-4a30-afd4-e84f47471c15",
            "cluster_name": "cluster4",
            "enabled": true,
            "id": "abddc45e-ac31-4f90-93cc-db55a7d8dd6d",
            "policy_id": "e026e09f-a3e9-4dad-a1b9-d7ba316026a1",
            "policy_name": "sp1",
            "policy_type": "senlin.policy.scaling-1.0"
        }
    ]
}
```

3.5.2. Show cluster policy details

Method	URI	Description
GET	`/v1/clusters/{cluster_id}/policies/{policy_id}`	Shows details for a policy for a cluster.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

3.5.2.1. Request

This table shows the URI parameters for the show cluster policy details request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster.
`{policy_id}`	UUID	The UUID of the policy.

This operation does not accept a request body.

3.5.2.2. Response

Example 3.12. Show cluster policy details: JSON response

A `cluster_policy` map with keys such as `cluster_id`, `policy_id`, `policy_type` , and so on. The key values are attributes for the policy that is attached to the cluster.

```
{
    "cluster_policy": {
        "cluster_id": "7d85f602-a948-4a30-afd4-e84f47471c15",
        "cluster_name": "cluster4",
        "enabled": true,
        "id": "06be3a1f-b238-4a96-a737-ceec5714087e",
        "policy_id": "714fe676-a08f-4196-b7af-61d52eeded15",
        "policy_name": "dp01",
        "policy_type": "senlin.policy.deletion-1.0"
    }
}
```

3.6. Events

Lists all events and shows information for an event.

Method	URI	Description
GET	`/v1/events{?limit,marker,sort,global_project,obj_id,obj_type,obj_name,cluster_id,action}`	Lists events.
GET	`/v1/events/{event_id}`	Shows details for an event.

3.6.1. List events

Method	URI	Description
GET	`/v1/events{?limit,marker,sort,global_project,obj_id,obj_type,obj_name,cluster_id,action}`	Lists events.

Normal response codes: 200

3.6.1.1. Request

This table shows the query parameters for the list events request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`obj_id`	String *(Optional)*	Filters the response by the object ID for an event. Use this filter multiple times to filter by multiple objects.
`obj_type`	String *(Optional)*	Filters the response by the type of object associated with an event. Use this filter multiple times to filter by multiple objects. A valid value is `CLUSTER` or `NODE`.
`obj_name`	String *(Optional)*	Filters the response by the name of object associated with an event. Use this filter multiple times to filter by multiple objects.
`cluster_id`	String *(Optional)*	Filters the response by the cluster ID associated with an event. Use this filter multiple times to filter by multiple clusters.
`action`	String *(Optional)*	Filters the response by the action name associated with an event. Use this filter multiple times to filter by multiple actions.

This operation does not accept a request body.

3.6.1.2. Response

Example 3.13. List events: JSON response

```
{
    "events": [
        {
            "action": "create",
            "cluster_id": null,
            "id": "2d255b9c-8f36-41a2-a137-c0175ccc29c3",
            "level": "20",
            "obj_id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
            "obj_name": "node009",
            "obj_type": "NODE",
            "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
            "status": "CREATING",
            "status_reason": "Initializing",
            "timestamp": "2015-03-05T08:53:15",
            "user": "a21ded6060534d99840658a777c2af5a"
        }
    ]
}
```

3.6.2. Show event details

Method	URI	Description
GET	`/v1/events/{event_id}`	Shows details for an event.

Normal response codes: 200

3.6.2.1. Request

This table shows the URI parameters for the show event details request:

Name	Type	Description
`{event_id}`	UUID	The UUID of the event.

This operation does not accept a request body.

3.6.2.2. Response

Example 3.14. Show event details: JSON response

An `event` map, which contains the details of an event such as `TIMESTAMP` or `LEVEL`.

```
{
    "event": {
        "action": "create",
        "cluster_id": null,
        "id": "2d255b9c-8f36-41a2-a137-c0175ccc29c3",
        "level": "20",
        "obj_id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
        "obj_name": "node009",
        "obj_type": "NODE",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "status": "CREATING",
        "status_reason": "Initializing",
        "timestamp": "2015-03-05T08:53:15",
        "user": "a21ded6060534d99840658a777c2af5a"
    }
}
```

3.7. Nodes

Lists all nodes, and creates, shows information for, updates, deletes a node.

Method	URI	Description
GET	`/v1/nodes{?limit,marker,sort,global_project,cluster_id,name,status}`	Lists all nodes.
POST	`/v1/nodes`	Creates a node.
GET	`/v1/nodes/{node_id}{?show_details}`	Shows data for a node.
PATCH	`/v1/nodes/{node_id}`	Updates a node.
DELETE	`/v1/nodes/{node_id}`	Deletes a node.
POST	`/v1/nodes/{node_id}/actions`	Triggers an action on a node.

3.7.1. List nodes

Method	URI	Description
GET	`/v1/nodes{?limit,marker,sort,global_project,cluster_id,name,status}`	Lists all nodes.

Normal response codes: 200

3.7.1.1. Request

This table shows the query parameters for the list nodes request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`cluster_id`	String *(Optional)*	Filters the response by the cluster that owns a node.
`name`	String *(Optional)*	Filters the response by the name of a node.
`status`	String *(Optional)*	Filters the response by the status of a node.

This operation does not accept a request body.

3.7.1.2. Response

Example 3.15. List nodes: JSON response


```
{
    "nodes": [
        {
            "cluster_id": null,
            "created_at": "2015-02-27T04:39:21",
            "data": {},
            "details": {},
            "domain": null,
            "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
            "index": -1,
            "init_at": "2015-02-27T04:39:18",
            "metadata": {},
            "name": "node00a",
            "physical_id": "cc028275-d078-4729-bf3e-154b7359814b",
            "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
            "profile_name": "mystack",
            "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
            "role": null,
            "status": "ACTIVE",
            "status_reason": "Creation succeeded",
            "updated_at": null,
            "user": "5e5bf8027826429c96af157f68dc9072"
        }
    ]
}
```

3.7.2. Create node

Method	URI	Description
POST	/v1/nodes	Creates a node.

Normal response codes: 202

3.7.2.1. Request

Example 3.16. Create node: JSON request

```
{
    "node": {
        "cluster_id": null,
        "metadata": {},
        "name": "node009",
        "profile_id": "mystack",
        "role": "master"
    }
}
```

3.7.2.2. Response

Example 3.17. Create node: JSON response

```
{
    "node": {
        "cluster_id": null,
        "created_at": null,
        "data": {},
        "domain": null,
        "id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
        "index": -1,
        "init_at": "2015-03-05T08:53:15",
        "metadata": {},
        "name": "node009",
        "physical_id": "",
        "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
        "profile_name": "mystack",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "role": "master",
        "status": "INIT",
        "status_reason": "Initializing",
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.7.3. Show node details

Method	URI	Description
GET	`/v1/nodes/{node_id}{?show_details}`	Shows data for a node.

Normal response codes: 200

3.7.3.1. Request

This table shows the URI parameters for the show node details request:

Name	Type	Description
`{node_id}`	UUID	The UUID of the node.

This table shows the query parameters for the show node details request:

Name	Type	Description
`show_details`	Boolean *(Optional)*	Indicates whether the node details are returned. Default is `false`.

This operation does not accept a request body.

3.7.3.2. Response

Example 3.18. Show node details: JSON response

A node record. Each record contains fields such as `id`, `cluster_id`, `name`, `physical_id`, `profile_id`, `created_at`, `index`, `status`, `status_reason`, `metadata`, `updated_at`, and so on.

```
{
    "node": {
        "cluster_id": null,
        "created_at": "2015-02-10T12:03:16",
        "data": {},
        "domain": null,
        "id": "d5779bb0-f0a0-49c9-88cc-6f078adb5a0b",
        "index": -1,
        "init_at": "2015-02-10T12:03:13",
        "metadata": {},
        "name": "node1",
        "physical_id": "f41537fa-22ab-4bea-94c0-c874e19d0c80",
        "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
        "profile_name": "mystack",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "role": null,
        "status": "ACTIVE",
        "status_reason": "Creation succeeded",
        "updated_at": "2015-03-04T04:58:27",
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.7.4. Update node

Method	URI	Description
PATCH	`/v1/nodes/{node_id}`	Updates a node.

Normal response codes: 202

3.7.4.1. Request

This table shows the URI parameters for the update node request:

Name	Type	Description
`{node_id}`	UUID	The UUID of the node.

Example 3.19. Update node: JSON request

```
{
    "node": {
        "name": "new_node_name"
    }
}
```

3.7.4.2. Response

Example 3.20. Update node: JSON response

```
{
    "node": {
        "action": "2366d440-c73e-4961-9254-6d1c3af7c167",
        "cluster_id": null,
        "created_at": null,
        "data": {},
        "domain": null,
        "id": "0df0931b-e251-4f2e-8719-4ebfda3627ba",
        "index": -1,
        "init_at": "2015-03-05T08:53:15",
        "metadata": {},
        "name": "node009",
        "physical_id": "",
        "profile_id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
        "profile_name": "mystack",
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "role": "master",
        "status": "ACTIVE",
        "status_reason": "Update succeeded",
        "updated_at": "2015-09-01T18:53:15",
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.7.5. Delete node

Method	URI	Description
DELETE	`/v1/nodes/{node_id}`	Deletes a node.

Normal response codes: 202

3.7.5.1. Request

This table shows the URI parameters for the delete node request:

Name	Type	Description
`{node_id}`	UUID	The UUID of the node.

This operation does not accept a request body.

3.7.5.2. Response

The `Location` response header contains the action that the request initiated.

This operation does not return a response body.

3.7.6. Trigger node action

Method	URI	Description
POST	`/v1/nodes/{node_id}/actions`	Triggers an action on a node.

Normal response codes: 202

3.7.6.1. Request

This table shows the URI parameters for the trigger node action request:

Name	Type	Description
`{node_id}`	UUID	The UUID of the node.

3.7.6.2. Response

The `Location` response header contains the action that the request initiates.

This operation does not return a response body.

3.8. Policies

Lists all policies and creates, shows information for, updates, and deletes a policy.

Method	URI	Description
GET	`/v1/policies{?limit,marker,sort,global_project,name,type}`	Lists all policies.
POST	`/v1/policies`	Creates a policy.
GET	`/v1/policies/{policy_id}`	Shows details for a policy.
PATCH	`/v1/policies/{policy_id}`	Updates a policy.
DELETE	`/v1/policies/{policy_id}`	Deletes a policy.

3.8.1. List policies

Method	URI	Description
GET	`/v1/policies{?limit,marker,sort,global_project,name,type}`	Lists all policies.

Normal response codes: 200

3.8.1.1. Request

This table shows the query parameters for the list policies request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`name`	String *(Optional)*	Filters the response by the name of a policy.
`type`	String *(Optional)*	Filters the response by the type of a policy.

This operation does not accept a request body.

3.8.1.2. Response

Example 3.21. List policies: JSON response

```
{
    "policies": [
```

```
        {
            "created_at": "2015-02-15T08:33:13.000000",
            "data": {},
            "domain": null,
            "id": "7192d8df-73be-4e98-ab99-1cf6d5066729",
            "name": "test_policy_1",
            "project": "42d9e9663331431f97b75e25136307ff",
            "spec": {
                "description": "A test policy",
                "properties": {
                    "criteria": "OLDEST_FIRST",
                    "destroy_after_deletion": true,
                    "grace_period": 60,
                    "reduce_desired_capacity": false
                },
                "type": "senlin.policy.deletion",
                "version": "1.0"
            },
            "type": "senlin.policy.deletion-1.0",
            "updated_at": null,
            "user": "5e5bf8027826429c96af157f68dc9072"
        }
    ]
}
```

3.8.2. Create policy

Method	URI	Description
POST	/v1/policies	Creates a policy.

Normal response codes: 201

3.8.2.1. Request

Example 3.22. Create policy: JSON request

```
{
    "policy": {
        "name": "sp001",
        "spec": {
            "properties": {
                "adjustment": {
                    "min_step": 1,
                    "number": 1,
                    "type": "CHANGE_IN_CAPACITY"
                },
                "event": "CLUSTER_SCALE_IN"
            },
            "type": "senlin.policy.scaling",
            "version": "1.0"
        }
    }
}
```

3.8.2.2. Response

Example 3.23. Create policy: JSON response

```
{
    "policy": {
        "created_at": "2015-03-02T07:40:31",
        "data": {},
        "domain": null,
        "id": "02f62195-2198-4797-b0a9-877632208527",
        "name": "sp001",
        "project": "42d9e9663331431f97b75e25136307ff",
        "spec": {
            "properties": {
                "adjustment": {
                    "best_effort": true,
                    "min_step": 1,
                    "number": 1,
                    "type": "CHANGE_IN_CAPACITY"
                },
                "event": "CLUSTER_SCALE_IN"
            },
            "type": "senlin.policy.scaling",
            "version": "1.0"
        },
        "type": "senlin.policy.scaling-1.0",
        "updated_at": null,
```

```
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.8.3. Show policy details

Method	URI	Description
GET	`/v1/policies/{policy_id}`	Shows details for a policy.

Normal response codes: 200

3.8.3.1. Request

This table shows the URI parameters for the show policy details request:

Name	Type	Description
`{policy_id}`	UUID	The UUID of the policy.

This operation does not accept a request body.

3.8.3.2. Response

Example 3.24. Show policy details: JSON response

A policy record with fields such as `id`, `name`, `type`, `spec`, `created_at`, `updated_at`, and so on.

```
{
    "policy": {
        "created_at": "2015-03-02T07:40:31",
        "data": {},
        "domain": null,
        "id": "02f62195-2198-4797-b0a9-877632208527",
        "name": "sp001",
        "project": "42d9e9663331431f97b75e25136307ff",
        "spec": {
            "properties": {
                "adjustment": {
                    "best_effort": true,
                    "min_step": 1,
                    "number": 1,
                    "type": "CHANGE_IN_CAPACITY"
                },
                "event": "CLUSTER_SCALE_IN"
            },
            "type": "senlin.policy.scaling",
            "version": "1.0"
        },
        "type": "senlin.policy.scaling-1.0",
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.8.4. Update policy

Method	URI	Description
PATCH	`/v1/policies/{policy_id}`	Updates a policy.

Normal response codes: 200

3.8.4.1. Request

This table shows the URI parameters for the update policy request:

Name	Type	Description
`{policy_id}`	UUID	The UUID of the policy.

Example 3.25. Update policy: JSON request

```
{
    "policy": {
        "name": "new_name"
    }
}
```

3.8.4.2. Response

Example 3.26. Update policy: JSON response

A policy record that includes fields such as `id`, `name`, `type`, `spec`, `created_at`, `updated_at`, and so on.

```
{
    "policy": {
        "created_at": "2015-10-14T09:14:53",
        "data": {},
        "domain": null,
        "id": "ac5415bd-f522-4160-8be0-f8853e4bc332",
        "name": "dp01",
        "project": "42d9e9663331431f97b75e25136307ff",
        "spec": {
            "description": "A policy for node deletion.",
            "properties": {
                "criteria": "OLDEST_FIRST",
                "destroy_after_deletion": true,
                "grace_period": 60,
                "reduce_desired_capacity": false
            },
            "type": "senlin.policy.deletion",
            "version": "1.0"
        },
        "type": "senlin.policy.deletion-1.0",
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.8.5. Delete policy

Method	URI	Description
DELETE	`/v1/policies/{policy_id}`	Deletes a policy.

Normal response codes: 204

3.8.5.1. Request

This table shows the URI parameters for the delete policy request:

Name	Type	Description
`{policy_id}`	UUID	The UUID of the policy.

This operation does not accept a request body.

3.9. Policy types

Lists all policy types and shows details for a policy type.

Method	URI	Description
GET	`/v1/policy-types`	Lists all supported policy types.
GET	`/v1/policy-types/{policy_type}`	Shows details for a policy type.

3.9.1. List policy types

Method	URI	Description
GET	`/v1/policy-types`	Lists all supported policy types.

Normal response codes: 200

3.9.1.1. Request

This operation does not accept a request body.

3.9.1.2. Response

Example 3.27. List policy types: JSON response

A list of supported policy types.

```
{
    "policy_types": [
        {
            "name": "ScalingPolicy"
        },
        {
            "name": "PlacementPolicy"
        },
        {
            "name": "DeletionPolicy"
        },
        {
            "name": "LoadBalancingPolicy"
        },
        {
            "name": "HealthPolicy"
        },
        {
            "name": "UpdatePolicy"
        }
    ]
}
```

3.9.2. Show policy type details

Method	URI	Description
GET	`/v1/policy-types/{policy_type}`	Shows details for a policy type.

Normal response codes: 200

3.9.2.1. Request

This table shows the URI parameters for the show policy type details request:

Name	Type	Description
`{policy_type}`	String	The name of the policy type.

This operation does not accept a request body.

3.9.2.2. Response

Example 3.28. Show policy type details: JSON response

```
{
    "policy_type": {
        "name": "senlin.policy.deletion",
        "schema": {
            "criteria": {
                "constraints": [
                    {
                        "constraint": [
                            "OLDEST_FIRST",
                            "OLDEST_PROFILE_FRIST",
                            "YOUNGEST_FIRST",
                            "RANDOM"
                        ],
                        "type": "AllowedValues"
                    }
                ],
                "default": "RANDOM",
                "description": "Criteria used in selecting candidates for
 deletion",
                "required": false,
                "type": "String"
            },
            "destroy_after_deletion": {
                "default": true,
                "description": "Whether a node should be completely destroyed
 after deletion. Default to true",
                "required": false,
                "type": "Boolean"
            },
            "grace_period": {
                "default": 0,
                "description": "Number of seconds before real deletion
 happens.",
                "required": false,
                "type": "Integer"
```

```
            },
            "reduce_desired_capacity": {
                "default": false,
                "description": "Whether the desired capacity of the cluster
 should be reduced along the deletion. Default to false.",
                "required": false,
                "type": "Boolean"
            }
        }
    }
}
```

3.10. Profiles

Lists all profiles and creates, shows information for, updates, and deletes a profile.

Method	URI	Description
GET	`/v1/profiles{?limit,marker,sort,global_project,name,type,metadata}`	Lists all profiles.
POST	`/v1/profiles`	Creates a profile.
GET	`/v1/profiles/{profile_id}`	Shows details for a profile.
PATCH	`/v1/profiles/{profile_id}`	Updates a profile.
DELETE	`/v1/profiles/{profile_id}`	Deletes a profile.

3.10.1. List profiles

Method	URI	Description
GET	`/v1/profiles{?limit,marker,sort,global_project,name,type,metadata}`	Lists all profiles.

Normal response codes: 200

3.10.1.1. Request

This table shows the query parameters for the list profiles request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`name`	String *(Optional)*	Filters the response by the name of a profile.
`type`	String *(Optional)*	Filters the response by the type of a profile.
`metadata`	Dict *(Optional)*	Filters the response by a metadata key and value pair.

This operation does not accept a request body.

3.10.1.2. Response

Example 3.29. List profiles: JSON response

```
{
```

```
    "profiles": [
        {
            "created_at": "2015-02-10T11:46:33.000000",
            "domain": null,
            "id": "edc63d0a-2ca4-48fa-9854-27926da76a4a",
            "metadata": {},
            "name": "mystack",
            "project": "42d9e9663331431f97b75e25136307ff",
            "spec": {
                "properties": {
                    "disable_rollback": false,
                    "environment": {
                        "resource_registry": {
                            "os.heat.server": "OS::Heat::Server"
                        }
                    },
                    "files": {
                        "file:///opt/stack/senlin/examples/profiles/
test_script.sh": "#!/bin/bash\n\necho \"this is a test script file\"\n"
                    },
                    "name": "random_string_stack",
                    "parameters": {},
                    "rollback": false,
                    "template": {
                        "heat_template_version": "2014-10-16",
                        "outputs": {
                            "result": {
                                "value": {
                                    "get_attr": [
                                        "random",
                                        "value"
                                    ]
                                }
                            }
                        },
                        "parameters": {
                            "file": {
                                "default": {
                                    "get_file": "file:///opt/stack/senlin/
examples/profiles/test_script.sh"
                                },
                                "type": "string"
                            }
                        },
                        "resources": {
                            "random": {
                                "properties": {
                                    "length": 64
                                },
                                "type": "OS::Heat::RandomString"
                            }
                        }
                    },
                    "timeout": 60
                },
                "type": "os.heat.stack",
                "version": "1.0"
            },
            "type": "os.heat.stack-1.0",
            "updated_at": null,
```

```
            "user": "5e5bf8027826429c96af157f68dc9072"
        }
    ]
}
```

3.10.2. Create profile

Method	URI	Description
POST	`/v1/profiles`	Creates a profile.

Normal response codes: 201

3.10.2.1. Request

Example 3.30. Create profile: JSON request

```
{
    "profile": {
        "metadata": {},
        "name": "test_prof1",
        "spec": {
            "properties": {
                "disable_rollback": false,
                "environment": {
                    "resource_registry": {
                        "os.heat.server": "OS::Heat::Server"
                    }
                },
                "files": {
                    "file:///usr/test_script.sh": "#!/bin/bash\n\necho \"this
 is a test script file\"\n"
                },
                "parameters": {},
                "template": {
                    "heat_template_version": "2014-10-16",
                    "outputs": {
                        "result": {
                            "value": {
                                "get_attr": [
                                    "random",
                                    "value"
                                ]
                            }
                        }
                    },
                    "parameters": {
                        "file": {
                            "default": {
                                "get_file": "file:///usr/test_script.sh"
                            },
                            "type": "string"
                        }
                    },
                    "resources": {
                        "random": {
                            "properties": {
                                "length": 64
                            },
                            "type": "OS::Heat::RandomString"
                        }
                    }
                },
```

```
                "timeout": 60
            },
            "type": "os.heat.stack",
            "version": "1.0"
        }
    }
}
```

3.10.2.2. Response

Example 3.31. Create profile: JSON response

```
{
    "profile": {
        "created_at": "2015-03-01T14:28:25",
        "domain": null,
        "id": "7fa885cd-fa39-4531-a42d-780af95c84a4",
        "metadata": {},
        "name": "test_prof1",
        "project": "42d9e9663331431f97b75e25136307ff",
        "spec": {
            "properties": {
                "disable_rollback": false,
                "environment": {
                    "resource_registry": {
                        "os.heat.server": "OS::Heat::Server"
                    }
                },
                "files": {
                    "file:///opt/stack/senlin/examples/profiles/test_script.
sh": "#!/bin/bash\n\necho \"this is a test script file\"\n"
                },
                "parameters": {},
                "template": {
                    "heat_template_version": "2014-10-16",
                    "outputs": {
                        "result": {
                            "value": {
                                "get_attr": [
                                    "random",
                                    "value"
                                ]
                            }
                        }
                    },
                    "parameters": {
                        "file": {
                            "default": {
                                "get_file": "file:///opt/stack/senlin/
examples/profiles/test_script.sh"
                            },
                            "type": "string"
                        }
                    },
                    "resources": {
                        "random": {
                            "properties": {
                                "length": 64
                            },
                            "type": "OS::Heat::RandomString"
```

```
                        }
                    }
                },
                "timeout": 60
            },
            "type": "os.heat.stack",
            "version": "1.0"
        },
        "type": "os.heat.stack-1.0",
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.10.3. Show profile details

Method	URI	Description
GET	`/v1/profiles/{profile_id}`	Shows details for a profile.

Normal response codes: 200

3.10.3.1. Request

This table shows the URI parameters for the show profile details request:

Name	Type	Description
`{profile_id}`	UUID	The UUID of the profile.

This operation does not accept a request body.

3.10.3.2. Response

Example 3.32. Show profile details: JSON response

A profile record. Contains fields such as `id`, `name`, `type`, `spec`, `metadata`, and so on.

```
{
    "profile": {
        "created_at": "2015-03-01T14:28:25",
        "domain": null,
        "id": "7fa885cd-fa39-4531-a42d-780af95c84a4",
        "metadata": {},
        "name": "test_prof1",
        "project": "42d9e9663331431f97b75e25136307ff",
        "spec": {
            "disable_rollback": false,
            "environment": {
                "resource_registry": {
                    "os.heat.server": "OS::Heat::Server"
                }
            },
            "files": {
                "file:///opt/stack/senlin/examples/profiles/test_script.sh":
 "#!/bin/bash\n\necho \"this is a test script file\"\n"
            },
            "parameters": {},
            "template": {
                "heat_template_version": "2014-10-16",
                "outputs": {
                    "result": {
                        "value": {
                            "get_attr": [
                                "random",
                                "value"
                            ]
                        }
                    }
                },
                "parameters": {
```

```
                    "file": {
                        "default": {
                            "get_file": "file:///opt/stack/senlin/examples/
profiles/test_script.sh"
                        },
                        "type": "string"
                    }
                },
                "resources": {
                    "random": {
                        "properties": {
                            "length": 64
                        },
                        "type": "OS::Heat::RandomString"
                    }
                },
                "timeout": 60
            },
            "type": "os.heat.stack",
            "version": "1.0"
        },
        "type": "os.heat.stack-1.0",
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.10.4. Update profile

Method	URI	Description
PATCH	`/v1/profiles/{profile_id}`	Updates a profile.

Normal response codes: 200

3.10.4.1. Request

This table shows the URI parameters for the update profile request:

Name	Type	Description
`{profile_id}`	UUID	The UUID of the profile.

Example 3.33. Update profile: JSON request

```
{
    "profile": {
        "metadata": {},
        "name": null
    }
}
```

3.10.4.2. Response

Example 3.34. Update profile: JSON response

A profile record. Contains fields such as `id`, `name`, `type`, `spec`, `metadata`, and so on.

```
{
    "profile": {
        "created_at": "2015-03-01T14:28:25",
        "domain": null,
        "id": "7fa885cd-fa39-4531-a42d-780af95c84a4",
        "metadata": {},
        "name": "test_prof1",
        "project": "42d9e9663331431f97b75e25136307ff",
        "spec": {
            "disable_rollback": false,
            "environment": {
                "resource_registry": {
                    "os.heat.server": "OS::Heat::Server"
                }
            },
            "files": {
                "file:///opt/stack/senlin/examples/profiles/test_script.sh":
 "#!/bin/bash\n\necho \"this is a test script file\"\n"
            },
            "parameters": {},
            "template": {
                "heat_template_version": "2014-10-16",
                "outputs": {
                    "result": {
                        "value": {
                            "get_attr": [
```

```
                                "random",
                                "value"
                            ]
                        }
                    }
                },
                "parameters": {
                    "file": {
                        "default": {
                            "get_file": "file:///opt/stack/senlin/examples/
profiles/test_script.sh"
                        },
                        "type": "string"
                    }
                },
                "resources": {
                    "random": {
                        "properties": {
                            "length": 64
                        },
                        "type": "OS::Heat::RandomString"
                    }
                },
                "timeout": 60
            },
            "type": "os.heat.stack",
            "version": "1.0"
        },
        "type": "os.heat.stack-1.0",
        "updated_at": null,
        "user": "5e5bf8027826429c96af157f68dc9072"
    }
}
```

3.10.5. Delete profile

Method	URI	Description
DELETE	`/v1/profiles/{profile_id}`	Deletes a profile.

Normal response codes: 204

3.10.5.1. Request

This table shows the URI parameters for the delete profile request:

Name	Type	Description
`{profile_id}`	UUID	The UUID of the profile.

This operation does not accept a request body.

3.11. Profile types

Lists all profile types and shows details for a profile type.

Method	URI	Description
GET	`/v1/profile-types`	Lists supported profile types.
GET	`/v1/profile-types/{profile_type}`	Shows details for a profile type.

3.11.1. List profile types

Method	URI	Description
GET	`/v1/profile-types`	Lists supported profile types.

Normal response codes: 200

3.11.1.1. Request

This operation does not accept a request body.

3.11.1.2. Response

Example 3.35. List profile types: JSON response

A list of supported profile types.

```
{
    "profile_types": [
        {
            "name": "os.heat.stack"
        },
        {
            "name": "os.heat.resource"
        },
        {
            "name": "os.nova.server"
        }
    ]
}
```

3.11.2. Show profile type details

Method	URI	Description
GET	`/v1/profile-types/{profile_type}`	Shows details for a profile type.

Normal response codes: 200

3.11.2.1. Request

This table shows the URI parameters for the show profile type details request:

Name	Type	Description
`{profile_type}`	String	The name of the profile type.

This operation does not accept a request body.

3.11.2.2. Response

Example 3.36. Show profile type details: JSON response

```
{
    "profile_type": {
        "name": "os.heat.stack",
        "schema": {
            "context": {
                "default": {},
                "description": "A dictionary for specifying the customized
context for stack operations",
                "required": false,
                "type": "Map"
            },
            "disable_rollback": {
                "default": true,
                "description": "A boolean specifying whether a stack operation
can be rolled back.",
                "required": false,
                "type": "Boolean"
            },
            "environment": {
                "default": {},
                "description": "A map that specifies the environment used for
stack operations.",
                "required": false,
                "type": "Map"
            },
            "files": {
                "default": {},
                "description": "Contents of files referenced by the template,
if any.",
                "required": false,
                "type": "Map"
            },
            "parameters": {
                "default": {},
                "description": "Parameters to be passed to Heat for stack
operations.",
```

```
                "required": false,
                "type": "Map"
            },
            "template": {
                "description": "Heat stack template.",
                "required": true,
                "type": "Map"
            },
            "timeout": {
                "description": "A integer that specifies the number of minutes
 that a stack operation times out.",
                "required": false,
                "type": "Integer"
            }
        }
    }
}
```

3.12. Receivers

Lists all receivers and creates, shows information for, and deletes a receiver.

Method	URI	Description
GET	`/v1/receivers{?limit,marker,sort,global_project,name,type,user,cluster_id,action}`	Lists all receivers.
POST	`/v1/receivers`	Creates a receiver.
GET	`/v1/receivers/{receiver_id}`	Shows details for a receiver.
DELETE	`/v1/receivers/{receiver_id}`	Deletes a receiver.

3.12.1. List receivers

Method	URI	Description
GET	`/v1/receivers{?limit,marker,sort,global_project,name,type,user,cluster_id,action}`	Lists all receivers.

Normal response codes: 200

3.12.1.1. Request

This table shows the query parameters for the list receivers request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

`sort`	String
	(Optional)
		Sorts the response by one or more attribute and optional sort direction combinations. A valid direction is `asc` (ascending) or `desc` (descending). Default direction is `asc` (ascending).
		
		Specify the list as <key>[:<direction>].
		
		For example, the following query parameters in the URI sort the objects in the response by `name` in ascending order and then by `status` in descending order:
		
		`GET /v2/images?sort=name:asc,status:desc`
		
		The following query parameters in the URI sort the objects in the response by `name` in descending order and then by `status` in ascending order.
		
		`GET /v2/images?sort=name:desc,status`

`global_project`	Boolean *(Optional)*	Indicates whether to include objects for all projects or objects for the current project in the response. If you are an administrative user and you set this value to `true`, the call returns all objects from all projects. Default is `false`, which returns only objects in the current project.
`name`	String *(Optional)*	Filters the response by the name of a receiver.
`type`	String *(Optional)*	Filters the response by the type of a receiver.
`user`	String *(Optional)*	Filters the response by the user name of a receiver.
`cluster_id`	String *(Optional)*	Filters the response by the ID of the targeted cluster of a receiver.
`action`	String *(Optional)*	Filters the response by the targeted action of a receiver.

This operation does not accept a request body.

3.12.1.2. Response

Example 3.37. List receivers: JSON response

List of receiver records. Each record contains fields such as `id`, `name`, `user`, `project>`, `domain`, `cluster_id`, `type`, `action`, `actor`, `params`, `created_at` and `updated_at`, and so on.

```
{
    "receivers": [
        {
            "action": "CLUSTER_SCALE_OUT",
            "actor": {
                "trust_id": [
                    "6dc6d336e3fc4c0a951b5698cd1236d9"
                ]
            },
            "channel": {
                "alarm_url": "http://node1:8778/v1/webhooks/
e03dd2e5-8f2e-4ec1-8c6a-74ba891e5422/trigger?V=1&count=1"
            },
            "cluster_id": "ae63a10b-4a90-452c-aef1-113a0b255ee3",
            "created_at": "2015-06-27T05:09:43",
            "domain": "Default",
            "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
            "name": "cluster_inflate",
            "params": {
                "count": "1"
            },
            "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
            "type": "webhook",
            "updated_at": null,
            "user": "b4ad2d6e18cc2b9c48049f6dbe8a5b3c"
        }
    ]
}
```

3.12.2. Create receiver

Method	URI	Description
POST	`/v1/receivers`	Creates a receiver.

Normal response codes: 201

3.12.2.1. Request

Example 3.38. Create receiver: JSON request

```
{
    "receiver": {
        "action": "CLUSTER_SCALE_OUT",
        "cluster_id": "cf99d754-3cdc-47f4-8a29-cd14f02f5436",
        "name": "cluster_inflate",
        "params": {
            "count": "1"
        },
        "type": "webhook"
    }
}
```

3.12.2.2. Response

Example 3.39. Create receiver: JSON response

```
{
    "receiver": {
        "action": "CLUSTER_SCALE_OUT",
        "actor": {
            "trust_id": [
                "6dc6d336e3fc4c0a951b5698cd1236d9"
            ]
        },
        "channel": {
            "alarm_url": "http://node1:8778/v1/webhooks/
e03dd2e5-8f2e-4ec1-8c6a-74ba891e5422/trigger?V=1&count=1"
        },
        "cluster_id": "ae63a10b-4a90-452c-aef1-113a0b255ee3",
        "created_at": "2015-06-27T05:09:43",
        "domain": "Default",
        "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
        "name": "cluster_inflate",
        "params": {
            "count": "1"
        },
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "type": "webhook",
        "updated_at": null,
        "user": "b4ad2d6e18cc2b9c48049f6dbe8a5b3c"
    }
}
```

3.12.3. Show receiver details

Method	URI	Description
GET	`/v1/receivers/{receiver_id}`	Shows details for a receiver.

Normal response codes: 200

3.12.3.1. Request

This table shows the URI parameters for the show receiver details request:

Name	Type	Description
`{receiver_id}`	UUID	The UUID of the receiver.

This operation does not accept a request body.

3.12.3.2. Response

Example 3.40. Show receiver details: JSON response

A receiver record. The record includes fields such as `id`, `name`, `type`, `cluster_id`, `action`, `actor`, `params`, `channel`, `created_at`, `updated_at`, `user`, `project`, `domain`, and so on.

```
{
    "receiver": {
        "action": "CLUSTER_SCALE_OUT",
        "actor": {
            "trust_id": [
                "6dc6d336e3fc4c0a951b5698cd1236d9"
            ]
        },
        "channel": {
            "alarm_url": "http://node1:8778/v1/webhooks/
e03dd2e5-8f2e-4ec1-8c6a-74ba891e5422/trigger?V=1&count=1"
        },
        "cluster_id": "ae63a10b-4a90-452c-aef1-113a0b255ee3",
        "created_at": "2015-06-27T05:09:43",
        "domain": "Default",
        "id": "573aa1ba-bf45-49fd-907d-6b5d6e6adfd3",
        "name": "cluster_inflate",
        "params": {
            "count": "1"
        },
        "project": "6e18cc2bdbeb48a5b3cad2dc499f6804",
        "type": "webhook",
        "updated_at": null,
        "user": "b4ad2d6e18cc2b9c48049f6dbe8a5b3c"
    }
}
```

3.12.4. Delete receiver

Method	URI	Description
DELETE	`/v1/receivers/{receiver_id}`	Deletes a receiver.

Normal response codes: 204

3.12.4.1. Request

This table shows the URI parameters for the delete receiver request:

Name	Type	Description
`{receiver_id}`	UUID	The UUID of the receiver.

This operation does not accept a request body.

3.13. Webhooks

Triggers an action represented by a webhook.

Method	URI	Description
POST	`/v1/webhooks/{webhook_id}/trigger{?V,params}`	Triggers a webhook receiver.

3.13.1. Trigger webhook action

Method	URI	Description
POST	`/v1/webhooks/{webhook_id}/trigger{?V,params}`	Triggers a webhook receiver.

Normal response codes: 202

3.13.1.1. Request

This table shows the URI parameters for the trigger webhook action request:

Name	Type	Description
`{webhook_id}`	UUID	The UUID of the webhook.

This table shows the query parameters for the trigger webhook action request:

Name	Type	Description
V	String *(Required)*	The webhook implementation version requested.
`params`	Dict *(Optional)*	The query string that forms the inputs to use for the targeted action.

This operation does not accept a request body.

3.13.1.2. Response

The `Location` response header contains the action that the request initiated.

This operation does not return a response body.

4. Database Service API v1.0 (CURRENT)

Method	URI	Description
API versions		
GET	`/`	Lists information about all Database Service API versions.
GET	`/v1.0`	Shows details for the Database Service API v1.0.
Database instances (instances)		
POST	`/v1.0/{accountId}/instances`	Creates a database instance.
GET	`/v1.0/{accountId}/instances`	Lists information, including status, for all database instances.
GET	`/v1.0/{accountId}/instances/{instanceId}`	Shows database instance details.
PUT	`/v1.0/{accountId}/instances/{instanceId}`	Attaches a configuration group to an instance.
PUT	`/v1.0/{accountId}/instances/{instanceId}`	Detaches a configuration group from an instance.
DELETE	`/v1.0/{accountId}/instances/{instanceId}`	Deletes a database instance, including any associated data.
PATCH	`/v1.0/{accountId}/instances/{instanceId}`	Detaches a replica from its replication source.
GET	`/v1.0/{accountId}/instances/{instanceId}/configuration`	Lists the configuration defaults for an instance.
Database instance actions (action)		
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Resizes the memory for an instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Resizes the volume that is attached to an instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Restarts the database service for an instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Promotes a replica.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Deletes the base instance in a replication set.
Databases (databases)		
POST	`/v1.0/{accountId}/instances/{instanceId}/databases`	Creates a database within an instance.
GET	`/v1.0/{accountId}/instances/{instanceId}/databases`	Lists databases for an instance.
DELETE	`/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}`	Deletes a database.
Users (users)		
POST	`/v1.0/{accountId}/instances/{instanceId}/users`	Creates a user for a database instance.
GET	`/v1.0/{accountId}/instances/{instanceId}/users`	Lists the users in a database instance and the associated databases for that user.
DELETE	`/v1.0/{accountId}/instances/{instanceId}/users/{name}`	Deletes a user for a database instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/root`	Enables the root user for a database instance and returns the root password.
GET	`/v1.0/{accountId}/instances/{instanceId}/root`	Shows root-enabled status for a database instance.

Method	URI	Description
DELETE	`/v1.0/{accountId}/instances/{instanceId}/root`	Disables the root user.
Flavors (flavors)		
GET	`/v1.0/{accountId}/flavors`	Lists information for all available flavors.
GET	`/v1.0/{accountId}/flavors/{flavorId}`	Shows flavor details with details of the RAM.
Data stores (datastores)		
GET	`/v1.0/{accountId}/datastores/{datastore_name}/versions`	Lists the available versions of a data store.
GET	`/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters`	Lists the available configuration parameters for a data store version.
GET	`/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters/{parameter_name}`	Displays details for a configuration parameter associated with a data store version.
Configuration groups (configurations)		
POST	`/v1.0/{accountId}/configurations`	Creates a configuration group.
GET	`/v1.0/{accountId}/configurations`	Lists all configuration groups.
DELETE	`/v1.0/{accountId}/configurations/{configId}`	Deletes a configuration group.
PATCH	`/v1.0/{accountId}/configurations/{configId}`	Sets new values for a configuration group.
GET	`/v1.0/{accountId}/configurations/{configId}`	Lists details about a configuration group, including its values.
PUT	`/v1.0/{accountId}/configurations/{configId}`	Sets new values for a configuration group. Also lets you change the name and description of the configuration group.
GET	`/v1.0/{accountId}/configurations/{configId}/instances`	Lists the instances associated with the specified configuration group.

4.1. API versions

Lists information for all Database Service API versions and shows Database Service v1.0 details.

Method	URI	Description
GET	`/`	Lists information about all Database Service API versions.
GET	`/v1.0`	Shows details for the Database Service API v1.0.

4.1.1. List versions

Method	URI	Description
GET	/	Lists information about all Database Service API versions.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.1.1.1. Request

Example 4.1. List versions: JSON request

```
GET / HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.1.1.2. Response

Example 4.2. List versions: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 153
Date: Wed, 25 Jan 2012 21:53:04 GMT
```

```
{
    "versions": [
        {
            "id": "v1.0",
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/",
                    "rel": "self"
                }
            ],
            "status": "CURRENT",
            "updated": "2012-01-01T00:00:00Z"
        }
    ]
}
```

4.1.2. Show version details

Method	URI	Description
GET	`/v1.0`	Shows details for the Database Service API v1.0.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.1.2.1. Request

Example 4.3. Show version details: JSON request

```
GET /v1.0/ HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.1.2.2. Response

Example 4.4. Show version details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 158
Date: Tue, 22 Apr 2014 19:02:58 GMT
```

```
{
    "versions": [
        {
            "status": "CURRENT",
            "updated": "2012-08-01T00:00:00Z",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8779/v1.0/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

4.2. Database instances (instances)

Creates, lists, shows details for, attaches a configuration group to, detaches a configuration group from, deletes, lists configuration defaults, creates root, and determines whether root is enables for instances.

Method	URI	Description
POST	`/v1.0/{accountId}/instances`	Creates a database instance.
GET	`/v1.0/{accountId}/instances`	Lists information, including status, for all database instances.
GET	`/v1.0/{accountId}/instances/{instanceId}`	Shows database instance details.
PUT	`/v1.0/{accountId}/instances/{instanceId}`	Attaches a configuration group to an instance.
PUT	`/v1.0/{accountId}/instances/{instanceId}`	Detaches a configuration group from an instance.
DELETE	`/v1.0/{accountId}/instances/{instanceId}`	Deletes a database instance, including any associated data.
PATCH	`/v1.0/{accountId}/instances/{instanceId}`	Detaches a replica from its replication source.
GET	`/v1.0/{accountId}/instances/{instanceId}/configuration`	Lists the configuration defaults for an instance.

4.2.1. Create database instance

Method	URI	Description
POST	`/v1.0/{accountId}/instances`	Creates a database instance.

Asynchronously provisions a database instance. You must specify a flavor and a volume size. The service provisions the instance with a volume of the requested size, which serves as storage for the database instance.

Notes

- You can create only one database instance per **POST** request.
- You can create a database instance with one or more databases. You associate users with each database.
- The default binding for the MySQL instance is port 3306.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.2.1.1. Request

This table shows the URI parameters for the create database instance request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.

Example 4.5. Create database instance: JSON request

```
POST /v1.0/1234/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
    "instance": {
        "databases": [
            {
                "character_set": "utf8",
                "collate": "utf8_general_ci",
                "name": "sampledb"
            },
            {
                "name": "nextround"
            }
        ],
        "flavorRef": "https://openstack.example.com/v1.0/1234/flavors/1",
        "name": "json_rack_instance",
```

```
        "users": [
            {
                "databases": [
                    {
                        "name": "sampledb"
                    }
                ],
                "name": "demouser",
                "password": "secretsecret"
            }
        ],
        "volume": {
            "size": 2
        }
    }
}
```

4.2.1.2. Response

Example 4.6. Create database instance: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 636
Date: Wed, 25 Jan 2012 21:53:10 GMT
```

```
{
    "instance": {
        "created": "2012-01-25T21:53:09Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/flavors/
1",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/flavors/1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostname": "e09ad9a3f73309469cf1f43d11e79549caf9acf2.
rackspaceclouddb.com",
        "id": "dea5a2f7-3ec7-4496-adab-0abb5a42d635",
        "links": [
            {
                "href": "https://openstack.example.com/v1.0/1234/instances/
dea5a2f7-3ec7-4496-adab-0abb5a42d635",
                "rel": "self"
            },
            {
                "href": "https://openstack.example.com/instances/
dea5a2f7-3ec7-4496-adab-0abb5a42d635",
                "rel": "bookmark"
            }
        ],
        "name": "json_rack_instance",
        "status": "BUILD",
```

```
        "updated": "2012-01-25T21:53:10Z",
        "volume": {
            "size": 2
        }
    }
}
```

The previous response examples show resources that contain links to themselves that enable a client to easily obtain resource URIs rather than construct them. Resources can have these link relations:

- A `self` link contains a *versioned* link to the resource. Use these links in cases where the link will be followed immediately.
- A `bookmark` link provides a permanent link to a resource that is appropriate for long-term storage.

4.2.2. List database instances

Method	URI	Description
GET	`/v1.0/{accountId}/instances`	Lists information, including status, for all database instances.

Lists status and information for all database instances.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.2.2.1. Request

This table shows the URI parameters for the list database instances request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.

Example 4.7. List database instances: JSON request

```
GET /v1.0/1234/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.2.2.2. Response

Example 4.8. List database instances: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1150
Date: Tue, 19 Jun 2012 19:53:04 GMT
```

```
{
    "instances": [
        {
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "https://openstack.example.com/v1.0/1234/
flavors/1",
                        "rel": "self"
                    },
                    {
                        "href": "https://openstack.example.com/flavors/1",
```

```
                        "rel": "bookmark"
                    }
                ]
            },
            "id": "28d1b8f3-172a-4f6d-983d-36021508444a",
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/
instances/28d1b8f3-172a-4f6d-983d-36021508444a",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/instances/
28d1b8f3-172a-4f6d-983d-36021508444a",
                    "rel": "bookmark"
                }
            ],
            "name": "json_rack_instance",
            "status": "ACTIVE",
            "volume": {
                "size": 2
            }
        },
        {
            "flavor": {
                "id": "1",
                "links": [
                    {
                        "href": "https://openstack.example.com/v1.0/1234/
flavors/1",
                        "rel": "self"
                    },
                    {
                        "href": "https://openstack.example.com/flavors/1",
                        "rel": "bookmark"
                    }
                ]
            },
            "id": "8fb081af-f237-44f5-80cc-b46be1840ca9",
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/
instances/8fb081af-f237-44f5-80cc-b46be1840ca9",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/instances/8fb081af-
f237-44f5-80cc-b46be1840ca9",
                    "rel": "bookmark"
                }
            ],
            "name": "xml_rack_instance",
            "status": "ACTIVE",
            "volume": {
                "size": 2
            }
        }
    ]
}
```

4.2.3. Show database instance details

Method	URI	Description
GET	`/v1.0/{accountId}/instances/{instanceId}`	Shows database instance details.

Lists the status and details of the database instance.

Lists the volume size in gigabytes (GB) and the approximate GB used.

Note

After instance creation, the `used` value is greater than 0, which is expected and due to the automatic creation of non-empty transaction logs for MySQL optimization. The response does not include the `used` attribute when the instance status is `BUILD`, `REBOOT`, `RESIZE`, or `ERROR`.

The list operations return a DNS-resolvable host name for the database instance rather than an IP address. Because the host name always resolves to the correct IP address for the database instance, you do not need to maintain the mapping. Although the IP address might change when you resize, migrate, or perform other operations, the host name always resolves to the correct database instance.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.2.3.1. Request

This table shows the URI parameters for the show database instance details request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.9. Show database instance details: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.2.3.2. Response

Example 4.10. Show database instance details: JSON response

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
Content-Length: 685
Date: Wed, 28 Mar 2012 21:37:29 GMT
```


```
{
    "instance": {
        "created": "2012-03-28T21:31:02Z",
        "flavor": {
            "id": "1",
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/flavors/
1",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/flavors/1",
                    "rel": "bookmark"
                }
            ]
        },
        "hostname": "e09ad9a3f73309469cf1f43d11e79549caf9acf2.
rackspaceclouddb.com",
        "id": "2450c73f-7805-4afe-a42c-4094ab42666b",
        "links": [
            {
                "href": "https://openstack.example.com/v1.0/1234/instances/
2450c73f-7805-4afe-a42c-4094ab42666b",
                "rel": "self"
            },
            {
                "href": "https://openstack.example.com/instances/
2450c73f-7805-4afe-a42c-4094ab42666b",
                "rel": "bookmark"
            }
        ],
        "name": "xml_rack_instance",
        "status": "ACTIVE",
        "updated": "2012-03-28T21:34:25Z",
        "volume": {
            "size": 2,
            "used": 0.124542236328125
        }
    }
}
```

4.2.4. Attach configuration group

Method	URI	Description
PUT	`/v1.0/{accountId}/instances/{instanceId}`	Attaches a configuration group to an instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.2.4.1. Request

This table shows the URI parameters for the attach configuration group request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.11. Attach configuration group: JSON request

```
PUT /v1.0/1234/instances/4c93c73b-d6d0-47d7-b8c6-b699d19d7de9 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "instance": {
        "configuration": "2aa51628-5c42-4086-8682-137caffd2ba6"
    }
}
```

4.2.4.2. Response

Example 4.12. Attach configuration group: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 13 Jul 2015 19:53:04 GMT
```

This operation does not return a response body.

4.2.5. Detach configuration group

Method	URI	Description
PUT	`/v1.0/{accountId}/instances/{instanceId}`	Detaches a configuration group from an instance.

When you pass in only an instance ID and omit the configuration ID, this operation detaches any configuration group that was attached to the instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.2.5.1. Request

This table shows the URI parameters for the detach configuration group request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.13. Detach configuration group: JSON request

```
PUT /v1.0/1234/instances/4c93c73b-d6d0-47d7-b8c6-b699d19d7de9 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "instance": {}
}
```

4.2.5.2. Response

Example 4.14. Detach configuration group: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 13 Jul 2015 19:53:04 GMT
```

This operation does not return a response body.

4.2.6. Delete database instance

Method	URI	Description
DELETE	`/v1.0/{accountId}/instances/{instanceId}`	Deletes a database instance, including any associated data.

Note

This operation does not delete any read slaves.

Note

You cannot complete this operation when the instance state is either `REBUILDING` or `BUILDING`.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), 422, itemNotFound (404)

4.2.6.1. Request

This table shows the URI parameters for the delete database instance request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.15. Delete database instance: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.2.7. Detach replica

Method	URI	Description
PATCH	`/v1.0/{accountId}/instances/{instanceId}`	Detaches a replica from its replication source.

If you created an instance that is a replica of a source instance, you can detach the replica from the source. This can be useful if the source becomes unavailable. In this case, you can detach the replica from the source, making the replica a standalone database instance. You can then take the new standalone instance and create a new replica of that instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.2.7.1. Request

This table shows the URI parameters for the detach replica request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.16. Detach replica: JSON request

```
PATCH /v1.0/1234/instances/4c93c73b-d6d0-47d7-b8c6-b699d19d7de9 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "instance": {
        "replica_of": null,
        "slave_of": null
    }
}
```

4.2.7.2. Response

Example 4.17. Detach replica: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 13 Jul 2015 19:53:04 GMT
```

This operation does not return a response body.

4.2.8. List configuration defaults

Method	URI	Description
GET	`/v1.0/{accountId}/instances/{instanceId}/configuration`	Lists the configuration defaults for an instance.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.2.8.1. Request

This table shows the URI parameters for the list configuration defaults request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.18. List configuration defaults: JSON request

```
GET /v1.0/1234/instances/instance_1 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.2.8.2. Response

Example 4.19. List configuration defaults: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1559
Date: Wed, 08 Jul 2015 19:53:04 GMT
```

```
{
    "instance": {
        "configuration": {
            "tmp_table_size": "16M",
            "innodb_log_files_in_group": "2",
            "skip-external-locking": "1",
            "read_rnd_buffer_size": "512 K",
            "max_user_connections": "100",
            "max_heap_table_size": "16M",
            "port": "3306",
            "tmpdir": "/var/tmp",
            "pid_file": "/var/run/mysqld/mysqld.pid",
```

```
            "myisam-recover ": "BACKUP",
            "server_id": "334596",
            "innodb_buffer_pool_size": "150M",
            "basedir": "/usr",
            "max_allowed_packet": "1024K",
            "datadir": "/var/lib/mysql/data",
            "inno db_log_buffer_size": "25M",
            "max_connections": "100",
            "table_open_cache": "256",
            "connect_timeout": "15",
            "query_cache_type": "1",
            "local-infile": "0",
            "innodb_ log_file_size": "50M",
            "thread_stack": "192K",
            "query_cache_limit": "1M",
            "wait_timeout": "120",
            "user": "mysql",
            "thread_cache_size": "4",
            "query_cache_size": "8M",
            "innodb_data_file_path": "ibdata1:10M:autoextend",
            "default_storage_engine": "innodb",
            "sort_buffer_size": "1M",
            "table_definition_cache": "256",
            "read_bu ffer_size": "512K",
            "open_files_limit": "512",
            "innodb_file_per_table": "1",
            "key_buffer_size": "50M",
            "join_buffer_size": "1M"
        }
    }
}
```

4.3. Database instance actions (action)

Resizes instances and volumes and restarts instances.

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Resizes the memory for an instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Resizes the volume that is attached to an instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Restarts the database service for an instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Promotes a replica.
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Deletes the base instance in a replication set.

4.3.1. Resize instance

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Resizes the memory for an instance.

If you provide a valid `flavorRef`, this operation changes the memory size of the instance, and restarts MySQL.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.3.1.1. Request

This table shows the URI parameters for the resize instance request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.20. Resize instance: JSON request

```
POST /v1.0/1234/instances/23a3d4fb-3731-497b-afd4-bf25bde2b5fc/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 2eeb3252-0164-40f5-8fb7-85df5faa2698
Accept: application/json
Content-Type: application/json
```

```
{
    "resize": {
        "flavorRef": "https://openstack.example.com/v1.0/1234/flavors/2"
    }
}
```

4.3.1.2. Response

Example 4.21. Resize instance: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: text/plain; charset=UTF-8
Content-Length: 58
Date: Mon, 06 Feb 2012 21:28:10 GMT
```

This operation does not return a response body.

4.3.2. Resize instance volume

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Resizes the volume that is attached to an instance.

You can use this operation to increase but not decrease the volume size. A valid volume size is an integer value in gigabytes (GB).

Note

You cannot increase the volume to a size that is larger than the API volume size limit.

If this operation succeeds, it returns a 202 Accepted response.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.3.2.1. Request

This table shows the URI parameters for the resize instance volume request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.22. Resize instance volume: JSON request

```
POST /v1.0/1234/instances/23a3d4fb-3731-497b-afd4-bf25bde2b5fc/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 2eeb3252-0164-40f5-8fb7-85df5faa2698
Accept: application/json
Content-Type: application/json
```

```
{
    "resize": {
        "volume": {
            "size": 4
        }
    }
}
```

4.3.2.2. Response

Example 4.23. Resize instance volume: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
```

```
Content-Length: 0
Date: Wed, 27 Jun 2012 23:12:20 GMT
```

This operation does not return a response body.

4.3.3. Restart instance

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Restarts the database service for an instance.

The restart operation restarts only the MySQL instance. Restarting MySQL erases any dynamic configuration settings that you make in MySQL.

Note

The MySQL service is unavailable until the instance restarts.

If the operation succeeds, it returns the `Accepted (202)` response code.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.3.3.1. Request

This table shows the URI parameters for the restart instance request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.24. Restart instance: JSON request

```
POST /v1.0/1234/instances/13d940c4-70bb-4ff4-8866-6ee9ab5e5cae/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "restart": {}
}
```

4.3.3.2. Response

Example 4.25. Restart instance: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:19 GMT
```

This operation does not return a response body.

4.3.4. Promote instance to replica source

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Promotes a replica.

If you have set up replication, and the base instance is still reachable, you can use this operation to promote a replica to be the new base instance.

This can be useful if you want to make a configuration change to the base instance that your replicas are replicating from. For example, you might want to increase the disk or CPU capacity. If you made the change on the base instance directly, you would need to take the base instance down for the duration of the operation. Instead, you can create a replica, make the configuration change on the replica, and then promote the replica to become the new base instance.

For `instanceId`, pass in the instance ID of the replica you want to promote.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.3.4.1. Request

This table shows the URI parameters for the promote instance to replica source request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.26. Promote instance to replica source: JSON request

```
POST /v1.0/1234/instances/13d940c4-70bb-4ff4-8866-6ee9ab5e5cae/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "promote_to_replica_source": {}
}
```

4.3.4.2. Response

Example 4.27. Promote instance to replica source: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Fri, 08 Apr 2016 23:11:19 GMT
```

This operation does not return a response body.

4.3.5. Delete replication base instance

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/action`	Deletes the base instance in a replication set.

If the base instance becomes unreachable, you can use this operation to delete the base instance.

This operation:

- Finds the replica that has processed the greatest number of transactions and picks that replica to use as the new base instance.
- Transfers the public IP of the old base instance to the new base instance (which is the newly-promoted replica).
- Deletes the old base instance.
- Takes all the instances in the replication set and makes them start replicating from the new base instance.

For `instanceId`, pass in the instance ID of the unreachable base instance.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404), badMediaType (415)

4.3.5.1. Request

This table shows the URI parameters for the delete replication base instance request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.28. Delete replication base instance: JSON request

```
POST /v1.0/1234/instances/13d940c4-70bb-4ff4-8866-6ee9ab5e5cae/action HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "eject_replica_source": {}
}
```

4.3.5.2. Response

Example 4.29. Delete replication base instance: JSON response

```
HTTP/1.1 202 Accepted
```

```
Content-Type: application/json
Content-Length: 0
Date: Fri, 08 Apr 2016 23:11:19 GMT
```

This operation does not return a response body.

4.4. Databases (databases)

Creates, lists all, and deletes databases.

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/databases`	Creates a database within an instance.
GET	`/v1.0/{accountId}/instances/{instanceId}/databases`	Lists databases for an instance.
DELETE	`/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}`	Deletes a database.

4.4.1. Create database

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/databases`	Creates a database within an instance.

The `name` of the database is a required attribute.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.4.1.1. Request

This table shows the URI parameters for the create database request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.30. Create database: JSON request

```
POST /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/databases HTTP/
1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

```
{
    "databases": [
        {
            "character_set": "utf8",
            "collate": "utf8_general_ci",
            "name": "testingdb"
        },
        {
            "name": "sampledb"
        }
    ]
}
```

4.4.1.2. Response

Example 4.31. Create database: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

4.4.2. List instance databases

Method	URI	Description
GET	`/v1.0/{accountId}/instances/{instanceId}/databases`	Lists databases for an instance.

Note

This operation returns only the user-defined databases and not the system databases. Only the database administrator can view the `mysql`, `information_schema`, and `lost+found` system databases.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.4.2.1. Request

This table shows the URI parameters for the list instance databases request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.32. List instance databases: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/databases HTTP/
1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.4.2.2. Response

Example 4.33. List instance databases: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 136
Date: Wed, 25 Jan 2012 21:58:01 GMT
```

```
{
    "databases": [
        {
            "name": "anotherexampledb"
        },
        {
```

```
            "name": "exampledb"
        },
        {
            "name": "nextround"
        },
        {
            "name": "sampledb"
        },
        {
            "name": "testingdb"
        }
    ]
}
```

4.4.3. Delete database

Method	URI	Description
DELETE	`/v1.0/{accountId}/instances/{instanceId}/databases/{databaseName}`	Deletes a database.

This operation also deletes all data that is associated with the database.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.4.3.1. Request

This table shows the URI parameters for the delete database request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.
`{databaseName}`	String	The name for the database.

Example 4.34. Delete database: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/databases/
exampledb HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.4.3.2. Response

Example 4.35. Delete database: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

4.5. Users (users)

Creates, lists all, and deletes users.

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/users`	Creates a user for a database instance.
GET	`/v1.0/{accountId}/instances/{instanceId}/users`	Lists the users in a database instance and the associated databases for that user.
DELETE	`/v1.0/{accountId}/instances/{instanceId}/users/{name}`	Deletes a user for a database instance.
POST	`/v1.0/{accountId}/instances/{instanceId}/root`	Enables the root user for a database instance and returns the root password.
GET	`/v1.0/{accountId}/instances/{instanceId}/root`	Shows root-enabled status for a database instance.
DELETE	`/v1.0/{accountId}/instances/{instanceId}/root`	Disables the root user.

4.5.1. Create user

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/users`	Creates a user for a database instance.

Asynchronously provisions a new user for the database instance by using the configuration that you define in the request object. After the API validates the request and starts progress on the provisioning process, the call returns the `Accepted (202)` response code.

If the API cannot fulfill the corresponding request due to insufficient data or data that is not valid, the API returns the `Bad Request (400)`response code with information about the nature of the failure. You cannot recover from validation errors. You must correct the cause of the failure and the request again.

This table lists the required attributes for creating users:

Table 4.1. Required attributes for user

Applies to	Name	Description	Required
User	name	Name of the user for the database.	Yes
	password	User password for database access.	Yes
	(database) name	Name of the database that the user can access. You must specify one or more database names.	No

Notes

- The operation grants the user all privileges on the databases.
- Do not use the `root` user name, which is reserved.

These tables list the valid characters for database names, user names, and passwords.

Table 4.2. Valid characters in database name, user name, and password

Character
Letters (upper and lower cases allowed)
Numbers
@, ?, #, and spaces are allowed, but *not* at the beginning and end of the database name, user name, and password
_ is allowed anywhere in the database name, user name, and password

Table 4.3. Characters that are not allowed in database name, user name, and password

Character
Single quotes

Character
Double quotes
Back quotes
Semicolons
Commas
Back slashes
Forward slashes
Spaces at the front or end of the user name or password

Table 4.4. Length restrictions for database name, user name, and password

Restriction	Value
Database name maximum length	64
User name maximum length	16
Password maximum length	unlimited (no restrictions)

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.5.1.1. Request

This table shows the URI parameters for the create user request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.36. Create user: JSON request

```
POST /v1.0/1234/instances/1c59bdb8-03b6-4079-a7db-ba92d23a98b3/users HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: bb64d788-2dec-4a6b-a670-7151d108cacf
Accept: application/json
Content-Type: application/json
```

```
{
    "users": [
        {
            "databases": [
                {
                    "name": "databaseA"
                }
            ],
            "name": "dbuser3",
            "password": "secretsecret"
        },
        {
            "databases": [
                {
```

```
                    "name": "databaseB"
                },
                {
                    "name": "databaseC"
                }
            ],
            "name": "dbuser4",
            "password": "secretsecret"
        }
    ]
}
```

4.5.1.2. Response

Example 4.37. Create user: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

4.5.2. List database instance users

Method	URI	Description
GET	`/v1.0/{accountId}/instances/{instanceId}/users`	Lists the users in a database instance and the associated databases for that user.

Note

This operation does not return system users. A system user is a database administrator who administers the health of the database. Also, this operation returns the `root` user only if it is enabled.

The following notes apply to MySQL users:

- User names can be up to 16 characters long.
- When you create accounts with INSERT, you must use FLUSH PRIVILEGES to tell the server to reload the grant tables.
- For additional information, See: http://dev.mysql.com/doc/refman/5.1/en/user-account-management.html

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.5.2.1. Request

This table shows the URI parameters for the list database instance users request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.38. List database instance users: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/users HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.5.2.2. Response

Example 4.39. List database instance users: JSON response

```
HTTP/1.1 200 OK
```

```
Content-Type: application/json
Content-Length: 152
Date: Wed, 21 Mar 2012 17:46:46 GMT
```


```
{
    "users": [
        {
            "databases": [
                {
                    "name": "databaseA"
                }
            ],
            "name": "dbuser3"
        },
        {
            "databases": [
                {
                    "name": "databaseB"
                },
                {
                    "name": "databaseC"
                }
            ],
            "name": "dbuser4"
        }
    ]
}
```

4.5.3. Delete user

Method	URI	Description
DELETE	`/v1.0/{accountId}/instances/{instanceId}/users/{name}`	Deletes a user for a database instance.

Warning

Do not use periods in user names. A bug in a Python library that Rackspace uses that can cause incorrect user deletions to occur if you use a period (.) in the user name. In this case, the bug in the library truncates the user name to the portion from the beginning up to the period. For example, for the `my.userA` user, the bug truncates the user name to `my`, and if the `user` exists, that user is incorrectly deleted.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.5.3.1. Request

This table shows the URI parameters for the delete user request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.
`{userName}`	String	The name for the user.

Example 4.40. Delete user: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/users/
testuser HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.5.3.2. Response

Example 4.41. Delete user: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:19 GMT
```

This operation does not return a response body.

4.5.4. Enable root user

Method	URI	Description
POST	`/v1.0/{accountId}/instances/{instanceId}/root`	Enables the root user for a database instance and returns the root password.

This operation generates a root password for the root user and enables the root user to log in from any host.

Note

Changes that you make as a root user can impact the database instance and API operations in unpredictable and detrimental ways. When you enable the root user, you accept the possibility that we cannot support your database instance. We might not be able to assist you if you change core MySQL settings. These changes can be, but are not limited to, turning off bin logs, removing users that we use to access your instance, and so on.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.5.4.1. Request

This table shows the URI parameters for the enable root user request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.42. Enable root user: JSON request

```
POST /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/root HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.5.4.2. Response

Example 4.43. Enable root user: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 78
```

```
Date: Wed, 25 Jan 2012 21:58:11 GMT
```

```
{
    "user": {
        "name": "root",
        "password": "secretsecret"
    }
}
```

4.5.5. Show root-enabled status for database instance

Method	URI	Description
GET	`/v1.0/{accountId}/instances/{instanceId}/root`	Shows root-enabled status for a database instance.

Returns `true` if root user is enabled for a database instance. Otherwise, returns `false`.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.5.5.1. Request

This table shows the URI parameters for the show root-enabled status for database instance request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.44. Show root-enabled status for database instance: JSON request

```
GET /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/root HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.5.5.2. Response

Example 4.45. Show root-enabled status for database instance: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 21
Date: Wed, 25 Jan 2012 21:58:13 GMT
```

```
{
    "rootEnabled": true
}
```

4.5.6. Disable root user

Method	URI	Description
DELETE	`/v1.0/{accountId}/instances/{instanceId}/root`	Disables the root user.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.5.6.1. Request

This table shows the URI parameters for the disable root user request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{instanceId}`	String	The ID for the database instance.

Example 4.46. Disable root user: JSON request

```
DELETE /v1.0/1234/instances/692d8418-7a8f-47f1-8060-59846c6e024f/root HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.5.6.2. Response

Example 4.47. Disable root user: JSON response

```
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 0
Date: Wed, 27 Jun 2012 23:11:18 GMT
```

This operation does not return a response body.

4.6. Flavors (flavors)

Lists all flavors and shows details for a flavor, by ID.

Method	URI	Description
GET	`/v1.0/{accountId}/flavors`	Lists information for all available flavors.
GET	`/v1.0/{accountId}/flavors/{flavorId}`	Shows flavor details with details of the RAM.

4.6.1. List flavors

Method	URI	Description
GET	`/v1.0/{accountId}/flavors`	Lists information for all available flavors.

This operation lists information for all available flavors.

This resource is identical to the flavors found in the OpenStack Nova API, but without the disk property.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.6.1.1. Request

This table shows the URI parameters for the list flavors request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.

Example 4.48. List flavors: JSON request

```
GET /v1.0/1234/flavors HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.6.1.2. Response

Example 4.49. List flavors: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1768
Date: Tue, 19 Jun 2012 19:52:45 GMT
```

```
{
    "flavors": [
        {
            "id": 1,
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/flavors/
1",
                    "rel": "self"
```

```
                },
                {
                    "href": "https://openstack.example.com/flavors/1",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.tiny",
            "ram": 512
        },
        {
            "id": 2,
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/flavors/
2",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/flavors/2",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.small",
            "ram": 1024
        },
        {
            "id": 3,
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/flavors/
3",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/flavors/3",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.medium",
            "ram": 2048
        },
        {
            "id": 4,
            "links": [
                {
                    "href": "https://openstack.example.com/v1.0/1234/flavors/
4",
                    "rel": "self"
                },
                {
                    "href": "https://openstack.example.com/flavors/4",
                    "rel": "bookmark"
                }
            ],
            "name": "m1.large",
            "ram": 4096
        }
    ]
}
```

4.6.2. Show flavor details

Method	URI	Description
GET	`/v1.0/{accountId}/flavors/{flavorId}`	Shows flavor details with details of the RAM.

This resource is identical to the flavors found in the OpenStack Compute API, but without the disk property.

Note

The `flavorId` parameter must be an integer value. If you use a floating point value for this parameter, this call truncates the decimal portion and uses the integer portion as the `flavorId` value.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.6.2.1. Request

This table shows the URI parameters for the show flavor details request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{flavorId}`	String	The ID for the flavor.

Example 4.50. Show flavor details: JSON request

```
GET /v1.0/1234/flavors/1 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.6.2.2. Response

Example 4.51. Show flavor details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 209
Date: Wed, 25 Jan 2012 21:53:05 GMT
```

```
{
    "flavor": {
```

```
        "id": 1,
        "links": [
            {
                "href": "https://openstack.example.com/v1.0/1234/flavors/1",
                "rel": "self"
            },
            {
                "href": "https://openstack.example.com/flavors/1",
                "rel": "bookmark"
            }
        ],
        "name": "m1.tiny",
        "ram": 512
    }
}
```

4.7. Data stores (datastores)

Lists data store versions, lists parameters for data stores, and shows parameter details for a data store version.

Method	URI	Description
GET	`/v1.0/{accountId}/datastores/{datastore_name}/versions`	Lists the available versions of a data store.
GET	`/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters`	Lists the available configuration parameters for a data store version.
GET	`/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters/{parameter_name}`	Displays details for a configuration parameter associated with a data store version.

4.7.1. List datastore versions

Method	URI	Description
GET	`/v1.0/{accountId}/datas-tores/{datastore_name}/versions`	Lists the available versions of a data store.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.7.1.1. Request

This table shows the URI parameters for the list datastore versions request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{datastore_name}`	String	The name of the data store.

Example 4.52. List datastore versions: JSON request

```
GET /v1.0/1234/datastores/mysql/versions HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.7.1.2. Response

Example 4.53. List datastore versions: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 580
Date: Tue, 23 Jun 2015 21:58:13 GMT
```

```
{
    "name": "5.6",
    "links": [
        {
            "href": "https://10.240.28.38:8779/v1.0/
27bee406abb5486e81ef3ff4382aabaf/datastores/versions/2dc7faa0-efff-4c2b-8cff-
bcd949c518a5",
            "rel": "self"
        },
        {
            "href": "https://10.240.28.38:8779/datastores/versions/2dc7faa0-
efff-4c2b-8cff-bcd949c518a5",
            "rel": "bookmark"
        }
```

```
    ],
    "image": "b69fbd9e-b31d-46ff-8afb-cbf452f6f835",
    "active": 1,
    "datastore": "3a8968d8-e5f5-4452-83ca-f6c90b5de06a",
    "packages": "mysql-server-5.6",
    "id": "2dc7faa0-efff-4c2b-8cff-bcd949c518a5"
}
```

4.7.2. List configuration parameters

Method	URI	Description
GET	`/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters`	Lists the available configuration parameters for a data store version.

Parameter information includes the type, minimum and maximum values, and whether you must restart the instance after you change a parameter value.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.7.2.1. Request

This table shows the URI parameters for the list configuration parameters request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{datastore_version_id}`	UUID	The UUID of the data store version.

Example 4.54. List configuration parameters: JSON request

```
GET /v1.0/1234/datastores/versions/692d8418-7a8f-47f1-8060-59846c6e024f/
parameters HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.7.2.2. Response

Example 4.55. List configuration parameters: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 8454
Date: Fri, 12 Jun 2015 21:58:13 GMT
```

```
{
    "configuration-parameters": [
        {
            "name": "myisam_sort_buffer_size",
            "min": 4096,
            "max": 18446744073709552000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "sync_binlog",
```

```
            "min": 0,
            "max": 4294967295,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "max_allowed_packet",
            "min": 1024,
            "max": 1073741824,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "type": "string",
            "name": "character_set_connection",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "name": "autocommit",
            "min": 0,
            "max": 1,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "type": "string",
            "name": "character_set_client",
            "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "name": "join_buffer_size",
            "min": 128,
            "max": 18446744073709548000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "local_infile",
            "min": 0,
            "max": 1,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "auto_increment_offset",
            "min": 1,
            "max": 65535,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "max_connections",
```

```
            "min": 1,
            "max": 100000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "bulk_insert_buffer_size",
            "min": 0,
            "max": 18446744073709552000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef474dac18-2c6abd35a758"
        },
        {
            "name": "sort_buffer_size",
            "min": 32768,
            "max": 18446744073709552000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "innodb_log_buffer_size",
            "min": 262144,
            "max": 4294967295,
            "restart_required": true,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "innodb_file_per_table",
            "min": 0,
            "max": 1,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "type": "string",
            "name": "character_set_server",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "name": "innodb_buffer_pool_size",
            "min": 5242880,
            "max": 18446744073709552000,
            "restart_required": true,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef474dac18-2c6abd35a758"
        },
        {
            "type": "string",
            "name": "collation_server",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "type": "string",
```

```
            "name": "character_set_filesystem",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "type": "string",
            "name": "collation_database",
            "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "name": "innodb_flush_log_at_trx_commit",
            "min": 0,
            "max": 2,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "interactive_timeout",
            "min": 1,
            "max": 65535,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "max_user_connections",
            "min": 0,
            "max": 4294967295,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "innodb_thread_concurrency",
            "min": 0,
            "max": 1000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "innodb_open_files",
            "min": 10,
            "max": 4294967295,
            "restart_required": true,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "key_buffer_size",
            "min": 8,
            "max": 4294967295,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758"
        },
        {
            "name": "connect_timeout",
```

```
            "min": 2,
            "max": 31536000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758"
        },
        {
            "type": "string",
            "name": "collation_connection",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "type": "string",
            "name": "character_set_database",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "name": "auto_increment_increment",
            "min": 1,
            "max": 65535,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "max_connect_errors",
            "min": 1,
            "max": 18446744073709552000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "type": "string",
            "name": "character_set_results",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758",
            "restart_required": false
        },
        {
            "name": "expire_logs_days",
            "min": 0,
            "max": 99,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        },
        {
            "name": "wait_timeout",
            "min": 1,
            "max": 31536000,
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef474d-ac18-2c6abd35a758"
        },
        {
            "name": "server_id",
            "min": 0,
            "max": 4294967295,
```

```
            "restart_required": false,
            "type": "integer",
            "datastore_version_id": "f597f709-70ef-474d-ac18-2c6abd35a758"
        }
    ]
}
```

4.7.3. Show configuration parameter details

Method	URI	Description
GET	`/v1.0/{accountId}/datastores/versions/{datastore_version_id}/parameters/{parameter_name}`	Displays details for a configuration parameter associated with a data store version.

Details include the type, minimum and maximum values, and whether you must restart the instance after you change the parameter value.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.7.3.1. Request

This table shows the URI parameters for the show configuration parameter details request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{datastore_version_id}`	UUID	The UUID of the data store version.
`{parameter_name}`	String	The name of the parameter for which to show details.

Example 4.56. Show configuration parameter details: JSON request

```
GET /v1.0/1234/datastores/versions/f8e67741-e767-4137-b394-3fb8a3fafd2f/
parameters/connect_timeout HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.7.3.2. Response

Example 4.57. Show configuration parameter details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 180
Date: Tue, 16 Jun 2015 21:58:13 GMT
```

```
{
    "name": "connect_timeout",
    "min": 2,
    "max": 31536000,
    "restart_required": false,
    "type": "integer",
    "datastore_version_id": "f8e67741-e767-4137-b394-3fb8a3fafd2f"
}
```

4.8. Configuration groups (configurations)

Creates and lists all configuration groups.

Method	URI	Description
POST	`/v1.0/{accountId}/configurations`	Creates a configuration group.
GET	`/v1.0/{accountId}/configurations`	Lists all configuration groups.
DELETE	`/v1.0/{accountId}/configurations/{configId}`	Deletes a configuration group.
PATCH	`/v1.0/{accountId}/configurations/{configId}`	Sets new values for a configuration group.
GET	`/v1.0/{accountId}/configurations/{configId}`	Lists details about a configuration group, including its values.
PUT	`/v1.0/{accountId}/configurations/{configId}`	Sets new values for a configuration group. Also lets you change the name and description of the configuration group.
GET	`/v1.0/{accountId}/configurations/{configId}/instances`	Lists the instances associated with the specified configuration group.

4.8.1. Create configuration group

Method	URI	Description
POST	`/v1.0/{accountId}/configurations`	Creates a configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.8.1.1. Request

This table shows the URI parameters for the create configuration group request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.

Example 4.58. Create configuration group: JSON request

```
POST /v1.0/1234/configurations HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
    "configuration": {
        "datastore": [
            {
                "type": "mysql"
            }
        ],
        "values": [
            {
                "sync_binlog": 1
            }
        ],
        "name": "group1"
    }
}
```

4.8.1.2. Response

Example 4.59. Create configuration group: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 360
Date: Mon, 6 Jul 2015 21:53:10 GMT
```

```
{
```

```
    "updated": "2015-07-01T16:38:27",
    "name": "group1",
    "created": "2015-07-01T16:38:27",
    "instance_count": 0,
    "values": {
        "sync_binlog": 1
    },
    "datastore_version_id": "2dc7faa0-efff-4c2b-8cff-bcd949c518a5",
    "id": "2aa51628-5c42-4086-8682-137caffd2ba6",
    "datastore_name": "mysql",
    "datastore_version_name": "5.6",
    "description": null
}
```

4.8.2. List configuration groups

Method	URI	Description
GET	`/v1.0/{accountId}/configurations`	Lists all configuration groups.

The list includes the associated data store and data store version for each configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.8.2.1. Request

This table shows the URI parameters for the list configuration groups request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.

Example 4.60. List configuration groups: JSON request

```
GET /v1.0/1234/configurations HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.8.2.2. Response

Example 4.61. List configuration groups: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 426
Date: Tue, 07 Jul 2012 19:53:04 GMT
```

```
{
    "configurations": [
        {
            "datastore_name": "mysql",
            "updated": "2015-07-01T16:38:27",
            "name": "group1",
            "created": "2015-07-01T16:38:27",
            "datastore_version_name": "5.6",
            "id": "2aa51628-5c42-4086-8682-137caffd2ba6",
            "datastore_version_id": "2dc7faa0-efff-4c2b-8cff-bcd949c518a5",
            "description": null
```

```
        }
    ]
}
```

4.8.3. Delete configuration group

Method	URI	Description
DELETE	`/v1.0/{accountId}/configurations/{configId}`	Deletes a configuration group.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), 422, itemNotFound (404)

4.8.3.1. Request

This table shows the URI parameters for the delete configuration group request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{configId}`	String	The ID of the configuration group.

Example 4.62. Delete configuration group: JSON request

```
DELETE /v1.0/1234/configurations/692d8418-7a8f-47f1-8060-59846c6e024f HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.8.4. Patch configuration group

Method	URI	Description
PATCH	`/v1.0/{accountId}/configurations/{configId}`	Sets new values for a configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.8.4.1. Request

This table shows the URI parameters for the patch configuration group request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{configId}`	String	The ID of the configuration group.

Example 4.63. Patch configuration group: JSON request

```
PATCH /v1.0/1234/configurations/5678 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
    "configuration": {
        "values": {
            "connect_timeout": 17
        }
    }
}
```

4.8.4.2. Response

Example 4.64. Patch configuration group: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 6 Jul 2015 21:53:10 GMT
```

This operation does not return a response body.

4.8.5. Show configuration group details

Method	URI	Description
GET	`/v1.0/{accountId}/configurations/{configId}`	Lists details about a configuration group, including its values.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.8.5.1. Request

This table shows the URI parameters for the show configuration group details request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{configId}`	String	The ID of the configuration group.

Example 4.65. Show configuration group details: JSON request

```
GET /v1.0/1234/configurations/5678 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.8.5.2. Response

Example 4.66. Show configuration group details: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 138
Date: Fri, 20 Nov 2015 19:53:04 GMT
```

```
{
    "configuration": {
        "datastore_name": "mysql",
        "updated": "2015-11-22T19:07:20",
        "values": {
            "connect_timeout": 17
        },
        "name": "group1",
        "created": "2015-11-20T20:51:24",
        "datastore_version_name": "5.6",
        "instance_count": 1,
        "id": "1c8a4fdd-690c-4e6e-b2e1-148b8d738770",
        "datastore_version_id": "b9f97132-467b-4f8e-b12d-947cfc223ac3",
        "description": null
```

```
    }
}
```

4.8.6. Update configuration group

Method	URI	Description
PUT	`/v1.0/{accountId}/configurations/{configId}`	Sets new values for a configuration group. Also lets you change the name and description of the configuration group.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.8.6.1. Request

This table shows the URI parameters for the update configuration group request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{configId}`	String	The ID of the configuration group.

Example 4.67. Update configuration group: JSON request

```
PUT /v1.0/1234/configurations/5678 HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: d6cafa5b-e0c7-4ab8-948e-7c95f2acd031
Accept: application/json
Content-Type: application/json
```

```
{
    "configuration": {
        "values": {
            "connect_timeout": 18
        },
        "name": "new_name",
        "description": "New description"
    }
}
```

4.8.6.2. Response

Example 4.68. Update configuration group: JSON response

```
HTTP/1.1 202 OK
Content-Type: application/json
Content-Length: 0
Date: Mon, 6 Jul 2015 21:53:10 GMT
```

This operation does not return a response body.

4.8.7. List configuration group instances

Method	URI	Description
GET	`/v1.0/{accountId}/configurations/{configId}/instances`	Lists the instances associated with the specified configuration group.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), forbidden (403), badMethod (405), overLimit (413), unprocessableEntity (422), instanceFault (500), notImplemented (501), serviceUnavailable (503), itemNotFound (404)

4.8.7.1. Request

This table shows the URI parameters for the list configuration group instances request:

Name	Type	Description
`{accountId}`	String	The account ID of the owner of the instance.
`{configId}`	String	The ID of the configuration group.

Example 4.69. List configuration group instances: JSON request

```
GET /v1.0/1234/configurations/5678/instances HTTP/1.1
User-Agent: python-example-client
Host: openstack.example.com
X-Auth-Token: 87c6033c-9ff6-405f-943e-2deb73f278b7
Accept: application/json
Content-Type: application/json
```

This operation does not accept a request body.

4.8.7.2. Response

Example 4.70. List configuration group instances: JSON response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 138
Date: Fri, 20 Nov 2015 19:53:04 GMT
```

```
{
    "instances": [
        {
            "id": "7fd2d1d6-a2ef-4a76-8c03-e233db4d86da",
            "name": "master_1"
        }
    ]
}
```

5. Data Processing API v1.1 (CURRENT)

Produce data processing operations.

Method	URI	Description
Plugins		
GET	`/v1.1/{tenant_id}/plugins`	Lists all registered plugins.
GET	`/v1.1/{tenant_id}/plugins/{plugin_name}`	Shows details for a plugin.
GET	`/v1.1/{tenant_id}/plugins/{plugin_name}/{version}`	Shows details for a plugin version.
Image registry		
GET	`/v1.1/{tenant_id}/images`	Lists all images registered in the registry.
GET	`/v1.1/{tenant_id}/images/{image_id}`	Shows details for an image.
POST	`/v1.1/{tenant_id}/images/{image_id}`	Registers an image in the registry.
DELETE	`/v1.1/{tenant_id}/images/{image_id}`	Removes an image from the registry.
POST	`/v1.1/{tenant_id}/images/{image_id}/tag`	Adds tags to an image.
POST	`/v1.1/{tenant_id}/images/{image_id}/untag`	Removes tags from an image.
Node group templates		
GET	`/v1.1/{tenant_id}/node-group-templates`	Lists available node group templates.
POST	`/v1.1/{tenant_id}/node-group-templates`	Creates a node group template.
GET	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Shows a node group template, by ID.
DELETE	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Deletes a node group template.
PUT	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Updates a node group template.
Cluster templates		
GET	`/v1.1/{tenant_id}/cluster-templates`	Lists available cluster templates.
POST	`/v1.1/{tenant_id}/cluster-templates`	Creates a cluster template.
GET	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Shows details for a cluster template.
PUT	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Updates a cluster template.
DELETE	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Deletes a cluster template.
Clusters		
GET	`/v1.1/{tenant_id}/clusters`	Lists available clusters.
POST	`/v1.1/{tenant_id}/clusters`	Creates a cluster.
POST	`/v1.1/{tenant_id}/clusters/multiple`	Creates multiple clusters.
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows details for a cluster, by ID.

Method	URI	Description
PUT	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Scales a cluster.
PATCH	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Updates a cluster.
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows provisioning progress for a cluster.
DELETE	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Deletes a cluster.
Event log		
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows provisioning progress of cluster.
Data sources		
GET	`/v1.1/{tenant_id}/data-sources`	Lists all data sources.
POST	`/v1.1/{tenant_id}/data-sources`	Creates a data source.
GET	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Shows details for a data source.
PUT	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Updates a data source.
DELETE	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Deletes a data source.
Job binary internals		
GET	`/v1.1/{tenant_id}/job-binary-internals`	Lists the available job binary internals.
PUT	`/v1.1/{tenant_id}/job-binary-internals/{name}`	Creates a job binary internal.
GET	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Shows details for a job binary internal.
DELETE	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Deletes a job binary internal.
PATCH	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Updates a job binary internal.
GET	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data`	Shows data for a job binary internal.
Job binaries		
GET	`/v1.1/{tenant_id}/job-binaries`	Lists the available job binaries.
GET	`/v1.1/{tenant_id}/job-binaries`	Shows details for a job binary.
POST	`/v1.1/{tenant_id}/job-binaries`	Creates a job binary.
PUT	`/v1.1/{tenant_id}/job-binaries`	Updates a job binary.
DELETE	`/v1.1/{tenant_id}/job-binaries`	Deletes a job binary.
GET	`/v1.1/{tenant_id}/job-binaries/{job_binary_id}/data`	Shows data for a job binary.
Jobs		
GET	`/v1.1/{tenant_id}/jobs`	Lists all jobs.
POST	`/v1.1/{tenant_id}/jobs`	Creates a job object.
GET	`/v1.1/{tenant_id}/jobs/{job_id}`	Shows details for a job.
DELETE	`/v1.1/{tenant_id}/jobs/{job_id}`	Removes a job.
PATCH	`/v1.1/{tenant_id}/jobs/{job_id}`	Updates a job object.
POST	`/v1.1/{tenant_id}/jobs/{job_id}/execute`	Runs a job.

Method	URI	Description
Job executions		
GET	`/v1.1/{tenant_id}/job-executions`	Lists available job executions.
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Shows details for a job execution, by ID.
DELETE	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Deletes a job execution.
PATCH	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Updates a job execution.
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}/refresh-status`	Refreshes the status of and shows information for a job execution.
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}/cancel`	Cancels a job execution.
Job types		
GET	`/v1.1/{tenant_id}/job-types`	Lists all job types.

5.1. Plugins

A plugin object defines the Hadoop or Spark version that it can install and which configurations can be set for the cluster.

Method	URI	Description
GET	`/v1.1/{tenant_id}/plugins`	Lists all registered plugins.
GET	`/v1.1/{tenant_id}/plugins/{plugin_name}`	Shows details for a plugin.
GET	`/v1.1/{tenant_id}/plugins/{plugin_name}/{version}`	Shows details for a plugin version.

5.1.1. List plugins

Method	URI	Description
GET	/v1.1/{tenant_id}/plugins	Lists all registered plugins.

Normal response codes: 200

5.1.1.1. Request

This operation does not accept a request body.

5.1.1.2. Response

Example 5.1. List plugins: JSON response

```
{
    "plugins": [
        {
            "name": "vanilla",
            "description": "The Apache Vanilla plugin provides the ability
 to launch upstream Vanilla Apache Hadoop cluster without any management
 consoles. It can also deploy the Oozie component.",
            "versions": [
                "1.2.1",
                "2.4.1",
                "2.6.0"
            ],
            "title": "Vanilla Apache Hadoop"
        },
        {
            "name": "hdp",
            "description": "The Hortonworks Sahara plugin automates the
 deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
            "versions": [
                "1.3.2",
                "2.0.6"
            ],
            "title": "Hortonworks Data Platform"
        },
        {
            "name": "spark",
            "description": "This plugin provides an ability to launch Spark on
 Hadoop CDH cluster without any management consoles.",
            "versions": [
                "1.0.0",
                "0.9.1"
            ],
            "title": "Apache Spark"
        },
        {
            "name": "cdh",
            "description": "The Cloudera Sahara plugin provides the ability to
 launch the Cloudera distribution of Apache Hadoop (CDH) with Cloudera Manager
 management console.",
            "versions": [
                "5",
```

```
                "5.3.0"
            ],
            "title": "Cloudera Plugin"
        }
    ]
}
```

5.1.2. Show plugin details

Method	URI	Description
GET	`/v1.1/{tenant_id}/plug-ins/{plugin_name}`	Shows details for a plugin.

Normal response codes: 200

5.1.2.1. Request

This table shows the URI parameters for the show plugin details request:

Name	Type	Description
`{plugin_name}`	String	Name of the plugin.

This operation does not accept a request body.

5.1.2.2. Response

Example 5.2. Show plugin details: JSON response

```
{
    "plugin": {
        "name": "vanilla",
        "versions": [
            "1.2.1",
            "2.4.1",
            "2.6.0"
        ],
        "title": "Vanilla Apache Hadoop",
        "description": "The Apache Vanilla plugin provides the ability
 to launch upstream Vanilla Apache Hadoop cluster without any management
 consoles. It can also deploy the Oozie component."
    }
}
```

5.1.3. Show plugin version details

Method	URI	Description
GET	`/v1.1/{tenant_id}/plug-ins/{plugin_name}/{version}`	Shows details for a plugin version.

Normal response codes: 200

5.1.3.1. Request

This table shows the URI parameters for the show plugin version details request:

Name	Type	Description
`{plugin_name}`	String	Name of the plugin.
`{version}`	String	Version of the plugin.

This operation does not accept a request body.

5.1.3.2. Response

Example 5.3. Show plugin version details: JSON response

```
{
    "plugin": {
        "name": "vanilla",
        "versions": [
            "1.2.1",
            "2.4.1",
            "2.6.0"
        ],
        "description": "The Apache Vanilla plugin provides the ability
 to launch upstream Vanilla Apache Hadoop cluster without any management
 consoles. It can also deploy the Oozie component.",
        "required_image_tags": [
            "vanilla",
            "2.6.0"
        ],
        "node_processes": {
            "JobFlow": [
                "oozie"
            ],
            "HDFS": [
                "namenode",
                "datanode",
                "secondarynamenode"
            ],
            "YARN": [
                "resourcemanager",
                "nodemanager"
            ],
            "MapReduce": [
                "historyserver"
            ],
            "Hadoop": [],
            "Hive": [
                "hiveserver"
            ]
```

```
        },
        "configs": [
            {
                "default_value": "/tmp/hadoop-${user.name}",
                "name": "hadoop.tmp.dir",
                "priority": 2,
                "config_type": "string",
                "applicable_target": "HDFS",
                "is_optional": true,
                "scope": "node",
                "description": "A base for other temporary directories."
            },
            {
                "default_value": true,
                "name": "hadoop.native.lib",
                "priority": 2,
                "config_type": "bool",
                "applicable_target": "HDFS",
                "is_optional": true,
                "scope": "node",
                "description": "Should native hadoop libraries, if present, be
 used."
            },
            {
                "default_value": 1024,
                "name": "NodeManager Heap Size",
                "config_values": null,
                "priority": 1,
                "config_type": "int",
                "applicable_target": "YARN",
                "is_optional": false,
                "scope": "node",
                "description": null
            },
            {
                "default_value": true,
                "name": "Enable Swift",
                "config_values": null,
                "priority": 1,
                "config_type": "bool",
                "applicable_target": "general",
                "is_optional": false,
                "scope": "cluster",
                "description": null
            },
            {
                "default_value": true,
                "name": "Enable MySQL",
                "config_values": null,
                "priority": 1,
                "config_type": "bool",
                "applicable_target": "general",
                "is_optional": true,
                "scope": "cluster",
                "description": null
            }
        ],
        "title": "Vanilla Apache Hadoop"
    }
}
```

5.2. Image registry

Use the image registry tool to manage images, add tags to and remove tags from images, and define the user name for an instance operating system. Each plugin lists required tags for an image. To run remote operations, the Data Processing service requires a user name with which to log in to the operating system for an instance.

Method	URI	Description
GET	`/v1.1/{tenant_id}/images`	Lists all images registered in the registry.
GET	`/v1.1/{tenant_id}/images/{image_id}`	Shows details for an image.
POST	`/v1.1/{tenant_id}/images/{image_id}`	Registers an image in the registry.
DELETE	`/v1.1/{tenant_id}/images/{image_id}`	Removes an image from the registry.
POST	`/v1.1/{tenant_id}/images/{image_id}/tag`	Adds tags to an image.
POST	`/v1.1/{tenant_id}/images/{image_id}/untag`	Removes tags from an image.

5.2.1. List images

Method	URI	Description
GET	`/v1.1/{tenant_id}/images`	Lists all images registered in the registry.

Normal response codes: 200

5.2.1.1. Request

This operation does not accept a request body.

5.2.1.2. Response

Example 5.4. List images: JSON response

```
{
    "images": [
        {
            "name": "ubuntu-vanilla-2.7.1",
            "id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
            "created": "2015-08-06T08:17:14Z",
            "metadata": {
                "_sahara_tag_2.7.1": true,
                "_sahara_username": "ubuntu",
                "_sahara_tag_vanilla": true
            },
            "username": "ubuntu",
            "progress": 100,
            "OS-EXT-IMG-SIZE:size": 998716928,
            "status": "ACTIVE",
            "minDisk": 0,
            "tags": [
                "vanilla",
                "2.7.1"
            ],
            "updated": "2015-09-04T09:35:09Z",
            "minRam": 0,
            "description": null
        },
        {
            "name": "cdh-latest",
            "id": "ff74035b-9da7-4edf-981d-57f270ed337d",
            "created": "2015-09-04T11:56:44Z",
            "metadata": {
                "_sahara_username": "ubuntu",
                "_sahara_tag_5.4.0": true,
                "_sahara_tag_cdh": true
            },
            "username": "ubuntu",
            "progress": 100,
            "OS-EXT-IMG-SIZE:size": 3281453056,
            "status": "ACTIVE",
            "minDisk": 0,
            "tags": [
                "5.4.0",
                "cdh"
```

```
            ],
            "updated": "2015-09-04T12:46:42Z",
            "minRam": 0,
            "description": null
        }
    ]
}
```

5.2.2. Show image details

Method	URI	Description
GET	`/v1.1/{tenant_id}/images/{image_id}`	Shows details for an image.

Normal response codes: 200

5.2.2.1. Request

This table shows the URI parameters for the show image details request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

5.2.2.2. Response

Example 5.5. Show image details: JSON response

```
{
    "image": {
        "updated": "2015-02-03T10:29:32Z",
        "metadata": {
            "_sahara_username": "ubuntu",
            "_sahara_tag_vanilla": true,
            "_sahara_tag_2.6.0": true
        },
        "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
        "minDisk": 0,
        "status": "ACTIVE",
        "tags": [
            "vanilla",
            "2.6.0"
        ],
        "minRam": 0,
        "progress": 100,
        "username": "ubuntu",
        "created": "2015-02-03T10:28:39Z",
        "name": "sahara-vanilla-2.6.0-ubuntu-14.04",
        "description": null,
        "OS-EXT-IMG-SIZE:size": 1101856768
    }
}
```

5.2.3. Register image

Method	URI	Description
POST	`/v1.1/{tenant_id}/images/{image_id}`	Registers an image in the registry.

Normal response codes: 202

5.2.3.1. Request

This table shows the URI parameters for the register image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

Example 5.6. Register image: JSON request

```
{
    "username": "ubuntu",
    "description": "Ubuntu image for Hadoop 2.7.1"
}
```

5.2.3.2. Response

Example 5.7. Register image: JSON response

```
{
    "image": {
        "updated": "2015-03-24T10:05:10Z",
        "metadata": {
            "_sahara_description": "Ubuntu image for Hadoop 2.7.1",
            "_sahara_username": "ubuntu",
            "_sahara_tag_vanilla": true,
            "_sahara_tag_2.7.1": true
        },
        "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
        "minDisk": 0,
        "status": "ACTIVE",
        "tags": [
            "vanilla",
            "2.7.1"
        ],
        "minRam": 0,
        "progress": 100,
        "username": "ubuntu",
        "created": "2015-02-03T10:28:39Z",
        "name": "sahara-vanilla-2.7.1-ubuntu-14.04",
        "description": "Ubuntu image for Hadoop 2.7.1",
        "OS-EXT-IMG-SIZE:size": 1101856768
    }
}
```

5.2.4. Unregister image

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/images/{image_id}`	Removes an image from the registry.

Normal response codes: 204

5.2.4.1. Request

This table shows the URI parameters for the unregister image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

5.2.5. Add tags to image

Method	URI	Description
POST	`/v1.1/{tenant_id}/images/{image_id}/tag`	Adds tags to an image.

Normal response codes: 202

5.2.5.1. Request

This table shows the URI parameters for the add tags to image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

Example 5.8. Add tags to image: JSON request

```
{
    "tags": [
        "vanilla",
        "2.7.1",
        "some_other_tag"
    ]
}
```

5.2.5.2. Response

Example 5.9. Add tags to image: JSON response

```
{
    "image": {
        "updated": "2015-03-24T10:18:33Z",
        "metadata": {
            "_sahara_tag_vanilla": true,
            "_sahara_description": "Ubuntu image for Hadoop 2.7.1",
            "_sahara_username": "ubuntu",
            "_sahara_tag_some_other_tag": true,
            "_sahara_tag_2.7.1": true
        },
        "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
        "minDisk": 0,
        "status": "ACTIVE",
        "tags": [
            "vanilla",
            "some_other_tag",
            "2.7.1"
        ],
        "minRam": 0,
        "progress": 100,
        "username": "ubuntu",
        "created": "2015-02-03T10:28:39Z",
        "name": "sahara-vanilla-2.6.0-ubuntu-14.04",
        "description": "Ubuntu image for Hadoop 2.7.1",
        "OS-EXT-IMG-SIZE:size": 1101856768
    }
```

```
}
```

5.2.6. Remove tags from image

Method	URI	Description
POST	`/v1.1/{tenant_id}/images/{image_id}/untag`	Removes tags from an image.

Normal response codes: 202

5.2.6.1. Request

This table shows the URI parameters for the remove tags from image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

Example 5.10. Remove tags from image: JSON request

```
{
    "tags": [
        "some_other_tag"
    ]
}
```

5.2.6.2. Response

Example 5.11. Remove tags from image: JSON response

```
{
    "image": {
        "updated": "2015-03-24T10:18:33Z",
        "metadata": {
            "_sahara_tag_vanilla": true,
            "_sahara_description": "Ubuntu image for Hadoop 2.7.1",
            "_sahara_username": "ubuntu",
            "_sahara_tag_some_other_tag": true,
            "_sahara_tag_2.7.1": true
        },
        "id": "bb8d12b5-f9bb-49f0-aecb-739b8a9bec89",
        "minDisk": 0,
        "status": "ACTIVE",
        "tags": [
            "vanilla",
            "some_other_tag",
            "2.7.1"
        ],
        "minRam": 0,
        "progress": 100,
        "username": "ubuntu",
        "created": "2015-02-03T10:28:39Z",
        "name": "sahara-vanilla-2.6.0-ubuntu-14.04",
        "description": "Ubuntu image for Hadoop 2.7.1",
        "OS-EXT-IMG-SIZE:size": 1101856768
    }
}
```

5.3. Node group templates

A cluster is a group of nodes with the same configuration. A node group template configures a node in the cluster.

A template configures Hadoop processes and VM characteristics, such as the number of reduced slots for task tracker, the number of CPUs, and the amount of RAM. The template specifies the VM characteristics through an OpenStack flavor.

Method	URI	Description
GET	`/v1.1/{tenant_id}/node-group-templates`	Lists available node group templates.
POST	`/v1.1/{tenant_id}/node-group-templates`	Creates a node group template.
GET	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Shows a node group template, by ID.
DELETE	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Deletes a node group template.
PUT	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Updates a node group template.

5.3.1. List node group templates

Method	URI	Description
GET	`/v1.1/{tenant_id}/node-group-templates`	Lists available node group templates.

Normal response codes: 200

5.3.1.1. Request

This operation does not accept a request body.

5.3.1.2. Response

Example 5.12. List node group templates: JSON response


```
{
    "node_group_templates": [
        {
            "is_public": false,
            "image_id": null,
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "shares": null,
            "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
            "node_configs": {},
            "auto_security_group": false,
            "is_default": false,
            "availability_zone": null,
            "plugin_name": "vanilla",
            "flavor_id": "2",
            "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
            "description": null,
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "volumes_availability_zone": null,
            "created_at": "2015-09-14T10:20:11",
            "is_protected": false,
            "updated_at": null,
            "volumes_per_node": 0,
            "is_proxy_gateway": false,
            "name": "master",
            "volume_mount_prefix": "/volumes/disk",
            "node_processes": [
                "namenode",
                "resourcemanager",
                "oozie",
                "historyserver"
            ],
            "volumes_size": 0,
            "volume_local_to_instance": false,
            "security_groups": null,
            "volume_type": null
        },
        {
            "is_public": false,
            "image_id": null,
```

```
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "shares": null,
            "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
            "node_configs": {},
            "auto_security_group": false,
            "is_default": false,
            "availability_zone": null,
            "plugin_name": "vanilla",
            "flavor_id": "2",
            "id": "846edb31-add5-46e6-a4ee-a4c339f99251",
            "description": null,
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "volumes_availability_zone": null,
            "created_at": "2015-09-14T10:27:00",
            "is_protected": false,
            "updated_at": null,
            "volumes_per_node": 0,
            "is_proxy_gateway": false,
            "name": "worker",
            "volume_mount_prefix": "/volumes/disk",
            "node_processes": [
                "datanode",
                "nodemanager"
            ],
            "volumes_size": 0,
            "volume_local_to_instance": false,
            "security_groups": null,
            "volume_type": null
        }
    ]
}
```

5.3.2. Create node group template

Method	URI	Description
POST	`/v1.1/{tenant_id}/node-group-templates`	Creates a node group template.

Normal response codes: 202

5.3.2.1. Request

Example 5.13. Create node group template: JSON request

```
{
    "plugin_name": "vanilla",
    "hadoop_version": "2.7.1",
    "node_processes": [
        "namenode",
        "resourcemanager",
        "oozie",
        "historyserver"
    ],
    "name": "master",
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "flavor_id": "2"
}
```

5.3.2.2. Response

Example 5.14. Create node group template: JSON response

```
{
    "node_group_template": {
        "is_public": false,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {},
        "auto_security_group": false,
        "is_default": false,
        "availability_zone": null,
        "plugin_name": "vanilla",
        "is_protected": false,
        "flavor_id": "2",
        "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:20:11",
        "security_groups": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "master",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "namenode",
            "resourcemanager",
            "oozie",
```

```
            "historyserver"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    }
}
```

5.3.3. Show node group template details

Method	URI	Description
GET	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Shows a node group template, by ID.

Normal response codes: 200

5.3.3.1. Request

This table shows the URI parameters for the show node group template details request:

Name	Type	Description
`{node_group_template_id}`	UUID	The UUID of the node group template.

This operation does not accept a request body.

5.3.3.2. Response

Example 5.15. Show node group template details: JSON response

```
{
    "node_group_template": {
        "is_public": false,
        "image_id": null,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "shares": null,
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {},
        "auto_security_group": false,
        "is_default": false,
        "availability_zone": null,
        "plugin_name": "vanilla",
        "flavor_id": "2",
        "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
        "description": null,
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:20:11",
        "is_protected": false,
        "updated_at": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "master",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
            "namenode",
            "resourcemanager",
            "oozie",
            "historyserver"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "security_groups": null,
```

```
        "volume_type": null
    }
}
```

5.3.4. Delete node group template

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Deletes a node group template.

Normal response codes: 204

5.3.4.1. Request

This table shows the URI parameters for the delete node group template request:

Name	Type	Description
`{node_group_template_id}`	UUID	The UUID of the node group template.

This operation does not accept a request body.

5.3.5. Update node group template

Method	URI	Description
PUT	`/v1.1/{tenant_id}/node-group-templates/{node_group_template_id}`	Updates a node group template.

Normal response codes: 202

5.3.5.1. Request

This table shows the URI parameters for the update node group template request:

Name	Type	Description
`{node_group_template_id}`	UUID	The UUID of the node group template.

Example 5.16. Update node group template: JSON request

```
{
    "plugin_name": "vanilla",
    "hadoop_version": "2.7.1",
    "node_processes": [
        "datanode"
    ],
    "name": "new",
    "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
    "flavor_id": "2"
}
```

5.3.5.2. Response

Example 5.17. Update node group template: JSON response

```
{
    "node_group_template": {
        "is_public": false,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
        "node_configs": {},
        "auto_security_group": false,
        "is_default": false,
        "availability_zone": null,
        "plugin_name": "vanilla",
        "is_protected": false,
        "flavor_id": "2",
        "id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "volumes_availability_zone": null,
        "created_at": "2015-09-14T10:20:11",
        "security_groups": null,
        "volumes_per_node": 0,
        "is_proxy_gateway": false,
        "name": "new",
        "volume_mount_prefix": "/volumes/disk",
        "node_processes": [
```

```
            "datanode"
        ],
        "volumes_size": 0,
        "volume_local_to_instance": false,
        "volume_type": null
    }
}
```

5.4. Cluster templates

A cluster template configures a Hadoop cluster. A cluster template lists node groups with the number of instances in each group. You can also define cluster-scoped configurations in a cluster template.

Method	URI	Description
GET	`/v1.1/{tenant_id}/cluster-templates`	Lists available cluster templates.
POST	`/v1.1/{tenant_id}/cluster-templates`	Creates a cluster template.
GET	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Shows details for a cluster template.
PUT	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Updates a cluster template.
DELETE	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Deletes a cluster template.

5.4.1. List cluster templates

Method	URI	Description
GET	`/v1.1/{tenant_id}/cluster-templates`	Lists available cluster templates.

Normal response codes: 200

5.4.1.1. Request

This operation does not accept a request body.

5.4.1.2. Response

Example 5.18. List cluster templates: JSON response


```
{
    "cluster_templates": [
        {
            "is_public": false,
            "anti_affinity": [],
            "name": "cluster-template",
            "created_at": "2015-09-14T10:38:44",
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "cluster_configs": {},
            "shares": null,
            "id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
            "default_image_id": null,
            "is_default": false,
            "updated_at": null,
            "plugin_name": "vanilla",
            "node_groups": [
                {
                    "image_id": null,
                    "shares": null,
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {},
                    "auto_security_group": false,
                    "availability_zone": null,
                    "count": 1,
                    "flavor_id": "2",
                    "id": "1751c04e-8f39-467e-a421-480961172d4b",
                    "security_groups": null,
                    "use_autoconfig": true,
                    "volumes_availability_zone": null,
                    "created_at": "2015-09-14T10:38:44",
                    "node_group_template_id":
 "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                    "updated_at": null,
                    "volumes_per_node": 0,
                    "is_proxy_gateway": false,
                    "name": "master",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "namenode",
                        "resourcemanager",
```

```
                        "oozie",
                        "historyserver"
                    ],
                    "volumes_size": 0,
                    "volume_local_to_instance": false,
                    "volume_type": null
                },
                {
                    "image_id": null,
                    "shares": null,
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {},
                    "auto_security_group": false,
                    "availability_zone": null,
                    "count": 3,
                    "flavor_id": "2",
                    "id": "3ee85068-c455-4391-9db2-b54a20b99df3",
                    "security_groups": null,
                    "use_autoconfig": true,
                    "volumes_availability_zone": null,
                    "created_at": "2015-09-14T10:38:44",
                    "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                    "updated_at": null,
                    "volumes_per_node": 0,
                    "is_proxy_gateway": false,
                    "name": "worker",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "datanode",
                        "nodemanager"
                    ],
                    "volumes_size": 0,
                    "volume_local_to_instance": false,
                    "volume_type": null
                }
            ],
            "neutron_management_network": "b1610452-2933-46b0-
bf31-660cfa5621bd",
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "description": null,
            "is_protected": false
        },
        {
            "is_public": true,
            "anti_affinity": [],
            "name": "asd",
            "created_at": "2015-08-18T08:39:39",
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "cluster_configs": {
                "general": {}
            },
            "shares": null,
            "id": "5a9c787c-2078-4f7d-9a66-27759be9051b",
            "default_image_id": null,
            "is_default": false,
            "updated_at": "2015-09-14T08:41:15",
            "plugin_name": "vanilla",
```

```
            "node_groups": [
                {
                    "image_id": null,
                    "shares": null,
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {},
                    "auto_security_group": true,
                    "availability_zone": "",
                    "count": 1,
                    "flavor_id": "2",
                    "id": "a65864dd-3f99-4d29-a011-f7711cc23fa0",
                    "security_groups": [],
                    "use_autoconfig": true,
                    "volumes_availability_zone": null,
                    "created_at": "2015-08-18T08:39:39",
                    "node_group_template_id":
 "42ce49de-1b8f-41d5-8f4a-244ec0826d92",
                    "updated_at": null,
                    "volumes_per_node": 1,
                    "is_proxy_gateway": false,
                    "name": "asd",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "namenode",
                        "jobtracker"
                    ],
                    "volumes_size": 10,
                    "volume_local_to_instance": false,
                    "volume_type": null
                }
            ],
            "neutron_management_network": null,
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "description": "",
            "is_protected": false
        }
    ]
}
```

5.4.2. Create cluster templates

Method	URI	Description
POST	`/v1.1/{tenant_id}/cluster-templates`	Creates a cluster template.

Normal response codes: 202

5.4.2.1. Request

Example 5.19. Create cluster template: JSON request

```
{
    "plugin_name": "vanilla",
    "hadoop_version": "2.7.1",
    "node_groups": [
        {
            "name": "worker",
            "count": 3,
            "node_group_template_id": "846edb31-add5-46e6-a4ee-a4c339f99251"
        },
        {
            "name": "master",
            "count": 1,
            "node_group_template_id": "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae"
        }
    ],
    "name": "cluster-template"
}
```

5.4.2.2. Response

Example 5.20. Create cluster template: JSON response

```
{
    "cluster_template": {
        "is_public": false,
        "anti_affinity": [],
        "name": "cluster-template",
        "created_at": "2015-09-14T10:38:44",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "cluster_configs": {},
        "shares": null,
        "id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
        "default_image_id": null,
        "is_default": false,
        "updated_at": null,
        "plugin_name": "vanilla",
        "node_groups": [
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {},
                "auto_security_group": false,
                "availability_zone": null,
                "count": 1,
```

```
                "flavor_id": "2",
                "id": "1751c04e-8f39-467e-a421-480961172d4b",
                "security_groups": null,
                "use_autoconfig": true,
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:38:44",
                "node_group_template_id":
 "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                "updated_at": null,
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "master",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "resourcemanager",
                    "oozie",
                    "historyserver"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            },
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {},
                "auto_security_group": false,
                "availability_zone": null,
                "count": 3,
                "flavor_id": "2",
                "id": "3ee85068-c455-4391-9db2-b54a20b99df3",
                "security_groups": null,
                "use_autoconfig": true,
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:38:44",
                "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                "updated_at": null,
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "worker",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "datanode",
                    "nodemanager"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "neutron_management_network": null,
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "description": null,
        "is_protected": false
    }
}
```

5.4.3. Show cluster template details

Method	URI	Description
GET	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Shows details for a cluster template.

Normal response codes: 200

5.4.3.1. Request

This table shows the URI parameters for the show cluster template details request:

Name	Type	Description
`{cluster_template_id}`	UUID	The unique identifier of the cluster template.

This operation does not accept a request body.

5.4.3.2. Response

Example 5.21. Show cluster template details: JSON response

```
{
    "cluster_templates": [
        {
            "is_public": false,
            "anti_affinity": [],
            "name": "cluster-template",
            "created_at": "2015-09-14T10:38:44",
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "cluster_configs": {},
            "shares": null,
            "id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
            "default_image_id": null,
            "is_default": false,
            "updated_at": null,
            "plugin_name": "vanilla",
            "node_groups": [
                {
                    "image_id": null,
                    "shares": null,
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {},
                    "auto_security_group": false,
                    "availability_zone": null,
                    "count": 1,
                    "flavor_id": "2",
                    "id": "1751c04e-8f39-467e-a421-480961172d4b",
                    "security_groups": null,
                    "use_autoconfig": true,
                    "volumes_availability_zone": null,
                    "created_at": "2015-09-14T10:38:44",
                    "node_group_template_id":
 "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                    "updated_at": null,
                    "volumes_per_node": 0,
                    "is_proxy_gateway": false,
```

```
                    "name": "master",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "namenode",
                        "resourcemanager",
                        "oozie",
                        "historyserver"
                    ],
                    "volumes_size": 0,
                    "volume_local_to_instance": false,
                    "volume_type": null
                },
                {
                    "image_id": null,
                    "shares": null,
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {},
                    "auto_security_group": false,
                    "availability_zone": null,
                    "count": 3,
                    "flavor_id": "2",
                    "id": "3ee85068-c455-4391-9db2-b54a20b99df3",
                    "security_groups": null,
                    "use_autoconfig": true,
                    "volumes_availability_zone": null,
                    "created_at": "2015-09-14T10:38:44",
                    "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                    "updated_at": null,
                    "volumes_per_node": 0,
                    "is_proxy_gateway": false,
                    "name": "worker",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "datanode",
                        "nodemanager"
                    ],
                    "volumes_size": 0,
                    "volume_local_to_instance": false,
                    "volume_type": null
                }
            ],
            "neutron_management_network": "b1610452-2933-46b0-
bf31-660cfa5621bd",
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "description": null,
            "is_protected": false
        },
        {
            "is_public": true,
            "anti_affinity": [],
            "name": "asd",
            "created_at": "2015-08-18T08:39:39",
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "cluster_configs": {
                "general": {}
            },
            "shares": null,
```

```
            "id": "5a9c787c-2078-4f7d-9a66-27759be9051b",
            "default_image_id": null,
            "is_default": false,
            "updated_at": "2015-09-14T08:41:15",
            "plugin_name": "vanilla",
            "node_groups": [
                {
                    "image_id": null,
                    "shares": null,
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {},
                    "auto_security_group": true,
                    "availability_zone": "",
                    "count": 1,
                    "flavor_id": "2",
                    "id": "a65864dd-3f99-4d29-a011-f7711cc23fa0",
                    "security_groups": [],
                    "use_autoconfig": true,
                    "volumes_availability_zone": null,
                    "created_at": "2015-08-18T08:39:39",
                    "node_group_template_id":
 "42ce49de-1b8f-41d5-8f4a-244ec0826d92",
                    "updated_at": null,
                    "volumes_per_node": 1,
                    "is_proxy_gateway": false,
                    "name": "asd",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "namenode",
                        "jobtracker"
                    ],
                    "volumes_size": 10,
                    "volume_local_to_instance": false,
                    "volume_type": null
                }
            ],
            "neutron_management_network": null,
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "description": "",
            "is_protected": false
        }
    ]
}
```

5.4.4. Update cluster templates

Method	URI	Description
PUT	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Updates a cluster template.

Normal response codes: 202

5.4.4.1. Request

This table shows the URI parameters for the update cluster templates request:

Name	Type	Description
`{cluster_template_id}`	UUID	The unique identifier of the cluster template.

Example 5.22. Update the cluster template: JSON request

```
{
    "description": "Updated template",
    "plugin_name": "vanilla",
    "hadoop_version": "2.7.1",
    "name": "vanilla-updated",
    "cluster_configs": {
        "HDFS": {
            "dfs.replication": 2
        }
    }
}
```

5.4.4.2. Response

Example 5.23. Update cluster templates: JSON response

```
{
    "cluster_template": {
        "is_public": false,
        "anti_affinity": [],
        "name": "vanilla-updated",
        "created_at": "2015-08-21T08:41:24",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "cluster_configs": {
            "HDFS": {
                "dfs.replication": 2
            }
        },
        "shares": null,
        "id": "84d47e85-6094-473f-bf6d-5a7e6e86564e",
        "default_image_id": null,
        "is_default": false,
        "updated_at": "2015-09-14T10:45:57",
        "plugin_name": "vanilla",
        "node_groups": [
            {
                "image_id": null,
                "shares": null,
```

```
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {},
                    "JobFlow": {},
                    "MapReduce": {},
                    "Hive": {},
                    "Hadoop": {},
                    "HDFS": {}
                },
                "auto_security_group": true,
                "availability_zone": "",
                "count": 1,
                "flavor_id": "3",
                "id": "57b966ab-617e-4735-bf60-0cb991208a52",
                "security_groups": [],
                "use_autoconfig": true,
                "volumes_availability_zone": null,
                "created_at": "2015-08-21T08:41:24",
                "node_group_template_id": "a5533187-3f14-42c3-
ba3a-196c13fe0fb5",
                "updated_at": null,
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "all",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "datanode",
                    "historyserver",
                    "resourcemanager",
                    "nodemanager",
                    "oozie"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "neutron_management_network": null,
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "description": "Updated template",
        "is_protected": false
    }
}
```

5.4.5. Delete cluster template

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/cluster-templates/{cluster_template_id}`	Deletes a cluster template.

Normal response codes: 204

5.4.5.1. Request

This table shows the URI parameters for the delete cluster template request:

Name	Type	Description
`{cluster_template_id}`	UUID	The unique identifier of the cluster template.

This operation does not accept a request body.

5.5. Clusters

A cluster is a group of nodes with the same configuration.

Method	URI	Description
GET	`/v1.1/{tenant_id}/clusters`	Lists available clusters.
POST	`/v1.1/{tenant_id}/clusters`	Creates a cluster.
POST	`/v1.1/{tenant_id}/clusters/multiple`	Creates multiple clusters.
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows details for a cluster, by ID.
PUT	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Scales a cluster.
PATCH	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Updates a cluster.
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows provisioning progress for a cluster.
DELETE	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Deletes a cluster.

5.5.1. List available clusters

Method	URI	Description
GET	/v1.1/{tenant_id}/clusters	Lists available clusters.

Normal response codes: 200

5.5.1.1. Request

This operation does not accept a request body.

5.5.1.2. Response

Example 5.24. List available clusters: JSON response


```
{
    "clusters": [
        {
            "is_public": false,
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "shares": null,
            "status_description": "",
            "plugin_name": "vanilla",
            "neutron_management_network": "b1610452-2933-46b0-
bf31-660cfa5621bd",
            "info": {
                "YARN": {
                    "Web UI": "http://172.18.168.115:8088",
                    "ResourceManager": "http://172.18.168.115:8032"
                },
                "HDFS": {
                    "Web UI": "http://172.18.168.115:50070",
                    "NameNode": "hdfs://vanilla-cluster-master-0:9000"
                },
                "JobFlow": {
                    "Oozie": "http://172.18.168.115:11000"
                },
                "MapReduce JobHistory Server": {
                    "Web UI": "http://172.18.168.115:19888"
                }
            },
            "user_keypair_id": "apavlov",
            "management_public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCfe9ARO
+t9CybtuC1+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rjOwAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhOSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
            "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
            "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
            "node_groups": [
                {
                    "image_id": null,
                    "shares": null,
```

```
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {
                        "YARN": {
                            "yarn.nodemanager.vmem-check-enabled": "false",
                            "yarn.scheduler.maximum-allocation-mb": 2048,
                            "yarn.scheduler.minimum-allocation-mb": 256,
                            "yarn.nodemanager.resource.memory-mb": 2048
                        },
                        "MapReduce": {
                            "yarn.app.mapreduce.am.resource.mb": 256,
                            "mapreduce.task.io.sort.mb": 102,
                            "mapreduce.reduce.java.opts": "-Xmx409m",
                            "mapreduce.reduce.memory.mb": 512,
                            "mapreduce.map.memory.mb": 256,
                            "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                            "mapreduce.map.java.opts": "-Xmx204m"
                        }
                    },
                    "auto_security_group": false,
                    "availability_zone": null,
                    "count": 1,
                    "flavor_id": "2",
                    "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
                    "security_groups": null,
                    "use_autoconfig": true,
                    "instances": [
                        {
                            "created_at": "2015-09-14T10:57:36",
                            "id": "4867d92e-cc7b-4cde-9a1a-149e91caa491",
                            "management_ip": "172.18.168.115",
                            "updated_at": "2015-09-14T10:57:39",
                            "instance_id":
 "b9f16a07-88fc-423e-83a3-489598fe6737",
                            "internal_ip": "10.50.0.60",
                            "instance_name": "vanilla-cluster-master-0"
                        }
                    ],
                    "volumes_availability_zone": null,
                    "created_at": "2015-09-14T10:57:11",
                    "node_group_template_id":
 "0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                    "updated_at": "2015-09-14T10:57:36",
                    "volumes_per_node": 0,
                    "is_proxy_gateway": false,
                    "name": "master",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "namenode",
                        "resourcemanager",
                        "oozie",
                        "historyserver"
                    ],
                    "volumes_size": 0,
                    "volume_local_to_instance": false,
                    "volume_type": null
                },
                {
                    "image_id": null,
                    "shares": null,
```

```
                    "floating_ip_pool": "033debed-aeb8-488c-b7d0-
adb74c61faa5",
                    "node_configs": {
                        "YARN": {
                            "yarn.nodemanager.vmem-check-enabled": "false",
                            "yarn.scheduler.maximum-allocation-mb": 2048,
                            "yarn.scheduler.minimum-allocation-mb": 256,
                            "yarn.nodemanager.resource.memory-mb": 2048
                        },
                        "MapReduce": {
                            "yarn.app.mapreduce.am.resource.mb": 256,
                            "mapreduce.task.io.sort.mb": 102,
                            "mapreduce.reduce.java.opts": "-Xmx409m",
                            "mapreduce.reduce.memory.mb": 512,
                            "mapreduce.map.memory.mb": 256,
                            "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                            "mapreduce.map.java.opts": "-Xmx204m"
                        }
                    },
                    "auto_security_group": false,
                    "availability_zone": null,
                    "count": 3,
                    "flavor_id": "2",
                    "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
                    "security_groups": null,
                    "use_autoconfig": true,
                    "instances": [
                        {
                            "created_at": "2015-09-14T10:57:37",
                            "id": "f3633b30-c1e4-4144-930b-ab5b780b87be",
                            "management_ip": "172.18.168.118",
                            "updated_at": "2015-09-14T10:57:39",
                            "instance_id": "0cf1ee81-aa72-48da-
be2c-65bc2fa51f8f",
                            "internal_ip": "10.50.0.63",
                            "instance_name": "vanilla-cluster-worker-0"
                        },
                        {
                            "created_at": "2015-09-14T10:57:37",
                            "id": "0d66fd93-f277-4a94-b46a-f5866aa0c38f",
                            "management_ip": "172.18.168.117",
                            "updated_at": "2015-09-14T10:57:40",
                            "instance_id": "4a937391-
b594-4ad0-9a53-00a99a691383",
                            "internal_ip": "10.50.0.62",
                            "instance_name": "vanilla-cluster-worker-1"
                        },
                        {
                            "created_at": "2015-09-14T10:57:37",
                            "id": "0982cefd-5c58-436e-8f1e-c1d0830f18a7",
                            "management_ip": "172.18.168.116",
                            "updated_at": "2015-09-14T10:57:40",
                            "instance_id":
 "839b1d56-6d0d-4aa4-9d05-30e029c276f8",
                            "internal_ip": "10.50.0.61",
                            "instance_name": "vanilla-cluster-worker-2"
                        }
                    ],
                    "volumes_availability_zone": null,
                    "created_at": "2015-09-14T10:57:11",
```

```
                    "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                    "updated_at": "2015-09-14T10:57:37",
                    "volumes_per_node": 0,
                    "is_proxy_gateway": false,
                    "name": "worker",
                    "volume_mount_prefix": "/volumes/disk",
                    "node_processes": [
                        "datanode",
                        "nodemanager"
                    ],
                    "volumes_size": 0,
                    "volume_local_to_instance": false,
                    "volume_type": null
                }
            ],
            "provision_progress": [
                {
                    "created_at": "2015-09-14T10:57:18",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "0a6d95f9-30f4-4434-823a-a38a7999a5af",
                    "step_type": "Engine: create cluster",
                    "step_name": "Create Heat stack",
                    "updated_at": "2015-09-14T10:57:38",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:58:16",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "29f2b587-c34c-4871-9ed9-9235b411cd9a",
                    "step_type": "Engine: create cluster",
                    "step_name": "Configure instances",
                    "updated_at": "2015-09-14T10:58:22",
                    "successful": true,
                    "total": 4,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T11:00:27",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "36f1efde-90f9-41c1-b409-aa1cf9623e3e",
                    "step_type": "Plugin: start cluster",
                    "step_name": "Start the following process(es): Oozie",
                    "updated_at": "2015-09-14T11:01:15",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:58:22",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "602bcc27-3a2d-42c8-8aca-ebc475319c72",
                    "step_type": "Plugin: configure cluster",
                    "step_name": "Configure instances",
                    "updated_at": "2015-09-14T10:59:21",
                    "successful": true,
                    "total": 4,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
```

```
                },
                {
                    "created_at": "2015-09-14T10:59:21",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "7e291df1-2d32-410d-ae89-33ab6f83cf17",
                    "step_type": "Plugin: configure cluster",
                    "step_name": "Configure topology data",
                    "updated_at": "2015-09-14T10:59:37",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T11:00:01",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "8ab7933c-ad61-4a4f-88db-23ce78ee10f6",
                    "step_type": "Plugin: start cluster",
                    "step_name": "Start the following process(es): DataNodes,
NodeManagers",
                    "updated_at": "2015-09-14T11:00:11",
                    "successful": true,
                    "total": 3,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T11:00:11",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "9c8dc016-8c5b-4e80-9857-80c41f6bd971",
                    "step_type": "Plugin: start cluster",
                    "step_name": "Await DataNodes start up",
                    "updated_at": "2015-09-14T11:00:21",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T11:00:21",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "c6327532-222b-416c-858f-73dbb32b8e97",
                    "step_type": "Plugin: start cluster",
                    "step_name": "Start the following process(es):
HistoryServer",
                    "updated_at": "2015-09-14T11:00:27",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:57:41",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "d3eca726-8b44-473a-ac29-fba45a893725",
                    "step_type": "Engine: create cluster",
                    "step_name": "Wait for instance accessibility",
                    "updated_at": "2015-09-14T10:58:14",
                    "successful": true,
                    "total": 4,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:58:14",
```

```
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "d7a875ff-64bf-41aa-882d-b5061c8ee152",
                    "step_type": "Engine: create cluster",
                    "step_name": "Mount volumes to instances",
                    "updated_at": "2015-09-14T10:58:15",
                    "successful": true,
                    "total": 0,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:59:55",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "ded7d227-10b8-4cb0-ab6c-25da1462bb7a",
                    "step_type": "Plugin: start cluster",
                    "step_name": "Start the following process(es):
ResourceManager",
                    "updated_at": "2015-09-14T11:00:00",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:59:38",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "e1701ff5-930a-4212-945a-43515dfe24d1",
                    "step_type": "Plugin: start cluster",
                    "step_name": "Start the following process(es): NameNode",
                    "updated_at": "2015-09-14T10:59:54",
                    "successful": true,
                    "total": 1,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                },
                {
                    "created_at": "2015-09-14T10:57:38",
                    "tenant_id": "808d5032ea0446889097723bfc8e919d",
                    "id": "eaf0ab1b-bf8f-48f0-8f2c-fa4f82f539b9",
                    "step_type": "Engine: create cluster",
                    "step_name": "Assign IPs",
                    "updated_at": "2015-09-14T10:57:41",
                    "successful": true,
                    "total": 4,
                    "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42"
                }
            ],
            "hadoop_version": "2.7.1",
            "use_autoconfig": true,
            "trust_id": null,
            "description": null,
            "created_at": "2015-09-14T10:57:11",
            "is_protected": false,
            "updated_at": "2015-09-14T11:01:15",
            "is_transient": false,
            "cluster_configs": {
                "HDFS": {
                    "dfs.replication": 3
                }
            },
            "anti_affinity": [],
            "name": "vanilla-cluster",
            "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
```

```
            "status": "Active"
        }
    ]
}
```

5.5.2. Create cluster

Method	URI	Description
POST	`/v1.1/{tenant_id}/clusters`	Creates a cluster.

Normal response codes: 202

5.5.2.1. Request

Example 5.25. Create cluster: JSON request

```
{
    "plugin_name": "vanilla",
    "hadoop_version": "2.7.1",
    "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
    "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
    "user_keypair_id": "test",
    "name": "vanilla-cluster",
    "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd"
}
```

5.5.2.2. Response

Example 5.26. Create cluster: JSON response

```
{
    "cluster": {
        "is_public": false,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "shares": null,
        "status_description": "",
        "plugin_name": "vanilla",
        "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
        "info": {},
        "user_keypair_id": "test",
        "management_public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCfe9ARO
+t9CybtuC1+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rjOwAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhOSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
        "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
        "node_groups": [
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.maximum-allocation-mb": 2048,
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048
                    },
```

```
                    "MapReduce": {
                        "yarn.app.mapreduce.am.resource.mb": 256,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m"
                    }
                },
                "auto_security_group": false,
                "availability_zone": null,
                "count": 1,
                "flavor_id": "2",
                "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
                "security_groups": null,
                "use_autoconfig": true,
                "instances": [],
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:57:11",
                "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                "updated_at": "2015-09-14T10:57:12",
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "master",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "resourcemanager",
                    "oozie",
                    "historyserver"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            },
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.maximum-allocation-mb": 2048,
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048
                    },
                    "MapReduce": {
                        "yarn.app.mapreduce.am.resource.mb": 256,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m"
                    }
                },
                "auto_security_group": false,
                "availability_zone": null,
```

```
                "count": 3,
                "flavor_id": "2",
                "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
                "security_groups": null,
                "use_autoconfig": true,
                "instances": [],
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:57:11",
                "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                "updated_at": "2015-09-14T10:57:12",
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "worker",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "datanode",
                    "nodemanager"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "provision_progress": [],
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "trust_id": null,
        "description": null,
        "created_at": "2015-09-14T10:57:11",
        "is_protected": false,
        "updated_at": "2015-09-14T10:57:12",
        "is_transient": false,
        "cluster_configs": {
            "HDFS": {
                "dfs.replication": 3
            }
        },
        "anti_affinity": [],
        "name": "vanilla-cluster",
        "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
        "status": "Validating"
    }
}
```

5.5.3. Create multiple clusters

Method	URI	Description
POST	`/v1.1/{tenant_id}/clusters/multiple`	Creates multiple clusters.

Normal response codes: 202

5.5.3.1. Request

Example 5.27. Create multiple clusters: JSON request

```
{
    "plugin_name": "vanilla",
    "hadoop_version": "2.6.0",
    "cluster_template_id": "9951f86d-57ba-43d6-9cb0-14ed2ec7a6cf",
    "default_image_id": "bc3c3d3c-2684-4bf8-a9fa-388fb71288a9",
    "user_keypair_id": "test",
    "name": "def-cluster",
    "count": 2,
    "cluster_configs": {},
    "neutron_management_network": "7e31648b-4b2e-4f32-9b0a-113581c27076"
}
```

5.5.3.2. Response

Example 5.28. Create multiple clusters: JSON response

```
{
    "clusters": [
        "a007a3e7-658f-4568-b0f2-fe2fd5efc554",
        "b012a6et-65hf-4566-b0f2-fe3fd7efc567"
    ]
}
```

5.5.4. Show details of a cluster

Method	URI	Description
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows details for a cluster, by ID.

Normal response codes: 200

5.5.4.1. Request

This table shows the URI parameters for the show details of a cluster request:

Name	Type	Description
`{cluster_id}`	UUID	The ID of the cluster

This operation does not accept a request body.

5.5.4.2. Response

Example 5.29. Show details of a cluster: JSON response


```
{
    "cluster": {
        "is_public": false,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "shares": null,
        "status_description": "",
        "plugin_name": "vanilla",
        "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
        "info": {},
        "user_keypair_id": "test",
        "management_public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCfe9ARO
+t9CybtuC1+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rjOwAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhOSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
        "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
        "node_groups": [
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.maximum-allocation-mb": 2048,
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048
                    },
                    "MapReduce": {
                        "yarn.app.mapreduce.am.resource.mb": 256,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
```

```
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m"
                    }
                },
                "auto_security_group": false,
                "availability_zone": null,
                "count": 1,
                "flavor_id": "2",
                "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
                "security_groups": null,
                "use_autoconfig": true,
                "instances": [],
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:57:11",
                "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                "updated_at": "2015-09-14T10:57:12",
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "master",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "resourcemanager",
                    "oozie",
                    "historyserver"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            },
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.maximum-allocation-mb": 2048,
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048
                    },
                    "MapReduce": {
                        "yarn.app.mapreduce.am.resource.mb": 256,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m"
                    }
                },
                "auto_security_group": false,
                "availability_zone": null,
                "count": 3,
                "flavor_id": "2",
                "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
                "security_groups": null,
```

```
                "use_autoconfig": true,
                "instances": [],
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:57:11",
                "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                "updated_at": "2015-09-14T10:57:12",
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "worker",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "datanode",
                    "nodemanager"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "provision_progress": [],
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "trust_id": null,
        "description": null,
        "created_at": "2015-09-14T10:57:11",
        "is_protected": false,
        "updated_at": "2015-09-14T10:57:12",
        "is_transient": false,
        "cluster_configs": {
            "HDFS": {
                "dfs.replication": 3
            }
        },
        "anti_affinity": [],
        "name": "vanilla-cluster",
        "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
        "status": "Validating"
    }
}
```

5.5.5. Scale cluster

Method	URI	Description
PUT	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Scales a cluster.

Normal response codes: 202

5.5.5.1. Request

This table shows the URI parameters for the scale cluster request:

Name	Type	Description
`{cluster_id}`	UUID	The ID of the cluster

Example 5.30. Scale cluster: JSON request

```
{
    "add_node_groups": [
        {
            "count": 1,
            "name": "b-worker",
            "node_group_template_id": "bc270ffe-a086-4eeb-9baa-2f5a73504622"
        }
    ],
    "resize_node_groups": [
        {
            "count": 4,
            "name": "worker"
        }
    ]
}
```

5.5.5.2. Response

Example 5.31. Scale cluster: JSON response

```
{
    "cluster": {
        "info": {
            "YARN": {
                "Web UI": "http://172.18.168.115:8088",
                "ResourceManager": "http://172.18.168.115:8032"
            },
            "HDFS": {
                "Web UI": "http://172.18.168.115:50070",
                "NameNode": "hdfs://vanilla-cluster-master-0:9000"
            },
            "MapReduce JobHistory Server": {
                "Web UI": "http://172.18.168.115:19888"
            },
            "JobFlow": {
                "Oozie": "http://172.18.168.115:11000"
            }
        },
        "plugin_name": "vanilla",
        "hadoop_version": "2.7.1",
```

```
        "updated_at": "2015-09-14T11:01:15",
        "name": "vanilla-cluster",
        "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "management_public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCfe9ARO
+t9CybtuC1+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rjOwAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhOSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
        "trust_id": null,
        "status_description": "",
        "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
        "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
        "is_protected": false,
        "is_transient": false,
        "provision_progress": [
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Create Heat stack",
                "step_type": "Engine: create cluster",
                "updated_at": "2015-09-14T10:57:38",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:57:18",
                "id": "0a6d95f9-30f4-4434-823a-a38a7999a5af"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 4,
                "successful": true,
                "step_name": "Configure instances",
                "step_type": "Engine: create cluster",
                "updated_at": "2015-09-14T10:58:22",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:58:16",
                "id": "29f2b587-c34c-4871-9ed9-9235b411cd9a"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Start the following process(es): Oozie",
                "step_type": "Plugin: start cluster",
                "updated_at": "2015-09-14T11:01:15",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T11:00:27",
                "id": "36f1efde-90f9-41c1-b409-aa1cf9623e3e"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 4,
                "successful": true,
                "step_name": "Configure instances",
                "step_type": "Plugin: configure cluster",
                "updated_at": "2015-09-14T10:59:21",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:58:22",
```

```
                "id": "602bcc27-3a2d-42c8-8aca-ebc475319c72"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Configure topology data",
                "step_type": "Plugin: configure cluster",
                "updated_at": "2015-09-14T10:59:37",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:59:21",
                "id": "7e291df1-2d32-410d-ae89-33ab6f83cf17"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 3,
                "successful": true,
                "step_name": "Start the following process(es): DataNodes,
NodeManagers",
                "step_type": "Plugin: start cluster",
                "updated_at": "2015-09-14T11:00:11",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T11:00:01",
                "id": "8ab7933c-ad61-4a4f-88db-23ce78ee10f6"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Await DataNodes start up",
                "step_type": "Plugin: start cluster",
                "updated_at": "2015-09-14T11:00:21",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T11:00:11",
                "id": "9c8dc016-8c5b-4e80-9857-80c41f6bd971"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Start the following process(es): HistoryServer",
                "step_type": "Plugin: start cluster",
                "updated_at": "2015-09-14T11:00:27",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T11:00:21",
                "id": "c6327532-222b-416c-858f-73dbb32b8e97"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 4,
                "successful": true,
                "step_name": "Wait for instance accessibility",
                "step_type": "Engine: create cluster",
                "updated_at": "2015-09-14T10:58:14",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:57:41",
                "id": "d3eca726-8b44-473a-ac29-fba45a893725"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
```

```
                "total": 0,
                "successful": true,
                "step_name": "Mount volumes to instances",
                "step_type": "Engine: create cluster",
                "updated_at": "2015-09-14T10:58:15",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:58:14",
                "id": "d7a875ff-64bf-41aa-882d-b5061c8ee152"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Start the following process(es):
 ResourceManager",
                "step_type": "Plugin: start cluster",
                "updated_at": "2015-09-14T11:00:00",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:59:55",
                "id": "ded7d227-10b8-4cb0-ab6c-25da1462bb7a"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 1,
                "successful": true,
                "step_name": "Start the following process(es): NameNode",
                "step_type": "Plugin: start cluster",
                "updated_at": "2015-09-14T10:59:54",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:59:38",
                "id": "e1701ff5-930a-4212-945a-43515dfe24d1"
            },
            {
                "cluster_id": "e172d86c-906d-418e-a29c-6189f53bfa42",
                "total": 4,
                "successful": true,
                "step_name": "Assign IPs",
                "step_type": "Engine: create cluster",
                "updated_at": "2015-09-14T10:57:41",
                "tenant_id": "808d5032ea0446889097723bfc8e919d",
                "created_at": "2015-09-14T10:57:38",
                "id": "eaf0ab1b-bf8f-48f0-8f2c-fa4f82f539b9"
            }
        ],
        "status": "Active",
        "description": null,
        "use_autoconfig": true,
        "shares": null,
        "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
        "is_public": false,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "node_groups": [
            {
                "volumes_per_node": 0,
                "volume_type": null,
                "updated_at": "2015-09-14T10:57:37",
                "name": "b-worker",
                "id": "b7a6dea4-c898-446b-8c67-4f378d4c06c4",
                "node_group_template_id": "bc270ffe-
a086-4eeb-9baa-2f5a73504622",
```

```
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048,
                        "yarn.scheduler.maximum-allocation-mb": 2048
                    },
                    "MapReduce": {
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "yarn.app.mapreduce.am.resource.mb": 256
                    }
                },
                "auto_security_group": false,
                "volumes_availability_zone": null,
                "use_autoconfig": true,
                "security_groups": null,
                "shares": null,
                "node_processes": [
                    "datanode",
                    "nodemanager"
                ],
                "availability_zone": null,
                "flavor_id": "2",
                "image_id": null,
                "volume_local_to_instance": false,
                "count": 1,
                "volumes_size": 0,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "volume_mount_prefix": "/volumes/disk",
                "instances": [],
                "is_proxy_gateway": false,
                "created_at": "2015-09-14T10:57:11"
            },
            {
                "volumes_per_node": 0,
                "volume_type": null,
                "updated_at": "2015-09-14T10:57:36",
                "name": "master",
                "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
                "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048,
                        "yarn.scheduler.maximum-allocation-mb": 2048
                    },
                    "MapReduce": {
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
```

```
                        "yarn.app.mapreduce.am.resource.mb": 256
                    }
                },
                "auto_security_group": false,
                "volumes_availability_zone": null,
                "use_autoconfig": true,
                "security_groups": null,
                "shares": null,
                "node_processes": [
                    "namenode",
                    "resourcemanager",
                    "oozie",
                    "historyserver"
                ],
                "availability_zone": null,
                "flavor_id": "2",
                "image_id": null,
                "volume_local_to_instance": false,
                "count": 1,
                "volumes_size": 0,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "volume_mount_prefix": "/volumes/disk",
                "instances": [
                    {
                        "instance_id": "b9f16a07-88fc-423e-83a3-489598fe6737",
                        "internal_ip": "10.50.0.60",
                        "instance_name": "vanilla-cluster-master-0",
                        "updated_at": "2015-09-14T10:57:39",
                        "management_ip": "172.18.168.115",
                        "created_at": "2015-09-14T10:57:36",
                        "id": "4867d92e-cc7b-4cde-9a1a-149e91caa491"
                    }
                ],
                "is_proxy_gateway": false,
                "created_at": "2015-09-14T10:57:11"
            },
            {
                "volumes_per_node": 0,
                "volume_type": null,
                "updated_at": "2015-09-14T10:57:37",
                "name": "worker",
                "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
                "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048,
                        "yarn.scheduler.maximum-allocation-mb": 2048
                    },
                    "MapReduce": {
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "yarn.app.mapreduce.am.resource.mb": 256
                    }
```

```
                },
                "auto_security_group": false,
                "volumes_availability_zone": null,
                "use_autoconfig": true,
                "security_groups": null,
                "shares": null,
                "node_processes": [
                    "datanode",
                    "nodemanager"
                ],
                "availability_zone": null,
                "flavor_id": "2",
                "image_id": null,
                "volume_local_to_instance": false,
                "count": 4,
                "volumes_size": 0,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "volume_mount_prefix": "/volumes/disk",
                "instances": [
                    {
                        "instance_id": "0cf1ee81-aa72-48da-be2c-65bc2fa51f8f",
                        "internal_ip": "10.50.0.63",
                        "instance_name": "vanilla-cluster-worker-0",
                        "updated_at": "2015-09-14T10:57:39",
                        "management_ip": "172.18.168.118",
                        "created_at": "2015-09-14T10:57:37",
                        "id": "f3633b30-c1e4-4144-930b-ab5b780b87be"
                    },
                    {
                        "instance_id": "4a937391-b594-4ad0-9a53-00a99a691383",
                        "internal_ip": "10.50.0.62",
                        "instance_name": "vanilla-cluster-worker-1",
                        "updated_at": "2015-09-14T10:57:40",
                        "management_ip": "172.18.168.117",
                        "created_at": "2015-09-14T10:57:37",
                        "id": "0d66fd93-f277-4a94-b46a-f5866aa0c38f"
                    },
                    {
                        "instance_id": "839b1d56-6d0d-4aa4-9d05-30e029c276f8",
                        "internal_ip": "10.50.0.61",
                        "instance_name": "vanilla-cluster-worker-2",
                        "updated_at": "2015-09-14T10:57:40",
                        "management_ip": "172.18.168.116",
                        "created_at": "2015-09-14T10:57:37",
                        "id": "0982cefd-5c58-436e-8f1e-c1d0830f18a7"
                    }
                ],
                "is_proxy_gateway": false,
                "created_at": "2015-09-14T10:57:11"
            }
        ],
        "cluster_configs": {
            "HDFS": {
                "dfs.replication": 3
            }
        },
        "user_keypair_id": "apavlov",
        "anti_affinity": [],
        "created_at": "2015-09-14T10:57:11"
    }
```

```
}
```

5.5.6. Update cluster

Method	URI	Description
PATCH	`/v1.1/{tenant_id}/clus-ters/{cluster_id}`	Updates a cluster.

Normal response codes: 202

5.5.6.1. Request

This table shows the URI parameters for the update cluster request:

Name	Type	Description
`{cluster_id}`	UUID	The ID of the cluster

Example 5.32. Update cluster: JSON request

```
{
    "name": "public-vanilla-cluster",
    "is_public": true
}
```

5.5.6.2. Response

Example 5.33. Update cluster: JSON response

```
{
    "cluster": {
        "is_public": true,
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "shares": null,
        "status_description": "",
        "plugin_name": "vanilla",
        "neutron_management_network": "b1610452-2933-46b0-bf31-660cfa5621bd",
        "info": {},
        "user_keypair_id": "test",
        "management_public_key": "ssh-rsa
 AAAAB3NzaC1yc2EAAAADAQABAAABAQCfe9ARO
+t9CybtuC1+cusDTeQL7wos1+U2dKPlCUJvNUn0PcunGefqWI4MUZPY9yGmvRqfINy7/
xRQCzL0AwgqzwcCXamcK8JCC80uH7j8Vxa4kJheG1jxMoz/FpDSdRnzNZ
+m7H5rjOwAQANhL7KatGLyCPQg9fqOoaIyCZE/A3fztm/XjJMpWnuANpUZubZtISEfu4UZKVk/
DPSlBrbTZkTOvEog1LwZCZoTt0rq6a7PJFzJJkq0YecRudu/
f3tpXbNe/F84sd9PhOSqcrRbm72WzglyEE8PuS1kuWpEz8G+Y5/
0tQxnoh6khj9mgflrdCFuvpdutFLH4eN5MFDh Generated-by-Sahara\n",
        "id": "e172d86c-906d-418e-a29c-6189f53bfa42",
        "cluster_template_id": "57c92a7c-5c6a-42ea-9c6f-9f40a5aa4b36",
        "node_groups": [
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.maximum-allocation-mb": 2048,
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048
```

```
                    },
                    "MapReduce": {
                        "yarn.app.mapreduce.am.resource.mb": 256,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m"
                    }
                },
                "auto_security_group": false,
                "availability_zone": null,
                "count": 1,
                "flavor_id": "2",
                "id": "0fe07f2a-0275-4bc0-93b2-c3c1e48e2815",
                "security_groups": null,
                "use_autoconfig": true,
                "instances": [],
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:57:11",
                "node_group_template_id":
"0bb9f1a4-0c44-4dc5-9452-6741c62ed9ae",
                "updated_at": "2015-09-14T10:57:12",
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "master",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "namenode",
                    "resourcemanager",
                    "oozie",
                    "historyserver"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            },
            {
                "image_id": null,
                "shares": null,
                "floating_ip_pool": "033debed-aeb8-488c-b7d0-adb74c61faa5",
                "node_configs": {
                    "YARN": {
                        "yarn.nodemanager.vmem-check-enabled": "false",
                        "yarn.scheduler.maximum-allocation-mb": 2048,
                        "yarn.scheduler.minimum-allocation-mb": 256,
                        "yarn.nodemanager.resource.memory-mb": 2048
                    },
                    "MapReduce": {
                        "yarn.app.mapreduce.am.resource.mb": 256,
                        "mapreduce.task.io.sort.mb": 102,
                        "mapreduce.reduce.java.opts": "-Xmx409m",
                        "mapreduce.reduce.memory.mb": 512,
                        "mapreduce.map.memory.mb": 256,
                        "yarn.app.mapreduce.am.command-opts": "-Xmx204m",
                        "mapreduce.map.java.opts": "-Xmx204m"
                    }
                },
                "auto_security_group": false,
```

```
                "availability_zone": null,
                "count": 3,
                "flavor_id": "2",
                "id": "c7a3bea4-c898-446b-8c67-6d378d4c06c4",
                "security_groups": null,
                "use_autoconfig": true,
                "instances": [],
                "volumes_availability_zone": null,
                "created_at": "2015-09-14T10:57:11",
                "node_group_template_id": "846edb31-add5-46e6-a4ee-
a4c339f99251",
                "updated_at": "2015-09-14T10:57:12",
                "volumes_per_node": 0,
                "is_proxy_gateway": false,
                "name": "worker",
                "volume_mount_prefix": "/volumes/disk",
                "node_processes": [
                    "datanode",
                    "nodemanager"
                ],
                "volumes_size": 0,
                "volume_local_to_instance": false,
                "volume_type": null
            }
        ],
        "provision_progress": [],
        "hadoop_version": "2.7.1",
        "use_autoconfig": true,
        "trust_id": null,
        "description": null,
        "created_at": "2015-09-14T10:57:11",
        "is_protected": false,
        "updated_at": "2015-09-14T10:57:12",
        "is_transient": false,
        "cluster_configs": {
            "HDFS": {
                "dfs.replication": 3
            }
        },
        "anti_affinity": [],
        "name": "public-vanilla-cluster",
        "default_image_id": "4118a476-dfdc-4b0e-8d5c-463cba08e9ae",
        "status": "Validating"
    }
}
```

5.5.7. Show progress

Method	URI	Description
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows provisioning progress for a cluster.

Normal response codes: 200

5.5.7.1. Request

This table shows the URI parameters for the show progress request:

Name	Type	Description
`{cluster_id}`	UUID	The ID of the cluster

5.5.7.2. Response

Example 5.34. Show progress: JSON response


```
{
    "status": "Error",
    "neutron_management_network": "7e31648b-4b2e-4f32-9b0a-113581c27076",
    "is_transient": false,
    "description": "",
    "user_keypair_id": "vgridnev",
    "updated_at": "2015-03-31 14:10:59",
    "plugin_name": "spark",
    "provision_progress": [
        {
            "successful": false,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-03-31 14:10:20",
            "step_type": "Engine: create cluster",
            "updated_at": "2015-03-31 14:10:35",
            "events": [
                {
                    "instance_name": "sample-worker-spark-004",
                    "successful": false,
                    "created_at": "2015-03-31 14:10:35",
                    "updated_at": null,
                    "event_info": "Node sample-worker-spark-004 has error
 status\nError ID: 3e238c82-d1f5-4560-8ed8-691e923e16a0",
                    "instance_id": "b5ba5ba8-e9c1-47f7-9355-3ce0ec0e449d",
                    "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
                    "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
                    "id": "34afcfc7-bdb0-43cb-b142-283d560dc6ad"
                },
                {
                    "instance_name": "sample-worker-spark-001",
                    "successful": true,
                    "created_at": "2015-03-31 14:10:35",
                    "updated_at": null,
                    "event_info": null,
                    "instance_id": "c532ab71-38da-475a-95f8-f8eb93b8f1c2",
                    "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
```

```
                    "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
                    "id": "4ba50414-5216-4161-bc7a-12716122b99d"
                }
            ],
            "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
            "step_name": "Wait for instances to become active",
            "total": 5,
            "id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6"
        },
        {
            "successful": true,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-03-31 14:10:12",
            "step_type": "Engine: create cluster",
            "updated_at": "2015-03-31 14:10:19",
            "events": [],
            "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
            "step_name": "Run instances",
            "total": 5,
            "id": "407ba50a-c799-46af-9dfb-6aa5f6ade426"
        }
    ],
    "anti_affinity": [],
    "node_groups": [],
    "management_public_key": "Sahara",
    "status_description": "Creating cluster failed for the following
 reason(s): Node sample-worker-spark-004 has error status\nError ID: 3e238c82-
d1f5-4560-8ed8-691e923e16a0",
    "hadoop_version": "1.0.0",
    "id": "c26ec982-ba6b-4d75-1f8c-a50240164af0",
    "trust_id": null,
    "info": {},
    "cluster_template_id": "5a9a09a3-9349-43bd-9058-16c401fad2d5",
    "name": "sample",
    "cluster_configs": {},
    "created_at": "2015-03-31 14:10:07",
    "default_image_id": "e6a6c5da-67be-4017-a7d2-81f466efe67e",
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda"
}
```

5.5.8. Delete a cluster

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Deletes a cluster.

Normal response codes: 204

5.5.8.1. Request

This table shows the URI parameters for the delete a cluster request:

Name	Type	Description
`{cluster_id}`	UUID	The ID of the cluster

This operation does not accept a request body.

5.6. Event log

The event log feature provides information about cluster provisioning. In the event of errors, the event log shows the reason for the failure.

Method	URI	Description
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows provisioning progress of cluster.

5.6.1. Show progress

Method	URI	Description
GET	`/v1.1/{tenant_id}/clusters/{cluster_id}`	Shows provisioning progress of cluster.

Normal response codes: 200

5.6.1.1. Request

This table shows the URI parameters for the show progress request:

Name	Type	Description
`{cluster_id}`	UUID	The UUID of the cluster

5.6.1.2. Response

Example 5.35. Show progress: JSON response


```
{
    "status": "Error",
    "neutron_management_network": "7e31648b-4b2e-4f32-9b0a-113581c27076",
    "is_transient": false,
    "description": "",
    "user_keypair_id": "vgridnev",
    "updated_at": "2015-03-31 14:10:59",
    "plugin_name": "spark",
    "provision_progress": [
        {
            "successful": false,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-03-31 14:10:20",
            "step_type": "Engine: create cluster",
            "updated_at": "2015-03-31 14:10:35",
            "events": [
                {
                    "instance_name": "sample-worker-spark-004",
                    "successful": false,
                    "created_at": "2015-03-31 14:10:35",
                    "updated_at": null,
                    "event_info": "Node sample-worker-spark-004 has error
 status\nError ID: 3e238c82-d1f5-4560-8ed8-691e923e16a0",
                    "instance_id": "b5ba5ba8-e9c1-47f7-9355-3ce0ec0e449d",
                    "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
                    "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
                    "id": "34afcfc7-bdb0-43cb-b142-283d560dc6ad"
                },
                {
                    "instance_name": "sample-worker-spark-001",
                    "successful": true,
                    "created_at": "2015-03-31 14:10:35",
                    "updated_at": null,
                    "event_info": null,
                    "instance_id": "c532ab71-38da-475a-95f8-f8eb93b8f1c2",
                    "node_group_id": "145cf2fb-dcdf-42af-a4b9-a4047d2919d4",
```

```
                    "step_id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6",
                    "id": "4ba50414-5216-4161-bc7a-12716122b99d"
                }
            ],
            "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
            "step_name": "Wait for instances to become active",
            "total": 5,
            "id": "3f243c67-2c27-47c7-a0c0-0834ad17f8b6"
        },
        {
            "successful": true,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-03-31 14:10:12",
            "step_type": "Engine: create cluster",
            "updated_at": "2015-03-31 14:10:19",
            "events": [],
            "cluster_id": "c26ec982-ba6b-4d75-818c-a50240164af0",
            "step_name": "Run instances",
            "total": 5,
            "id": "407ba50a-c799-46af-9dfb-6aa5f6ade426"
        }
    ],
    "anti_affinity": [],
    "node_groups": [],
    "management_public_key": "Sahara",
    "status_description": "Creating cluster failed for the following
 reason(s): Node sample-worker-spark-004 has error status\nError ID: 3e238c82-
d1f5-4560-8ed8-691e923e16a0",
    "hadoop_version": "1.0.0",
    "id": "c26ec982-ba6b-4d75-1f8c-a50240164af0",
    "trust_id": null,
    "info": {},
    "cluster_template_id": "5a9a09a3-9349-43bd-9058-16c401fad2d5",
    "name": "sample",
    "cluster_configs": {},
    "created_at": "2015-03-31 14:10:07",
    "default_image_id": "e6a6c5da-67be-4017-a7d2-81f466efe67e",
    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda"
}
```


5.7. Data sources

A data source object defines the location of input or output for MapReduce jobs and might reference different types of storage.

The Data Processing service does not validate data source locations.

Method	URI	Description
GET	`/v1.1/{tenant_id}/data-sources`	Lists all data sources.
POST	`/v1.1/{tenant_id}/data-sources`	Creates a data source.
GET	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Shows details for a data source.
PUT	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Updates a data source.
DELETE	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Deletes a data source.

5.7.1. List data sources

Method	URI	Description
GET	`/v1.1/{tenant_id}/data-sources`	Lists all data sources.

Normal response codes: 200

5.7.1.1. Request

This operation does not accept a request body.

5.7.1.2. Response

Example 5.36. List data sources: JSON response

```
{
    "data_sources": [
        {
            "is_public": false,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "is_protected": false,
            "created_at": "2015-03-26 11:18:10",
            "id": "953831f2-0852-49d8-ac71-af5805e25256",
            "name": "swift_input",
            "updated_at": null,
            "description": "This is input",
            "url": "swift://container/text",
            "type": "swift"
        },
        {
            "is_public": false,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "is_protected": false,
            "created_at": "2015-03-26 11:09:36",
            "id": "d7fffe9c-3b42-46a9-8be8-e98f586fa7a9",
            "name": "hdfs_input",
            "updated_at": null,
            "description": "This is hdfs input",
            "url": "hdfs://test-master-node:8020/user/hadoop/input",
            "type": "hdfs"
        }
    ]
}
```

5.7.2. Create data source

Method	URI	Description
POST	`/v1.1/{tenant_id}/data-sources`	Creates a data source.

Normal response codes: 202

5.7.2.1. Request

Example 5.37. Create data source: JSON request

```
{
    "description": "This is input",
    "url": "swift://container/text",
    "credentials": {
        "password": "swordfish",
        "user": "dev"
    },
    "type": "swift",
    "name": "swift_input"
}
```

Example 5.38. Create data source: JSON request

```
{
    "description": "This is hdfs input",
    "url": "hdfs://test-master-node:8020/user/hadoop/input",
    "type": "hdfs",
    "name": "hdfs_input"
}
```

5.7.2.2. Response

Example 5.39. Create data source: JSON response

```
{
    "data_source": {
        "is_public": false,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "is_protected": false,
        "created_at": "2015-03-26 11:18:10.691493",
        "id": "953831f2-0852-49d8-ac71-af5805e25256",
        "updated_at": null,
        "name": "swift_input",
        "description": "This is input",
        "url": "swift://container/text",
        "type": "swift"
    }
}
```

Example 5.40. Create data source: JSON response

```
{
    "data_source": {
        "is_public": false,
```

```
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "is_protected": false,
        "created_at": "2015-03-26 11:09:36.148464",
        "id": "d7fffe9c-3b42-46a9-8be8-e98f586fa7a9",
        "updated_at": null,
        "name": "hdfs_input",
        "description": "This is hdfs input",
        "url": "hdfs://test-master-node:8020/user/hadoop/input",
        "type": "hdfs"
    }
}
```

5.7.3. Show data source details

Method	URI	Description
GET	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Shows details for a data source.

Normal response codes: 200

5.7.3.1. Request

This table shows the URI parameters for the show data source details request:

Name	Type	Description
`{data_source_id}`	UUID	The UUID of the data source.

This operation does not accept a request body.

5.7.3.2. Response

Example 5.41. Show data source details: JSON response

```
{
    "data_source": {
        "is_public": false,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "is_protected": false,
        "created_at": "2015-03-26 11:18:10.691493",
        "id": "953831f2-0852-49d8-ac71-af5805e25256",
        "updated_at": null,
        "name": "swift_input",
        "description": "This is input",
        "url": "swift://container/text",
        "type": "swift"
    }
}
```

5.7.4. Update data source

Method	URI	Description
PUT	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Updates a data source.

Normal response codes: 202

5.7.4.1. Request

This table shows the URI parameters for the update data source request:

Name	Type	Description
`{data_source_id}`	UUID	The UUID of the data source.

Example 5.42. Update data source: JSON request

```
{
    "description": "This is public input",
    "is_protected": true
}
```

5.7.4.2. Response

Example 5.43. Update data source: JSON response

```
{
    "data_source": {
        "is_public": true,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "is_protected": false,
        "created_at": "2015-09-15 12:32:24.847493",
        "id": "953831f2-0852-49d8-ac71-af5805e25256",
        "updated_at": "2015-09-15 12:34:42.597435",
        "name": "swift_input",
        "description": "This is public input",
        "url": "swift://container/text",
        "type": "swift"
    }
}
```

5.7.5. Delete data source

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/data-sources/{data_source_id}`	Deletes a data source.

Normal response codes: 204

5.7.5.1. Request

This table shows the URI parameters for the delete data source request:

Name	Type	Description
`{data_source_id}`	UUID	The UUID of the data source.

This operation does not accept a request body.

5.8. Job binary internals

Job binary internal objects represent data processing applications and libraries that are stored in the internal database.

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binary-internals`	Lists the available job binary internals.
PUT	`/v1.1/{tenant_id}/job-binary-internals/{name}`	Creates a job binary internal.
GET	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Shows details for a job binary internal.
DELETE	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Deletes a job binary internal.
PATCH	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Updates a job binary internal.
GET	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data`	Shows data for a job binary internal.

5.8.1. List job binary internals

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binary-internals`	Lists the available job binary internals.

Normal response codes: 200

5.8.1.1. Request

This operation does not accept a request body.

5.8.1.2. Response

Example 5.44. List job binary internals: JSON response

```
{
    "binaries": [
        {
            "is_public": false,
            "name": "example.pig",
            "tenant_id": "11587919cc534bcbb1027a161c82cf58",
            "created_at": "2013-10-15 12:36:59.329034",
            "updated_at": null,
            "datasize": 161,
            "id": "d2498cbf-4589-484a-a814-81436c18beb3",
            "is_protected": false
        },
        {
            "is_public": false,
            "name": "udf.jar",
            "tenant_id": "11587919cc534bcbb1027a161c82cf58",
            "created_at": "2013-10-15 12:43:52.008620",
            "updated_at": null,
            "datasize": 3745,
            "id": "22f1d87a-23c8-483e-a0dd-cb4a16dde5f9",
            "is_protected": false
        }
    ]
}
```

5.8.2. Create job binary internal

Method	URI	Description
PUT	`/v1.1/{tenant_id}/job-binary-internals/{name}`	Creates a job binary internal.

Job binary internals are objects that represent data processing applications and libraries that are stored in the internal database.

Specify the file contents (raw data or script text) in the request body. Specify the file name in the URI.

Normal response codes: 202

5.8.2.1. Request

This table shows the URI parameters for the create job binary internal request:

Name	Type	Description
`{name}`	String	The name of the job binary internal.

This operation does not accept a request body.

5.8.2.2. Response

Example 5.45. Create job binary internal: JSON response

```
{
    "job_binary_internal": {
        "is_public": false,
        "name": "script.pig",
        "tenant_id": "11587919cc534bcbb1027a161c82cf58",
        "created_at": "2013-10-15 13:17:35.994466",
        "updated_at": null,
        "datasize": 160,
        "id": "4833dc4b-8682-4d5b-8a9f-2036b47a0996",
        "is_protected": false
    }
}
```

5.8.3. Show job binary internal details

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Shows details for a job binary internal.

Normal response codes: 200

5.8.3.1. Request

This table shows the URI parameters for the show job binary internal details request:

Name	Type	Description
`{job_binary_internals_id}`	UUID	The UUID of the job binary internal.

This operation does not accept a request body.

5.8.3.2. Response

Example 5.46. Show job binary internal details: JSON response

```
{
    "job_binary_internal": {
        "is_public": false,
        "name": "script.pig",
        "tenant_id": "11587919cc534bcbb1027a161c82cf58",
        "created_at": "2013-10-15 13:17:35.994466",
        "updated_at": null,
        "datasize": 160,
        "id": "4833dc4b-8682-4d5b-8a9f-2036b47a0996",
        "is_protected": false
    }
}
```

5.8.4. Delete job binary internal

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Deletes a job binary internal.

Normal response codes: 204

5.8.4.1. Request

This table shows the URI parameters for the delete job binary internal request:

Name	Type	Description
`{job_binary_internals_id}`	UUID	The UUID of the job binary internal.

This operation does not accept a request body.

5.8.5. Update job binary internal

Method	URI	Description
PATCH	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}`	Updates a job binary internal.

Normal response codes: 202

5.8.5.1. Request

This table shows the URI parameters for the update job binary internal request:

Name	Type	Description
`{job_binary_internals_id}`	UUID	The UUID of the job binary internal.

Example 5.47. Update job binary internal : JSON request

```
{
    "name": "public-jbi",
    "is_public": true
}
```

5.8.5.2. Response

Example 5.48. Update job binary internal: JSON response

```
{
    "job_binary_internal": {
        "is_public": true,
        "name": "public-jbi",
        "tenant_id": "11587919cc534bcbb1027a161c82cf58",
        "created_at": "2015-09-15 13:21:54.485912",
        "updated_at": "2015-09-15 13:24:24.590124",
        "datasize": 200,
        "id": "2433dc4b-8682-4d5b-8a9f-2036d47a0996",
        "is_protected": false
    }
}
```

5.8.6. Show job binary internal data

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binary-internals/{job_binary_internals_id}/data`	Shows data for a job binary internal.

The response body shows the job binary raw data and the response headers show the data length.

Example response:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 161
Content-Type: text/html; charset=utf-8
Date: Sat, 28 Mar 2016 02:21:13 GMT

A = load '$INPUT' using PigStorage(':') as (fruit: chararray);
B = foreach A generate com.hadoopbook.pig.Trim(fruit);
store B into '$OUTPUT' USING PigStorage();
```

Normal response codes: 200

5.8.6.1. Request

This table shows the URI parameters for the show job binary internal data request:

Name	Type	Description
`{job_binary_internals_id}`	UUID	The UUID of the job binary internal.

This operation does not accept a request body.

5.8.6.2. Response

This operation does not return a response body.

5.9. Job binaries

Job binary objects represent data processing applications and libraries that are stored in either the internal database or the Object Storage service.

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binaries`	Lists the available job binaries.
GET	`/v1.1/{tenant_id}/job-binaries`	Shows details for a job binary.
POST	`/v1.1/{tenant_id}/job-binaries`	Creates a job binary.
PUT	`/v1.1/{tenant_id}/job-binaries`	Updates a job binary.
DELETE	`/v1.1/{tenant_id}/job-binaries`	Deletes a job binary.
GET	`/v1.1/{tenant_id}/job-binaries/{job_binary_id}/data`	Shows data for a job binary.

5.9.1. List job binaries

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binaries`	Lists the available job binaries.

Normal response codes: 200

5.9.1.1. Request

This operation does not accept a request body.

5.9.1.2. Response

Example 5.49. List job binaries: JSON response

```
{
    "binaries": [
        {
            "is_public": false,
            "description": "",
            "url": "internal-db://d2498cbf-4589-484a-a814-81436c18beb3",
            "tenant_id": "11587919cc534bcbb1027a161c82cf58",
            "created_at": "2013-10-15 12:36:59.375060",
            "updated_at": null,
            "id": "84248975-3c82-4206-a58d-6e7fb3a563fd",
            "name": "example.pig",
            "is_protected": false
        },
        {
            "is_public": false,
            "description": "",
            "url": "internal-db://22f1d87a-23c8-483e-a0dd-cb4a16dde5f9",
            "tenant_id": "11587919cc534bcbb1027a161c82cf58",
            "created_at": "2013-10-15 12:43:52.265899",
            "updated_at": null,
            "id": "508fc62d-1d58-4412-b603-bdab307bb926",
            "name": "udf.jar",
            "is_protected": false
        },
        {
            "is_public": false,
            "description": "",
            "url": "swift://container/jar-example.jar",
            "tenant_id": "11587919cc534bcbb1027a161c82cf58",
            "created_at": "2013-10-15 14:25:04.970513",
            "updated_at": null,
            "id": "a716a9cd-9add-4b12-b1b6-cdb71aaef350",
            "name": "jar-example.jar",
            "is_protected": false
        }
    ]
}
```

5.9.2. Show job binary details

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binaries`	Shows details for a job binary.

Normal response codes: 200

5.9.2.1. Request

This operation does not accept a request body.

5.9.2.2. Response

Example 5.50. Show job binary details: JSON response

```
{
    "job_binary": {
        "is_public": false,
        "description": "an example jar file",
        "url": "swift://container/jar-example.jar",
        "tenant_id": "11587919cc534bcbb1027a161c82cf58",
        "created_at": "2013-10-15 14:25:04.970513",
        "updated_at": null,
        "id": "a716a9cd-9add-4b12-b1b6-cdb71aaef350",
        "name": "jar-example.jar",
        "is_protected": false
    }
}
```

5.9.3. Create job binary

Method	URI	Description
POST	`/v1.1/{tenant_id}/job-binaries`	Creates a job binary.

Normal response codes: 202

5.9.3.1. Request

Example 5.51. Create job binary: JSON request

```
{
    "url": "swift://container/jar-example.jar",
    "name": "jar-example.jar",
    "description": "This is a job binary",
    "extra": {
        "password": "swordfish",
        "user": "admin"
    }
}
```

5.9.3.2. Response

Example 5.52. Create job binary: JSON response

```
{
    "job_binary": {
        "is_public": false,
        "description": "This is a job binary",
        "url": "swift://container/jar-example.jar",
        "tenant_id": "11587919cc534bcbb1027a161c82cf58",
        "created_at": "2013-10-15 14:49:20.106452",
        "id": "07f86352-ee8a-4b08-b737-d705ded5ff9c",
        "updated_at": null,
        "name": "jar-example.jar",
        "is_protected": false
    }
}
```

5.9.4. Update job binary

Method	URI	Description
PUT	`/v1.1/{tenant_id}/job-binaries`	Updates a job binary.

Normal response codes: 202

5.9.4.1. Request

Example 5.53. Update job binary : JSON request

```
{
    "url": "swift://container/new-jar-example.jar",
    "name": "new-jar-example.jar",
    "description": "This is a new job binary"
}
```

5.9.4.2. Response

Example 5.54. Update job binary: JSON response

```
{
    "job_binary": {
        "is_public": false,
        "description": "This is a new job binary",
        "url": "swift://container/new-jar-example.jar",
        "tenant_id": "11587919cc534bcbb1027a161c82cf58",
        "created_at": "2015-09-15 12:42:51.421542",
        "updated_at": null,
        "id": "b713d7ad-4add-4f12-g1b6-cdg71aaef350",
        "name": "new-jar-example.jar",
        "is_protected": false
    }
}
```

5.9.5. Delete job binary

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/job-binaries`	Deletes a job binary.

Normal response codes: 204

5.9.5.1. Request

This operation does not accept a request body.

5.9.6. Show job binary data

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-binaries/{job_binary_id}/data`	Shows data for a job binary.

The response body shows the job binary raw data and the response headers show the data length.

Example response:

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Length: 161
Content-Type: text/html; charset=utf-8
Date: Sat, 28 Mar 2016 02:42:48 GMT

A = load '$INPUT' using PigStorage(':') as (fruit: chararray);
B = foreach A generate com.hadoopbook.pig.Trim(fruit);
store B into '$OUTPUT' USING PigStorage();
```

Normal response codes: 200

5.9.6.1. Request

This table shows the URI parameters for the show job binary data request:

Name	Type	Description
`{job_binary_id}`	UUID	The UUID of the job binary.

This operation does not accept a request body.

5.9.6.2. Response

This operation does not return a response body.

5.10. Jobs

A job object lists the binaries that a job needs to run. To run a job, you must specify data sources and job parameters.

You can run a job on an existing or new transient cluster.

Method	URI	Description
GET	`/v1.1/{tenant_id}/jobs`	Lists all jobs.
POST	`/v1.1/{tenant_id}/jobs`	Creates a job object.
GET	`/v1.1/{tenant_id}/jobs/{job_id}`	Shows details for a job.
DELETE	`/v1.1/{tenant_id}/jobs/{job_id}`	Removes a job.
PATCH	`/v1.1/{tenant_id}/jobs/{job_id}`	Updates a job object.
POST	`/v1.1/{tenant_id}/jobs/{job_id}/execute`	Runs a job.

5.10.1. List jobs

Method	URI	Description
GET	/v1.1/{tenant_id}/jobs	Lists all jobs.

Normal response codes: 200

5.10.1.1. Request

This operation does not accept a request body.

5.10.1.2. Response

Example 5.55. List jobs: JSON response


```
{
    "jobs": [
        {
            "is_public": false,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-02-10 14:25:48",
            "id": "1a674c31-9aaa-4d07-b844-2bf200a1b836",
            "name": "Edp-test-job-3d60854e",
            "updated_at": null,
            "description": "",
            "interface": [],
            "libs": [
                {
                    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                    "created_at": "2015-02-10 14:25:48",
                    "id": "0ff4ac10-94a4-4e25-9ac9-603afe27b100",
                    "name": "binary-job-339c2d1a.jar",
                    "updated_at": null,
                    "description": "",
                    "url": "swift://Edp-test-c71e6bce.sahara/binary-
job-339c2d1a.jar"
                }
            ],
            "type": "MapReduce",
            "mains": [],
            "is_protected": false
        },
        {
            "is_public": false,
            "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
            "created_at": "2015-02-10 14:25:44",
            "id": "4d1f3759-3497-4927-8352-910bacf24e62",
            "name": "Edp-test-job-6b6953c8",
            "updated_at": null,
            "description": "",
            "interface": [],
            "libs": [
                {
                    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                    "created_at": "2015-02-10 14:25:44",
                    "id": "e0d47800-4ac1-4d63-a2e1-c92d669a44e2",
```

```
                    "name": "binary-job-6f21a2f8.jar",
                    "updated_at": null,
                    "description": "",
                    "url": "swift://Edp-test-b409ec68.sahara/binary-
job-6f21a2f8.jar"
                }
            ],
            "type": "Pig",
            "mains": [
                {
                    "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                    "created_at": "2015-02-10 14:25:44",
                    "id": "e073e896-f123-4b76-995f-901d786262df",
                    "name": "binary-job-d4f8bd75.pig",
                    "updated_at": null,
                    "description": "",
                    "url": "swift://Edp-test-b409ec68.sahara/binary-job-
d4f8bd75.pig"
                }
            ],
            "is_protected": false
        }
    ]
}
```

5.10.2. Create job

Method	URI	Description
POST	`/v1.1/{tenant_id}/jobs`	Creates a job object.

Normal response codes: 202

5.10.2.1. Request

Example 5.56. Create job: JSON request

```
{
    "description": "This is pig job example",
    "mains": [
        "90d9d5ec-11aa-48bd-bc8c-34936ce0db6e"
    ],
    "libs": [
        "320a2ca7-25fd-4b48-9bc3-4fb1b6c4ff27"
    ],
    "type": "Pig",
    "name": "pig-job-example"
}
```

5.10.2.2. Response

Example 5.57. Create job: JSON response

```
{
    "job": {
        "is_public": false,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-03-27 08:48:38.630827",
        "id": "71defc8f-d005-484f-9d86-1aedf644d1ef",
        "name": "pig-job-example",
        "description": "This is pig job example",
        "interface": [],
        "libs": [
            {
                "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                "created_at": "2015-02-10 14:25:53",
                "id": "320a2ca7-25fd-4b48-9bc3-4fb1b6c4ff27",
                "name": "binary-job",
                "updated_at": null,
                "description": "",
                "url": "internal-db://c6a925fa-ac1d-4b2e-b88a-7054e1927521"
            }
        ],
        "type": "Pig",
        "is_protected": false,
        "mains": [
            {
                "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                "created_at": "2015-02-03 10:47:51",
                "id": "90d9d5ec-11aa-48bd-bc8c-34936ce0db6e",
                "name": "pig",
                "updated_at": null,
```

```
                "description": "",
                "url": "internal-db://872878f6-72ea-44db-8d1d-e6a6396d2df0"
            }
        ]
    }
}
```

5.10.3. Show job details

Method	URI	Description
GET	`/v1.1/{tenant_id}/jobs/{job_id}`	Shows details for a job.

Normal response codes: 200

5.10.3.1. Request

This table shows the URI parameters for the show job details request:

Name	Type	Description
`{job_id}`	UUID	The UUID of the job.

This operation does not accept a request body.

5.10.3.2. Response

Example 5.58. Show job details: JSON response

```
{
    "job": {
        "is_public": false,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-02-10 14:25:48",
        "id": "1a674c31-9aaa-4d07-b844-2bf200a1b836",
        "name": "Edp-test-job",
        "updated_at": null,
        "description": "",
        "interface": [],
        "libs": [
            {
                "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                "created_at": "2015-02-10 14:25:48",
                "id": "0ff4ac10-94a4-4e25-9ac9-603afe27b100",
                "name": "binary-job.jar",
                "updated_at": null,
                "description": "",
                "url": "swift://Edp-test-c71e6bce.sahara/binary-job.jar"
            }
        ],
        "type": "MapReduce",
        "mains": [],
        "is_protected": false
    }
}
```

5.10.4. Remove job

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/jobs/{job_id}`	Removes a job.

Normal response codes: 204

5.10.4.1. Request

This table shows the URI parameters for the remove job request:

Name	Type	Description
`{job_id}`	UUID	The UUID of the job.

This operation does not accept a request body.

5.10.5. Update job object

Method	URI	Description
PATCH	`/v1.1/{tenant_id}/jobs/{job_id}`	Updates a job object.

Normal response codes: 202

5.10.5.1. Request

This table shows the URI parameters for the update job object request:

Name	Type	Description
`{job_id}`	UUID	The UUID of the job.

Example 5.59. Update job object: JSON request

```
{
    "description": "This is public pig job example",
    "name": "public-pig-job-example"
}
```

5.10.5.2. Response

Example 5.60. Update job object: JSON response

```
{
    "job": {
        "is_public": false,
        "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
        "created_at": "2015-02-10 14:25:48",
        "id": "1a674c31-9aaa-4d07-b844-2bf200a1b836",
        "name": "public-pig-job-example",
        "updated_at": null,
        "description": "This is public pig job example",
        "interface": [],
        "libs": [
            {
                "tenant_id": "9cd1314a0a31493282b6712b76a8fcda",
                "created_at": "2015-02-10 14:25:48",
                "id": "0ff4ac10-94a4-4e25-9ac9-603afe27b100",
                "name": "binary-job.jar",
                "updated_at": null,
                "description": "",
                "url": "swift://Edp-test-c71e6bce.sahara/binary-job.jar"
            }
        ],
        "type": "MapReduce",
        "mains": [],
        "is_protected": false
    }
}
```

5.10.6. Run job

Method	URI	Description
POST	`/v1.1/{tenant_id}/jobs/{job_id}/execute`	Runs a job.

Normal response codes: 202

5.10.6.1. Request

This table shows the URI parameters for the run job request:

Name	Type	Description
`{job_id}`	UUID	The UUID of the job.

Example 5.61. Run job: JSON request

```
{
    "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
    "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
    "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
    "job_configs": {
        "configs": {
            "mapred.map.tasks": "1",
            "mapred.reduce.tasks": "1"
        },
        "args": [
            "arg1",
            "arg2"
        ],
        "params": {
            "param2": "value2",
            "param1": "value1"
        }
    }
}
```

5.10.6.2. Response

Example 5.62. Run job: JSON response

```
{
    "job_execution": {
        "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
        "is_protected": false,
        "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
        "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
        "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
        "created_at": "2015-09-15T09:49:24",
        "is_public": false,
        "id": "20da9edb-12ce-4b45-a473-41baeefef997",
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "job_configs": {
            "configs": {
                "mapred.reduce.tasks": "1",
```

```
                "mapred.map.tasks": "1"
            },
            "args": [
                "arg1",
                "arg2"
            ],
            "params": {
                "param2": "value2",
                "param1": "value1"
            }
        },
        "info": {
            "status": "PENDING"
        }
    }
}
```

5.11. Job executions

A job execution object represents a Hadoop job that runs on a cluster. A job execution polls the status of a running job and reports it to the user. Also a user can cancel a running job.

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-executions`	Lists available job executions.
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Shows details for a job execution, by ID.
DELETE	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Deletes a job execution.
PATCH	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Updates a job execution.
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}/refresh-status`	Refreshes the status of and shows information for a job execution.
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}/cancel`	Cancels a job execution.

5.11.1. List job executions

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-executions`	Lists available job executions.

Normal response codes: 200

5.11.1.1. Request

This operation does not accept a request body.

5.11.1.2. Response

Example 5.63. List job executions: JSON response

```
{
    "job_executions": [
        {
            "job_configs": {
                "configs": {
                    "mapred.reduce.tasks": "1",
                    "mapred.map.tasks": "1"
                },
                "args": [
                    "arg1",
                    "arg2"
                ],
                "params": {
                    "param2": "value2",
                    "param1": "value1"
                }
            },
            "is_protected": false,
            "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
            "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
            "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
            "created_at": "2015-09-15T09:49:24",
            "end_time": "2015-09-15T12:50:46",
            "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
            "is_public": false,
            "updated_at": "2015-09-15T09:50:46",
            "return_code": null,
            "data_source_urls": {
                "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/
input",
                "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/
output"
            },
            "tenant_id": "808d5032ea0446889097723bfc8e919d",
            "start_time": "2015-09-15T12:49:43",
            "id": "20da9edb-12ce-4b45-a473-41baeefef997",
            "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
            "info": {
                "user": "hadoop",
                "actions": [
                    {
                        "name": ":start:",
```

```
                        "trackerUri": "-",
                        "externalStatus": "OK",
                        "status": "OK",
                        "externalId": "-",
                        "transition": "job-node",
                        "data": null,
                        "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                        "errorCode": null,
                        "id": "0000001-150915094349962-oozie-hado-W@:start:",
                        "consoleUrl": "-",
                        "errorMessage": null,
                        "toString": "Action name[:start:] status[OK]",
                        "stats": null,
                        "type": ":START:",
                        "retries": 0,
                        "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                        "externalChildIDs": null,
                        "cred": "null"
                    },
                    {
                        "name": "job-node",
                        "trackerUri": "http://172.18.168.119:8032",
                        "externalStatus": "FAILED/KILLED",
                        "status": "ERROR",
                        "externalId": "job_1442310173665_0002",
                        "transition": "fail",
                        "data": null,
                        "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                        "errorCode": "JA018",
                        "id": "0000001-150915094349962-oozie-hado-W@job-node",
                        "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
                        "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
                        "toString": "Action name[job-node] status[ERROR]",
                        "stats": null,
                        "type": "pig",
                        "retries": 0,
                        "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                        "externalChildIDs": null,
                        "cred": "null"
                    },
                    {
                        "name": "fail",
                        "trackerUri": "-",
                        "externalStatus": "OK",
                        "status": "OK",
                        "externalId": "-",
                        "transition": null,
                        "data": null,
                        "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                        "errorCode": "E0729",
                        "id": "0000001-150915094349962-oozie-hado-W@fail",
                        "consoleUrl": "-",
                        "errorMessage": "Workflow failed, error message[Main
 class [org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
                        "toString": "Action name[fail] status[OK]",
                        "stats": null,
                        "type": ":KILL:",
                        "retries": 0,
```

```
                        "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                        "externalChildIDs": null,
                        "cred": "null"
                    }
                ],
                "createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
                "status": "KILLED",
                "group": null,
                "externalId": null,
                "acl": null,
                "run": 0,
                "appName": "job-wf",
                "parentId": null,
                "conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
  <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
 <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
                "id": "0000001-150915094349962-oozie-hado-W",
                "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
                "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
 status[KILLED]",
                "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?
job=0000001-150915094349962-oozie-hado-W"
            }
        }
    ]
}
```

5.11.2. Show job execution details

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Shows details for a job execution, by ID.

Normal response codes: 200

5.11.2.1. Request

This table shows the URI parameters for the show job execution details request:

Name	Type	Description
`{job_execution_id}`	UUID	The UUID of the job execution.

This operation does not accept a request body.

5.11.2.2. Response

Example 5.64. Show job execution details: JSON response


```
{
    "job_execution": {
        "job_configs": {
            "configs": {
                "mapred.reduce.tasks": "1",
                "mapred.map.tasks": "1"
            },
            "args": [
                "arg1",
                "arg2"
            ],
            "params": {
                "param2": "value2",
                "param1": "value1"
            }
        },
        "is_protected": false,
        "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
        "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
        "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
        "created_at": "2015-09-15T09:49:24",
        "end_time": "2015-09-15T12:50:46",
        "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
        "is_public": false,
        "updated_at": "2015-09-15T09:50:46",
        "return_code": null,
        "data_source_urls": {
            "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
            "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
        },
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "start_time": "2015-09-15T12:49:43",
        "id": "20da9edb-12ce-4b45-a473-41baeefef997",
        "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
        "info": {
            "user": "hadoop",
```

```
            "actions": [
                {
                    "name": ":start:",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": "job-node",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "errorCode": null,
                    "id": "0000001-150915094349962-oozie-hado-W@:start:",
                    "consoleUrl": "-",
                    "errorMessage": null,
                    "toString": "Action name[:start:] status[OK]",
                    "stats": null,
                    "type": ":START:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "job-node",
                    "trackerUri": "http://172.18.168.119:8032",
                    "externalStatus": "FAILED/KILLED",
                    "status": "ERROR",
                    "externalId": "job_1442310173665_0002",
                    "transition": "fail",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "JA018",
                    "id": "0000001-150915094349962-oozie-hado-W@job-node",
                    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
                    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
                    "toString": "Action name[job-node] status[ERROR]",
                    "stats": null,
                    "type": "pig",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "fail",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": null,
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "E0729",
                    "id": "0000001-150915094349962-oozie-hado-W@fail",
                    "consoleUrl": "-",
                    "errorMessage": "Workflow failed, error message[Main class
 [org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
                    "toString": "Action name[fail] status[OK]",
```

```
                    "stats": null,
                    "type": ":KILL:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                }
            ],
            "createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
            "status": "KILLED",
            "group": null,
            "externalId": null,
            "acl": null,
            "run": 0,
            "appName": "job-wf",
            "parentId": null,
            "conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
  <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
 <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
            "id": "0000001-150915094349962-oozie-hado-W",
            "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
            "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
            "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
 status[KILLED]",
            "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
0000001-150915094349962-oozie-hado-W"
        }
    }
}
```

5.11.3. Delete job execution

Method	URI	Description
DELETE	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Deletes a job execution.

Normal response codes: 204

5.11.3.1. Request

This table shows the URI parameters for the delete job execution request:

Name	Type	Description
`{job_execution_id}`	UUID	The UUID of the job execution.

This operation does not accept a request body.

5.11.4. Update job execution

Method	URI	Description
PATCH	`/v1.1/{tenant_id}/job-executions/{job_execution_id}`	Updates a job execution.

Normal response codes: 202

5.11.4.1. Request

This table shows the URI parameters for the update job execution request:

Name	Type	Description
`{job_execution_id}`	UUID	The UUID of the job execution.

Example 5.65. Update job execution: JSON request

```
{
    "is_public": true
}
```

5.11.4.2. Response

Example 5.66. Update job execution: JSON response

```
{
    "job_execution": {
        "job_configs": {
            "configs": {
                "mapred.reduce.tasks": "1",
                "mapred.map.tasks": "1"
            },
            "args": [
                "arg1",
                "arg2"
            ],
            "params": {
                "param2": "value2",
                "param1": "value1"
            }
        },
        "is_protected": false,
        "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
        "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
        "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
        "created_at": "2015-09-15T09:49:24",
        "end_time": "2015-09-15T12:50:46",
        "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
        "is_public": true,
        "updated_at": "2015-09-15T09:50:46",
        "return_code": null,
        "data_source_urls": {
            "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
            "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
        },
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "start_time": "2015-09-15T12:49:43",
```

```
        "id": "20da9edb-12ce-4b45-a473-41baeefef997",
        "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
        "info": {
            "user": "hadoop",
            "actions": [
                {
                    "name": ":start:",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": "job-node",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "errorCode": null,
                    "id": "0000001-150915094349962-oozie-hado-W@:start:",
                    "consoleUrl": "-",
                    "errorMessage": null,
                    "toString": "Action name[:start:] status[OK]",
                    "stats": null,
                    "type": ":START:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "job-node",
                    "trackerUri": "http://172.18.168.119:8032",
                    "externalStatus": "FAILED/KILLED",
                    "status": "ERROR",
                    "externalId": "job_1442310173665_0002",
                    "transition": "fail",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "JA018",
                    "id": "0000001-150915094349962-oozie-hado-W@job-node",
                    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
                    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
                    "toString": "Action name[job-node] status[ERROR]",
                    "stats": null,
                    "type": "pig",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "fail",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": null,
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "E0729",
                    "id": "0000001-150915094349962-oozie-hado-W@fail",
```

```
                    "consoleUrl": "-",
                    "errorMessage": "Workflow failed, error message[Main class
 [org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
                    "toString": "Action name[fail] status[OK]",
                    "stats": null,
                    "type": ":KILL:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                }
            ],
            "createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
            "status": "KILLED",
            "group": null,
            "externalId": null,
            "acl": null,
            "run": 0,
            "appName": "job-wf",
            "parentId": null,
            "conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
  <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
 <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
            "id": "0000001-150915094349962-oozie-hado-W",
            "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
            "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
            "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
 status[KILLED]",
            "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
0000001-150915094349962-oozie-hado-W"
        }
    }
}
```

5.11.5. Refresh job execution status

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}/refresh-status`	Refreshes the status of and shows information for a job execution.

Normal response codes: 200

5.11.5.1. Request

This table shows the URI parameters for the refresh job execution status request:

Name	Type	Description
`{job_execution_id}`	UUID	The UUID of the job execution.

This operation does not accept a request body.

5.11.5.2. Response

Example 5.67. Refresh job execution status: JSON response

```
{
    "job_execution": {
        "job_configs": {
            "configs": {
                "mapred.reduce.tasks": "1",
                "mapred.map.tasks": "1"
            },
            "args": [
                "arg1",
                "arg2"
            ],
            "params": {
                "param2": "value2",
                "param1": "value1"
            }
        },
        "is_protected": false,
        "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
        "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
        "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
        "created_at": "2015-09-15T09:49:24",
        "end_time": "2015-09-15T12:50:46",
        "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
        "is_public": false,
        "updated_at": "2015-09-15T09:50:46",
        "return_code": null,
        "data_source_urls": {
            "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
            "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
        },
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "start_time": "2015-09-15T12:49:43",
        "id": "20da9edb-12ce-4b45-a473-41baeefef997",
        "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
        "info": {
```

```
            "user": "hadoop",
            "actions": [
                {
                    "name": ":start:",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": "job-node",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "errorCode": null,
                    "id": "0000001-150915094349962-oozie-hado-W@:start:",
                    "consoleUrl": "-",
                    "errorMessage": null,
                    "toString": "Action name[:start:] status[OK]",
                    "stats": null,
                    "type": ":START:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "job-node",
                    "trackerUri": "http://172.18.168.119:8032",
                    "externalStatus": "FAILED/KILLED",
                    "status": "ERROR",
                    "externalId": "job_1442310173665_0002",
                    "transition": "fail",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "JA018",
                    "id": "0000001-150915094349962-oozie-hado-W@job-node",
                    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
                    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
                    "toString": "Action name[job-node] status[ERROR]",
                    "stats": null,
                    "type": "pig",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "fail",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": null,
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "E0729",
                    "id": "0000001-150915094349962-oozie-hado-W@fail",
                    "consoleUrl": "-",
                    "errorMessage": "Workflow failed, error message[Main class
 [org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
```

```
                    "toString": "Action name[fail] status[OK]",
                    "stats": null,
                    "type": ":KILL:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                }
            ],
            "createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
            "status": "KILLED",
            "group": null,
            "externalId": null,
            "acl": null,
            "run": 0,
            "appName": "job-wf",
            "parentId": null,
            "conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
  <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
 <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
            "id": "0000001-150915094349962-oozie-hado-W",
            "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
            "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
            "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
 status[KILLED]",
            "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
0000001-150915094349962-oozie-hado-W"
        }
    }
}
```

5.11.6. Cancel job execution

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-executions/{job_execution_id}/cancel`	Cancels a job execution.

Normal response codes: 200

5.11.6.1. Request

This table shows the URI parameters for the cancel job execution request:

Name	Type	Description
`{job_execution_id}`	UUID	The UUID of the job execution.

This operation does not accept a request body.

5.11.6.2. Response

Example 5.68. Cancel job execution: JSON response


```
{
    "job_execution": {
        "job_configs": {
            "configs": {
                "mapred.reduce.tasks": "1",
                "mapred.map.tasks": "1"
            },
            "args": [
                "arg1",
                "arg2"
            ],
            "params": {
                "param2": "value2",
                "param1": "value1"
            }
        },
        "is_protected": false,
        "input_id": "3e1bc8e6-8c69-4749-8e52-90d9341d15bc",
        "job_id": "310b0fc6-e1db-408e-8798-312e7500f3ac",
        "cluster_id": "811e1134-666f-4c48-bc92-afb5b10c9d8c",
        "created_at": "2015-09-15T09:49:24",
        "end_time": "2015-09-15T12:50:46",
        "output_id": "52146b52-6540-4aac-a024-fee253cf52a9",
        "is_public": false,
        "updated_at": "2015-09-15T09:50:46",
        "return_code": null,
        "data_source_urls": {
            "3e1bc8e6-8c69-4749-8e52-90d9341d15bc": "swift://ap-cont/input",
            "52146b52-6540-4aac-a024-fee253cf52a9": "swift://ap-cont/output"
        },
        "tenant_id": "808d5032ea0446889097723bfc8e919d",
        "start_time": "2015-09-15T12:49:43",
        "id": "20da9edb-12ce-4b45-a473-41baeefef997",
        "oozie_job_id": "0000001-150915094349962-oozie-hado-W",
        "info": {
            "user": "hadoop",
```

```
            "actions": [
                {
                    "name": ":start:",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": "job-node",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "errorCode": null,
                    "id": "0000001-150915094349962-oozie-hado-W@:start:",
                    "consoleUrl": "-",
                    "errorMessage": null,
                    "toString": "Action name[:start:] status[OK]",
                    "stats": null,
                    "type": ":START:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "job-node",
                    "trackerUri": "http://172.18.168.119:8032",
                    "externalStatus": "FAILED/KILLED",
                    "status": "ERROR",
                    "externalId": "job_1442310173665_0002",
                    "transition": "fail",
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "JA018",
                    "id": "0000001-150915094349962-oozie-hado-W@job-node",
                    "consoleUrl": "http://ap-cluster-all-0:8088/proxy/
application_1442310173665_0002/",
                    "errorMessage": "Main class [org.apache.oozie.action.
hadoop.PigMain], exit code [2]",
                    "toString": "Action name[job-node] status[ERROR]",
                    "stats": null,
                    "type": "pig",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                },
                {
                    "name": "fail",
                    "trackerUri": "-",
                    "externalStatus": "OK",
                    "status": "OK",
                    "externalId": "-",
                    "transition": null,
                    "data": null,
                    "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "errorCode": "E0729",
                    "id": "0000001-150915094349962-oozie-hado-W@fail",
                    "consoleUrl": "-",
                    "errorMessage": "Workflow failed, error message[Main class
 [org.apache.oozie.action.hadoop.PigMain], exit code [2]]",
                    "toString": "Action name[fail] status[OK]",
```

```
                    "stats": null,
                    "type": ":KILL:",
                    "retries": 0,
                    "startTime": "Tue, 15 Sep 2015 09:50:17 GMT",
                    "externalChildIDs": null,
                    "cred": "null"
                }
            ],
            "createdTime": "Tue, 15 Sep 2015 09:49:58 GMT",
            "status": "KILLED",
            "group": null,
            "externalId": null,
            "acl": null,
            "run": 0,
            "appName": "job-wf",
            "parentId": null,
            "conf": "<configuration>\r\n  <property>\r\n    <name>user.
name</name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>
\r\n    <name>oozie.use.system.libpath</name>\r\n    <value>true</value>
\r\n  </property>\r\n  <property>\r\n    <name>mapreduce.job.user.name</
name>\r\n    <value>hadoop</value>\r\n  </property>\r\n  <property>\r\n
  <name>nameNode</name>\r\n    <value>hdfs://ap-cluster-all-0:9000</value>
\r\n  </property>\r\n  <property>\r\n    <name>jobTracker</name>\r\n
 <value>http://172.18.168.119:8032</value>\r\n  </property>\r\n  <property>\r
\n    <name>oozie.wf.application.path</name>\r\n    <value>hdfs://ap-cluster-
all-0:9000/user/hadoop/pig-job-example/3038025d-9974-4993-a778-26a074cdfb8d/
workflow.xml</value>\r\n  </property>\r\n</configuration>",
            "id": "0000001-150915094349962-oozie-hado-W",
            "startTime": "Tue, 15 Sep 2015 09:49:59 GMT",
            "appPath": "hdfs://ap-cluster-all-0:9000/user/hadoop/pig-job-
example/3038025d-9974-4993-a778-26a074cdfb8d/workflow.xml",
            "endTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "toString": "Workflow id[0000001-150915094349962-oozie-hado-W]
 status[KILLED]",
            "lastModTime": "Tue, 15 Sep 2015 09:50:17 GMT",
            "consoleUrl": "http://ap-cluster-all-0.novalocal:11000/oozie?job=
0000001-150915094349962-oozie-hado-W"
        }
    }
}
```


5.12. Job types

Each plugin that supports EDP also supports specific job types. Different versions of a plug-in might actually support different job types. Configuration options vary by plugin, version, and job type.

The job types provide information about which plugins support which job types and how to configure the job types.

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-types`	Lists all job types.

5.12.1. List job types

Method	URI	Description
GET	`/v1.1/{tenant_id}/job-types`	Lists all job types.

You can use query parameters to filter the response.

Normal response codes: 200

5.12.1.1. Request

This operation does not accept a request body.

5.12.1.2. Response

Example 5.69. List job types: JSON response

```
{
    "job_types": [
        {
            "plugins": [
                {
                    "description": "The Apache Vanilla plugin provides
 the ability to launch upstream Vanilla Apache Hadoop cluster without any
 management consoles. It can also deploy the Oozie component.",
                    "versions": {
                        "1.2.1": {},
                        "2.6.0": {}
                    },
                    "title": "Vanilla Apache Hadoop",
                    "name": "vanilla"
                },
                {
                    "description": "The Hortonworks Sahara plugin automates
 the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                    "versions": {
                        "1.3.2": {},
                        "2.0.6": {}
                    },
                    "title": "Hortonworks Data Platform",
                    "name": "hdp"
                },
                {
                    "description": "The Cloudera Sahara plugin provides the
 ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
 Cloudera Manager management console.",
                    "versions": {
                        "5": {},
                        "5.3.0": {}
                    },
                    "title": "Cloudera Plugin",
                    "name": "cdh"
                }
            ],
            "name": "Hive"
        },
        {
```

```
            "plugins": [
                {
                    "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                    "versions": {
                        "1.2.1": {},
                        "2.6.0": {}
                    },
                    "title": "Vanilla Apache Hadoop",
                    "name": "vanilla"
                },
                {
                    "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                    "versions": {
                        "1.3.2": {},
                        "2.0.6": {}
                    },
                    "title": "Hortonworks Data Platform",
                    "name": "hdp"
                },
                {
                    "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                    "versions": {
                        "5": {},
                        "5.3.0": {}
                    },
                    "title": "Cloudera Plugin",
                    "name": "cdh"
                }
            ],
            "name": "Java"
        },
        {
            "plugins": [
                {
                    "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                    "versions": {
                        "1.2.1": {},
                        "2.6.0": {}
                    },
                    "title": "Vanilla Apache Hadoop",
                    "name": "vanilla"
                },
                {
                    "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                    "versions": {
                        "1.3.2": {},
                        "2.0.6": {}
                    },
                    "title": "Hortonworks Data Platform",
                    "name": "hdp"
                },
                {
```

```
                    "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                    "versions": {
                        "5": {},
                        "5.3.0": {}
                    },
                    "title": "Cloudera Plugin",
                    "name": "cdh"
                }
            ],
            "name": "MapReduce"
        },
        {
            "plugins": [
                {
                    "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                    "versions": {
                        "1.2.1": {},
                        "2.6.0": {}
                    },
                    "title": "Vanilla Apache Hadoop",
                    "name": "vanilla"
                },
                {
                    "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                    "versions": {
                        "1.3.2": {},
                        "2.0.6": {}
                    },
                    "title": "Hortonworks Data Platform",
                    "name": "hdp"
                },
                {
                    "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                    "versions": {
                        "5": {},
                        "5.3.0": {}
                    },
                    "title": "Cloudera Plugin",
                    "name": "cdh"
                }
            ],
            "name": "MapReduce.Streaming"
        },
        {
            "plugins": [
                {
                    "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                    "versions": {
                        "1.2.1": {},
                        "2.6.0": {}
                    },
```

```
                    "title": "Vanilla Apache Hadoop",
                    "name": "vanilla"
                },
                {
                    "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                    "versions": {
                        "1.3.2": {},
                        "2.0.6": {}
                    },
                    "title": "Hortonworks Data Platform",
                    "name": "hdp"
                },
                {
                    "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                    "versions": {
                        "5": {},
                        "5.3.0": {}
                    },
                    "title": "Cloudera Plugin",
                    "name": "cdh"
                }
            ],
            "name": "Pig"
        },
        {
            "plugins": [
                {
                    "description": "The Apache Vanilla plugin provides
the ability to launch upstream Vanilla Apache Hadoop cluster without any
management consoles. It can also deploy the Oozie component.",
                    "versions": {
                        "1.2.1": {},
                        "2.6.0": {}
                    },
                    "title": "Vanilla Apache Hadoop",
                    "name": "vanilla"
                },
                {
                    "description": "The Hortonworks Sahara plugin automates
the deployment of the Hortonworks Data Platform (HDP) on OpenStack.",
                    "versions": {
                        "1.3.2": {},
                        "2.0.6": {}
                    },
                    "title": "Hortonworks Data Platform",
                    "name": "hdp"
                },
                {
                    "description": "The Cloudera Sahara plugin provides the
ability to launch the Cloudera distribution of Apache Hadoop (CDH) with
Cloudera Manager management console.",
                    "versions": {
                        "5": {},
                        "5.3.0": {}
                    },
                    "title": "Cloudera Plugin",
                    "name": "cdh"
```

```
                }
            ],
            "name": "Shell"
        },
        {
            "plugins": [
                {
                    "description": "This plugin provides an ability to launch
 Spark on Hadoop CDH cluster without any management consoles.",
                    "versions": {
                        "1.0.0": {}
                    },
                    "title": "Apache Spark",
                    "name": "spark"
                }
            ],
            "name": "Spark"
        }
    ]
}
```

6. Image service API v2 (CURRENT)

Image service API v2.0, API v2.1, API v2.2, and API v2.3.

Cloud providers can configure property protections that prevent non-administrative users from updating and deleting protected properties. For more information, see Image property protection in the *OpenStack Cloud Administrator Guide*.

Method	URI	Description
API versions		
GET	`/`	Lists version information for all Image service API versions.
Images (images)		
GET	`/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag,created_at,updated_at}`	(Since Image API v2.0) Lists public virtual machine (VM) images.
POST	`/v2/images`	(Since Image API v2.0) Creates a virtual machine (VM) image.
GET	`/v2/images/{image_id}`	(Since Image API v2.0) Shows details for an image.
PATCH	`/v2/images/{image_id}`	(Since Image API v2.0) Updates an image.
DELETE	`/v2/images/{image_id}`	(Since Image API v2.0) Deletes an image.
POST	`/v2/images/{image_id}/actions/reactivate`	(Since Image API v2.0) Reactivates an image.
POST	`/v2/images/{image_id}/actions/deactivate`	(Since Image API v2.0) Deactivates an image.
Image data (images, file)		
PUT	`/v2/images/{image_id}/file`	(Since Image API v2.0) Uploads binary image data.
GET	`/v2/images/{image_id}/file`	(Since Image API v2.0) Downloads binary image data.
Image tags (images, tags)		
PUT	`/v2/images/{image_id}/tags/{tag}`	(Since Image API v2.0) Adds a tag to an image.
DELETE	`/v2/images/{image_id}/tags/{tag}`	(Since Image API v2.0) Deletes a tag from an image.
Members (images, members)		
GET	`/v2/images/{image_id}/members`	(Since Image API v2.1) Lists the tenants that share this image.
POST	`/v2/images/{image_id}/members`	(Since Image API v2.1) Adds a tenant ID as an image member.
GET	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.2) Shows image member details.
PUT	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.1) Sets the status for an image member.
DELETE	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.
Image schemas (schemas, image, images, member, members)		
GET	`/v2/schemas/images`	(Since Images v2.0) Shows a JSON schema document that represents an images entity.
GET	`/v2/schemas/image`	(Since Images v2.0) Shows a JSON schema document that represents an image entity.
GET	`/v2/schemas/members`	(Since Images v2.1) Shows a JSON schema document that represents an image members entity.

Method	URI	Description
GET	`/v2/schemas/member`	(Since Images v2.1) Shows a JSON schema document that represents an image member entity.
Metadata definition resource types (since API v2.0) (metadefs, namespaces, resource_types)		
GET	`/v2/metadefs/resource_types`	Lists all resource types.
POST	`/v2/metadefs/namespaces/{namespace_id}/resource_types`	Creates a resource type association in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/resource_types`	Lists resource type associations in a namespace.
DELETE	`/v2/metadefs/namespaces/{namespace_id}/resource_types/{name}`	Removes a resource type association in a namespace.
Metadata definition namespaces (since API v2.0) (metadefs, namespaces)		
POST	`/v2/metadefs/namespaces`	Creates a namespace.
GET	`/v2/metadefs/namespaces{?limit,marker,visibility,resource_types,sort_key,sort_dir}`	Lists public namespaces.
GET	`/v2/metadefs/namespaces/{namespace_id}`	Gets details for a namespace.
PUT	`/v2/metadefs/namespaces/{namespace_id}`	Updates a namespace.
DELETE	`/v2/metadefs/namespaces/{namespace_id}`	Deletes a namespace and its properties, objects, and any resource type associations.
Metadata definition properties (since API v2.0) (metadefs, namespaces, properties)		
POST	`/v2/metadefs/namespaces/{namespace_id}/properties`	Creates a property definition in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/properties`	Lists property definitions in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}{?resource_type}`	Shows the definition for a property.
PUT	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}`	Updates a property definition.
DELETE	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}`	Removes a property definition in a namespace.
Metadata definition objects (since API v2.0) (metadefs, namespaces, objects)		
POST	`/v2/metadefs/namespaces/{namespace_id}/objects`	Creates an object definition in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/objects{?visibility,resource_types,sort_key,sort_dir}`	Lists object definitions in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Shows the definition for an object.
PUT	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Updates an object definition in a namespace.

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Deletes an object definition from a namespace.
Metadata definition tags (since API v2.0) (metadefs, namespaces, tags)		
POST	`/v2/metadefs/namespaces/tags/{namespace_id}`	Creates one or more tag definitions in a namespace.
GET	`/v2/metadefs/namespaces/tags/{namespace_id}{?limit,marker,sort_key,sort_dir}`	Lists the tag definitions within a namespace.
DELETE	`/v2/metadefs/namespaces/tags/{namespace_id}`	Deletes all tag definitions within a namespace.
POST	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Adds a tag to the list of namespace tag definitions.
GET	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Gets a definition for a tag.
PUT	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Renames a tag definition.
DELETE	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Deletes a tag definition within a namespace.
Metadata definition schemas (schemas, metadefs, namespace, namespaces, object, objects, property, properties, tag, tags, resource_type, resource_types)		
GET	`/v2/schemas/metadefs/namespace`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace entity.
GET	`/v2/schemas/metadefs/namespaces`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespaces entity.
GET	`/v2/schemas/metadefs/object`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition object entity.
GET	`/v2/schemas/metadefs/objects`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition objects entity.
GET	`/v2/schemas/metadefs/property`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition property entity.
GET	`/v2/schemas/metadefs/properties`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition properties entity.
GET	`/v2/schemas/metadefs/tag`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tag entity.
GET	`/v2/schemas/metadefs/tags`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tags entity.
GET	`/v2/schemas/metadefs/resource_type`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type association entity.
GET	`/v2/schemas/metadefs/resource_types`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type associations entity.
Tasks (since API v2.2) (tasks)		
POST	`/v2/tasks`	Creates a task.
GET	`/v2/tasks{?sort_dir,sort_key,type,status}`	Lists tasks.
GET	`/v2/tasks/{task_id}`	Shows details for a task.

6.1. API versions

Method	URI	Description
GET	/	Lists version information for all Image service API versions.

6.1.1. List API versions

Method	URI	Description
GET	/	Lists version information for all Image service API versions.

Normal response codes: 300

6.1.1.1. Request

This operation does not accept a request body.

6.2. Images (images)

Creates, lists, updates, and deletes images.

Image operations show all fields in the response body. Any field with no value is set to `null` value (JSON null data type).

The possible status values for images are:

Table 6.1. Image status

Status	Description
queued	The Image service reserved an image ID for the image in the registry but did not yet upload any image data.
saving	The Image service is currently uploading the raw data for the image.
active	The image is active and fully available in the Image service.
killed	An image data upload error occurred.
deleted	The Image service retains information about the image but the image is no longer available for use.
pending_delete	Similar to the `deleted` status. An image in this state is not recoverable.

Method	URI	Description
GET	`/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag,created_at,updated_at}`	(Since Image API v2.0) Lists public virtual machine (VM) images.
POST	`/v2/images`	(Since Image API v2.0) Creates a virtual machine (VM) image.
GET	`/v2/images/{image_id}`	(Since Image API v2.0) Shows details for an image.
PATCH	`/v2/images/{image_id}`	(Since Image API v2.0) Updates an image.
DELETE	`/v2/images/{image_id}`	(Since Image API v2.0) Deletes an image.
POST	`/v2/images/{image_id}/actions/reactivate`	(Since Image API v2.0) Reactivates an image.
POST	`/v2/images/{image_id}/actions/deactivate`	(Since Image API v2.0) Deactivates an image.

6.2.1. List images

Method	URI	Description
GET	`/v2/images{?limit,marker,name,visibility,member_status,owner,status,size_min,size_max,sort_key,sort_dir,sort,tag,created_at,updated_at}`	(Since Image API v2.0) Lists public virtual machine (VM) images.

Returns a subset of the larger collection of images and a link that you can use to get the next set of images. You should always check for the presence of a `next` link and use it as the URI in a subsequent HTTP GET request. You should follow this pattern until a `next` link is no longer provided. The next link preserves any query parameters that you send in your initial request. You can use the `first` link to jump back to the first page of the collection. If you prefer to paginate through images manually, use the `limit` and `marker` parameters.

The list operation accepts query parameters to filter the response.

A client can provide direct comparison filters by using most image attributes, such as `name=Ubuntu`, `visibility=public`, and so on. A client cannot use tags or any `link` in the json-schema, such as self, file, or schema, to filter the response.

You can use the `size_min` and `size_max` query parameters to filter images that are greater than or less than the image size. The size, in bytes, is the size of an image on disk.

For example, to filter the container to include only images that are from 1 to 4 MB, set the `size_min` query parameter to `1048576` and the `size_max` query parameter to `4194304`.

You can list VM images that have a status of `active`, `queued`, or `saving`.

You can use query parameters to sort the results of this operation.

- `sort_key`. Sorts by an image attribute. Sorts in the natural sorting direction of the image attribute.
- `sort_dir`. Sorts in a sort direction.
- `sort`. Sorts by one or more sets of attribute and sort direction combinations. If you omit the sort direction in a set, the default is `desc`.

To sort the response, use the `sort_key` and `sort_dir` query parameters:

```
GET /v2/images?sort_key=name&sort_dir=asc&sort_key=status&sort_dir=desc
```

Alternatively, specify the `sort` query parameter:

```
GET /v2/images?sort=name:asc,status:desc
```

Normal response codes: 200

6.2.1.1. Request

This table shows the query parameters for the list images request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`name`	String *(Optional)*	Filters the response by a name, as a string. A valid value is the name of an image.
`visibility`	String *(Optional)*	Filters the response by an image visibility value or values. Use the comma (`,`) character to separate multiple values. A valid value is `public`, `private`, or `shared`. If you omit this parameter, the response shows `public`, `private`, and `shared` images with a member status of `accepted`.
`member_status`	String *(Optional)*	Filters the response by a member status. A valid value is `accepted`, `pending`, `rejected`, or `all`. Default is `accepted`.
`owner`	String *(Optional)*	Filters the response by a tenant ID. Shows only images that are shared with this tenant owner.
`status`	Int *(Optional)*	Filters the response by an image status. A valid value is `queued`, `saving`, `active`, `killed`, `deleted`, or `pending_delete`.
`size_min`	String *(Optional)*	Filters the response by a minimum image size, in bytes.
`size_max`	String *(Optional)*	Filters the response by a maximum image size, in bytes.
`sort_key`	String *(Optional)*	Sorts the response by an attribute, such as `name`, `id`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` image attribute.
`sort_dir`	String *(Optional)*	Sorts the response by a set of one or more sort direction and attribute (`sort_key`) combinations. A valid value for the sort direction is `asc` (ascending) or `desc` (descending). If you omit the sort direction in a set, the default is `desc`.
`sort`	String *(Optional)*	Sorts the response by one or more attribute and sort direction combinations. You can also set multiple sort keys and directions. Default direction is `desc`. Use the comma (`,`) character to separate multiple values. For example: `GET /v2/images?sort=name:asc,status:desc`
`tag`	String *(Optional)*	Filters the response by an image tag. For example, `?tag="cirros"`.
`created_at`	DateTime *(Optional)*	Filters the response by the date and time when the resource was created. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm`

Name	Type	Description
		The ±hh:mm value, if included, returns the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the created_at date and time stamp, its value is null.
updated_at	DateTime *(Optional)*	Filters the response by the date and time when the resource was updated. The date and time stamp format is ISO 8601: CCYY-MM-DDThh:mm:ss±hh:mm The ±hh:mm value, if included, is the time zone as an offset from UTC. For example, 2015-08-27T09:49:58-05:00. If you omit the time zone, the UTC time zone is assumed. If you omit the updated_at date and time stamp, its value is null.

This operation does not accept a request body.

6.2.1.2. Response

Example 6.1. List images: JSON response

```
{
    "images": [
        {
            "status": "active",
            "name": "cirros-0.3.2-x86_64-disk",
            "tags": [],
            "container_format": "bare",
            "created_at": "2014-11-07T17:07:06Z",
            "disk_format": "qcow2",
            "updated_at": "2014-11-07T17:19:09Z",
            "visibility": "public",
            "self": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27",
            "min_disk": 0,
            "protected": false,
            "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
            "file": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27/file",
            "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
            "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
            "size": 13167616,
            "min_ram": 0,
            "schema": "/v2/schemas/image",
            "virtual_size": null
        },
        {
            "status": "active",
            "name": "F17-x86_64-cfntools",
            "tags": [],
            "container_format": "bare",
            "created_at": "2014-10-30T08:23:39Z",
            "disk_format": "qcow2",
            "updated_at": "2014-11-03T16:40:10Z",
            "visibility": "public",
            "self": "/v2/images/781b3762-9469-4cec-b58d-3349e5de4e9c",
            "min_disk": 0,
```

```
            "protected": false,
            "id": "781b3762-9469-4cec-b58d-3349e5de4e9c",
            "file": "/v2/images/781b3762-9469-4cec-b58d-3349e5de4e9c/file",
            "checksum": "afab0f79bac770d61d24b4d0560b5f70",
            "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
            "size": 476704768,
            "min_ram": 0,
            "schema": "/v2/schemas/image",
            "virtual_size": null
        }
    ],
    "schema": "/v2/schemas/images",
    "first": "/v2/images"
}
```

6.2.2. Create image

Method	URI	Description
POST	`/v2/images`	(Since Image API v2.0) Creates a virtual machine (VM) image.

The `Location` response header contains the URI for the image. The response body contains the new image entity.

Synchronous Postconditions

- With correct permissions, you can see the image status as `queued` through API calls.

Normal response codes: 201

6.2.2.1. Request

Example 6.2. Create image: JSON request

```
{
    "container_format": "bare",
    "disk_format": "raw",
    "name": "Ubuntu",
    "id": "b2173dd3-7ad6-4362-baa6-a68bce3565cb"
}
```

6.2.2.2. Response

Example 6.3. Create image: JSON response

```
{
    "status": "queued",
    "name": "Ubuntu",
    "tags": [],
    "container_format": "bare",
    "created_at": "2015-11-29T22:21:42Z",
    "size": null,
    "disk_format": "raw",
    "updated_at": "2015-11-29T22:21:42Z",
    "visibility": "private",
    "locations": [],
    "self": "/v2/images/b2173dd3-7ad6-4362-baa6-a68bce3565cb",
    "min_disk": 0,
    "protected": false,
    "id": "b2173dd3-7ad6-4362-baa6-a68bce3565cb",
    "file": "/v2/images/b2173dd3-7ad6-4362-baa6-a68bce3565cb/file",
    "checksum": null,
    "owner": "bab7d5c60cd041a0a36f7c4b6e1dd978",
    "virtual_size": null,
    "min_ram": 0,
    "schema": "/v2/schemas/image"
}
```

6.2.3. Show image details

Method	URI	Description
GET	`/v2/images/{image_id}`	(Since Image API v2.0) Shows details for an image.

The response body contains a single image entity.

Preconditions

- The image must exist.

Normal response codes: 200

Error response codes: itemNotFound (404)

6.2.3.1. Request

This table shows the URI parameters for the show image details request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.2.3.2. Response

Example 6.4. Show image details: JSON response

```
{
    "status": "active",
    "name": "cirros-0.3.2-x86_64-disk",
    "tags": [],
    "container_format": "bare",
    "created_at": "2014-05-05T17:15:10Z",
    "disk_format": "qcow2",
    "updated_at": "2014-05-05T17:15:11Z",
    "visibility": "public",
    "self": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27",
    "min_disk": 0,
    "protected": false,
    "id": "1bea47ed-f6a9-463b-b423-14b9cca9ad27",
    "file": "/v2/images/1bea47ed-f6a9-463b-b423-14b9cca9ad27/file",
    "checksum": "64d7c1cd2b6f60c92c14662941cb7913",
    "owner": "5ef70662f8b34079a6eddb8da9d75fe8",
    "size": 13167616,
    "min_ram": 0,
    "schema": "/v2/schemas/image",
    "virtual_size": null
}
```

6.2.4. Update image

Method	URI	Description
PATCH	`/v2/images/{image_id}`	(Since Image API v2.0) Updates an image.

Depending on the referenced target location, this operation performs one of these actions:

Table 6.2. Image update actions

Target location	Update action
An array index	The API inserts a new value into the array at the index.
An object member that does not exist	The API adds a member to the object.
An object member that exists	The member value is replaced.

The operation object must contain a `value` member that contains the value to add. For example:

```
{
    "op": "add",
    "path": "/a/b/c",
    "value": [
        "foo",
        "bar"
    ]
}
```

The target location must reference one of these values:

- The root of the target document. The value is the entire content of the target document.
- A member value to add to an object. The API adds the value to the object at the location. If the member already exists, the API replaces it with the value.
- An element to add to the array. The API adds the element value to the array at the location. The API shifts any element that is at or above the index one position to the right. The index must not be greater than the number of elements in the array. If you use the hyphen (-) character to index the end of the array, the API appends the value to the array. See JavaScript Object Notation (JSON) Pointer.

Because this operation adds to existing objects and arrays, its target location often does not exist.

The request body must conform to one of these media types:

- `application/openstack-images-v2.0-json-patch`
- `application/openstack-images-v2.1-json-patch` (since Image API v2.2)

You can also use the **PATCH** method to add or remove image properties.

For information about the **PATCH** method and the available media types, see Image API v2 HTTP PATCH media types.

Preconditions

- When you add a location to or replace a location in the image, you must set the `disk_format` and `container_format` parameters in the image.
- When you replace a location, that location must be previously set in the image.

Synchronous Postconditions

- With correct permissions, you can view the updated values of the attributes of the image.
- After you add a location to an image that had no location and with correct permissions, you can use API calls to view the image status as `active`.
- After you remove all locations from the image and with correct permissions, you can use API calls to view the image status as `queued`.

Troubleshooting

- If you cannot update locations, your request might be missing some information. Make sure that you meet the preconditions and run the request again. If the request fails again, review your API request.

Normal response codes: 200

Error response codes: itemNotFound (404)

6.2.4.1. Request

This table shows the URI parameters for the update image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

Example 6.5. Update image: JSON request

```
[
    {
        "op": "replace",
        "path": "/name",
        "value": "Fedora 17"
    },
    {
        "op": "replace",
        "path": "/tags",
        "value": [
            "fedora",
            "beefy"
        ]
    }
]
```

6.2.4.2. Response

Example 6.6. Update image: JSON response

```
{
```

```
    "id": "da3b75d9-3f4a-40e7-8a2c-bfab23927dea",
    "name": "Fedora 17",
    "status": "active",
    "visibility": "public",
    "size": 2254249,
    "checksum": "2cec138d7dae2aa59038ef8c9aec2390",
    "tags": [
        "fedora",
        "beefy"
    ],
    "created_at": "2012-08-10T19:23:50Z",
    "updated_at": "2012-08-12T11:11:33Z",
    "self": "/v2/images/da3b75d9-3f4a-40e7-8a2c-bfab23927dea",
    "file": "/v2/images/da3b75d9-3f4a-40e7-8a2c-bfab23927dea/file",
    "schema": "/v2/schemas/image",
    "owner": null,
    "min_ram": null,
    "min_disk": null,
    "disk_format": null,
    "virtual_size": null,
    "container_format": null
}
```

6.2.5. Delete image

Method	URI	Description
DELETE	`/v2/images/{image_id}`	(Since Image API v2.0) Deletes an image.

You cannot delete images with the `protected` attribute set to `true` (boolean).

Preconditions

- You can delete an image in any status except `deleted`.
- First, set the `protected` attribute to `false` (boolean). Then, perform the delete.

Synchronous Postconditions

- The response is empty and returns the HTTP `204` response code.
- The API deletes the image from the images index.
- If the image stores binary image data in the storage node, the OpenStack Image service deletes the data from the node.

Troubleshooting

- The call returns the HTTP `403` response code when the `protected` attribute is set to `true` even if you have correct permissions. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request.

Normal response codes: 204

Error response codes: forbidden (403), itemNotFound (404)

6.2.5.1. Request

This table shows the URI parameters for the delete image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.2.6. Reactivate image

Method	URI	Description
POST	`/v2/images/{image_id}/actions/reactivate`	(Since Image API v2.0) Reactivates an image.

The reactivate operation returns an error if the image status is not `active` or `deactivated`.

Preconditions

- The image must exist.

Normal response codes: 204

Error response codes: itemNotFound (404)

6.2.6.1. Request

This table shows the URI parameters for the reactivate image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.2.7. Deactivate image

Method	URI	Description
POST	`/v2/images/{image_id}/actions/deactivate`	(Since Image API v2.0) Deactivates an image.

If you try to download a deactivated image, the call returns the `Forbidden (403)` response code. Also, only administrative users can view image locations for deactivated images.

The deactivate operation returns an error if the image status is not `active` or `deactivated`.

Preconditions

- The image must exist.

Normal response codes: 204

Error response codes: forbidden (403), itemNotFound (404)

6.2.7.1. Request

This table shows the URI parameters for the deactivate image request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.3. Image data (images, file)

Uploads and downloads raw image data.

Method	URI	Description
PUT	`/v2/images/{image_id}/file`	(Since Image API v2.0) Uploads binary image data.
GET	`/v2/images/{image_id}/file`	(Since Image API v2.0) Downloads binary image data.

6.3.1. Upload binary image data

Method	URI	Description
PUT	`/v2/images/{image_id}/file`	(Since Image API v2.0) Uploads binary image data.

Set the `Content-Type` request header to `application/octet-stream`.

Example call:

```
curl -i -X PUT -H "X-Auth-Token: $token" -H "Content-Type: application/octet-
stream" \
   -d @/home/glance/ubuntu-12.10.qcow2 $image_url/v2/images/{image_id}/file
```

Preconditions

Before you can store binary image data, you must meet the following preconditions:

- The image must exist.
- You must set the disk and container formats in the image.
- The image status must be `queued`.
- Your image storage quota must be sufficient.
- The size of the data that you want to store must not exceed the size that the OpenStack Image service allows.

Synchronous Postconditions

- With correct permissions, you can see the image status as `active` through API calls.
- With correct access, you can see the stored data in the storage system that OpenStack Image service manages.

Troubleshooting

- If you cannot store the data, either your request lacks required information or you exceeded your allotted quota. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request.
- The storage back ends for storing the data must have enough free storage space to accommodate the size of the data.

Normal response codes: 204

6.3.1.1. Request

This table shows the URI parameters for the upload binary image data request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.3.2. Download binary image data

Method	URI	Description
GET	`/v2/images/{image_id}/file`	(Since Image API v2.0) Downloads binary image data.

Example call: `curl -i -X GET -H "X-Auth-Token: $token" $image_url/v2/images/{image_id}/file`

The response body contains the raw binary data that represents the actual virtual disk. The `Content-Type` header contains the `application/octet-stream` value. The `Content-MD5` header contains an MD5 checksum of the image data. Use this checksum to verify the integrity of the image data.

Preconditions

- The images must exist.

Synchronous Postconditions

- You can download the binary image data in your machine if the image has image data.
- If image data exists, the call returns the HTTP `200` response code.
- If no image data exists, the call returns the HTTP `204` response code.

Normal response codes: 200204

Error response codes: forbidden (403), itemNotFound (404)

6.3.2.1. Request

This table shows the header parameters for the download binary image data request:

Name	Type	Description
`Content-Range`	String *(Optional)*	The content range of image data. For details, see Hypertext Transfer Protocol (HTTP/1.1): Range Requests.

This table shows the URI parameters for the download binary image data request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.4. Image tags (images, tags)

Adds and deletes image tags.

Method	URI	Description
PUT	`/v2/images/{image_id}/tags/{tag}`	(Since Image API v2.0) Adds a tag to an image.

Method	URI	Description
DELETE	`/v2/images/{image_id}/tags/{tag}`	(Since Image API v2.0) Deletes a tag from an image.

6.4.1. Add image tag

Method	URI	Description
PUT	`/v2/images/{image_id}/tags/{tag}`	(Since Image API v2.0) Adds a tag to an image.

Normal response codes: 204

Error response codes: itemNotFound (404)

6.4.1.1. Request

This table shows the URI parameters for the add image tag request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.
`{tag}`	String	The image tag.

This operation does not accept a request body.

6.4.2. Delete image tag

Method	URI	Description
DELETE	`/v2/images/{image_id}/tags/{tag}`	(Since Image API v2.0) Deletes a tag from an image.

Normal response codes: 204

Error response codes: itemNotFound (404)

6.4.2.1. Request

This table shows the URI parameters for the delete image tag request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.
`{tag}`	String	The image tag.

This operation does not accept a request body.

6.5. Members (images, members)

Creates, lists, updates, and deletes image members.

Method	URI	Description
GET	`/v2/images/{image_id}/members`	(Since Image API v2.1) Lists the tenants that share this image.
POST	`/v2/images/{image_id}/members`	(Since Image API v2.1) Adds a tenant ID as an image member.
GET	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.2) Shows image member details.
PUT	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.1) Sets the status for an image member.
DELETE	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.

6.5.1. List image members

Method	URI	Description
GET	`/v2/images/{image_id}/members`	(Since Image API v2.1) Lists the tenants that share this image.

If a user who shares this image makes this call, the member list contains only information for that user.

If a user who does not share this image makes this call, the call returns the HTTP `404` response code.

Preconditions

- The image must exist.
- You must be the owner or a member of the image.

Normal response codes: 200

Error response codes: itemNotFound (404)

6.5.1.1. Request

This table shows the URI parameters for the list image members request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

This operation does not accept a request body.

6.5.1.2. Response

Example 6.7. List image members: JSON response

```
{
    "members": [
        {
            "created_at": "2013-10-07T17:58:03Z",
            "image_id": "dbc999e3-c52f-4200-bedd-3b18fe7f87fe",
            "member_id": "123456789",
            "schema": "/v2/schemas/member",
            "status": "pending",
            "updated_at": "2013-10-07T17:58:03Z"
        },
        {
            "created_at": "2013-10-07T17:58:55Z",
            "image_id": "dbc999e3-c52f-4200-bedd-3b18fe7f87fe",
            "member_id": "987654321",
            "schema": "/v2/schemas/member",
            "status": "accepted",
            "updated_at": "2013-10-08T12:08:55Z"
        }
    ],
```

```
    "schema": "/v2/schemas/members"
}
```

6.5.2. Create image member

Method	URI	Description
POST	`/v2/images/{image_id}/members`	(Since Image API v2.1) Adds a tenant ID as an image member.

This call accepts either the `member_id` or `member` attribute in the request body. If you specify both attributes, the API uses the `member_id` value and ignores the `member` value. Use of the `member` attribute is supported but deprecated. Use of the `member_id` attribute is preferred.

Preconditions

- The images must exist.
- You can add a member to a private image.
- You must be the owner of the image.

Synchronous Postconditions

- With correct permissions, you can see the member status of the image as `pending` through API calls.

Troubleshooting

- Even if you have correct permissions, if the `visibility` attribute is set to `public`, the request returns the HTTP `403` response code. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request.
- If the member is already a member for the image, the service returns the `Conflict (409)` response code. If you meant to specify a different member, run the request again.

Normal response codes: 200

Error response codes: invalidVisibility (403), memberConflict (409)

6.5.2.1. Request

This table shows the URI parameters for the create image member request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.

Example 6.8. Create image member: JSON request

```
{
    "member": "8989447062e04a818baf9e073fd04fa7"
}
```

6.5.2.2. Response

Example 6.9. Create image member: JSON response

```
{
    "created_at": "2013-09-20T19:22:19Z",
    "image_id": "a96be11e-8536-4910-92cb-de50aa19dfe6",
    "member_id": "8989447062e04a818baf9e073fd04fa7",
    "schema": "/v2/schemas/member",
    "status": "pending",
    "updated_at": "2013-09-20T19:25:31Z"
}
```

6.5.3. Show image member details

Method	URI	Description
GET	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.2) Shows image member details.

Response body is a single image member entity.

Preconditions

- The image must exist.
- You must be the owner or a member of the image.

Normal response codes: 200

6.5.3.1. Request

This table shows the URI parameters for the show image member details request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.
`{member_id}`	String	The ID of the image member. An image member is a tenant with whom the image is shared.

This operation does not accept a request body.

6.5.3.2. Response

Example 6.10. Show image member details: JSON response

```
{
    "status": "pending",
    "created_at": "2013-11-26T07:21:21Z",
    "updated_at": "2013-11-26T07:21:21Z",
    "image_id": "0ae74cc5-5147-4239-9ce2-b0c580f7067e",
    "member_id": "8989447062e04a818baf9e073fd04fa7",
    "schema": "/v2/schemas/member"
}
```

6.5.4. Update image member

Method	URI	Description
PUT	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.1) Sets the status for an image member.

Preconditions

- The images must exist.
- You must be a member of the image.

Synchronous Postconditions

- If you update the member status to `accepted` and have the correct permissions, you see the image in list images responses.
- With correct permissions, you can make API calls to see the updated member status of the image.

Normal response codes: 200

6.5.4.1. Request

This table shows the URI parameters for the update image member request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.
`{member_id}`	String	The ID of the image member. An image member is a tenant with whom the image is shared.

Example 6.11. Update image member: JSON request

```
{
    "status": "accepted"
}
```

6.5.4.2. Response

Example 6.12. Update image member: JSON response

```
{
    "created_at": "2013-09-20T19:22:19Z",
    "image_id": "a96be11e-8536-4910-92cb-de50aa19dfe6",
    "member_id": "8989447062e04a818baf9e073fd04fa7",
    "schema": "/v2/schemas/member",
    "status": "accepted",
    "updated_at": "2013-09-20T20:15:31Z"
}
```

6.5.5. Delete image member

Method	URI	Description
DELETE	`/v2/images/{image_id}/members/{member_id}`	(Since Image API v2.1) Deletes a tenant ID from the member list of an image.

Preconditions

- The image must exist.
- You must be the owner of the image.

Synchronous Postconditions

- The API removes the member from the image members.

Troubleshooting

- Even if you have correct permissions, if you are not the owner of the image or you specify an incorrect image ID or member ID, the call returns the HTTP `403` or `404` response code. Ensure that you meet the preconditions and run the request again. If the request fails again, review your API request URI.

Normal response codes: 204

Error response codes: forbidden (403), itemNotFound (404)

6.5.5.1. Request

This table shows the URI parameters for the delete image member request:

Name	Type	Description
`{image_id}`	UUID	The UUID of the image.
`{member_id}`	String	The ID of the image member. An image member is a tenant with whom the image is shared.

This operation does not accept a request body.

6.6. Image schemas (schemas, image, images, member, members)

Gets a JSON-schema document that represents an images or image entity.

Method	URI	Description
GET	`/v2/schemas/images`	(Since Images v2.0) Shows a JSON schema document that represents an images entity.
GET	`/v2/schemas/image`	(Since Images v2.0) Shows a JSON schema document that represents an image entity.
GET	`/v2/schemas/members`	(Since Images v2.1) Shows a JSON schema document that represents an image members entity.

Method	URI	Description
GET	`/v2/schemas/member`	(Since Images v2.1) Shows a JSON schema document that represents an image member entity.

6.6.1. Show images schema

Method	URI	Description
GET	`/v2/schemas/images`	(Since Images v2.0) Shows a JSON schema document that represents an images entity.

An images entity is a container of image entities.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

6.6.1.1. Request

This operation does not accept a request body.

6.6.1.2. Response

Example 6.13. Show images schema: JSON response

```
{
    "links": [
        {
            "href": "{first}",
            "rel": "first"
        },
        {
            "href": "{next}",
            "rel": "next"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "name": "images",
    "properties": {
        "first": {
            "type": "string"
        },
        "images": {
            "items": {
                "additionalProperties": {
                    "type": "string"
                },
                "links": [
                    {
                        "href": "{self}",
                        "rel": "self"
                    },
                    {
                        "href": "{file}",
                        "rel": "enclosure"
                    },
                    {
                        "href": "{schema}",
```

```
                        "rel": "describedby"
                    }
                ],
                "name": "image",
                "properties": {
                    "architecture": {
                        "description": "Operating system architecture as
 specified in http://docs.openstack.org/trunk/openstack-compute/admin/content/
adding-images.html",
                        "is_base": false,
                        "type": "string"
                    },
                    "checksum": {
                        "description": "md5 hash of image contents. (READ-
ONLY)",
                        "maxLength": 32,
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "container_format": {
                        "description": "Format of the container",
                        "enum": [
                            null,
                            "ami",
                            "ari",
                            "aki",
                            "bare",
                            "ovf",
                            "ova"
                        ],
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "created_at": {
                        "description": "Date and time of image registration
 (READ-ONLY)",
                        "type": "string"
                    },
                    "direct_url": {
                        "description": "URL to access the image file kept in
 external store (READ-ONLY)",
                        "type": "string"
                    },
                    "disk_format": {
                        "description": "Format of the disk",
                        "enum": [
                            null,
                            "ami",
                            "ari",
                            "aki",
                            "vhd",
                            "vmdk",
                            "raw",
                            "qcow2",
                            "vdi",
                            "iso"
```

```
                        ],
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "file": {
                        "description": "(READ-ONLY)",
                        "type": "string"
                    },
                    "id": {
                        "description": "An identifier for the image",
                        "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-
fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
                        "type": "string"
                    },
                    "instance_uuid": {
                        "description": "ID of instance used to create this
 image.",
                        "is_base": false,
                        "type": "string"
                    },
                    "kernel_id": {
                        "description": "ID of image stored in Glance that
 should be used as the kernel when booting an AMI-style image.",
                        "is_base": false,
                        "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-
fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
                        "type": "string"
                    },
                    "locations": {
                        "description": "A set of URLs to access the image file
 kept in external store",
                        "items": {
                            "properties": {
                                "metadata": {
                                    "type": "object"
                                },
                                "url": {
                                    "maxLength": 255,
                                    "type": "string"
                                }
                            },
                            "required": [
                                "url",
                                "metadata"
                            ],
                            "type": "object"
                        },
                        "type": "array"
                    },
                    "min_disk": {
                        "description": "Amount of disk space (in GB) required
 to boot image.",
                        "type": "integer"
                    },
                    "min_ram": {
                        "description": "Amount of ram (in MB) required to boot
 image.",
                        "type": "integer"
```

```
                    },
                    "name": {
                        "description": "Descriptive name for the image",
                        "maxLength": 255,
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "os_distro": {
                        "description": "Common name of operating system
 distribution as specified in http://docs.openstack.org/trunk/openstack-
compute/admin/content/adding-images.html",
                        "is_base": false,
                        "type": "string"
                    },
                    "os_version": {
                        "description": "Operating system version as specified
 by the distributor",
                        "is_base": false,
                        "type": "string"
                    },
                    "owner": {
                        "description": "Owner of the image",
                        "maxLength": 255,
                        "type": [
                            "null",
                            "string"
                        ]
                    },
                    "protected": {
                        "description": "If true, image will not be deletable.
",
                        "type": "boolean"
                    },
                    "ramdisk_id": {
                        "description": "ID of image stored in Glance that
 should be used as the ramdisk when booting an AMI-style image.",
                        "is_base": false,
                        "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-
fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
                        "type": "string"
                    },
                    "schema": {
                        "description": "(READ-ONLY)",
                        "type": "string"
                    },
                    "self": {
                        "description": "(READ-ONLY)",
                        "type": "string"
                    },
                    "size": {
                        "description": "Size of image file in bytes (READ-
ONLY)",
                        "type": [
                            "null",
                            "integer"
                        ]
                    },
                    "status": {
```

```
                        "description": "Status of the image (READ-ONLY)",
                        "enum": [
                            "queued",
                            "saving",
                            "active",
                            "killed",
                            "deleted",
                            "pending_delete"
                        ],
                        "type": "string"
                    },
                    "tags": {
                        "description": "List of strings related to the image",
                        "items": {
                            "maxLength": 255,
                            "type": "string"
                        },
                        "type": "array"
                    },
                    "updated_at": {
                        "description": "Date and time of the last image
 modification (READ-ONLY)",
                        "type": "string"
                    },
                    "virtual_size": {
                        "description": "Virtual size of image in bytes (READ-
ONLY)",
                        "type": [
                            "null",
                            "integer"
                        ]
                    },
                    "visibility": {
                        "description": "Scope of image accessibility",
                        "enum": [
                            "public",
                            "private"
                        ],
                        "type": "string"
                    }
                }
            },
            "type": "array"
        },
        "next": {
            "type": "string"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

6.6.2. Show image schema

Method	URI	Description
GET	`/v2/schemas/image`	(Since Images v2.0) Shows a JSON schema document that represents an image entity.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

6.6.2.1. Request

This operation does not accept a request body.

6.6.2.2. Response

Example 6.14. Show image schema: JSON response

```
{
    "additionalProperties": {
        "type": "string"
    },
    "links": [
        {
            "href": "{self}",
            "rel": "self"
        },
        {
            "href": "{file}",
            "rel": "enclosure"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "name": "image",
    "properties": {
        "architecture": {
            "description": "Operating system architecture as specified in
 http://docs.openstack.org/trunk/openstack-compute/admin/content/adding-
images.html",
            "is_base": false,
            "type": "string"
        },
        "checksum": {
            "description": "md5 hash of image contents. (READ-ONLY)",
            "maxLength": 32,
            "type": [
                "null",
                "string"
            ]
        },
        "container_format": {
            "description": "Format of the container",
            "enum": [
```

```
                null,
                "ami",
                "ari",
                "aki",
                "bare",
                "ovf",
                "ova"
            ],
            "type": [
                "null",
                "string"
            ]
        },
        "created_at": {
            "description": "Date and time of image registration (READ-ONLY)",
            "type": "string"
        },
        "direct_url": {
            "description": "URL to access the image file kept in external
 store (READ-ONLY)",
            "type": "string"
        },
        "disk_format": {
            "description": "Format of the disk",
            "enum": [
                null,
                "ami",
                "ari",
                "aki",
                "vhd",
                "vmdk",
                "raw",
                "qcow2",
                "vdi",
                "iso"
            ],
            "type": [
                "null",
                "string"
            ]
        },
        "file": {
            "description": "(READ-ONLY)",
            "type": "string"
        },
        "id": {
            "description": "An identifier for the image",
            "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
            "type": "string"
        },
        "instance_uuid": {
            "description": "ID of instance used to create this image.",
            "is_base": false,
            "type": "string"
        },
        "kernel_id": {
            "description": "ID of image stored in Glance that should be used
 as the kernel when booting an AMI-style image.",
            "is_base": false,
```

```
            "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
            "type": "string"
        },
        "locations": {
            "description": "A set of URLs to access the image file kept in
 external store",
            "items": {
                "properties": {
                    "metadata": {
                        "type": "object"
                    },
                    "url": {
                        "maxLength": 255,
                        "type": "string"
                    }
                },
                "required": [
                    "url",
                    "metadata"
                ],
                "type": "object"
            },
            "type": "array"
        },
        "min_disk": {
            "description": "Amount of disk space (in GB) required to boot
 image.",
            "type": "integer"
        },
        "min_ram": {
            "description": "Amount of ram (in MB) required to boot image.",
            "type": "integer"
        },
        "name": {
            "description": "Descriptive name for the image",
            "maxLength": 255,
            "type": [
                "null",
                "string"
            ]
        },
        "os_distro": {
            "description": "Common name of operating system distribution as
 specified in http://docs.openstack.org/trunk/openstack-compute/admin/content/
adding-images.html",
            "is_base": false,
            "type": "string"
        },
        "os_version": {
            "description": "Operating system version as specified by the
 distributor",
            "is_base": false,
            "type": "string"
        },
        "owner": {
            "description": "Owner of the image",
            "maxLength": 255,
            "type": [
                "null",
```

```
                "string"
            ]
        },
        "protected": {
            "description": "If true, image will not be deletable.",
            "type": "boolean"
        },
        "ramdisk_id": {
            "description": "ID of image stored in Glance that should be used
 as the ramdisk when booting an AMI-style image.",
            "is_base": false,
            "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
            "type": "string"
        },
        "schema": {
            "description": "(READ-ONLY)",
            "type": "string"
        },
        "self": {
            "description": "(READ-ONLY)",
            "type": "string"
        },
        "size": {
            "description": "Size of image file in bytes (READ-ONLY)",
            "type": [
                "null",
                "integer"
            ]
        },
        "status": {
            "description": "Status of the image (READ-ONLY)",
            "enum": [
                "queued",
                "saving",
                "active",
                "killed",
                "deleted",
                "pending_delete"
            ],
            "type": "string"
        },
        "tags": {
            "description": "List of strings related to the image",
            "items": {
                "maxLength": 255,
                "type": "string"
            },
            "type": "array"
        },
        "updated_at": {
            "description": "Date and time of the last image modification
 (READ-ONLY)",
            "type": "string"
        },
        "virtual_size": {
            "description": "Virtual size of image in bytes (READ-ONLY)",
            "type": [
                "null",
                "integer"
```

```
            ]
        },
        "visibility": {
            "description": "Scope of image accessibility",
            "enum": [
                "public",
                "private"
            ],
            "type": "string"
        }
    }
}
```

6.6.3. Show image members schema

Method	URI	Description
GET	`/v2/schemas/members`	(Since Images v2.1) Shows a JSON schema document that represents an image members entity.

An image members entity is a container of image member entities.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

6.6.3.1. Request

This operation does not accept a request body.

6.6.3.2. Response

Example 6.15. Show image members schema: JSON response

```
{
    "links": [
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "name": "members",
    "properties": {
        "members": {
            "items": {
                "name": "member",
                "properties": {
                    "created_at": {
                        "description": "Date and time of image member
 creation",
                        "type": "string"
                    },
                    "image_id": {
                        "description": "An identifier for the image",
                        "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-
fA-F]){4}-([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
                        "type": "string"
                    },
                    "member_id": {
                        "description": "An identifier for the image member
 (tenantId)",
                        "type": "string"
                    },
                    "schema": {
                        "type": "string"
                    },
                    "status": {
                        "description": "The status of this image member",
                        "enum": [
```

```
                            "pending",
                            "accepted",
                            "rejected"
                        ],
                        "type": "string"
                    },
                    "updated_at": {
                        "description": "Date and time of last modification of
 image member",
                        "type": "string"
                    }
                }
            },
            "type": "array"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

6.6.4. Show image member schema

Method	URI	Description
GET	`/v2/schemas/member`	(Since Images v2.1) Shows a JSON schema document that represents an image member entity.

The following schema is solely an example. Consider only the response to the API call as authoritative.

Normal response codes: 200

6.6.4.1. Request

This operation does not accept a request body.

6.6.4.2. Response

Example 6.16. Show image member schema: JSON response

```
{
    "name": "member",
    "properties": {
        "created_at": {
            "description": "Date and time of image member creation",
            "type": "string"
        },
        "image_id": {
            "description": "An identifier for the image",
            "pattern": "^([0-9a-fA-F]){8}-([0-9a-fA-F]){4}-([0-9a-fA-F]){4}-
([0-9a-fA-F]){4}-([0-9a-fA-F]){12}$",
            "type": "string"
        },
        "member_id": {
            "description": "An identifier for the image member (tenantId)",
            "type": "string"
        },
        "schema": {
            "type": "string"
        },
        "status": {
            "description": "The status of this image member",
            "enum": [
                "pending",
                "accepted",
                "rejected"
            ],
            "type": "string"
        },
        "updated_at": {
            "description": "Date and time of last modification of image
 member",
            "type": "string"
        }
    }
}
```

6.7. Metadata definition resource types (since API v2.0) (metadefs, namespaces, resource_types)

Lists resource types. Also, creates, lists, and removes resource type associations in a namespace.

Method	URI	Description
GET	`/v2/metadefs/resource_types`	Lists all resource types.
POST	`/v2/metadefs/namespaces/{namespace_id}/resource_types`	Creates a resource type association in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/resource_types`	Lists resource type associations in a namespace.
DELETE	`/v2/metadefs/namespaces/{namespace_id}/resource_types/{name}`	Removes a resource type association in a namespace.

6.7.1. List resource types

Method	URI	Description
GET	`/v2/metadefs/resource_types`	Lists all resource types.

You can assign metadata definition namespaces to these resource types. See the metadata definition resource types section.

Normal response codes: 200

6.7.1.1. Request

This operation does not accept a request body.

6.7.1.2. Response

Example 6.17. List resource types: JSON response

```
{
    "resource_types": [
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Glance::Image",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Cinder::Volume",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Nova::Flavor",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Nova::Aggregate",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Nova::Instance",
            "updated_at": "2014-08-28T18:13:04Z"
        }
    ]
}
```

6.7.2. Create resource type association

Method	URI	Description
POST	`/v2/metadefs/namespaces/{namespace_id}/resource_types`	Creates a resource type association in a namespace.

Normal response codes: 201

6.7.2.1. Request

This table shows the URI parameters for the create resource type association request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

Example 6.18. Create resource type association: JSON request

```
{
    "name": "OS::Cinder::Volume",
    "prefix": "hw_",
    "properties_target": "image"
}
```

6.7.2.2. Response

Example 6.19. Create resource type association: JSON response

```
{
    "created_at": "2014-09-19T16:09:13Z",
    "name": "OS::Cinder::Volume",
    "prefix": "hw_",
    "properties_target": "image",
    "updated_at": "2014-09-19T16:09:13Z"
}
```

6.7.3. List resource type associations

Method	URI	Description
GET	`/v2/metadefs/namespaces/{namespace_id}/resource_types`	Lists resource type associations in a namespace.

The response body lists resource type association entities.

Normal response codes: 200

6.7.3.1. Request

This table shows the URI parameters for the list resource type associations request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This operation does not accept a request body.

6.7.3.2. Response

Example 6.20. List resource type associations: JSON response

```
{
    "resource_types": [
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Glance::Image",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Cinder::Volume",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Nova::Flavor",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Nova::Aggregate",
            "updated_at": "2014-08-28T18:13:04Z"
        },
        {
            "created_at": "2014-08-28T18:13:04Z",
            "name": "OS::Nova::Instance",
            "updated_at": "2014-08-28T18:13:04Z"
        }
    ]
}
```

6.7.4. Remove resource type association

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/{namespace_id}/resource_types/{name}`	Removes a resource type association in a namespace.

To remove an association, first make an update namespace request to set the `protected` attribute to false (boolean) on the namespace. Then, remove the association. If the operation succeeds, the response returns the HTTP 204 status code.

If you try to remove resource type associations in a namespace with the `protected` attribute set to true (boolean), the operation fails and the response returns the HTTP 403 error code.

Normal response codes: 204

Error response codes: forbidden (403)

6.7.4.1. Request

This table shows the URI parameters for the remove resource type association request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{name}`	String	Name of the resource type.

This operation does not accept a request body.

6.8. Metadata definition namespaces (since API v2.0) (metadefs, namespaces)

Creates, lists, shows details for, updates, and deletes metadata definition namespaces. Defines namespaces that can contain property definitions, object definitions, and resource type associations.

Method	URI	Description
POST	`/v2/metadefs/namespaces`	Creates a namespace.
GET	`/v2/metadefs/namespaces{?limit,marker,visibility,resource_types,sort_key,sort_dir}`	Lists public namespaces.
GET	`/v2/metadefs/namespaces/{namespace_id}`	Gets details for a namespace.
PUT	`/v2/metadefs/namespaces/{namespace_id}`	Updates a namespace.
DELETE	`/v2/metadefs/namespaces/{namespace_id}`	Deletes a namespace and its properties, objects, and any resource type associations.

6.8.1. Create namespace

Method	URI	Description
POST	`/v2/metadefs/namespaces`	Creates a namespace.

The `Location` response header contains the newly-created URI for the namespace.

Normal response codes: 201

6.8.1.1. Request

Example 6.21. Create namespace: JSON request

```
{
    "description": "Choose capabilities that should be provided by the Compute
 Host. This provides the ability to fine tune the hardware specification
 required when a new vm is requested.",
    "display_name": "Hypervisor Selection",
    "namespace": "OS::Compute::Hypervisor",
    "properties": {
        "hypervisor_type": {
            "description": "The hypervisor type.",
            "enum": [
                "xen",
                "qemu",
                "kvm",
                "lxc",
                "uml",
                "vmware",
                "hyperv"
            ],
            "title": "Hypervisor Type",
            "type": "string"
        },
        "vm_mode": {
            "description": "The virtual machine mode.",
            "enum": [
                "hvm",
                "xen",
                "uml",
                "exe"
            ],
            "title": "VM Mode",
            "type": "string"
        }
    },
    "protected": true,
    "resource_type_associations": [
        {
            "name": "OS::Glance::Image"
        }
    ],
    "visibility": "public"
}
```

6.8.1.2. Response

Example 6.22. Create namespace: JSON response

```
{
    "description": "Choose capabilities that should be provided by the Compute
 Host. This provides the ability to fine tune the hardware specification
 required when a new vm is requested.",
    "display_name": "Hypervisor Selection",
    "namespace": "OS::Compute::Hypervisor",
    "properties": {
        "hypervisor_type": {
            "description": "The hypervisor type.",
            "enum": [
                "xen",
                "qemu",
                "kvm",
                "lxc",
                "uml",
                "vmware",
                "hyperv"
            ],
            "title": "Hypervisor Type",
            "type": "string"
        },
        "vm_mode": {
            "description": "The virtual machine mode.",
            "enum": [
                "hvm",
                "xen",
                "uml",
                "exe"
            ],
            "title": "VM Mode",
            "type": "string"
        }
    },
    "protected": true,
    "resource_type_associations": [
        {
            "name": "OS::Glance::Image"
        }
    ],
    "schema": "/v2/schemas/metadefs/namespace",
    "self": "/v2/metadefs/namespaces/OS::Compute::Hypervisor",
    "visibility": "public"
}
```

6.8.2. List namespaces

Method	URI	Description
GET	`/v2/metadefs/namespaces{?limit,marker,visibility,resource_types,sort_key,sort_dir}`	Lists public namespaces.

Returns a subset in the larger collection of namespaces and a link that you can use to get the next set of namespaces. Check for the presence of a `next` link and use it as the URI in a subsequent HTTP GET request. Follow this pattern until a `next` link is no longer provided. The `next` link preserves any query parameters that you send in your initial request. You can use the `first` link to return to the first page in the collection. If you prefer to paginate through namespaces manually, use the `limit` and `marker` parameters.

The list operation accepts the `resource_types` and `visibility` query parameters, which you can use to filter the response.

To sort the results of this operation, use the `sort_key` and `sort_dir` parameters. The API uses the natural sorting order in the namespace attribute that you provide as the `sort_key` parameter.

Normal response codes: 200

6.8.2.1. Request

This table shows the query parameters for the list namespaces request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`visibility`	String *(Optional)*	Filters the response by an image visibility value or values. Use the comma (`,`) character to separate multiple values. A valid value is `public`, `private`, or `shared`. If you omit this parameter, the response shows `public`, `private`, and `shared` images with a member status of `accepted`.
`resource_types`	Int *(Optional)*	Filters the response by a resource type or types. Use the comma (`,`) character to separate multiple values. For example, `OS::Glance::Image,OS::Nova::Flavor` shows only namespaces for these resource types.
`sort_key`	String *(Optional)*	Sorts the response by an attribute, such as `name`, `id`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` image attribute.
`sort_dir`	String *(Optional)*	Sorts the response by a set of one or more sort direction and attribute (`sort_key`) combinations. A valid value for the sort direction is `asc` (ascending) or `desc` (descending). If you omit the sort direction in a set, the default is `desc`.

This operation does not accept a request body.

6.8.2.2. Response

Example 6.23. List namespaces: JSON response


```
{
    "first": "/v2/metadefs/namespaces?sort_key=created_at&sort_dir=asc",
    "namespaces": [
        {
            "created_at": "2014-08-28T17:13:06Z",
            "description": "The libvirt compute driver options. These are
 properties specific to compute drivers.  For a list of all hypervisors, see
 here: https://wiki.openstack.org/wiki/HypervisorSupportMatrix.",
            "display_name": "libvirt Driver Options",
            "namespace": "OS::Compute::Libvirt",
            "owner": "admin",
            "protected": true,
            "resource_type_associations": [
                {
                    "created_at": "2014-08-28T17:13:06Z",
                    "name": "OS::Glance::Image",
                    "updated_at": "2014-08-28T17:13:06Z"
                }
            ],
            "schema": "/v2/schemas/metadefs/namespace",
            "self": "/v2/metadefs/namespaces/OS::Compute::Libvirt",
            "updated_at": "2014-08-28T17:13:06Z",
            "visibility": "public"
        },
        {
            "created_at": "2014-08-28T17:13:06Z",
            "description": "Compute drivers may enable quotas on CPUs
 available to a VM, disk tuning, bandwidth I/O, and instance VIF traffic
 control.  See: http://docs.openstack.org/admin-guide-cloud/compute-flavors.
html",
            "display_name": "Flavor Quota",
            "namespace": "OS::Compute::Quota",
            "owner": "admin",
            "protected": true,
            "resource_type_associations": [
                {
                    "created_at": "2014-08-28T17:13:06Z",
                    "name": "OS::Nova::Flavor",
                    "updated_at": "2014-08-28T17:13:06Z"
                }
            ],
            "schema": "/v2/schemas/metadefs/namespace",
            "self": "/v2/metadefs/namespaces/OS::Compute::Quota",
            "updated_at": "2014-08-28T17:13:06Z",
            "visibility": "public"
        },
        {
            "created_at": "2014-08-28T17:13:06Z",
            "description": "Trusted compute pools with Intel\u00ae Trusted
 Execution Technology (Intel\u00ae TXT) support IT compliance by protecting
 virtualized data centers - private, public, and hybrid clouds against
 attacks toward hypervisor and BIOS, firmware, and other pre-launch software
 components.",
```

```
            "display_name": "Trusted Compute Pools (Intel\u00ae TXT)",
            "namespace": "OS::Compute::Trust",
            "owner": "admin",
            "protected": true,
            "resource_type_associations": [
                {
                    "created_at": "2014-08-28T17:13:06Z",
                    "name": "OS::Nova::Flavor",
                    "updated_at": "2014-08-28T17:13:06Z"
                }
            ],
            "schema": "/v2/schemas/metadefs/namespace",
            "self": "/v2/metadefs/namespaces/OS::Compute::Trust",
            "updated_at": "2014-08-28T17:13:06Z",
            "visibility": "public"
        },
        {
            "created_at": "2014-08-28T17:13:06Z",
            "description": "This provides the preferred socket/core/thread
 counts for the virtual CPU instance exposed to guests. This enables the
 ability to avoid hitting limitations on vCPU topologies that OS vendors place
 on their products. See also: http://git.openstack.org/cgit/openstack/nova-
specs/tree/specs/juno/virt-driver-vcpu-topology.rst",
            "display_name": "Virtual CPU Topology",
            "namespace": "OS::Compute::VirtCPUTopology",
            "owner": "admin",
            "protected": true,
            "resource_type_associations": [
                {
                    "created_at": "2014-08-28T17:13:06Z",
                    "name": "OS::Glance::Image",
                    "prefix": "hw_",
                    "updated_at": "2014-08-28T17:13:06Z"
                },
                {
                    "created_at": "2014-08-28T17:13:06Z",
                    "name": "OS::Cinder::Volume",
                    "prefix": "hw_",
                    "properties_target": "image",
                    "updated_at": "2014-08-28T17:13:06Z"
                },
                {
                    "created_at": "2014-08-28T17:13:06Z",
                    "name": "OS::Nova::Flavor",
                    "prefix": "hw:",
                    "updated_at": "2014-08-28T17:13:06Z"
                }
            ],
            "schema": "/v2/schemas/metadefs/namespace",
            "self": "/v2/metadefs/namespaces/OS::Compute::VirtCPUTopology",
            "updated_at": "2014-08-28T17:13:06Z",
            "visibility": "public"
        }
    ],
    "schema": "/v2/schemas/metadefs/namespaces"
}
```

6.8.3. Get namespaces details

Method	URI	Description
GET	`/v2/metadefs/namespaces/{namespace_id}`	Gets details for a namespace.

The response body shows a single namespace entity with all details including properties and objects.

Normal response codes: 200

6.8.3.1. Request

This table shows the URI parameters for the get namespaces details request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This operation does not accept a request body.

6.8.3.2. Response

Example 6.24. Get namespaces details: JSON response

```
{
    "created_at": "2014-08-28T17:13:06Z",
    "description": "The libvirt compute driver options. These are properties
 specific to compute drivers.  For a list of all hypervisors, see here:
 https://wiki.openstack.org/wiki/HypervisorSupportMatrix.",
    "display_name": "libvirt Driver Options",
    "namespace": "OS::Compute::Libvirt",
    "owner": "admin",
    "properties": {
        "hw_disk_bus": {
            "description": "Specifies the type of disk controller to attach
 disk devices to.",
            "enum": [
                "scsi",
                "virtio",
                "uml",
                "xen",
                "ide",
                "usb"
            ],
            "title": "Disk Bus",
            "type": "string"
        },
        "hw_machine_type": {
            "description": "Enables booting an ARM system using the specified
 machine type. By default, if an ARM image is used and its type is not
 specified, Compute uses vexpress-a15 (for ARMv7) or virt (for AArch64)
 machine types. Valid types can be viewed by using the virsh capabilities
 command (machine types are displayed in the machine tag).",
            "title": "Machine Type",
            "type": "string"
        },
        "hw_qemu_guest_agent": {
```

```
            "description": "It is a daemon program running inside the domain
which is supposed to help management applications with executing functions
which need assistance of the guest OS. For example, freezing and thawing
filesystems, entering suspend. However, guest agent (GA) is not bullet proof,
and hostile guest OS can send spurious replies.",
            "enum": [
                "yes",
                "no"
            ],
            "title": "QEMU Guest Agent",
            "type": "string"
        },
        "hw_rng_model": {
            "default": "virtio",
            "description": "Adds a random-number generator device to the
image's instances. The cloud administrator can enable and control device
behavior by configuring the instance's flavor. By default: The generator
device is disabled. /dev/random is used as the default entropy source. To
specify a physical HW RNG device, use the following option in the nova.conf
file: rng_dev_path=/dev/hwrng",
            "title": "Random Number Generator Device",
            "type": "string"
        },
        "hw_scsi_model": {
            "default": "virtio-scsi",
            "description": "Enables the use of VirtIO SCSI (virtio-scsi) to
provide block device access for compute instances; by default, instances use
VirtIO Block (virtio-blk). VirtIO SCSI is a para-virtualized SCSI controller
device that provides improved scalability and performance, and supports
advanced SCSI hardware.",
            "title": "SCSI Model",
            "type": "string"
        },
        "hw_video_model": {
            "description": "The video image driver used.",
            "enum": [
                "vga",
                "cirrus",
                "vmvga",
                "xen",
                "qxl"
            ],
            "title": "Video Model",
            "type": "string"
        },
        "hw_video_ram": {
            "description": "Maximum RAM for the video image. Used only if a
hw_video:ram_max_mb value has been set in the flavor's extra_specs and that
value is higher than the value set in hw_video_ram.",
            "title": "Max Video Ram",
            "type": "integer"
        },
        "hw_vif_model": {
            "description": "Specifies the model of virtual network interface
device to use. The valid options depend on the configured hypervisor. KVM
and QEMU: e1000, ne2k_pci, pcnet, rtl8139, and virtio. VMware: e1000, e1000e,
VirtualE1000, VirtualE1000e, VirtualPCNet32, VirtualSriovEthernetCard, and
VirtualVmxnet. Xen: e1000, netfront, ne2k_pci, pcnet, and rtl8139.",
            "enum": [
                "e1000",
```

```
                "ne2k_pci",
                "pcnet",
                "rtl8139",
                "virtio",
                "e1000",
                "e1000e",
                "VirtualE1000",
                "VirtualE1000e",
                "VirtualPCNet32",
                "VirtualSriovEthernetCard",
                "VirtualVmxnet",
                "netfront",
                "ne2k_pci"
            ],
            "title": "Virtual Network Interface",
            "type": "string"
        },
        "os_command_line": {
            "description": "The kernel command line to be used by the libvirt
 driver, instead of the default. For linux containers (LXC), the value is used
 as arguments for initialization. This key is valid only for Amazon kernel,
 ramdisk, or machine images (aki, ari, or ami).",
            "title": "Kernel Command Line",
            "type": "string"
        }
    },
    "protected": true,
    "resource_type_associations": [
        {
            "created_at": "2014-08-28T17:13:06Z",
            "name": "OS::Glance::Image",
            "updated_at": "2014-08-28T17:13:06Z"
        }
    ],
    "schema": "/v2/schemas/metadefs/namespace",
    "self": "/v2/metadefs/namespaces/OS::Compute::Libvirt",
    "updated_at": "2014-08-28T17:13:06Z",
    "visibility": "public"
}
```

6.8.4. Update namespace

Method	URI	Description
PUT	`/v2/metadefs/namespaces/{namespace_id}`	Updates a namespace.

Normal response codes: 200

6.8.4.1. Request

This table shows the URI parameters for the update namespace request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

Example 6.25. Update namespace: JSON request

```
{
    "description": "Choose capabilities that should be provided by the Compute
 Host. This provides the ability to fine tune the hardware specification
 required when a new vm is requested.",
    "display_name": "Hypervisor Selection",
    "namespace": "OS::Compute::Hypervisor",
    "protected": false,
    "visibility": "public"
}
```

6.8.4.2. Response

Example 6.26. Update namespace: JSON response

```
{
    "created_at": "2014-09-19T13:31:37Z",
    "description": "Choose capabilities that should be provided by the Compute
 Host. This provides the ability to fine tune the harware specification
 required when a new vm is requested.",
    "display_name": "Hypervisor Selection",
    "namespace": "OS::Compute::Hypervisor",
    "owner": "7ec22942411e427692e8a3436be1031a",
    "protected": false,
    "schema": "/v2/schemas/metadefs/namespace",
    "self": "/v2/metadefs/namespaces/OS::Compute::Hypervisor",
    "updated_at": "2014-09-19T13:31:37Z",
    "visibility": "public"
}
```

6.8.5. Delete namespace

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/{namespace_id}`	Deletes a namespace and its properties, objects, and any resource type associations.

You cannot delete namespaces with the `protected` attribute set to `true` (boolean); the response returns the HTTP `403` response code.

To delete a namespace, you must first make an update namespace request to set the `protected` attribute to false (boolean) on the namespace. Then, delete the namespace.

A successful operation returns the HTTP `204` response code.

If you try to remove a namespace with the `protected` attribute set to true (boolean), the operation fails and the response returns the HTTP `403` response code.

Normal response codes: 204

Error response codes: forbidden (403)

6.8.5.1. Request

This table shows the URI parameters for the delete namespace request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This operation does not accept a request body.

6.9. Metadata definition properties (since API v2.0) (metadefs, namespaces, properties)

Creates, lists, shows details for, updates, and deletes metadata definition properties.

Method	URI	Description
POST	`/v2/metadefs/namespaces/{namespace_id}/properties`	Creates a property definition in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/properties`	Lists property definitions in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}{?resource_type}`	Shows the definition for a property.
PUT	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}`	Updates a property definition.
DELETE	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}`	Removes a property definition in a namespace.

6.9.1. Create property

Method	URI	Description
POST	`/v2/metadefs/namespaces/{namespace_id}/properties`	Creates a property definition in a namespace.

The schema is a subset of the JSON property definition schema.

Normal response codes: 201

6.9.1.1. Request

This table shows the URI parameters for the create property request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

Example 6.27. Create property: JSON request

```
{
    "description": "The hypervisor type. It may be used by the host properties
 filter for scheduling. The ImagePropertiesFilter filters compute nodes that
 satisfy any architecture, hypervisor type, or virtual machine mode properties
 specified on the instance's image properties. Image properties are contained
 in the image dictionary in the request_spec.",
    "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
        "vmware",
        "hyperv"
    ],
    "name": "hypervisor_type",
    "title": "Hypervisor Type",
    "type": "string"
}
```

6.9.1.2. Response

Example 6.28. Create property: JSON response

```
{
    "description": "The hypervisor type. It may be used by the host properties
 filter for scheduling. The ImagePropertiesFilter filters compute nodes that
 satisfy any architecture, hypervisor type, or virtual machine mode properties
 specified on the instance's image properties. Image properties are contained
 in the image dictionary in the request_spec.",
    "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
```

```
        "vmware",
        "hyperv"
    ],
    "name": "hypervisor_type",
    "title": "Hypervisor Type",
    "type": "string"
}
```

6.9.2. List property definitions

Method	URI	Description
GET	`/v2/metadefs/namespaces/{namespace_id}/properties`	Lists property definitions in a namespace.

Normal response codes: 200

6.9.2.1. Request

This table shows the URI parameters for the list property definitions request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This operation does not accept a request body.

6.9.2.2. Response

Example 6.29. List property definitions: JSON response

```
{
    "properties": {
        "hw_disk_bus": {
            "description": "Specifies the type of disk controller to attach
 disk devices to.",
            "enum": [
                "scsi",
                "virtio",
                "uml",
                "xen",
                "ide",
                "usb"
            ],
            "title": "Disk Bus",
            "type": "string"
        },
        "hw_machine_type": {
            "description": "Enables booting an ARM system using the specified
 machine type. By default, if an ARM image is used and its type is not
 specified, Compute uses vexpress-a15 (for ARMv7) or virt (for AArch64)
 machine types. Valid types can be viewed by using the virsh capabilities
 command (machine types are displayed in the machine tag).",
            "title": "Machine Type",
            "type": "string"
        },
        "hw_qemu_guest_agent": {
            "description": "It is a daemon program running inside the domain
 which is supposed to help management applications with executing functions
 which need assistance of the guest OS. For example, freezing and thawing
 filesystems, entering suspend. However, guest agent (GA) is not bullet proof,
 and hostile guest OS can send spurious replies.",
            "enum": [
                "yes",
                "no"
            ],
            "title": "QEMU Guest Agent",
```

```
            "type": "string"
        },
        "hw_rng_model": {
            "default": "virtio",
            "description": "Adds a random-number generator device to the
image's instances. The cloud administrator can enable and control device
behavior by configuring the instance's flavor. By default: The generator
device is disabled. /dev/random is used as the default entropy source. To
specify a physical HW RNG device, use the following option in the nova.conf
file: rng_dev_path=/dev/hwrng",
            "title": "Random Number Generator Device",
            "type": "string"
        },
        "hw_scsi_model": {
            "default": "virtio-scsi",
            "description": "Enables the use of VirtIO SCSI (virtio-scsi) to
provide block device access for compute instances; by default, instances use
VirtIO Block (virtio-blk). VirtIO SCSI is a para-virtualized SCSI controller
device that provides improved scalability and performance, and supports
advanced SCSI hardware.",
            "title": "SCSI Model",
            "type": "string"
        },
        "hw_video_model": {
            "description": "The video image driver used.",
            "enum": [
                "vga",
                "cirrus",
                "vmvga",
                "xen",
                "qxl"
            ],
            "title": "Video Model",
            "type": "string"
        },
        "hw_video_ram": {
            "description": "Maximum RAM for the video image. Used only if a
hw_video:ram_max_mb value has been set in the flavor's extra_specs and that
value is higher than the value set in hw_video_ram.",
            "title": "Max Video Ram",
            "type": "integer"
        },
        "hw_vif_model": {
            "description": "Specifies the model of virtual network interface
device to use. The valid options depend on the configured hypervisor. KVM
and QEMU: e1000, ne2k_pci, pcnet, rtl8139, and virtio. VMware: e1000, e1000e,
VirtualE1000, VirtualE1000e, VirtualPCNet32, VirtualSriovEthernetCard, and
VirtualVmxnet. Xen: e1000, netfront, ne2k_pci, pcnet, and rtl8139.",
            "enum": [
                "e1000",
                "ne2k_pci",
                "pcnet",
                "rtl8139",
                "virtio",
                "e1000",
                "e1000e",
                "VirtualE1000",
                "VirtualE1000e",
                "VirtualPCNet32",
                "VirtualSriovEthernetCard",
```

```
                "VirtualVmxnet",
                "netfront",
                "ne2k_pci"
            ],
            "title": "Virtual Network Interface",
            "type": "string"
        },
        "os_command_line": {
            "description": "The kernel command line to be used by the libvirt
 driver, instead of the default. For linux containers (LXC), the value is used
 as arguments for initialization. This key is valid only for Amazon kernel,
 ramdisk, or machine images (aki, ari, or ami).",
            "title": "Kernel Command Line",
            "type": "string"
        }
    }
}
```

6.9.3. Show property definition

Method	URI	Description
GET	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}{?resource_type}`	Shows the definition for a property.

If you use the `resource_type` query parameter, the API removes the prefix of the resource type from the property name before it submits the query. This enables you to look for a property name that starts with a prefix from an associated resource type.

The response body shows a single property entity.

Normal response codes: 200

6.9.3.1. Request

This table shows the URI parameters for the show property definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{property_name}`	String	The name of the property.

This table shows the query parameters for the show property definition request:

Name	Type	Description
`resource_type`	Int *(Optional)*	Filters the response by property names that start with a prefix from an associated resource type. The API removes the prefix of the resource type from the property name in the response.

This operation does not accept a request body.

6.9.3.2. Response

Example 6.30. Show property definition: JSON response

```
{
    "description": "The hypervisor type. It may be used by the host properties
 filter for scheduling. The ImagePropertiesFilter filters compute nodes that
 satisfy any architecture, hypervisor type, or virtual machine mode properties
 specified on the instance's image properties. Image properties are contained
 in the image dictionary in the request_spec.",
    "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
        "vmware",
        "hyperv"
    ],
```

```
    "name": "hypervisor_type",
    "title": "Hypervisor Type",
    "type": "string"
}
```

6.9.4. Update property definition

Method	URI	Description
PUT	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}`	Updates a property definition.

Normal response codes: 200

6.9.4.1. Request

This table shows the URI parameters for the update property definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{property_name}`	String	The name of the property.

Example 6.31. Update property definition: JSON request

```
{
    "description": "The hypervisor type. It may be used by the host properties
 filter for scheduling. The ImagePropertiesFilter filters compute nodes that
 satisfy any architecture, hypervisor type, or virtual machine mode properties
 specified on the instance's image properties. Image properties are contained
 in the image dictionary in the request_spec.",
    "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
        "vmware",
        "hyperv"
    ],
    "name": "hypervisor_type",
    "title": "Hypervisor Type",
    "type": "string"
}
```

6.9.4.2. Response

Example 6.32. Update property definition: JSON response

```
{
    "description": "The hypervisor type. It may be used by the host properties
 filter for scheduling. The ImagePropertiesFilter filters compute nodes that
 satisfy any architecture, hypervisor type, or virtual machine mode properties
 specified on the instance's image properties. Image properties are contained
 in the image dictionary in the request_spec.",
    "enum": [
        "xen",
        "qemu",
        "kvm",
        "lxc",
        "uml",
```

```
        "vmware",
        "hyperv"
    ],
    "name": "hypervisor_type",
    "title": "Hypervisor Type",
    "type": "string"
}
```

6.9.5. Remove property definition

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/{namespace_id}/properties/{property_name}`	Removes a property definition in a namespace.

To remove a property, first make an update namespace request to set the `protected` attribute to false (boolean) on the namespace. Then, remove the property. If the operation succeeds, the response returns the HTTP 204 status code.

If you try to remove a property in a namespace with the `protected` attribute set to true (boolean), the operation fails and the response returns the HTTP 403 error code.

Normal response codes: 204

Error response codes: forbidden (403)

6.9.5.1. Request

This table shows the URI parameters for the remove property definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{property_name}`	String	The name of the property.

This operation does not accept a request body.

6.10. Metadata definition objects (since API v2.0) (metadefs, namespaces, objects)

Creates, lists, shows details for, updates, and deletes metadata definition objects.

Method	URI	Description
POST	`/v2/metadefs/namespaces/{namespace_id}/objects`	Creates an object definition in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/objects{?visibility,resource_types,sort_key,sort_dir}`	Lists object definitions in a namespace.
GET	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Shows the definition for an object.
PUT	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Updates an object definition in a namespace.
DELETE	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Deletes an object definition from a namespace.

6.10.1. Create object

Method	URI	Description
POST	`/v2/metadefs/namespaces/{namespace_id}/objects`	Creates an object definition in a namespace.

Normal response codes: 201

6.10.1.1. Request

This table shows the URI parameters for the create object request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

Example 6.33. Create object: JSON request

```
{
    "description": "You can configure the CPU limits with control parameters.
",
    "name": "CPU Limits",
    "properties": {
        "quota:cpu_period": {
            "description": "Specifies the enforcement interval (unit:
 microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
 the domain is not allowed to consume more than the quota worth of runtime.
 The value should be in range [1000, 1000000]. A period with value 0 means no
 value.",
            "maximum": 1000000,
            "minimum": 1000,
            "title": "Quota: CPU Period",
            "type": "integer"
        },
        "quota:cpu_quota": {
            "description": "Specifies the maximum allowed bandwidth (unit:
 microseconds). A domain with a negative-value quota indicates that the domain
 has infinite bandwidth, which means that it is not bandwidth controlled. The
 value should be in range [1000, 18446744073709551] or less than 0. A quota
 with value 0 means no value. You can use this feature to ensure that all
 vCPUs run at the same speed.",
            "title": "Quota: CPU Quota",
            "type": "integer"
        },
        "quota:cpu_shares": {
            "description": "Specifies the proportional weighted share for the
 domain. If this element is omitted, the service defaults to the OS provided
 defaults. There is no unit for the value; it is a relative measure based on
 the setting of other VMs. For example, a VM configured with value 2048 gets
 twice as much CPU time as a VM configured with value 1024.",
            "title": "Quota: CPU Shares",
            "type": "integer"
        }
    },
    "required": []
}
```

6.10.1.2. Response

Example 6.34. Create object: JSON response


```
{
    "created_at": "2014-09-19T18:20:56Z",
    "description": "You can configure the CPU limits with control parameters.
",
    "name": "CPU Limits",
    "properties": {
        "quota:cpu_period": {
            "description": "Specifies the enforcement interval (unit:
 microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
 the domain is not allowed to consume more than the quota worth of runtime.
 The value should be in range [1000, 1000000]. A period with value 0 means no
 value.",
            "maximum": 1000000,
            "minimum": 1000,
            "title": "Quota: CPU Period",
            "type": "integer"
        },
        "quota:cpu_quota": {
            "description": "Specifies the maximum allowed bandwidth (unit:
 microseconds). A domain with a negative-value quota indicates that the domain
 has infinite bandwidth, which means that it is not bandwidth controlled. The
 value should be in range [1000, 18446744073709551] or less than 0. A quota
 with value 0 means no value. You can use this feature to ensure that all
 vCPUs run at the same speed.",
            "title": "Quota: CPU Quota",
            "type": "integer"
        },
        "quota:cpu_shares": {
            "description": "Specifies the proportional weighted share for the
 domain. If this element is omitted, the service defaults to the OS provided
 defaults. There is no unit for the value; it is a relative measure based on
 the setting of other VMs. For example, a VM configured with value 2048 gets
 twice as much CPU time as a VM configured with value 1024.",
            "title": "Quota: CPU Shares",
            "type": "integer"
        }
    },
    "required": [],
    "schema": "/v2/schemas/metadefs/object",
    "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU Limits",
    "updated_at": "2014-09-19T18:20:56Z"
}
```

6.10.2. List object definitions

Method	URI	Description
GET	`/v2/metadefs/namespaces/{namespace_id}/objects{?visibility,resource_types,sort_key,sort_dir}`	Lists object definitions in a namespace.

Returns a subset of the larger collection of namespaces and a link that you can use to get the next set of namespaces. You should always check for the presence of a `next` link and use it as the URI in a subsequent HTTP GET request. You should follow this pattern until a `next` link is no longer provided. The next link preserves any query parameters that you send in your initial request. You can use the `first` link to jump back to the first page of the collection. If you prefer to paginate through namespaces manually, use the `limit` and `marker` parameters.

Use the `resource_types` and `visibility` query parameters to filter the response.

For example, set the `resource_types` query parameter to `OS::Glance::Image,OS::Nova::Flavor` to filter the response to include only namespaces that are associated with the given resource types.

You can sort the results of this operation by using the `sort_key` and `sort_dir` parameters. The API uses the natural sorting of whatever namespace attribute is provided as the `sort_key`.

Normal response codes: 200

6.10.2.1. Request

This table shows the URI parameters for the list object definitions request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This table shows the query parameters for the list object definitions request:

Name	Type	Description
`visibility`	String *(Optional)*	Filters the response by a namespace visibility value. A valid value is `public` or `private`. If you omit this parameter, the response shows `public` and `private` namespaces.
`resource_types`	Int *(Optional)*	Filters the response by a resource type or types. Use the comma (`,`) character to separate multiple values. For example, `OS::Glance::Image,OS::Nova::Flavor` shows only namespaces for these resource types.
`sort_key`	String *(Optional)*	Sorts the response by an attribute, such as `name`, `id`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` image attribute.
`sort_dir`	String *(Optional)*	Sorts the response by a set of one or more sort direction and attribute (`sort_key`) combinations. A valid value for the sort direction is `asc` (ascending) or `desc` (descending). If you omit the sort direction in a set, the default is `desc`.

This operation does not accept a request body.

6.10.2.2. Response

Example 6.35. List object definitions: JSON response


```
{
    "objects": [
        {
            "created_at": "2014-09-18T18:16:35Z",
            "description": "You can configure the CPU limits with control
 parameters.",
            "name": "CPU Limits",
            "properties": {
                "quota:cpu_period": {
                    "description": "Specifies the enforcement interval (unit:
 microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
 the domain is not allowed to consume more than the quota worth of runtime.
 The value should be in range [1000, 1000000]. A period with value 0 means no
 value.",
                    "maximum": 1000000,
                    "minimum": 1000,
                    "title": "Quota: CPU Period",
                    "type": "integer"
                },
                "quota:cpu_quota": {
                    "description": "Specifies the maximum allowed bandwidth
 (unit: microseconds). A domain with a negative-value quota indicates that
 the domain has infinite bandwidth, which means that it is not bandwidth
 controlled. The value should be in range [1000, 18446744073709551] or less
 than 0. A quota with value 0 means no value. You can use this feature to
 ensure that all vCPUs run at the same speed.",
                    "title": "Quota: CPU Quota",
                    "type": "integer"
                },
                "quota:cpu_shares": {
                    "description": "Specifies the proportional weighted share
 for the domain. If this element is omitted, the service defaults to the OS
 provided defaults. There is no unit for the value; it is a relative measure
 based on the setting of other VMs. For example, a VM configured with value
 2048 gets twice as much CPU time as a VM configured with value 1024.",
                    "title": "Quota: CPU Shares",
                    "type": "integer"
                }
            },
            "required": [],
            "schema": "/v2/schemas/metadefs/object",
            "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU
 Limits"
        },
        {
            "created_at": "2014-09-18T18:16:35Z",
            "description": "Using disk I/O quotas, you can set maximum disk
 write to 10 MB per second for a VM user.",
            "name": "Disk QoS",
            "properties": {
                "quota:disk_read_bytes_sec": {
                    "description": "Sets disk I/O quota for disk read bytes /
 sec.",
                    "title": "Quota: Disk read bytes / sec",
                    "type": "integer"
```

```
                },
                "quota:disk_read_iops_sec": {
                    "description": "Sets disk I/O quota for disk read IOPS /
sec.",
                    "title": "Quota: Disk read IOPS / sec",
                    "type": "integer"
                },
                "quota:disk_total_bytes_sec": {
                    "description": "Sets disk I/O quota for total disk bytes /
sec.",
                    "title": "Quota: Disk Total Bytes / sec",
                    "type": "integer"
                },
                "quota:disk_total_iops_sec": {
                    "description": "Sets disk I/O quota for disk total IOPS /
sec.",
                    "title": "Quota: Disk Total IOPS / sec",
                    "type": "integer"
                },
                "quota:disk_write_bytes_sec": {
                    "description": "Sets disk I/O quota for disk write bytes /
sec.",
                    "title": "Quota: Disk Write Bytes / sec",
                    "type": "integer"
                },
                "quota:disk_write_iops_sec": {
                    "description": "Sets disk I/O quota for disk write IOPS /
sec.",
                    "title": "Quota: Disk Write IOPS / sec",
                    "type": "integer"
                }
            },
            "required": [],
            "schema": "/v2/schemas/metadefs/object",
            "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/Disk
QoS"
        },
        {
            "created_at": "2014-09-18T18:16:35Z",
            "description": "Bandwidth QoS tuning for instance virtual
interfaces (VIFs) may be specified with these properties. Incoming and
outgoing traffic can be shaped independently. If not specified, no quality
of service (QoS) is applied on that traffic direction. So, if you want to
shape only the network's incoming traffic, use inbound only (and vice versa).
The OpenStack Networking service abstracts the physical implementation of
the network, allowing plugins to configure and manage physical resources.
Virtual Interfaces (VIF) in the logical model are analogous to physical
network interface cards (NICs). VIFs are typically owned a managed by an
external service; for instance when OpenStack Networking is used for building
OpenStack networks, VIFs would be created, owned, and managed in Nova. VIFs
are connected to OpenStack Networking networks via ports. A port is analogous
to a port on a network switch, and it has an administrative state. When a
VIF is attached to a port the OpenStack Networking API creates an attachment
object, which specifies the fact that a VIF with a given identifier is
plugged into the port.",
            "name": "Virtual Interface QoS",
            "properties": {
                "quota:vif_inbound_average": {
```

```
                    "description": "Network Virtual Interface (VIF) inbound
 average in kilobytes per second. Specifies average bit rate on the interface
 being shaped.",
                    "title": "Quota: VIF Inbound Average",
                    "type": "integer"
                },
                "quota:vif_inbound_burst": {
                    "description": "Network Virtual Interface (VIF) inbound
 burst in total kilobytes. Specifies the amount of bytes that can be burst at
 peak speed.",
                    "title": "Quota: VIF Inbound Burst",
                    "type": "integer"
                },
                "quota:vif_inbound_peak": {
                    "description": "Network Virtual Interface (VIF) inbound
 peak in kilobytes per second. Specifies maximum rate at which an interface
 can receive data.",
                    "title": "Quota: VIF Inbound Peak",
                    "type": "integer"
                },
                "quota:vif_outbound_average": {
                    "description": "Network Virtual Interface (VIF) outbound
 average in kilobytes per second. Specifies average bit rate on the interface
 being shaped.",
                    "title": "Quota: VIF Outbound Average",
                    "type": "integer"
                },
                "quota:vif_outbound_burst": {
                    "description": "Network Virtual Interface (VIF) outbound
 burst in total kilobytes. Specifies the amount of bytes that can be burst at
 peak speed.",
                    "title": "Quota: VIF Outbound Burst",
                    "type": "integer"
                },
                "quota:vif_outbound_peak": {
                    "description": "Network Virtual Interface (VIF) outbound
 peak in kilobytes per second. Specifies maximum rate at which an interface
 can send data.",
                    "title": "Quota: VIF Outbound Burst",
                    "type": "integer"
                }
            },
            "required": [],
            "schema": "/v2/schemas/metadefs/object",
            "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/
Virtual Interface QoS"
        }
    ],
    "schema": "v2/schemas/metadefs/objects"
}
```

6.10.3. Show object definition

Method	URI	Description
GET	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Shows the definition for an object.

The response body shows a single object entity.

Normal response codes: 200

6.10.3.1. Request

This table shows the URI parameters for the show object definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{object_name}`	String	The name of the object.

This operation does not accept a request body.

6.10.3.2. Response

Example 6.36. Show object definition: JSON response

```
{
    "created_at": "2014-09-19T18:20:56Z",
    "description": "You can configure the CPU limits with control parameters.
",
    "name": "CPU Limits",
    "properties": {
        "quota:cpu_period": {
            "description": "Specifies the enforcement interval (unit:
 microseconds) for QEMU and LXC hypervisors. Within a period, each VCPU of
 the domain is not allowed to consume more than the quota worth of runtime.
 The value should be in range [1000, 1000000]. A period with value 0 means no
 value.",
            "maximum": 1000000,
            "minimum": 1000,
            "title": "Quota: CPU Period",
            "type": "integer"
        },
        "quota:cpu_quota": {
            "description": "Specifies the maximum allowed bandwidth (unit:
 microseconds). A domain with a negative-value quota indicates that the domain
 has infinite bandwidth, which means that it is not bandwidth controlled. The
 value should be in range [1000, 18446744073709551] or less than 0. A quota
 with value 0 means no value. You can use this feature to ensure that all
 vCPUs run at the same speed.",
            "title": "Quota: CPU Quota",
            "type": "integer"
        },
        "quota:cpu_shares": {
            "description": "Specifies the proportional weighted share for the
 domain. If this element is omitted, the service defaults to the OS provided
```

```
 defaults. There is no unit for the value; it is a relative measure based on
 the setting of other VMs. For example, a VM configured with value 2048 gets
 twice as much CPU time as a VM configured with value 1024.",
            "title": "Quota: CPU Shares",
            "type": "integer"
        }
    },
    "required": [],
    "schema": "/v2/schemas/metadefs/object",
    "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU Limits",
    "updated_at": "2014-09-19T18:20:56Z"
}
```

6.10.4. Update object definition

Method	URI	Description
PUT	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Updates an object definition in a namespace.

Normal response codes: 200

6.10.4.1. Request

This table shows the URI parameters for the update object definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{object_name}`	String	The name of the object.

Example 6.37. Update object definition: JSON request

```
{
    "description": "You can configure the CPU limits with control parameters.
",
    "name": "CPU Limits",
    "properties": {
        "quota:cpu_shares": {
            "description": "Specifies the proportional weighted share for the
 domain. If this element is omitted, the service defaults to the OS provided
 defaults. There is no unit for the value; it is a relative measure based on
 the setting of other VMs. For example, a VM configured with value 2048 gets
 twice as much CPU time as a VM configured with value 1024.",
            "title": "Quota: CPU Shares",
            "type": "integer"
        }
    },
    "required": []
}
```

6.10.4.2. Response

Example 6.38. Update object definition: JSON response

```
{
    "created_at": "2014-09-19T19:20:56Z",
    "description": "You can configure the CPU limits with control parameters.
",
    "name": "CPU Limits",
    "properties": {
        "quota:cpu_shares": {
            "description": "Specifies the proportional weighted share for the
 domain. If this element is omitted, the service defaults to the OS provided
 defaults. There is no unit for the value; it is a relative measure based on
 the setting of other VMs. For example, a VM configured with value 2048 gets
 twice as much CPU time as a VM configured with value 1024.",
            "title": "Quota: CPU Shares",
            "type": "integer"
```

```
        }
    },
    "required": [],
    "schema": "/v2/schemas/metadefs/object",
    "self": "/v2/metadefs/namespaces/OS::Compute::Quota/objects/CPU Limits",
    "updated_at": "2014-09-19T19:20:56Z"
}
```

6.10.5. Delete property definition

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/{namespace_id}/objects/{object_name}`	Deletes an object definition from a namespace.

To delete a protected object from a namespace, you must first set the `protected` attribute to false (boolean) on the namespace and then perform the delete. If you try to delete a protected object, the call returns the `403` response code.

When you successfully delete an object from a namespace, the response is empty and the response code is `204`.

Normal response codes: 204

Error response codes: forbidden (403)

6.10.5.1. Request

This table shows the URI parameters for the delete property definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{object_name}`	String	The name of the object.

This operation does not accept a request body.

6.11. Metadata definition tags (since API v2.0) (metadefs, namespaces, tags)

Creates, lists, shows details for, updates, and deletes metadata definition tags.

Method	URI	Description
POST	`/v2/metadefs/namespaces/tags/{namespace_id}`	Creates one or more tag definitions in a namespace.
GET	`/v2/metadefs/namespaces/tags/{namespace_id}{?limit,marker,sort_key,sort_dir}`	Lists the tag definitions within a namespace.
DELETE	`/v2/metadefs/namespaces/tags/{namespace_id}`	Deletes all tag definitions within a namespace.
POST	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Adds a tag to the list of namespace tag definitions.
GET	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Gets a definition for a tag.
PUT	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Renames a tag definition.
DELETE	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Deletes a tag definition within a namespace.

6.11.1. Create tags

Method	URI	Description
POST	`/v2/metadefs/namespaces/tags/{namespace_id}`	Creates one or more tag definitions in a namespace.

Normal response codes: 201

6.11.1.1. Request

This table shows the URI parameters for the create tags request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This operation does not accept a request body.

6.11.1.2. Response

Example 6.39. Create tags: JSON response

```
{
    "tags": [
        {
            "name": "sample-tag1"
        },
        {
            "name": "sample-tag2"
        },
        {
            "name": "sample-tag3"
        }
    ]
}
```

6.11.2. List tags

Method	URI	Description
GET	`/v2/metadefs/namespaces/tags/{namespace_id}{?limit,marker,sort_key,sort_dir}`	Lists the tag definitions within a namespace.

To manually paginate through the list of tags, use the `limit` and `marker` parameters.

To sort the results of this operation use the `sort_key` and `sort_dir` parameters. The API uses the natural sort order of the tag attribute of the `sort_key` parameter.

Normal response codes: 200

6.11.2.1. Request

This table shows the URI parameters for the list tags request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This table shows the query parameters for the list tags request:

Name	Type	Description
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`sort_key`	String *(Optional)*	Sorts the response by an attribute, such as `name`, `id`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` image attribute.
`sort_dir`	String *(Optional)*	Sorts the response by a set of one or more sort direction and attribute (`sort_key`) combinations. A valid value for the sort direction is `asc` (ascending) or `desc` (descending). If you omit the sort direction in a set, the default is `desc`.

This operation does not accept a request body.

6.11.2.2. Response

Example 6.40. List tags: JSON response

```
{
    "tags": [
        {
            "name": "sample-tag1"
        },
        {
            "name": "sample-tag2"
        },
```

```
        {
            "name": "sample-tag3"
        }
    ]
}
```

6.11.3. Delete all tag definitions

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/tags/{namespace_id}`	Deletes all tag definitions within a namespace.

You cannot delete tags in a namespace with the 'protected' attribute set to true (boolean); the response returns the HTTP 403 status code.

You must first set the `protected` attribute to false (boolean) on the namespace and then perform the delete. The response is empty and returns the HTTP 204 status code.

Normal response codes: 204

Error response codes: forbidden (403)

6.11.3.1. Request

This table shows the URI parameters for the delete all tag definitions request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.

This operation does not accept a request body.

6.11.4. Add tag definition

Method	URI	Description
POST	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Adds a tag to the list of namespace tag definitions.

Normal response codes: 200

6.11.4.1. Request

This table shows the URI parameters for the add tag definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{tag_name}`	String	The name of the tag.

This operation does not accept a request body.

6.11.4.2. Response

Example 6.41. Add tag definition: JSON response

```
{
    "created_at": "2015-05-09T01:12:31Z",
    "name": "added-sample-tag",
    "updated_at": "2015-05-09T01:12:31Z"
}
```

6.11.5. Get tag definition

Method	URI	Description
GET	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Gets a definition for a tag.

The response body shows a single tag entity.

Normal response codes: 200

6.11.5.1. Request

This table shows the URI parameters for the get tag definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{tag_name}`	String	The name of the tag.

This operation does not accept a request body.

6.11.5.2. Response

Example 6.42. Get tag definition: JSON response

```
{
    "created_at": "2015-05-06T23:16:12Z",
    "name": "sample-tag2",
    "updated_at": "2015-05-06T23:16:12Z"
}
```

6.11.6. Update tag definition

Method	URI	Description
PUT	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Renames a tag definition.

Normal response codes: 200

6.11.6.1. Request

This table shows the URI parameters for the update tag definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{tag_name}`	String	The name of the tag.

This operation does not accept a request body.

6.11.6.2. Response

Example 6.43. Update tag definition: JSON response

```
{
    "name": "new-tag-name"
}
```

6.11.7. Delete tag definition

Method	URI	Description
DELETE	`/v2/metadefs/namespaces/tags/{namespace_id}/{tag_name}`	Deletes a tag definition within a namespace.

You cannot delete tags in a namespace with the 'protected' attribute set to true (boolean); the response returns the HTTP 403 status code.

You must first set the `protected` attribute to false (boolean) on the namespace and then perform the delete. The response is empty and returns the HTTP 204 status code.

Normal response codes: 204

Error response codes: forbidden (403)

6.11.7.1. Request

This table shows the URI parameters for the delete tag definition request:

Name	Type	Description
`{namespace_id}`	UUID	The UUID of the namespace.
`{tag_name}`	String	The name of the tag.

This operation does not accept a request body.

6.12. Metadata definition schemas (schemas, metadefs, namespace, namespaces, object, objects, property, properties, tag, tags, resource_type, resource_types)

Gets a JSON-schema document that represents a metadata definition entity.

Method	URI	Description
GET	`/v2/schemas/metadefs/namespace`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace entity.
GET	`/v2/schemas/metadefs/namespaces`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespaces entity.
GET	`/v2/schemas/metadefs/object`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition object entity.
GET	`/v2/schemas/metadefs/objects`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition objects entity.
GET	`/v2/schemas/metadefs/property`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition property entity.
GET	`/v2/schemas/metadefs/properties`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition properties entity.
GET	`/v2/schemas/metadefs/tag`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tag entity.

Method	URI	Description
GET	`/v2/schemas/metadefs/tags`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tags entity.
GET	`/v2/schemas/metadefs/resource_type`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type association entity.
GET	`/v2/schemas/metadefs/resource_types`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type associations entity.

6.12.1. Show metadata definition namespace schema

Method	URI	Description
GET	`/v2/schemas/metadefs/namespace`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.1.1. Request

This operation does not accept a request body.

6.12.1.2. Response

Example 6.44. Show metadata definition namespace schema: JSON response

```
{
    "additionalProperties": false,
    "definitions": {
        "positiveInteger": {
            "minimum": 0,
            "type": "integer"
        },
        "positiveIntegerDefault0": {
            "allOf": [
                {
                    "$ref": "#/definitions/positiveInteger"
                },
                {
                    "default": 0
                }
            ]
        },
        "property": {
            "additionalProperties": {
                "properties": {
                    "additionalItems": {
                        "type": "boolean"
                    },
                    "default": {},
                    "description": {
                        "type": "string"
                    },
                    "enum": {
                        "type": "array"
                    },
                    "items": {
                        "properties": {
                            "enum": {
                                "type": "array"
                            },
                            "type": {
                                "enum": [
                                    "array",
```

```
                                    "boolean",
                                    "integer",
                                    "number",
                                    "object",
                                    "string",
                                    null
                                ],
                                "type": "string"
                            }
                        },
                        "type": "object"
                    },
                    "maxItems": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maxLength": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maximum": {
                        "type": "number"
                    },
                    "minItems": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minLength": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minimum": {
                        "type": "number"
                    },
                    "name": {
                        "type": "string"
                    },
                    "pattern": {
                        "format": "regex",
                        "type": "string"
                    },
                    "readonly": {
                        "type": "boolean"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    },
                    "title": {
                        "type": "string"
                    },
                    "type": {
                        "enum": [
                            "array",
                            "boolean",
                            "integer",
                            "number",
                            "object",
                            "string",
                            null
                        ],
                        "type": "string"
                    },
                    "uniqueItems": {
                        "default": false,
```

```
                        "type": "boolean"
                    }
                },
                "required": [
                    "title",
                    "type"
                ],
                "type": "object"
            },
            "type": "object"
        },
        "stringArray": {
            "items": {
                "type": "string"
            },
            "type": "array",
            "uniqueItems": true
        }
    },
    "name": "namespace",
    "properties": {
        "created_at": {
            "description": "Date and time of namespace creation (READ-ONLY)",
            "format": "date-time",
            "type": "string"
        },
        "description": {
            "description": "Provides a user friendly description of the
namespace.",
            "maxLength": 500,
            "type": "string"
        },
        "display_name": {
            "description": "The user friendly name for the namespace. Used by
UI if available.",
            "maxLength": 80,
            "type": "string"
        },
        "namespace": {
            "description": "The unique namespace text.",
            "maxLength": 80,
            "type": "string"
        },
        "objects": {
            "items": {
                "properties": {
                    "description": {
                        "type": "string"
                    },
                    "name": {
                        "type": "string"
                    },
                    "properties": {
                        "$ref": "#/definitions/property"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    }
                },
                "type": "object"
```

```
            },
            "type": "array"
        },
        "owner": {
            "description": "Owner of the namespace.",
            "maxLength": 255,
            "type": "string"
        },
        "properties": {
            "$ref": "#/definitions/property"
        },
        "protected": {
            "description": "If true, namespace will not be deletable.",
            "type": "boolean"
        },
        "resource_type_associations": {
            "items": {
                "properties": {
                    "name": {
                        "type": "string"
                    },
                    "prefix": {
                        "type": "string"
                    },
                    "properties_target": {
                        "type": "string"
                    }
                },
                "type": "object"
            },
            "type": "array"
        },
        "schema": {
            "type": "string"
        },
        "self": {
            "type": "string"
        },
        "updated_at": {
            "description": "Date and time of the last namespace modification
 (READ-ONLY)",
            "format": "date-time",
            "type": "string"
        },
        "visibility": {
            "description": "Scope of namespace accessibility.",
            "enum": [
                "public",
                "private"
            ],
            "type": "string"
        }
    },
    "required": [
        "namespace"
    ]
}
```

6.12.2. Show metadata definition namespaces schema

Method	URI	Description
GET	`/v2/schemas/metadefs/namespaces`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespaces entity.

A namespaces entity is a container for namespace entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.2.1. Request

This operation does not accept a request body.

6.12.2.2. Response

Example 6.45. Show metadata definition namespaces schema: JSON response

```
{
    "definitions": {
        "positiveInteger": {
            "minimum": 0,
            "type": "integer"
        },
        "positiveIntegerDefault0": {
            "allOf": [
                {
                    "$ref": "#/definitions/positiveInteger"
                },
                {
                    "default": 0
                }
            ]
        },
        "property": {
            "additionalProperties": {
                "properties": {
                    "additionalItems": {
                        "type": "boolean"
                    },
                    "default": {},
                    "description": {
                        "type": "string"
                    },
                    "enum": {
                        "type": "array"
                    },
                    "items": {
                        "properties": {
                            "enum": {
                                "type": "array"
                            },
                            "type": {
                                "enum": [
```

```
                                    "array",
                                    "boolean",
                                    "integer",
                                    "number",
                                    "object",
                                    "string",
                                    null
                                ],
                                "type": "string"
                            }
                        },
                        "type": "object"
                    },
                    "maxItems": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maxLength": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maximum": {
                        "type": "number"
                    },
                    "minItems": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minLength": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minimum": {
                        "type": "number"
                    },
                    "name": {
                        "type": "string"
                    },
                    "pattern": {
                        "format": "regex",
                        "type": "string"
                    },
                    "readonly": {
                        "type": "boolean"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    },
                    "title": {
                        "type": "string"
                    },
                    "type": {
                        "enum": [
                            "array",
                            "boolean",
                            "integer",
                            "number",
                            "object",
                            "string",
                            null
                        ],
                        "type": "string"
                    },
                    "uniqueItems": {
```

```
                        "default": false,
                        "type": "boolean"
                    }
                },
                "required": [
                    "title",
                    "type"
                ],
                "type": "object"
            },
            "type": "object"
        },
        "stringArray": {
            "items": {
                "type": "string"
            },
            "type": "array",
            "uniqueItems": true
        }
    },
    "links": [
        {
            "href": "{first}",
            "rel": "first"
        },
        {
            "href": "{next}",
            "rel": "next"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "name": "namespaces",
    "properties": {
        "first": {
            "type": "string"
        },
        "namespaces": {
            "items": {
                "additionalProperties": false,
                "name": "namespace",
                "properties": {
                    "created_at": {
                        "description": "Date and time of namespace creation
(READ-ONLY)",
                        "format": "date-time",
                        "type": "string"
                    },
                    "description": {
                        "description": "Provides a user friendly description
of the namespace.",
                        "maxLength": 500,
                        "type": "string"
                    },
                    "display_name": {
                        "description": "The user friendly name for the
namespace. Used by UI if available.",
                        "maxLength": 80,
```

```
                        "type": "string"
                    },
                    "namespace": {
                        "description": "The unique namespace text.",
                        "maxLength": 80,
                        "type": "string"
                    },
                    "objects": {
                        "items": {
                            "properties": {
                                "description": {
                                    "type": "string"
                                },
                                "name": {
                                    "type": "string"
                                },
                                "properties": {
                                    "$ref": "#/definitions/property"
                                },
                                "required": {
                                    "$ref": "#/definitions/stringArray"
                                }
                            },
                            "type": "object"
                        },
                        "type": "array"
                    },
                    "owner": {
                        "description": "Owner of the namespace.",
                        "maxLength": 255,
                        "type": "string"
                    },
                    "properties": {
                        "$ref": "#/definitions/property"
                    },
                    "protected": {
                        "description": "If true, namespace will not be
deletable.",
                        "type": "boolean"
                    },
                    "resource_type_associations": {
                        "items": {
                            "properties": {
                                "name": {
                                    "type": "string"
                                },
                                "prefix": {
                                    "type": "string"
                                },
                                "properties_target": {
                                    "type": "string"
                                }
                            },
                            "type": "object"
                        },
                        "type": "array"
                    },
                    "schema": {
                        "type": "string"
                    },
```

```
                    "self": {
                        "type": "string"
                    },
                    "updated_at": {
                        "description": "Date and time of the last namespace
 modification (READ-ONLY)",
                        "format": "date-time",
                        "type": "string"
                    },
                    "visibility": {
                        "description": "Scope of namespace accessibility.",
                        "enum": [
                            "public",
                            "private"
                        ],
                        "type": "string"
                    }
                },
                "required": [
                    "namespace"
                ]
            },
            "type": "array"
        },
        "next": {
            "type": "string"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

6.12.3. Show metadata definition object schema

Method	URI	Description
GET	`/v2/schemas/metadefs/object`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition object entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.3.1. Request

This operation does not accept a request body.

6.12.3.2. Response

Example 6.46. Show metadata definition object schema: JSON response

```
{
    "additionalProperties": false,
    "definitions": {
        "positiveInteger": {
            "minimum": 0,
            "type": "integer"
        },
        "positiveIntegerDefault0": {
            "allOf": [
                {
                    "$ref": "#/definitions/positiveInteger"
                },
                {
                    "default": 0
                }
            ]
        },
        "property": {
            "additionalProperties": {
                "properties": {
                    "additionalItems": {
                        "type": "boolean"
                    },
                    "default": {},
                    "description": {
                        "type": "string"
                    },
                    "enum": {
                        "type": "array"
                    },
                    "items": {
                        "properties": {
                            "enum": {
                                "type": "array"
                            },
                            "type": {
                                "enum": [
                                    "array",
```

```
                                    "boolean",
                                    "integer",
                                    "number",
                                    "object",
                                    "string",
                                    null
                                ],
                                "type": "string"
                            }
                        },
                        "type": "object"
                    },
                    "maxItems": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maxLength": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maximum": {
                        "type": "number"
                    },
                    "minItems": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minLength": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minimum": {
                        "type": "number"
                    },
                    "name": {
                        "type": "string"
                    },
                    "pattern": {
                        "format": "regex",
                        "type": "string"
                    },
                    "readonly": {
                        "type": "boolean"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    },
                    "title": {
                        "type": "string"
                    },
                    "type": {
                        "enum": [
                            "array",
                            "boolean",
                            "integer",
                            "number",
                            "object",
                            "string",
                            null
                        ],
                        "type": "string"
                    },
                    "uniqueItems": {
                        "default": false,
```

```
                        "type": "boolean"
                    }
                },
                "required": [
                    "title",
                    "type"
                ],
                "type": "object"
            },
            "type": "object"
        },
        "stringArray": {
            "items": {
                "type": "string"
            },
            "type": "array",
            "uniqueItems": true
        }
    },
    "name": "object",
    "properties": {
        "created_at": {
            "description": "Date and time of object creation (READ-ONLY)",
            "format": "date-time",
            "type": "string"
        },
        "description": {
            "type": "string"
        },
        "name": {
            "type": "string"
        },
        "properties": {
            "$ref": "#/definitions/property"
        },
        "required": {
            "$ref": "#/definitions/stringArray"
        },
        "schema": {
            "type": "string"
        },
        "self": {
            "type": "string"
        },
        "updated_at": {
            "description": "Date and time of the last object modification
 (READ-ONLY)",
            "format": "date-time",
            "type": "string"
        }
    },
    "required": [
        "name"
    ]
}
```

6.12.4. Show metadata definition objects schema

Method	URI	Description
GET	`/v2/schemas/metadefs/objects`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition objects entity.

An objects entity is a container for object entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.4.1. Request

This operation does not accept a request body.

6.12.4.2. Response

Example 6.47. Show metadata definition objects schema: JSON response

```
{
    "definitions": {
        "positiveInteger": {
            "minimum": 0,
            "type": "integer"
        },
        "positiveIntegerDefault0": {
            "allOf": [
                {
                    "$ref": "#/definitions/positiveInteger"
                },
                {
                    "default": 0
                }
            ]
        },
        "property": {
            "additionalProperties": {
                "properties": {
                    "additionalItems": {
                        "type": "boolean"
                    },
                    "default": {},
                    "description": {
                        "type": "string"
                    },
                    "enum": {
                        "type": "array"
                    },
                    "items": {
                        "properties": {
                            "enum": {
                                "type": "array"
                            },
                            "type": {
                                "enum": [
```

```
                                    "array",
                                    "boolean",
                                    "integer",
                                    "number",
                                    "object",
                                    "string",
                                    null
                                ],
                                "type": "string"
                            }
                        },
                        "type": "object"
                    },
                    "maxItems": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maxLength": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maximum": {
                        "type": "number"
                    },
                    "minItems": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minLength": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minimum": {
                        "type": "number"
                    },
                    "name": {
                        "type": "string"
                    },
                    "pattern": {
                        "format": "regex",
                        "type": "string"
                    },
                    "readonly": {
                        "type": "boolean"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    },
                    "title": {
                        "type": "string"
                    },
                    "type": {
                        "enum": [
                            "array",
                            "boolean",
                            "integer",
                            "number",
                            "object",
                            "string",
                            null
                        ],
                        "type": "string"
                    },
                    "uniqueItems": {
```

```
                        "default": false,
                        "type": "boolean"
                    }
                },
                "required": [
                    "title",
                    "type"
                ],
                "type": "object"
            },
            "type": "object"
        },
        "stringArray": {
            "items": {
                "type": "string"
            },
            "type": "array",
            "uniqueItems": true
        }
    },
    "links": [
        {
            "href": "{first}",
            "rel": "first"
        },
        {
            "href": "{next}",
            "rel": "next"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "name": "objects",
    "properties": {
        "first": {
            "type": "string"
        },
        "next": {
            "type": "string"
        },
        "objects": {
            "items": {
                "additionalProperties": false,
                "name": "object",
                "properties": {
                    "created_at": {
                        "description": "Date and time of object creation
(READ-ONLY)",
                        "format": "date-time",
                        "type": "string"
                    },
                    "description": {
                        "type": "string"
                    },
                    "name": {
                        "type": "string"
                    },
                    "properties": {
```

```
                        "$ref": "#/definitions/property"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    },
                    "schema": {
                        "type": "string"
                    },
                    "self": {
                        "type": "string"
                    },
                    "updated_at": {
                        "description": "Date and time of the last object
 modification (READ-ONLY)",
                        "format": "date-time",
                        "type": "string"
                    }
                },
                "required": [
                    "name"
                ]
            },
            "type": "array"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

6.12.5. Show metadata definition property schema

Method	URI	Description
GET	`/v2/schemas/metadefs/property`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition property entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.5.1. Request

This operation does not accept a request body.

6.12.5.2. Response

Example 6.48. Show metadata definition property schema: JSON response

```
{
    "additionalProperties": false,
    "definitions": {
        "positiveInteger": {
            "minimum": 0,
            "type": "integer"
        },
        "positiveIntegerDefault0": {
            "allOf": [
                {
                    "$ref": "#/definitions/positiveInteger"
                },
                {
                    "default": 0
                }
            ]
        },
        "stringArray": {
            "items": {
                "type": "string"
            },
            "minItems": 1,
            "type": "array",
            "uniqueItems": true
        }
    },
    "name": "property",
    "properties": {
        "additionalItems": {
            "type": "boolean"
        },
        "default": {},
        "description": {
            "type": "string"
        },
        "enum": {
            "type": "array"
        },
```

```
        "items": {
            "properties": {
                "enum": {
                    "type": "array"
                },
                "type": {
                    "enum": [
                        "array",
                        "boolean",
                        "integer",
                        "number",
                        "object",
                        "string",
                        null
                    ],
                    "type": "string"
                }
            },
            "type": "object"
        },
        "maxItems": {
            "$ref": "#/definitions/positiveInteger"
        },
        "maxLength": {
            "$ref": "#/definitions/positiveInteger"
        },
        "maximum": {
            "type": "number"
        },
        "minItems": {
            "$ref": "#/definitions/positiveIntegerDefault0"
        },
        "minLength": {
            "$ref": "#/definitions/positiveIntegerDefault0"
        },
        "minimum": {
            "type": "number"
        },
        "name": {
            "type": "string"
        },
        "pattern": {
            "format": "regex",
            "type": "string"
        },
        "readonly": {
            "type": "boolean"
        },
        "required": {
            "$ref": "#/definitions/stringArray"
        },
        "title": {
            "type": "string"
        },
        "type": {
            "enum": [
                "array",
                "boolean",
                "integer",
                "number",
```

```
                "object",
                "string",
                null
            ],
            "type": "string"
        },
        "uniqueItems": {
            "default": false,
            "type": "boolean"
        }
    },
    "required": [
        "type",
        "title",
        "name"
    ]
}
```

6.12.6. Show metadata definition properties schema

Method	URI	Description
GET	`/v2/schemas/metadefs/properties`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition properties entity.

A properties entity is a container for property entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.6.1. Request

This operation does not accept a request body.

6.12.6.2. Response

Example 6.49. Show metadata definition properties schema: JSON response

```
{
    "definitions": {
        "positiveInteger": {
            "minimum": 0,
            "type": "integer"
        },
        "positiveIntegerDefault0": {
            "allOf": [
                {
                    "$ref": "#/definitions/positiveInteger"
                },
                {
                    "default": 0
                }
            ]
        },
        "stringArray": {
            "items": {
                "type": "string"
            },
            "minItems": 1,
            "type": "array",
            "uniqueItems": true
        }
    },
    "links": [
        {
            "href": "{first}",
            "rel": "first"
        },
        {
            "href": "{next}",
            "rel": "next"
        },
        {
            "href": "{schema}",
```

```
            "rel": "describedby"
        }
    ],
    "name": "properties",
    "properties": {
        "first": {
            "type": "string"
        },
        "next": {
            "type": "string"
        },
        "properties": {
            "additionalProperties": {
                "additionalProperties": false,
                "name": "property",
                "properties": {
                    "additionalItems": {
                        "type": "boolean"
                    },
                    "default": {},
                    "description": {
                        "type": "string"
                    },
                    "enum": {
                        "type": "array"
                    },
                    "items": {
                        "properties": {
                            "enum": {
                                "type": "array"
                            },
                            "type": {
                                "enum": [
                                    "array",
                                    "boolean",
                                    "integer",
                                    "number",
                                    "object",
                                    "string",
                                    null
                                ],
                                "type": "string"
                            }
                        },
                        "type": "object"
                    },
                    "maxItems": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maxLength": {
                        "$ref": "#/definitions/positiveInteger"
                    },
                    "maximum": {
                        "type": "number"
                    },
                    "minItems": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
                    },
                    "minLength": {
                        "$ref": "#/definitions/positiveIntegerDefault0"
```

```
                    },
                    "minimum": {
                        "type": "number"
                    },
                    "name": {
                        "type": "string"
                    },
                    "pattern": {
                        "format": "regex",
                        "type": "string"
                    },
                    "readonly": {
                        "type": "boolean"
                    },
                    "required": {
                        "$ref": "#/definitions/stringArray"
                    },
                    "title": {
                        "type": "string"
                    },
                    "type": {
                        "enum": [
                            "array",
                            "boolean",
                            "integer",
                            "number",
                            "object",
                            "string",
                            null
                        ],
                        "type": "string"
                    },
                    "uniqueItems": {
                        "default": false,
                        "type": "boolean"
                    }
                },
                "required": [
                    "type",
                    "title"
                ]
            },
            "type": "object"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

6.12.7. Show metadata definition tag schema

Method	URI	Description
GET	`/v2/schemas/metadefs/tag`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tag entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.7.1. Request

This operation does not accept a request body.

6.12.7.2. Response

Example 6.50. Show metadata definition tag schema: JSON response

```
{
    "additionalProperties": false,
    "required": [
        "name"
    ],
    "name": "tag",
    "properties": {
        "created_at": {
            "type": "string",
            "description": "Date and time of tag creation (READ-ONLY)",
            "format": "date-time"
        },
        "name": {
            "type": "string"
        },
        "updated_at": {
            "type": "string",
            "description": "Date and time of the last tag modification (READ-
ONLY)",
            "format": "date-time"
        }
    }
}
```

6.12.8. Show metadata definition tags schema

Method	URI	Description
GET	`/v2/schemas/metadefs/tags`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition tags entity.

A tags entity is a container for tag entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.8.1. Request

This operation does not accept a request body.

6.12.8.2. Response

Example 6.51. Show metadata definition tags schema: JSON response

```
{
    "name": "tags",
    "links": [
        {
            "href": "{first}",
            "rel": "first"
        },
        {
            "href": "{next}",
            "rel": "next"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "properties": {
        "next": {
            "type": "string"
        },
        "schema": {
            "type": "string"
        },
        "first": {
            "type": "string"
        },
        "tags": {
            "items": {
                "additionalProperties": false,
                "required": [
                    "name"
                ],
                "name": "tag",
                "properties": {
                    "created_at": {
```

```
                        "type": "string",
                        "description": "Date and time of tag creation (READ-
ONLY)",
                        "format": "date-time"
                    },
                    "name": {
                        "type": "string"
                    },
                    "updated_at": {
                        "type": "string",
                        "description": "Date and time of the last tag
 modification (READ-ONLY)",
                        "format": "date-time"
                    }
                }
            },
            "type": "array"
        }
    }
}
```

6.12.9. Show metadata definition namespace resource type association schema

Method	URI	Description
GET	`/v2/schemas/metadefs/resource_type`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type association entity.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.9.1. Request

This operation does not accept a request body.

6.12.9.2. Response

Example 6.52. Show metadata definition namespace resource type association schema: JSON response

```
{
    "additionalProperties": false,
    "name": "resource_type_association",
    "properties": {
        "created_at": {
            "description": "Date and time of resource type association (READ-
ONLY)",
            "format": "date-time",
            "type": "string"
        },
        "name": {
            "description": "Resource type names should be aligned with Heat
 resource types whenever possible: http://docs.openstack.org/developer/heat/
template_guide/openstack.html",
            "maxLength": 80,
            "type": "string"
        },
        "prefix": {
            "description": "Specifies the prefix to use for the given resource
 type. Any properties in the namespace should be prefixed with this prefix
 when being applied to the specified resource type. Must include prefix
 separator (e.g. a colon :). It is important to note that the same base
 property key can require different prefixes depending on the target resource
 type. For example: The desired virtual CPU topology can be set on both images
 and flavors via metadata. The keys have different prefixes on images than on
 flavors On flavors keys are prefixed with 'hw:', but on images the keys are
 prefixed with 'hw_'.",
            "maxLength": 80,
            "type": "string"
        },
        "properties_target": {
            "description": "Some resource types allow more than one key /
 value pair per instance.  For example, Cinder allows user and image
```

```
 metadata on volumes. Only the image properties metadata is evaluated by Nova
 (scheduling or drivers). This property allows a namespace target to remove
 the ambiguity.",
            "maxLength": 80,
            "type": "string"
        },
        "updated_at": {
            "description": "Date and time of the last resource type
 association modification (READ-ONLY)",
            "format": "date-time",
            "type": "string"
        }
    },
    "required": [
        "name"
    ]
}
```

6.12.10. Show metadata definition namespace resource type associations schema

Method	URI	Description
GET	`/v2/schemas/metadefs/resource_types`	(Since API v2.1) Shows a JSON schema document that represents a metadata definition namespace resource type associations entity.

A resource type associations entity is a container for resource type association entities.

The following schema document is an example. The authoritative response is the actual response to the API call.

Normal response codes: 200

6.12.10.1. Request

This operation does not accept a request body.

6.12.10.2. Response

Example 6.53. Show metadata definition namespace resource type associations schema: JSON response

```
{
    "links": [
        {
            "href": "{first}",
            "rel": "first"
        },
        {
            "href": "{next}",
            "rel": "next"
        },
        {
            "href": "{schema}",
            "rel": "describedby"
        }
    ],
    "name": "resource_type_associations",
    "properties": {
        "first": {
            "type": "string"
        },
        "next": {
            "type": "string"
        },
        "resource_type_associations": {
            "items": {
                "additionalProperties": false,
                "name": "resource_type_association",
                "properties": {
                    "created_at": {
```

```
                        "description": "Date and time of resource type
 association (READ-ONLY)",
                        "format": "date-time",
                        "type": "string"
                    },
                    "name": {
                        "description": "Resource type names should be aligned
 with Heat resource types whenever possible: http://docs.openstack.org/
developer/heat/template_guide/openstack.html",
                        "maxLength": 80,
                        "type": "string"
                    },
                    "prefix": {
                        "description": "Specifies the prefix to use for the
 given resource type. Any properties in the namespace should be prefixed with
 this prefix when being applied to the specified resource type. Must include
 prefix separator (e.g. a colon :). It is important to note that the same base
 property key can require different prefixes depending on the target resource
 type. For example: The desired virtual CPU topology can be set on both images
 and flavors via metadata. The keys have different prefixes on images than on
 flavors On flavors keys are prefixed with 'hw:', but on images the keys are
 prefixed with 'hw_'.",
                        "maxLength": 80,
                        "type": "string"
                    },
                    "properties_target": {
                        "description": "Some resource types allow more than
 one key / value pair per instance.  For example, Cinder allows user and image
 metadata on volumes. Only the image properties metadata is evaluated by Nova
 (scheduling or drivers). This property allows a namespace target to remove
 the ambiguity.",
                        "maxLength": 80,
                        "type": "string"
                    },
                    "updated_at": {
                        "description": "Date and time of the last resource
 type association modification (READ-ONLY)",
                        "format": "date-time",
                        "type": "string"
                    }
                },
                "required": [
                    "name"
                ]
            },
            "type": "array"
        },
        "schema": {
            "type": "string"
        }
    }
}
```

6.13. Tasks (since API v2.2) (tasks)

Creates, lists, and shows details for tasks.

Method	URI	Description
POST	/v2/tasks	Creates a task.

Method	URI	Description
GET	`/v2/tasks{?sort_dir,sort_key,type,status}`	Lists tasks.
GET	`/v2/tasks/{task_id}`	Shows details for a task.

6.13.1. Create task

Method	URI	Description
POST	`/v2/tasks`	Creates a task.

Normal response codes: 201

6.13.1.1. Request

Example 6.54. Create task: JSON request

```
{
    "type": "import",
    "input": {
        "import_from": "http://example.com",
        "import_from_format": "qcow2",
        "image_properties": {
            "disk_format": "vhd",
            "container_format": "ovf"
        }
    }
}
```

6.13.1.2. Response

Example 6.55. Create task: JSON response

```
{
    "id": "e7e59ff6-fa2e-4075-87d3-1a1398a07dc3",
    "type": "import",
    "status": "pending"
}
```

6.13.2. List tasks

Method	URI	Description
GET	`/v2/tasks{?sort_dir,sort_key,type,status}`	Lists tasks.

Normal response codes: 200

6.13.2.1. Request

This table shows the query parameters for the list tasks request:

Name	Type	Description
`sort_dir`	String *(Optional)*	Sorts the response by a set of one or more sort direction and attribute (`sort_key`) combinations. A valid value for the sort direction is `asc` (ascending) or `desc` (descending). If you omit the sort direction in a set, the default is `desc`.
`sort_key`	String *(Optional)*	Sorts the response by an attribute, such as `name`, `id`, or `updated_at`. Default is `created_at`. The API uses the natural sorting direction of the `sort_key` image attribute.
`type`	String *(Optional)*	Filters the response by a task type. A valid value is `import`.
`status`	String *(Optional)*	Filters the response by a task status. A valid value is `pending`, `processing`, `success`, or `failure`.

This operation does not accept a request body.

6.13.2.2. Response

Example 6.56. List tasks: JSON response

```
{
    "tasks": [
        {
            "id": "cbc36478b0bd8e67e89469c7749d4127",
            "type": "import",
            "status": "pending"
        },
        {
            "id": "bbc36578b0bd8e67e89469c7749d4126",
            "type": "import",
            "status": "processing"
        }
    ]
}
```

6.13.3. Show task details

Method	URI	Description
GET	`/v2/tasks/{task_id}`	Shows details for a task.

Normal response codes: 200

6.13.3.1. Request

This table shows the URI parameters for the show task details request:

Name	Type	Description
`{task_id}`	UUID	The UUID of the task.

This operation does not accept a request body.

6.13.3.2. Response

Example 6.57. Show task details: JSON response

```
{
    "id": "e7e59ff6-fa2e-4075-87d3-1a1398a07dc3",
    "type": "import",
    "status": "pending"
}
```

7. Image service API v1 (SUPPORTED)

Loads images for use at launch time by the Compute API. Also, assigns metadata to images.

Some cloud implementations do not expose this API and offer pretested images only.

Cloud providers can configure property protections that prevent non-administrative users from updating and deleting protected properties. For more information, see Image property protection in the *OpenStack Cloud Administrator Guide*.

Method	URI	Description
API versions		
GET	`/`	Lists information about all Image service API versions.
GET	`/versions`	Shows details for the Image service API v1.
Images		
GET	`/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}`	Lists all public VM images.
POST	`/v1/images`	Registers a virtual machine (VM) image.
GET	`/v1/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}`	Lists all available images with details.
GET	`/v1/images/{image_id}`	Shows the image details as headers and the image binary in the body of the response.
HEAD	`/v1/images/{image_id}`	Shows the image metadata information in the body of the response.
PUT	`/v1/images/{image_id}`	Updates an image, uploads an image file, or updates metadata for an image.
DELETE	`/v1/images/{image_id}`	Deletes an image.
Members		
PUT	`/v1/images/{image_id}/members`	Replaces a membership list for an image.
PUT	`/v1/images/{image_id}/members/{owner_id}`	Adds one or more members to an image.
DELETE	`/v1/images/{image_id}/members/{owner_id}`	Removes a member from an image.
Shared images		
GET	`/v1/shared-images/{owner_id}`	Lists the VM images that an owner shares. The owner ID is the tenant ID.

7.1. API versions

Method	URI	Description
GET	`/`	Lists information about all Image service API versions.
GET	`/versions`	Shows details for the Image service API v1.

7.1.1. List API versions

Method	URI	Description
GET	/	Lists information about all Image service API versions.

Normal response codes: 300

7.1.1.1. Request

This operation does not accept a request body.

7.1.2. List API versions

Method	URI	Description
GET	`/versions`	Shows details for the Image service API v1.

Normal response codes: 200

7.1.2.1. Request

This operation does not accept a request body.

7.1.2.2. Response

Example 7.1. List API versions: JSON response

```
{
    "versions": [
        {
            "status": "EXPERIMENTAL",
            "id": "v3.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v3/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "CURRENT",
            "id": "v2.3",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v2/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "SUPPORTED",
            "id": "v2.2",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v2/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "SUPPORTED",
            "id": "v2.1",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v2/",
                    "rel": "self"
                }
            ]
```

```
        },
        {
            "status": "SUPPORTED",
            "id": "v2.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v2/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "SUPPORTED",
            "id": "v1.1",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v1/",
                    "rel": "self"
                }
            ]
        },
        {
            "status": "SUPPORTED",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:9292/v1/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

7.2. Images

Method	URI	Description
GET	`/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}`	Lists all public VM images.
POST	`/v1/images`	Registers a virtual machine (VM) image.
GET	`/v1/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}`	Lists all available images with details.
GET	`/v1/images/{image_id}`	Shows the image details as headers and the image binary in the body of the response.
HEAD	`/v1/images/{image_id}`	Shows the image metadata information in the body of the response.
PUT	`/v1/images/{image_id}`	Updates an image, uploads an image file, or updates metadata for an image.
DELETE	`/v1/images/{image_id}`	Deletes an image.

7.2.1. List images

Method	URI	Description
GET	`/v1/images{?name,container_format,disk_format,status,size_min,size_max,changes-since}`	Lists all public VM images.

Normal response codes: 200

7.2.1.1. Request

This table shows the query parameters for the list images request:

Name	Type	Description
`name`	String *(Optional)*	Filters the image list by an image name, in string format.
`container_format`	String *(Optional)*	Filters the image list by a container format. A valid value is `aki`, `ami`, `ari`, `bare`, `docker`, `ova`, or `ovf`.
`disk_format`	String *(Optional)*	Filters the image list by a disk format. A valid value is `aki`, `ami`, `ari`, `iso`, `qcow2`, `raw`, `vhd`, `vdi`, or `vmdk`.
`status`	String *(Optional)*	Filters the image list by a status. A valid value is `queued`, `saving`, `active`, `killed`, `deleted`, or `pending_delete`.
`size_min`	String *(Optional)*	Filters the image list by a minimum image size, in bytes.
`size_max`	String *(Optional)*	Filters the image list by a maximum image size, in bytes.
`changes-since`	DateTime *(Optional)*	Filters the image list to those images that have changed since a time stamp value.

This operation does not accept a request body.

7.2.1.2. Response

Example 7.2. List images: JSON response

```
{
    "images": [
        {
            "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-
e12490b328d9",
            "name": "Ubuntu 10.04 Plain",
            "disk_format": "vhd",
            "container_format": "ovf",
            "size": "5368709120"
        },
        {
            "...": "..."
        }
    ]
```

```
}
```

7.2.2. Create image

Method	URI	Description
POST	`/v1/images`	Registers a virtual machine (VM) image.

Normal response codes: 202

7.2.2.1. Request

This operation does not accept a request body.

7.2.2.2. Response

This operation does not return a response body.

7.2.3. List images with details

Method	URI	Description
GET	`/v1/images/detail{?name,container_format,disk_format,status,size_min,size_max,changes-since}`	Lists all available images with details.

Normal response codes: 200

7.2.3.1. Request

This table shows the query parameters for the list images with details request:

Name	Type	Description
`name`	String *(Optional)*	Filters the image list by an image name, in string format.
`container_format`	String *(Optional)*	Filters the image list by a container format. A valid value is `aki`, `ami`, `ari`, `bare`, `docker`, `ova`, or `ovf`.
`disk_format`	String *(Optional)*	Filters the image list by a disk format. A valid value is `aki`, `ami`, `ari`, `iso`, `qcow2`, `raw`, `vhd`, `vdi`, or `vmdk`.
`status`	String *(Optional)*	Filters the image list by a status. A valid value is `queued`, `saving`, `active`, `killed`, `deleted`, or `pending_delete`.
`size_min`	String *(Optional)*	Filters the image list by a minimum image size, in bytes.
`size_max`	String *(Optional)*	Filters the image list by a maximum image size, in bytes.
`changes-since`	DateTime *(Optional)*	Filters the image list to those images that have changed since a time stamp value.

This operation does not accept a request body.

7.2.3.2. Response

Example 7.3. List images with details: JSON response

```
{
    "images": [
        {
            "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-
e12490b328d9",
            "name": "Ubuntu 10.04 Plain 5GB",
            "disk_format": "vhd",
            "container_format": "ovf",
            "size": "5368709120",
            "checksum": "c2e5db72bd7fd153f53ede5da5a06de3",
            "created_at": "2010-02-03 09:34:01",
            "updated_at": "2010-02-03 09:34:01",
            "deleted_at": "",
```

```
            "status": "active",
            "is_public": true,
            "min_ram": 256,
            "min_disk": 5,
            "owner": null,
            "properties": {
                "distro": "Ubuntu 10.04 LTS"
            }
        },
        {
            "...": "..."
        }
    ]
}
```

7.2.4. Show image details and image binary

Method	URI	Description
GET	`/v1/images/{image_id}`	Shows the image details as headers and the image binary in the body of the response.

Normal response codes: 200

Error response codes: forbidden (403)

7.2.4.1. Request

This table shows the URI parameters for the show image details and image binary request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

7.2.4.2. Response

Example 7.4. Show image details and image binary: JSON response

```
{
    "images": [
        {
            "uri": "http://glance.example.com/images/71c675ab-d94f-49cd-a114-
e12490b328d9",
            "name": "Ubuntu 10.04 Plain 5GB",
            "disk_format": "vhd",
            "container_format": "ovf",
            "size": "5368709120",
            "checksum": "c2e5db72bd7fd153f53ede5da5a06de3",
            "created_at": "2010-02-03 09:34:01",
            "updated_at": "2010-02-03 09:34:01",
            "deleted_at": "",
            "status": "active",
            "is_public": true,
            "min_ram": 256,
            "min_disk": 5,
            "owner": null,
            "properties": {
                "distro": "Ubuntu 10.04 LTS"
            }
        },
        {
            "...": "..."
        }
    ]
}
```

7.2.5. Show image metadata

Method	URI	Description
HEAD	`/v1/images/{image_id}`	Shows the image metadata information in the body of the response.

The Image system does not return a response body for the **HEAD** operation.

Example requests and responses:

- Show image metadata:

```
http://glance.example.com/v1/images/03bc0a8b-659c-4de9-b6bd-13c6e86e6455
```

```
X-Image-Meta-Checksum # 8a40c862b5735975d82605c1dd395796
X-Image-Meta-Container_format # aki
X-Image-Meta-Created_at # 2016-01-06T03:22:20.000000
X-Image-Meta-Deleted # false
X-Image-Meta-Disk_format # aki
X-Image-Meta-Id # 03bc0a8b-659c-4de9-b6bd-13c6e86e6455
X-Image-Meta-Is_public # true
X-Image-Meta-Min_disk # 0
X-Image-Meta-Min_ram # 0
X-Image-Meta-Name # cirros-0.3.4-x86_64-uec-kernel
X-Image-Meta-Owner # 13cc6052265b41529e2fd0fc461fa8ef
X-Image-Meta-Protected # false
X-Image-Meta-Size # 4979632
X-Image-Meta-Status # deactivated
X-Image-Meta-Updated_at # 2016-02-25T03:02:05.000000
X-Openstack-Request-Id # req-d5208320-28ed-4c22-b628-12dc6456d983
```

If the request succeeds, the operation returns the `200` response code.

If there is an image size mismatch detected with the `X-Image-Meta-Size`, the operation returns a `409` response code.

Normal response codes: 200

Error response codes: itemNotFound (404), 409

7.2.5.1. Request

This table shows the URI parameters for the show image metadata request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

7.2.6. Update image

Method	URI	Description
PUT	`/v1/images/{image_id}`	Updates an image, uploads an image file, or updates metadata for an image.

Normal response codes: 200

7.2.6.1. Request

This table shows the URI parameters for the update image request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

7.2.6.2. Response

This operation does not return a response body.

7.2.7. Delete image

Method	URI	Description
DELETE	`/v1/images/{image_id}`	Deletes an image.

Normal response codes: 204

7.2.7.1. Request

This table shows the URI parameters for the delete image request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.

This operation does not accept a request body.

7.3. Members

Method	URI	Description
PUT	`/v1/images/{image_id}/members`	Replaces a membership list for an image.
PUT	`/v1/images/{image_id}/members/{owner_id}`	Adds one or more members to an image.
DELETE	`/v1/images/{image_id}/members/{owner_id}`	Removes a member from an image.

7.3.1. Replace member

Method	URI	Description
PUT	`/v1/images/{image_id}/members`	Replaces a membership list for an image.

Normal response codes: 204

7.3.1.1. Request

This table shows the URI parameters for the replace member request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.

7.3.2. Add members to image

Method	URI	Description
PUT	`/v1/images/{image_id}/members/{owner_id}`	Adds one or more members to an image.

If you omit the request body, this call adds the membership to the image, leaves the existing memberships unmodified, and sets the `can_share` attribute to `false` for new memberships.

Normal response codes: 204

7.3.2.1. Request

This table shows the URI parameters for the add members to image request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.
`{owner_id}`	String	Owner ID, which is the tenant ID.

Example 7.5. Add members to image: JSON request

```
{
    "members": [
        {
            "member_id": "tenant1",
            "can_share": false
        },
        {
            "member_id": "tenant2",
            "can_share": false
        }
    ]
}
```

7.3.3. Remove member

Method	URI	Description
DELETE	`/v1/images/{image_id}/members/{owner_id}`	Removes a member from an image.

Normal response codes: 204

7.3.3.1. Request

This table shows the URI parameters for the remove member request:

Name	Type	Description
`{image_id}`	String	Image ID stored through the image API. Typically a UUID.
`{owner_id}`	String	Owner ID, which is the tenant ID.

This operation does not accept a request body.

7.4. Shared images

Method	URI	Description
GET	`/v1/shared-images/{owner_id}`	Lists the VM images that an owner shares. The owner ID is the tenant ID.

7.4.1. List shared images

Method	URI	Description
GET	`/v1/shared-images/{owner_id}`	Lists the VM images that an owner shares. The owner ID is the tenant ID.

Normal response codes: 200

7.4.1.1. Request

This table shows the URI parameters for the list shared images request:

Name	Type	Description
`{owner_id}`	String	Owner ID, which is the tenant ID.

This operation does not accept a request body.

7.4.1.2. Response

Example 7.6. List shared images: JSON response

```
{
    "shared_images": [
        {
            "image_id": "71c675ab-d94f-49cd-a114-e12490b328d9",
            "can_share": false
        },
        {
            "...": "..."
        }
    ]
}
```

8. Object Storage API v1 (SUPPORTED)

Manages the accounts, containers, and objects in the Object Storage system.

To run the cURL command examples for these requests, set these environment variables:

- `publicURL`. The public URL that is the HTTP endpoint from where you can access Object Storage. It includes the Object Storage API version number and your account name. For example, `https://23.253.72.207/v1/my_account`.
- `token`. The authentication token for Object Storage.

To obtain these values, run the **swift stat -v** command.

As shown in this example, the public URL appears in the `StorageURL` field, and the token appears in the `Auth Token` field:

```
StorageURL: https://23.253.72.207/v1/my_account
Auth Token: {token}
Account: my_account
Containers: 2
Objects: 3
Bytes: 47
Meta Book: MobyDick
X-Timestamp: 1389453423.35964
X-Trans-Id: txee55498935404a2caad89-0052dd3b77
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
```

For a complete description of HTTP 1.1 header definitions, see Header Field Definitions.

The API returns the `Range Not Satisfiable (416)` response code for any ranged **GET** requests that specify more than:

- Fifty ranges.
- Three overlapping ranges.
- Eight non-increasing ranges.

Method	URI	Description
Discoverability		
GET	`/info{?swiftinfo_sig, swiftinfo_expires}`	Lists the activated capabilities for this version of the OpenStack Object Storage API.
Endpoints		
GET	`/v1/endpoints`	Lists endpoints for an object, account, or container.
Accounts		
GET	`/v1/{account}{?limit,marker, end_marker,format,prefix,delim-iter}`	Shows details for an account and lists containers, sorted by name, in the account.
POST	`/v1/{account}`	Creates, updates, or deletes account metadata.
HEAD	`/v1/{account}`	Shows metadata for an account.
Containers		

Method	URI	Description
GET	`/v1/{account}/{container}{?limit,marker,end_marker,prefix,format,delimiter,path}`	Shows details for a container and lists objects, sorted by name, in the container.
PUT	`/v1/{account}/{container}`	Creates a container.
DELETE	`/v1/{account}/{container}`	Deletes an empty container.
POST	`/v1/{account}/{container}`	Creates, updates, or deletes custom metadata for a container.
HEAD	`/v1/{account}/{container}`	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.
Objects		
GET	`/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename,multipart-manifest}`	Downloads the object content and gets the object metadata.
PUT	`/v1/{account}/{container}/{object}{?multipart-manifest,temp_url_sig,temp_url_expires,filename}`	Creates an object with data content and metadata, or replaces an existing object with data content and metadata.
COPY	`/v1/{account}/{container}/{object}`	Copies an object to another object in the object store.
DELETE	`/v1/{account}/{container}/{object}{?multipart-manifest}`	Permanently deletes an object from the object store.
HEAD	`/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename}`	Shows object metadata.
POST	`/v1/{account}/{container}/{object}`	Creates or updates object metadata.

8.1. Discoverability

If configured, lists the activated capabilities for this version of the OpenStack Object Storage API.

Method	URI	Description
GET	`/info{?swiftinfo_sig,swiftinfo_expires}`	Lists the activated capabilities for this version of the OpenStack Object Storage API.

8.1.1. List activated capabilities

Method	URI	Description
GET	`/info{?swiftinfo_sig, swiftinfo_expires}`	Lists the activated capabilities for this version of the OpenStack Object Storage API.

Normal response codes: 200

8.1.1.1. Request

This table shows the query parameters for the list activated capabilities request:

Name	Type	Description
`swiftinfo_sig`	Char *(Optional)*	A hash-based message authentication code (HMAC) that enables access to administrator-only information. To use this parameter, the `swiftinfo_expires` parameter is also required.
`swiftinfo_expires`	Int *(Optional)*	Filters the response by the expiration date and time in UNIX Epoch time stamp format. For example, `1440619048` is equivalent to `Mon, Wed, 26 Aug 2015 19:57:28 GMT`.

This operation does not accept a request body.

8.1.1.2. Response

Example 8.1. List activated capabilities: JSON response

```
{
    "swift": {
        "version": "1.11.0"
    },
    "staticweb": {},
    "tempurl": {}
}
```

8.2. Endpoints

If configured, lists endpoints for an account.

Method	URI	Description
GET	`/v1/endpoints`	Lists endpoints for an object, account, or container.

8.2.1. List endpoints

Method	URI	Description
GET	`/v1/endpoints`	Lists endpoints for an object, account, or container.

When the cloud provider enables middleware to list the `/endpoints/` path, software that needs data location information can use this call to avoid network overhead. The cloud provider can map the `/endpoints/` path to another resource, so this exact resource might vary from provider to provider. Because it goes straight to the middleware, the call is not authenticated, so be sure you have tightly secured the environment and network when using this call.

Normal response codes: 201

8.2.1.1. Request

This operation does not accept a request body.

8.2.1.2. Response

Example 8.2. List endpoints: JSON response

```
{
    "endpoints": [
        "http://storage02.swiftdrive:6002/d2/617/AUTH_dev",
        "http://storage01.swiftdrive:6002/d8/617/AUTH_dev",
        "http://storage01.swiftdrive:6002/d11/617/AUTH_dev"
    ],
    "headers": {}
}
```

Example 8.3. List endpoints: JSON response

```
{
    "endpoints": [
        "http://storage01.swiftdrive.com:6008/d8/583/AUTH_dev/EC_cont1/obj",
        "http://storage02.swiftdrive.com:6008/d2/583/AUTH_dev/EC_cont1/obj",
        "http://storage02.swiftdrive.com:6006/d3/583/AUTH_dev/EC_cont1/obj",
        "http://storage02.swiftdrive.com:6008/d5/583/AUTH_dev/EC_cont1/obj",
        "http://storage01.swiftdrive.com:6007/d7/583/AUTH_dev/EC_cont1/obj",
        "http://storage02.swiftdrive.com:6007/d4/583/AUTH_dev/EC_cont1/obj",
        "http://storage01.swiftdrive.com:6006/d6/583/AUTH_dev/EC_cont1/obj"
    ],
    "headers": {
        "X-Backend-Storage-Policy-Index": "2"
    }
}
```

8.3. Accounts

Lists containers for an account. Creates, updates, shows, and deletes account metadata.

Account metadata operations work differently than container and object metadata operations work. Depending on the contents of your **POST** account metadata request, the Object Storage API updates the metadata in one of these ways:

Table 8.1. Account metadata operations

POST request body contains	Description
A metadata key without a value. The metadata key already exists for the account.	The API removes the metadata item from the account.
A metadata key without a value. The metadata key does not already exist for the account.	The API ignores the metadata key.
A metadata key value. The metadata key already exists for the account.	The API updates the metadata key value for the account.
A metadata key value. The metadata key does not already exist for the account.	The API adds the metadata key and value pair, or item, to the account.
One or more account metadata items are omitted. The metadata items already exist for the account.	The API does not change the existing metadata items.

For these requests, specifying the `X-Remove-Account-Meta-*` request header for the key with any value is equivalent to specifying the `X-Account-Meta-*` request header with an empty value.

Metadata keys must be treated as case-insensitive at all times. These keys can contain ASCII 7-bit characters that are not control (0-31) characters, DEL, or a separator character, according to HTTP/1.1. Also, Object Storage does not support the underscore character, which it silently converts to a hyphen.

The metadata values in Object Storage do not follow HTTP/1.1 rules for character encodings. You must use a UTF-8 encoding to get a byte array for any string that contains characters that are not in the 7-bit ASCII 0-127 range. Otherwise, Object Storage returns the 404 response code for ISO-8859-1 characters in the 128-255 range, which is a direct violation of the HTTP/1.1 basic rules.

Method	URI	Description
GET	`/v1/{account}{?limit,marker,end_marker,format,prefix,delimiter}`	Shows details for an account and lists containers, sorted by name, in the account.
POST	`/v1/{account}`	Creates, updates, or deletes account metadata.
HEAD	`/v1/{account}`	Shows metadata for an account.

8.3.1. Show account details and list containers

Method	URI	Description
GET	`/v1/{account}{?limit,marker,end_marker,format,prefix,delimiter}`	Shows details for an account and lists containers, sorted by name, in the account.

The sort order for the name is based on a binary comparison, a single built-in collating sequence that compares string data by using the SQLite `memcmp()` function, regardless of text encoding. See Collating Sequences.

Example requests and responses:

- Show account details and list containers and ask for a JSON response:

```
curl -i $publicURL?format=json -X GET -H "X-Auth-Token: $token"
```

- List containers and ask for an XML response:

```
curl -i $publicURL?format=xml -X GET -H "X-Auth-Token: $token"
```

The response body returns a list of containers. The default response (`text/plain`) returns one container per line.

If you use query parameters to page through a long list of containers, you have reached the end of the list if the number of items in the returned list is less than the request `limit` value. The list contains more items if the number of items in the returned list equals the `limit` value.

When asking for a list of containers and there are none, the response behavior changes depending on whether the request format is text, JSON, or XML. For a text response, you get a 204, because there is no content. However, for a JSON or XML response, you get a 200 with content indicating an empty array.

If the request succeeds, the operation returns one of these status codes:

- `OK (200)`. Success. The response body lists the containers.
- `No Content (204)`. Success. The response body shows no containers. Either the account has no containers or you are paging through a long list of names by using the `marker`, `limit`, or `end_marker` query parameter and you have reached the end of the list.

Normal response codes: 200204

8.3.1.1. Request

This table shows the header parameters for the show account details and list containers request:

Name	Type	Description
`X-Auth-Token`	String	Authentication token.

Name	Type	Description
	(Required)	
`X-Newest`	Boolean *(Optional)*	If set to `true`, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to `true` is more expensive for the back end, use it only when it is absolutely needed.
`Accept`	String *(Optional)*	Instead of using the `format` query parameter, set this header to `application/json`, `application/xml`, or `text/xml`.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show account details and list containers request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.

This table shows the query parameters for the show account details and list containers request:

Name	Type	Description
`limit`	Int *(Optional)*	For an integer value *n*, limits the number of results to *n*.
`marker`	String *(Optional)*	For a string value, *x*, returns container names that are greater than the marker value.
`end_marker`	String *(Optional)*	For a string value, *x*, returns container names that are less than the marker value.
`format`	String *(Optional)*	The response format. Valid values are `json`, `xml`, or `plain`. The default is `plain`. If you append the `format=xml` or `format=json` query parameter to the storage account URL, the response shows extended container information serialized in that format. If you append the `format=plain` query parameter, the response lists the container names separated by newlines.
`prefix`	String *(Optional)*	Prefix value. Named items in the response begin with this value.

Name	Type	Description
`delimiter`	Char *(Optional)*	Delimiter value, which returns the object names that are nested in the container. If you do not set a prefix and set the delimiter to "/" you may get unexpected results where all the objects are returned instead of only those with the delimiter set.

This operation does not accept a request body.

8.3.1.2. Response

Example 8.4. List containers response: HTTP and JSON

```
HTTP/1.1 200 OK
Content-Length: 96
X-Account-Object-Count: 1
X-Timestamp: 1389453423.35964
X-Account-Meta-Subject: Literature
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: application/json; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: tx274a77a8975c4a66aeb24-0052d95365
Date: Fri, 17 Jan 2014 15:59:33 GMT
```

```
[
    {
        "count": 0,
        "bytes": 0,
        "name": "janeausten"
    },
    {
        "count": 1,
        "bytes": 14,
        "name": "marktwain"
    }
]
```

This table shows the body parameters for the show account details and list containers response:

Name	Type	Description
`name`	String *(Required)*	The name of the container.
`count`	Int *(Required)*	The number of objects in the container.
`bytes`	Int *(Required)*	The total number of bytes that are stored in Object Storage for the account.

Example 8.5. List containers response: HTTP and XML

```
HTTP/1.1 200 OK
Content-Length: 262
X-Account-Object-Count: 1
X-Timestamp: 1389453423.35964
X-Account-Meta-Subject: Literature
```

```
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: application/xml; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: tx69f60bc9f7634a01988e6-0052d9544b
Date: Fri, 17 Jan 2014 16:03:23 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<account name="my_account">
    <container>
        <name>janeausten</name>
        <count>0</count>
        <bytes>0</bytes>
    </container>
    <container>
        <name>marktwain</name>
        <count>1</count>
        <bytes>14</bytes>
    </container>
</account>
```

This operation does not return a response body.

8.3.2. Create, update, or delete account metadata

Method	URI	Description
POST	`/v1/{account}`	Creates, updates, or deletes account metadata.

To create, update, or delete metadata, use the `X-Account-Meta-{name}` request header, where `{name}` is the name of the metadata item.

Subsequent requests for the same key and value pair overwrite the existing value.

To delete a metadata header, send an empty value for that header, such as for the `X-Account-Meta-Book` header. If the tool you use to communicate with Object Storage, such as an older version of cURL, does not support empty headers, send the `X-Remove-Account-Meta-{name}` header with an arbitrary value. For example, `X-Remove-Account-Meta-Book: x`. The operation ignores the arbitrary value.

If the container already has other custom metadata items, a request to create, update, or delete metadata does not affect those items.

This operation does not accept a request body.

Example requests and responses:

- Create account metadata:

```
curl -i $publicURL -X POST -H "X-Auth-Token: $token" -H "X-Account-Meta-
Book: MobyDick" -H "X-Account-Meta-Subject: Literature"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx8c2dd6aee35442a4a5646-0052d954fb
Date: Fri, 17 Jan 2014 16:06:19 GMT
```

- Update account metadata:

```
curl -i $publicURL -X POST -H "X-Auth-Token: $token" -H "X-Account-Meta-
Subject: AmericanLiterature"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx1439b96137364ab581156-0052d95532
Date: Fri, 17 Jan 2014 16:07:14 GMT
```

- Delete account metadata:

```
curl -i $publicURL -X POST -H "X-Auth-Token: $token" -H "X-Remove-Account-
Meta-Subject: x"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
```

```
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx411cf57701424da99948a-0052d9556f
Date: Fri, 17 Jan 2014 16:08:15 GMT
```

If the request succeeds, the operation returns the `No Content (204)` response code.

To confirm your changes, issue a show account metadata request.

Normal response codes: 204

8.3.2.1. Request

This table shows the header parameters for the create, update, or delete account metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.
`X-Account-Meta-Temp-URL-Key`	String *(Optional)*	The secret key value for temporary URLs.
`X-Account-Meta-Temp-URL-Key-2`	String *(Optional)*	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
`X-Account-Meta-name`	String *(Optional)*	The account metadata. The `{name}` is the name of metadata item that you want to add, update, or delete. To delete this item, send an empty value in this header. You must specify an `X-Account-Meta-{name}` header for each metadata item (for each `{name}`) that you want to add, update, or delete.
`Content-Type`	String *(Optional)*	Changes the MIME type for the object.
`X-Detect-Content-Type`	Boolean *(Optional)*	If set to `true`, Object Storage guesses the content type based on the file extension and ignores the value sent in the `Content-Type` header, if present.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create, update, or delete account metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.

This operation does not accept a request body.

8.3.2.2. Response

This operation does not return a response body.

8.3.3. Show account metadata

Method	URI	Description
HEAD	`/v1/{account}`	Shows metadata for an account.

Metadata for the account includes:

- Number of containers
- Number of objects
- Total number of bytes that are stored in Object Storage for the account

Because the storage system can store large amounts of data, take care when you represent the total bytes response as an integer; when possible, convert it to a 64-bit unsigned integer if your platform supports that primitive type.

Do not include metadata headers in this request.

Show account metadata request:

```
curl -i $publicURL -X HEAD -H "X-Auth-Token: $token"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
X-Account-Object-Count: 1
X-Account-Meta-Book: MobyDick
X-Timestamp: 1389453423.35964
X-Account-Bytes-Used: 14
X-Account-Container-Count: 2
Content-Type: text/plain; charset=utf-8
Accept-Ranges: bytes
X-Trans-Id: txafb3504870144b8ca40f7-0052d955d4
Date: Fri, 17 Jan 2014 16:09:56 GMT
```

If the account or authentication token is not valid, the operation returns the `Unauthorized (401)` response code.

Normal response codes: 204

Error response codes: unauthorized (401)

8.3.3.1. Request

This table shows the header parameters for the show account metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.
`X-Newest`	Boolean *(Optional)*	If set to `true`, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to `true` is more expensive for the back end, use it only when it is absolutely needed.

Name	Type	Description
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show account metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.

This operation does not accept a request body.

8.3.3.2. Response

This operation does not return a response body.

8.4. Containers

Lists objects in a container. Creates, shows details for, and deletes containers. Creates, updates, shows, and deletes container metadata.

Method	URI	Description
GET	`/v1/{account}/{container}{?limit,marker,end_marker,prefix,format,delimiter,path}`	Shows details for a container and lists objects, sorted by name, in the container.
PUT	`/v1/{account}/{container}`	Creates a container.
DELETE	`/v1/{account}/{container}`	Deletes an empty container.
POST	`/v1/{account}/{container}`	Creates, updates, or deletes custom metadata for a container.
HEAD	`/v1/{account}/{container}`	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.

8.4.1. Show container details and list objects

Method	URI	Description
GET	`/v1/{account}/{container}{?limit, marker,end_marker,prefix,format, delimiter,path}`	Shows details for a container and lists objects, sorted by name, in the container.

Specify query parameters in the request to filter the list and return a subset of object names. Omit query parameters to return the complete list of object names that are stored in the container, up to 10,000 names. The 10,000 maximum value is configurable. To view the value for the cluster, issue a **GET** `/info` request.

Example requests and responses:

- `OK (200)`. Success. The response body lists the objects.
- `No Content (204)`. Success. The response body shows no objects. Either the container has no objects or you are paging through a long list of names by using the `marker`, `limit`, or `end_marker` query parameter and you have reached the end of the list.

If the container does not exist, the call returns the `Not Found (404)` response code.

The operation returns the `Range Not Satisfiable (416)` response code for any ranged **GET** requests that specify more than:

- Fifty ranges.
- Three overlapping ranges.
- Eight non-increasing ranges.

Normal response codes: 200204

Error response codes: NotFound (404), 416

8.4.1.1. Request

This table shows the header parameters for the show container details and list objects request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.
`X-Newest`	Boolean *(Optional)*	If set to `true`, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to `true` is more expensive for the back end, use it only when it is absolutely needed.
`Accept`	String *(Optional)*	Instead of using the `format` query parameter, set this header to `application/json`, `application/xml`, or `text/xml`.
`X-Container-Meta-Temp-URL-Key`	String *(Optional)*	The secret key value for temporary URLs.

Name	Type	Description
`X-Container-Meta-Temp-URL-Key-2`	String *(Optional)*	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show container details and list objects request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.

This table shows the query parameters for the show container details and list objects request:

Name	Type	Description
`limit`	Int *(Optional)*	For an integer value *n*, limits the number of results to *n*.
`marker`	String *(Optional)*	For a string value, *x*, returns container names that are greater than the marker value.
`end_marker`	String *(Optional)*	For a string value, *x*, returns container names that are less than the marker value.
`prefix`	String *(Optional)*	Prefix value. Named items in the response begin with this value.
`format`	String *(Optional)*	The response format. Valid values are `json`, `xml`, or `plain`. The default is `plain`. If you append the `format=xml` or `format=json` query parameter to the storage account URL, the response shows extended container information serialized in that format. If you append the `format=plain` query parameter, the response lists the container names separated by newlines.

Name	Type	Description
delimiter	Char *(Optional)*	Delimiter value, which returns the object names that are nested in the container. If you do not set a prefix and set the delimiter to "/" you may get unexpected results where all the objects are returned instead of only those with the delimiter set.
path	String *(Optional)*	For a string value, returns the object names that are nested in the pseudo path.

This operation does not accept a request body.

8.4.1.2. Response

Example 8.6. Show container details response: HTTP and JSON

```
HTTP/1.1 200 OK
Content-Length: 341
X-Container-Object-Count: 2
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Bytes-Used: 26
Content-Type: application/json; charset=utf-8
X-Trans-Id: tx26377fe5fab74869825d1-0052d6bdff
Date: Wed, 15 Jan 2014 16:57:35 GMT
```

```
[
    {
        "hash": "451e372e48e0f6b1114fa0724aa79fa1",
        "last_modified": "2014-01-15T16:41:49.390270",
        "bytes": 14,
        "name": "goodbye",
        "content_type": "application/octet-stream"
    },
    {
        "hash": "ed076287532e86365e841e92bfc50d8c",
        "last_modified": "2014-01-15T16:37:43.427570",
        "bytes": 12,
        "name": "helloworld",
        "content_type": "application/octet-stream"
    }
]
```

This table shows the body parameters for the show container details and list objects response:

Name	Type	Description
name	String *(Required)*	The name of the container.
hash	String *(Required)*	The MD5 checksum value of the object content.
bytes	Int *(Required)*	The total number of bytes that are stored in Object Storage for the account.
content_type	String *(Required)*	The content type of the object.

Name	Type	Description
last_modified	DateTime *(Required)*	The date and time when the object was last modified. The date and time stamp format is ISO 8601: `CCYY-MM-DDThh:mm:ss±hh:mm` For example, `2015-08-27T09:49:58-05:00`. The `±hh:mm` value, if included, is the time zone as an offset from UTC. In the previous example, the offset value is `-05:00`.

Example 8.7. Show container details response: HTTP and XML

```
HTTP/1.1 200 OK
Content-Length: 500
X-Container-Object-Count: 2
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Bytes-Used: 26
Content-Type: application/xml; charset=utf-8
X-Trans-Id: txc75ea9a6e66f47d79e0c5-0052d6be76
Date: Wed, 15 Jan 2014 16:59:35 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<container name="marktwain">
    <object>
        <name>goodbye</name>
        <hash>451e372e48e0f6b1114fa0724aa79fa1</hash>
        <bytes>14</bytes>
        <content_type>application/octet-stream</content_type>
        <last_modified>2014-01-15T16:41:49.390270</last_modified>
    </object>
    <object>
        <name>helloworld</name>
        <hash>ed076287532e86365e841e92bfc50d8c</hash>
        <bytes>12</bytes>
        <content_type>application/octet-stream</content_type>
        <last_modified>2014-01-15T16:37:43.427570</last_modified>
    </object>
</container>
```

This operation does not return a response body.

8.4.2. Create container

Method	URI	Description
PUT	`/v1/{account}/{container}`	Creates a container.

You do not need to check whether a container already exists before issuing a **PUT** operation because the operation is idempotent: It creates a container or updates an existing container, as appropriate.

Example requests and responses:

- Create a container with no metadata:

```
curl -i $publicURL/steven -X PUT -H "Content-Length: 0" -H "X-Auth-Token:
 $token"
```

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7f6b7fa09bc2443a94df0-0052d58b56
Date: Tue, 14 Jan 2014 19:09:10 GMT
```

- Create a container with metadata:

```
curl -i $publicURL/marktwain -X PUT -H "X-Auth-Token: $token" -H "X-
Container-Meta-Book: TomSawyer"
```

```
HTTP/1.1 201 Created
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx06021f10fc8642b2901e7-0052d58f37
Date: Tue, 14 Jan 2014 19:25:43 GMT
```

Normal response codes: 201204

8.4.2.1. Request

This table shows the header parameters for the create container request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.

`X-Container-Read`	String
	
	(Optional)
		Sets a container access control list (ACL) that grants read access. Container ACLs are available on any Object Storage cluster, and are enabled by container rather than by cluster.
		
		To set the container read ACL:
		
		`$ curl -X {PUT|POST} -i -H "X-Auth-Token: TOKEN" -H \ "X-Container-Read: ACL" STORAGE_URL/CONTAINER`
		
		For example:
		
		`$ curl -X PUT -i \ -H "X-Auth-Token: 0101010101" \ -H "X-Container-Read: .r:*" \ http://swift.example.com/v1/AUTH_bob/read_container`

Name	Type	Description
		In the command, specify the ACL in the `X-Container-Read` header, as follows: • `.r:*`#All referrers. • `.r:example.com,swift.example.com`#Comma-separated list of referrers. • `.rlistings`#Container listing access. • AUTH_`username`#Access to a user who authenticates through a legacy or non-OpenStack-Identity-based authentication system. • LDAP_#Access to all users who authenticate through an LDAP-based legacy or non-OpenStack-Identity-based authentication system.
`X-Container-Write`	String *(Optional)*	Sets an ACL that grants write access.
`X-Container-Sync-To`	String *(Optional)*	Sets the destination for container synchronization. Used with the secret key indicated in the `X-Container-Sync-Key` header. If you want to stop a container from synchronizing, send a blank value for the `X-Container-Sync-Key` header.
`X-Container-Sync-Key`	String *(Optional)*	Sets the secret key for container synchronization. If you remove the secret key, synchronization is halted.
`X-Versions-Location`	String *(Optional)*	Enables versioning on this container. The value is the name of another container. You must UTF-8-encode and then URL-encode the name before you include it in the header. To disable versioning, set the header to an empty string.
`X-Container-Meta-name`	String *(Optional)*	The container metadata, where `{name}` is the name of metadata item. You must specify an `X-Container-Meta-{name}` header for each metadata item (for each `{name}`) that you want to add or update.
`X-Container-Meta-Access-Control-Allow-Origin`	String *(Optional)*	Originating URLs allowed to make cross-origin requests (CORS), separated by spaces. This heading applies to the container only, and all objects within the container with this header applied are CORS-enabled for the allowed origin URLs. A browser (user-agent) typically issues a preflighted request, which is an OPTIONS call that verifies the origin is allowed to make the request. The Object Storage service returns 200 if the originating URL is listed in this header parameter, and issues a 401 if the originating URL is not allowed to make a cross-origin request. Once a 200 is returned, the browser makes a second request to the Object Storage service to retrieve the CORS-enabled object.
`X-Container-Meta-Access-Control-Max-Age`	String *(Optional)*	Maximum time for the origin to hold the preflight results. A browser may make an OPTIONS call to verify the origin is allowed to make the request. Set the value to an integer number of seconds after the time that the request was received.
`X-Container-Meta-Access-Control-Expose-Headers`	String *(Optional)*	Headers the Object Storage service exposes to the browser (technically, through the `user-agent` setting), in the request response, separated by spaces. By default the Object Storage service returns the following values for this header: • All "simple response headers" as listed on http://www.w3.org/TR/cors/#simple-response-header. • The headers `etag`, `x-timestamp`, `x-trans-id`.

Name	Type	Description
		• All metadata headers (`X-Container-Meta-*` for containers and `X-Object-Meta-*` for objects) headers listed in `X-Container-Meta-Access-Control-Expose-Headers`.
`Content-Type`	String *(Optional)*	Changes the MIME type for the object.
`X-Detect-Content-Type`	Boolean *(Optional)*	If set to `true`, Object Storage guesses the content type based on the file extension and ignores the value sent in the `Content-Type` header, if present.
`X-Container-Meta-Temp-URL-Key`	String *(Optional)*	The secret key value for temporary URLs.
`X-Container-Meta-Temp-URL-Key-2`	String *(Optional)*	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create container request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.

This operation does not accept a request body.

8.4.2.2. Response

This operation does not return a response body.

8.4.3. Delete container

Method	URI	Description
DELETE	`/v1/{account}/{container}`	Deletes an empty container.

This operation fails unless the container is empty. An empty container has no objects.

Delete the `steven` container:

```
curl -i $publicURL/steven -X DELETE -H "X-Auth-Token: $token"
```

If the container does not exist, the response is:

```
HTTP/1.1 404 Not Found
Content-Length: 70
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d728126b17b43b598bf7-0052d81e34
Date: Thu, 16 Jan 2014 18:00:20 GMT
```

If the container exists and the deletion succeeds, the response is:

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txf76c375ebece4df19c84c-0052d81f14
Date: Thu, 16 Jan 2014 18:04:04 GMT
```

If the container exists but is not empty, the response is:

```
HTTP/1.1 409 Conflict
Content-Length: 95
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7782dc6a97b94a46956b5-0052d81f6b
Date: Thu, 16 Jan 2014 18:05:31 GMT

<html><h1>Conflict</h1><p>There was a conflict when trying to complete your
 request.</p></html>
```

Normal response codes: 204

Error response codes: NotFound (404), Conflict (409)

8.4.3.1. Request

This table shows the header parameters for the delete container request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.
`X-Container-Meta-Temp-URL-Key`	String *(Optional)*	The secret key value for temporary URLs.
`X-Container-Meta-Temp-URL-Key-2`	String *(Optional)*	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.

Name	Type	Description
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the delete container request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.

This operation does not accept a request body.

8.4.3.2. Response

This operation does not return a response body.

8.4.4. Create, update, or delete container metadata

Method	URI	Description
POST	`/v1/{account}/{container}`	Creates, updates, or deletes custom metadata for a container.

To create, update, or delete a custom metadata item, use the `X-Container-Meta-{name}` header, where `{name}` is the name of the metadata item.

Subsequent requests for the same key and value pair overwrite the previous value.

To delete container metadata, send an empty value for that header, such as for the `X-Container-Meta-Book` header. If the tool you use to communicate with Object Storage, such as an older version of cURL, does not support empty headers, send the `X-Remove-Container-Meta-{name}` header with an arbitrary value. For example, `X-Remove-Container-Meta-Book: x`. The operation ignores the arbitrary value.

If the container already has other custom metadata items, a request to create, update, or delete metadata does not affect those items.

Example requests and responses:

- Create container metadata:

```
curl -i $publicURL/marktwain -X POST -H "X-Auth-Token: $token" -H "X-
Container-Meta-Author: MarkTwain" -H "X-Container-Meta-Web-Directory-Type:
 text/directory" -H "X-Container-Meta-Century: Nineteenth"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx05dbd434c651429193139-0052d82635
Date: Thu, 16 Jan 2014 18:34:29 GMT
```

- Update container metadata:

```
curl -i $publicURL/marktwain -X POST -H "X-Auth-Token: $token" -H "X-
Container-Meta-Author: SamuelClemens"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txe60c7314bf614bb39dfe4-0052d82653
Date: Thu, 16 Jan 2014 18:34:59 GMT
```

- Delete container metadata:

```
curl -i $publicURL/marktwain -X POST -H "X-Auth-Token: $token" -H "X-Remove-
Container-Meta-Century: x"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx7997e18da2a34a9e84ceb-0052d826d0
```

```
Date: Thu, 16 Jan 2014 18:37:04 GMT
```

If the request succeeds, the operation returns the `No Content (204)` response code.

To confirm your changes, issue a show container metadata request.

Normal response codes: 204

8.4.4.1. Request

This table shows the header parameters for the create, update, or delete container metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.

`X-Container-Read`	String
	(Optional)
		Sets a container access control list (ACL) that grants read access. Container ACLs are available on any Object Storage cluster, and are enabled by container rather than by cluster.
		To set the container read ACL:
		`$ curl -X {PUT|POST} -i -H "X-Auth-Token: TOKEN" -H \ "X-Container-Read: ACL" STORAGE_URL/CONTAINER`
		For example:
		`$ curl -X PUT -i \ -H "X-Auth-Token: 0101010101" \ -H "X-Container-Read: .r:*" \ http://swift.example.com/v1/AUTH_bob/read_container`
		In the command, specify the ACL in the `X-Container-Read` header, as follows:
		• `.r:*`#All referrers.
		• `.r:example.com,swift.example.com`#Comma-separated list of referrers.
		• `.rlistings`#Container listing access.
		• `AUTH_username`#Access to a user who authenticates through a legacy or non-OpenStack-Identity-based authentication system.
		• `LDAP_`#Access to all users who authenticate through an LDAP-based legacy or non-OpenStack-Identity-based authentication system.

`X-Remove-Container-name`	String *(Optional)*	Removes the metadata item named `{name}`. For example, `X-Remove-Container-Read` removes the `X-Container-Read` metadata item.
`X-Container-Write`	String *(Optional)*	Sets an ACL that grants write access.
`X-Container-Sync-To`	String *(Optional)*	Sets the destination for container synchronization. Used with the secret key indicated in the `X-Container-Sync-Key` header. If you want to stop a container from synchronizing, send a blank value for the `X-Container-Sync-Key` header.
`X-Container-Sync-Key`	String *(Optional)*	Sets the secret key for container synchronization. If you remove the secret key, synchronization is halted.
`X-Versions-Location`	String *(Optional)*	Enables versioning on this container. The value is the name of another container. You must UTF-8-encode and then URL-encode the name before you include it in the header. To disable versioning, set the header to an empty string.

Name	Type	Description
`X-Remove-Versions-Location`	String *(Optional)*	Set to any value to disable versioning.
`X-Container-Meta-name`	String *(Optional)*	The container metadata, where `{name}` is the name of metadata item. You must specify an `X-Container-Meta-{name}` header for each metadata item (for each `{name}`) that you want to add or update.
`X-Container-Meta-Access-Control-Allow-Origin`	String *(Optional)*	Originating URLs allowed to make cross-origin requests (CORS), separated by spaces. This heading applies to the container only, and all objects within the container with this header applied are CORS-enabled for the allowed origin URLs. A browser (user-agent) typically issues a preflighted request, which is an OPTIONS call that verifies the origin is allowed to make the request. The Object Storage service returns 200 if the originating URL is listed in this header parameter, and issues a 401 if the originating URL is not allowed to make a cross-origin request. Once a 200 is returned, the browser makes a second request to the Object Storage service to retrieve the CORS-enabled object.
`X-Container-Meta-Access-Control-Max-Age`	String *(Optional)*	Maximum time for the origin to hold the preflight results. A browser may make an OPTIONS call to verify the origin is allowed to make the request. Set the value to an integer number of seconds after the time that the request was received.
`X-Container-Meta-Access-Control-Expose-Headers`	String *(Optional)*	Headers the Object Storage service exposes to the browser (technically, through the `user-agent` setting), in the request response, separated by spaces. By default the Object Storage service returns the following values for this header: • All "simple response headers" as listed on http://www.w3.org/TR/cors/#simple-response-header. • The headers `etag`, `x-timestamp`, `x-trans-id`. • All metadata headers (`X-Container-Meta-*` for containers and `X-Object-Meta-*` for objects) headers listed in `X-Container-Meta-Access-Control-Expose-Headers`.
`X-Container-Meta-Quota-Bytes`	String *(Optional)*	Sets maximum size of the container, in bytes. Typically these values are set by an administrator. Returns a 413 response (request entity too large) when an object PUT operation exceeds this quota value.
`X-Container-Meta-Quota-Count`	String *(Optional)*	Sets maximum object count of the container. Typically these values are set by an administrator. Returns a 413 response (request entity too large) when an object PUT operation exceeds this quota value.
`X-Container-Meta-Web-Directory-Type`	String *(Optional)*	Sets the content-type of directory marker objects. If the header is not set, default is `application/directory`. Directory marker objects are 0-byte objects that represent directories to create a simulated hierarchical structure. For example, if you set `"X-Container-Meta-Web-Directory-Type: text/directory"`, Object Storage treats 0-byte objects with a content-type of `text/directory` as directories rather than objects.
`Content-Type`	String *(Optional)*	Changes the MIME type for the object.
`X-Detect-Content-Type`	Boolean *(Optional)*	If set to `true`, Object Storage guesses the content type based on the file extension and ignores the value sent in the `Content-Type` header, if present.
`X-Container-Meta-Temp-URL-Key`	String	The secret key value for temporary URLs.

Name	Type	Description
	(Optional)	
`X-Container-Meta-Temp-URL-Key-2`	String *(Optional)*	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create, update, or delete container metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.

This operation does not accept a request body.

8.4.4.2. Response

This operation does not return a response body.

8.4.5. Show container metadata

Method	URI	Description
HEAD	`/v1/{account}/{container}`	Shows container metadata, including the number of objects and the total bytes of all objects stored in the container.

Show container metadata request:

```
curl -i $publicURL/marktwain -X HEAD -H "X-Auth-Token: $token"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
X-Container-Object-Count: 1
Accept-Ranges: bytes
X-Container-Meta-Book: TomSawyer
X-Timestamp: 1389727543.65372
X-Container-Meta-Author: SamuelClemens
X-Container-Bytes-Used: 14
Content-Type: text/plain; charset=utf-8
X-Trans-Id: tx0287b982a268461b9ec14-0052d826e2
Date: Thu, 16 Jan 2014 18:37:22 GMT
```

If the request succeeds, the operation returns the `No Content (204)` response code.

Normal response codes: 204

8.4.5.1. Request

This table shows the header parameters for the show container metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Optional)*	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
`X-Newest`	Boolean *(Optional)*	If set to `true`, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to `true` is more expensive for the back end, use it only when it is absolutely needed.
`X-Container-Meta-Temp-URL-Key`	String *(Optional)*	The secret key value for temporary URLs.
`X-Container-Meta-Temp-URL-Key-2`	String *(Optional)*	A second secret key value for temporary URLs. The second key enables you to rotate keys by having two active keys at the same time.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header.

Name	Type	Description
		For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show container metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.

This operation does not accept a request body.

8.4.5.2. Response

This operation does not return a response body.

8.5. Objects

Creates, replaces, shows details for, and deletes objects. Copies objects from another object with a new or different name. Updates object metadata.

Method	URI	Description
GET	`/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename,multipart-manifest}`	Downloads the object content and gets the object metadata.
PUT	`/v1/{account}/{container}/{object}{?multipart-manifest,temp_url_sig,temp_url_expires,filename}`	Creates an object with data content and metadata, or replaces an existing object with data content and metadata.
COPY	`/v1/{account}/{container}/{object}`	Copies an object to another object in the object store.
DELETE	`/v1/{account}/{container}/{object}{?multipart-manifest}`	Permanently deletes an object from the object store.
HEAD	`/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename}`	Shows object metadata.
POST	`/v1/{account}/{container}/{object}`	Creates or updates object metadata.

8.5.1. Get object content and metadata

Method	URI	Description
GET	`/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename,multipart-manifest}`	Downloads the object content and gets the object metadata.

This operation returns the object metadata in the response headers and the object content in the response body.

If this is a large object, the response body contains the concatenated content of the segment objects. To get the manifest instead of concatenated segment objects for a static large object, use the `multipart-manifest` query parameter.

Example requests and responses:

- Show object details for the `goodbye` object in the `marktwain` container:

```
curl -i $publicURL/marktwain/goodbye -X GET -H "X-Auth-Token: $token"
```

```
HTTP/1.1 200 OK
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Wed, 15 Jan 2014 16:41:49 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
X-Timestamp: 1389804109.39027
X-Object-Meta-Orig-Filename: goodbyeworld.txt
Content-Type: application/octet-stream
X-Trans-Id: tx8145a190241f4cf6b05f5-0052d82a34
Date: Thu, 16 Jan 2014 18:51:32 GMT

Goodbye World!
```

- Show object details for the `goodbye` object, which does not exist, in the `janeausten` container:

```
curl -i $publicURL/janeausten/goodbye -X GET -H "X-Auth-Token: $token"
```

```
HTTP/1.1 404 Not Found
Content-Length: 70
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx073f7cbb850c4c99934b9-0052d82b04
Date: Thu, 16 Jan 2014 18:55:00 GMT

<html><h1>Not Found</h1><p>The resource could not be found.</p></html>
```

The operation returns the `Range Not Satisfiable (416)` response code for any ranged **GET** requests that specify more than:

- Fifty ranges.
- Three overlapping ranges.
- Eight non-increasing ranges.

Normal response codes: 200

Error response codes: NotFound (404), 416

8.5.1.1. Request

This table shows the header parameters for the get object content and metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Optional)*	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
`X-Newest`	Boolean *(Optional)*	If set to `true`, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to `true` is more expensive for the back end, use it only when it is absolutely needed.
`Range`	String *(Optional)*	The ranges of content to get. You can use the `Range` header to get portions of data by using one or more range specifications. To specify many ranges, separate the range specifications with a comma. The types of range specifications are: • **Byte range specification**. Use FIRST_BYTE_OFFSET to specify the start of the data range, and LAST_BYTE_OFFSET to specify the end. You can omit the LAST_BYTE_OFFSET and if you do, the value defaults to the offset of the last byte of data. • **Suffix byte range specification**. Use LENGTH bytes to specify the length of the data range. The following forms of the header specify the following ranges of data: • `Range: bytes=-5`. The last five bytes. • `Range: bytes=10-15`. The five bytes of data after a 10-byte offset. • `Range: bytes=10-15,-5`. A multi-part response that contains the last five bytes and the five bytes of data after a 10-byte offset. The `Content-Type` response header contains `multipart/byteranges`. • `Range: bytes=4-6`. Bytes 4 to 6 inclusive. • `Range: bytes=2-2`. Byte 2, the third byte of the data. • `Range: bytes=6-`. Byte 6 and after. • `Range: bytes=1-3,2-5`. A multi-part response that contains bytes 1 to 3 inclusive, and bytes 2 to 5 inclusive. The `Content-Type` response header contains `multipart/byteranges`.
`If-Match`	String *(Optional)*	See Request for Comments: 2616.
`If-None-Match`	String *(Optional)*	In combination with `Expect: 100-Continue`, specify an `"If-None-Match: *"` header to query whether the server already has a copy of the object before any data is sent.
`If-Modified-Since`	Date *(Optional)*	See Request for Comments: 2616.
`If-Unmodified-Since`	Date	See Request for Comments: 2616.

Name	Type	Description
	(Optional)	
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the get object content and metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.
`{object}`	String	The unique name for the object.

This table shows the query parameters for the get object content and metadata request:

Name	Type	Description
`temp_url_sig`	String *(Required)*	Used with temporary URLs to sign the request with an HMAC-SHA1 cryptographic signature that defines the allowed HTTP method, expiration date, full path to the object, and the secret key for the temporary URL. For more information about temporary URLs, see Temporary URL middleware.
`temp_url_expires`	Int *(Required)*	The date and time in UNIX Epoch time stamp format when the signature for temporary URLs expires. For example, `1440619048` is equivalent to `Mon, Wed, 26 Aug 2015 19:57:28 GMT`. For more information about temporary URLs, see Temporary URL middleware.
`filename`	String *(Optional)*	Overrides the default file name. Object Storage generates a default file name for **GET** temporary URLs that is based on the object name. Object Storage returns this value in the `Content-Disposition` response header. Browsers can interpret this file name value as a file attachment to save. For more information about temporary URLs, see Temporary URL middleware.
`multipart-manifest`	String *(Optional)*	If you include the `multipart-manifest=get` query parameter and the object is a large object, the object contents are not returned. Instead, the manifest is returned in the `X-Object-Manifest` response

Name	Type	Description
		header for dynamic large objects or in the response body for static large objects.

This operation does not accept a request body.

8.5.1.2. Response

This operation does not return a response body.

8.5.2. Create or replace object

Method	URI	Description
PUT	`/v1/{account}/{container}/{object}{?multipart-manifest,temp_url_sig,temp_url_expires,filename}`	Creates an object with data content and metadata, or replaces an existing object with data content and metadata.

The **PUT** operation always creates an object. If you use this operation on an existing object, you replace the existing object and metadata rather than modifying the object. Consequently, this operation returns the `Created (201)` response code.

If you use this operation to copy a manifest object, the new object is a normal object and not a copy of the manifest. Instead it is a concatenation of all the segment objects. This means that you cannot copy objects larger than 5 GB.

Example requests and responses:

- Create object:

```
curl -i $publicURL/janeausten/helloworld.txt -X PUT -H "Content-Length: 1" -
H "Content-Type: text/html; charset=UTF-8" -H "X-Auth-Token: $token"
```

```
HTTP/1.1 201 Created
Last-Modified: Fri, 17 Jan 2014 17:28:35 GMT
Content-Length: 116
Etag: d41d8cd98f00b204e9800998ecf8427e
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d5e4f06d357462bb732f-0052d96843
Date: Fri, 17 Jan 2014 17:28:35 GMT
```

- Replace object:

```
curl -i $publicURL/janeausten/helloworld -X PUT -H "Content-Length: 0" -H
 "X-Auth-Token: $token"
```

```
HTTP/1.1 201 Created
Last-Modified: Fri, 17 Jan 2014 17:28:35 GMT
Content-Length: 116
Etag: d41d8cd98f00b204e9800998ecf8427e
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx4d5e4f06d357462bb732f-0052d96843
Date: Fri, 17 Jan 2014 17:28:35 GMT
```

The `Created (201)` response code indicates a successful write.

If the request times out, the operation returns the `Request Timeout (408)` response code.

The `Length Required (411)` response code indicates a missing `Transfer-Encoding` or `Content-Length` request header.

If the MD5 checksum of the data that is written to the object store does not match the optional `ETag` value, the operation returns the `Unprocessable Entity (422)` response code.

Normal response codes: 201

Error response codes: timeout (408), lengthRequired (411), unprocessableEntity (422)

8.5.2.1. Request

This table shows the header parameters for the create or replace object request:

Name	Type	Description
`X-Object-Manifest`	String *(Optional)*	Set to specify that this is a dynamic large object manifest object. The value is the container and object name prefix of the segment objects in the form `container/prefix`. You must UTF-8-encode and then URL-encode the names of the container and prefix before you include them in this header.
`X-Auth-Token`	String *(Optional)*	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
`Content-Length`	Int *(Optional)*	Set to the length of the object content. Do not set if chunked transfer encoding is being used.
`Transfer-Encoding`	String *(Optional)*	Set to `chunked` to enable chunked transfer encoding. If used, do not set the `Content-Length` header to a non-zero value.
`Content-Type`	String *(Optional)*	Changes the MIME type for the object.
`X-Detect-Content-Type`	Boolean *(Optional)*	If set to `true`, Object Storage guesses the content type based on the file extension and ignores the value sent in the `Content-Type` header, if present.
`X-Copy-From`	String *(Optional)*	If set, this is the name of an object used to create the new object by copying the `X-Copy-From` object. The value is in form `{container}/{object}`. You must UTF-8-encode and then URL-encode the names of the container and object before you include them in the header. Using **PUT** with `X-Copy-From` has the same effect as using the **COPY** operation to copy an object. Using `Range` header with `X-Copy-From` will create a new partial copied object with bytes set by `Range`.
`ETag`	String *(Optional)*	The MD5 checksum value of the request body. For example, the MD5 checksum value of the object content. You are strongly recommended to compute the MD5 checksum value of object content and include it in the request. This enables the Object Storage API to check the integrity of the upload. The value is not quoted.
`Content-Disposition`	String *(Optional)*	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.
`Content-Encoding`	String *(Optional)*	If set, the value of the `Content-Encoding` metadata.
`X-Delete-At`	Int *(Optional)*	The date and time in UNIX Epoch time stamp format when the system removes the object. For example, `1440619048` is equivalent to `Mon, Wed, 26 Aug 2015 19:57:28 GMT`.
`X-Delete-After`	Int *(Optional)*	The number of seconds after which the system removes the object. Internally, the Object Storage system stores this value in the `X-Delete-At` metadata item.
`X-Object-Meta-name`	String	The object metadata, where `{name}` is the name of the metadata item.

Name	Type	Description
	(Optional)	You must specify an `X-Object-Meta-{name}` header for each metadata `{name}` item that you want to add or update.
`If-None-Match`	String *(Optional)*	In combination with `Expect: 100-Continue`, specify an `"If-None-Match: *"` header to query whether the server already has a copy of the object before any data is sent.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create or replace object request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.
`{object}`	String	The unique name for the object.

This table shows the query parameters for the create or replace object request:

Name	Type	Description
`multipart-manifest`	String *(Optional)*	If `?multipart-manifest=put`, the object is a static large object manifest and the body contains the manifest.
`temp_url_sig`	String *(Required)*	Used with temporary URLs to sign the request with an HMAC-SHA1 cryptographic signature that defines the allowed HTTP method, expiration date, full path to the object, and the secret key for the temporary URL. For more information about temporary URLs, see Temporary URL middleware.
`temp_url_expires`	Int *(Required)*	The date and time in UNIX Epoch time stamp format when the signature for temporary URLs expires. For example, `1440619048` is equivalent to `Mon, Wed, 26 Aug 2015 19:57:28 GMT`. For more information about temporary URLs, see Temporary URL middleware.
`filename`	String	Overrides the default file name. Object Storage generates a default file name for **GET** temporary URLs that is based on the object name.

Name	Type	Description
	(Optional)	Object Storage returns this value in the `Content-Disposition` response header. Browsers can interpret this file name value as a file attachment to save. For more information about temporary URLs, see Temporary URL middleware.

This operation does not accept a request body.

8.5.2.2. Response

This operation does not return a response body.

8.5.3. Copy object

Method	URI	Description
COPY	`/v1/{account}/{container}/{object}`	Copies an object to another object in the object store.

You can copy an object to a new object with the same name. Copying to the same name is an alternative to using **POST** to add metadata to an object. With **POST**, you must specify all the metadata. With **COPY**, you can add additional metadata to the object.

With **COPY**, you can set the `X-Fresh-Metadata` header to `true` to copy the object without any existing metadata.

Alternatively, you can use **PUT** with the `X-Copy-From` request header to accomplish the same operation as the **COPY** object operation.

The **PUT** operation always creates an object. If you use this operation on an existing object, you replace the existing object and metadata rather than modifying the object. Consequently, this operation returns the `Created (201)` response code.

If you use this operation to copy a manifest object, the new object is a normal object and not a copy of the manifest. Instead it is a concatenation of all the segment objects. This means that you cannot copy objects larger than 5 GB in size. All metadata is preserved during the object copy. If you specify metadata on the request to copy the object, either **PUT** or **COPY**, the metadata overwrites any conflicting keys on the target (new) object.

Example requests and responses:

- Copy the `goodbye` object from the `marktwain` container to the `janeausten` container:

```
curl -i $publicURL/marktwain/goodbye -X COPY -H "X-Auth-Token: $token" -H
 "Destination: janeausten/goodbye"
```

```
HTTP/1.1 201 Created
Content-Length: 0
X-Copied-From-Last-Modified: Thu, 16 Jan 2014 21:19:45 GMT
X-Copied-From: marktwain/goodbye
Last-Modified: Fri, 17 Jan 2014 18:22:57 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
Content-Type: text/html; charset=UTF-8
X-Object-Meta-Movie: AmericanPie
X-Trans-Id: txdcb481ad49d24e9a81107-0052d97501
Date: Fri, 17 Jan 2014 18:22:57 GMT
```

- Alternatively, you can use **PUT** to copy the `goodbye` object from the `marktwain` container to the `janeausten` container. This request requires a `Content-Length` header, even if it is set to zero (0).

```
curl -i $publicURL/janeausten/goodbye -X PUT -H "X-Auth-Token: $token" -H
 "X-Copy-From: /marktwain/goodbye" -H "Content-Length: 0"
```

```
HTTP/1.1 201 Created
```

```
Content-Length: 0
X-Copied-From-Last-Modified: Thu, 16 Jan 2014 21:19:45 GMT
X-Copied-From: marktwain/goodbye
Last-Modified: Fri, 17 Jan 2014 18:22:57 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
Content-Type: text/html; charset=UTF-8
X-Object-Meta-Movie: AmericanPie
X-Trans-Id: txdcb481ad49d24e9a81107-0052d97501
Date: Fri, 17 Jan 2014 18:22:57 GMT
```

When several replicas exist, the system copies from the most recent replica. That is, the **COPY** operation behaves as though the `X-Newest` header is in the request.

Normal response codes: 201

8.5.3.1. Request

This table shows the header parameters for the copy object request:

Name	Type	Description
`X-Auth-Token`	String *(Optional)*	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
`Destination`	String *(Required)*	The container and object name of the destination object in the form of `/container/object`. You must UTF-8-encode and then URL-encode the names of the destination container and object before you include them in this header.
`Content-Type`	String *(Optional)*	Changes the MIME type for the object.
`Content-Encoding`	String *(Optional)*	If set, the value of the `Content-Encoding` metadata.
`Content-Disposition`	String *(Optional)*	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.
`X-Object-Meta-name`	String *(Optional)*	The object metadata, where `{name}` is the name of the metadata item. You must specify an `X-Object-Meta-{name}` header for each metadata `{name}` item that you want to add or update.
`X-Fresh-Metadata`	Boolean *(Optional)*	Enables object creation that omits existing user metadata. If set to `true`, the **COPY** request creates an object without existing user metadata. Default value is `false`.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra`

Name	Type	Description
		request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the copy object request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.
`{object}`	String	The unique name for the object.

This operation does not accept a request body.

8.5.3.2. Response

This operation does not return a response body.

8.5.4. Delete object

Method	URI	Description
DELETE	`/v1/{account}/{container}/{object}{?multipart-manifest}`	Permanently deletes an object from the object store.

You can use the **COPY** method to copy the object to a new location. Then, use the **DELETE** method to delete the original object.

Object deletion occurs immediately at request time. Any subsequent **GET**, **HEAD**, **POST**, or **DELETE** operations return a `404 Not Found` error code.

For static large object manifests, you can add the `?multipart-manifest=delete` query parameter. This operation deletes the segment objects and if all deletions succeed, this operation deletes the manifest object.

Example request and response:

- Delete the `helloworld` object from the `marktwain` container:

```
curl -i $publicURL/marktwain/helloworld -X DELETE -H "X-Auth-Token: $token"
```

```
HTTP/1.1 204 No Content
Content-Length: 0
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx36c7606fcd1843f59167c-0052d6fdac
Date: Wed, 15 Jan 2014 21:29:16 GMT
```

Typically, the **DELETE** operation does not return a response body. However, with the `multipart-manifest=delete` query parameter, the response body contains a list of manifest and segment objects and the status of their **DELETE** operations.

Normal response codes: 204

8.5.4.1. Request

This table shows the header parameters for the delete object request:

Name	Type	Description
`X-Auth-Token`	String *(Optional)*	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment

Name	Type	Description
		and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the delete object request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.
`{object}`	String	The unique name for the object.

This table shows the query parameters for the delete object request:

Name	Type	Description
`multipart-manifest`	String *(Optional)*	If you include the `multipart-manifest=delete` query parameter and the object is a static large object, the segment objects and manifest object are deleted. If you omit the `multipart-manifest=delete` query parameter and the object is a static large object, the manifest object is deleted but the segment objects are not deleted. For a bulk delete, the response body looks the same as it does for a normal bulk delete. In contrast, a plain object **DELETE** response has an empty body.

This operation does not accept a request body.

8.5.4.2. Response

This operation does not return a response body.

8.5.5. Show object metadata

Method	URI	Description
HEAD	`/v1/{account}/{container}/{object}{?temp_url_sig,temp_url_expires,filename}`	Shows object metadata.

If the `Content-Length` response header is non-zero, the example cURL command stalls after it prints the response headers because it is waiting for a response body. However, the Object Storage system does not return a response body for the **HEAD** operation.

Example requests and responses:

- Show object metadata:

```
curl -i $publicURL/marktwain/goodbye -X HEAD -H "X-Auth-Token: $token"
```

```
HTTP/1.1 200 OK
Content-Length: 14
Accept-Ranges: bytes
Last-Modified: Thu, 16 Jan 2014 21:12:31 GMT
Etag: 451e372e48e0f6b1114fa0724aa79fa1
X-Timestamp: 1389906751.73463
X-Object-Meta-Book: GoodbyeColumbus
Content-Type: application/octet-stream
X-Trans-Id: tx37ea34dcd1ed48ca9bc7d-0052d84b6f
Date: Thu, 16 Jan 2014 21:13:19 GMT
```

If the request succeeds, the operation returns the `200` response code.

Normal response codes: 200204

8.5.5.1. Request

This table shows the header parameters for the show object metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Required)*	Authentication token.
`X-Newest`	Boolean *(Optional)*	If set to `true`, Object Storage queries all replicas to return the most recent one. If you omit this header, Object Storage responds faster after it finds one valid replica. Because setting this header to `true` is more expensive for the back end, use it only when it is absolutely needed.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header.

Name	Type	Description
		For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs. You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the show object metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.
`{object}`	String	The unique name for the object.

This table shows the query parameters for the show object metadata request:

Name	Type	Description
`temp_url_sig`	String *(Required)*	Used with temporary URLs to sign the request with an HMAC-SHA1 cryptographic signature that defines the allowed HTTP method, expiration date, full path to the object, and the secret key for the temporary URL. For more information about temporary URLs, see Temporary URL middleware.
`temp_url_expires`	Int *(Required)*	The date and time in UNIX Epoch time stamp format when the signature for temporary URLs expires. For example, `1440619048` is equivalent to `Mon, Wed, 26 Aug 2015 19:57:28 GMT`. For more information about temporary URLs, see Temporary URL middleware.
`filename`	String *(Optional)*	Overrides the default file name. Object Storage generates a default file name for **GET** temporary URLs that is based on the object name. Object Storage returns this value in the `Content-Disposition` response header. Browsers can interpret this file name value as a file attachment to save. For more information about temporary URLs, see Temporary URL middleware.

This operation does not accept a request body.

8.5.5.2. Response

This operation does not return a response body.

8.5.6. Create or update object metadata

Method	URI	Description
POST	`/v1/{account}/{container}/{object}`	Creates or updates object metadata.

To create or update custom metadata, use the `X-Object-Meta-{name}` header, where `{name}` is the name of the metadata item.

In addition to the custom metadata, you can update the `Content-Type`, `Content-Encoding`, `Content-Disposition`, and `X-Delete-At` system metadata items. However you cannot update other system metadata, such as `Content-Length` or `Last-Modified`.

You can use **COPY** as an alternate to the **POST** operation by copying to the same object. With the **POST** operation you must specify all metadata items, whereas with the **COPY** operation, you need to specify only changed or additional items.

All metadata is preserved during the object copy. If you specify metadata on the request to copy the object, either **PUT** or **COPY**, the metadata overwrites any conflicting keys on the target (new) object.

A **POST** request deletes any existing custom metadata that you added with a previous **PUT** or **POST** request. Consequently, you must specify all custom metadata in the request. However, system metadata is unchanged by the **POST** request unless you explicitly supply it in a request header.

You can also set the `X-Delete-At` or `X-Delete-After` header to define when to expire the object.

When used as described in this section, the **POST** operation creates or replaces metadata. This form of the operation has no request body.

You can also use the form **POST** feature to upload objects.

Example requests and responses:

- Create object metadata:

```
curl -i $publicURL/marktwain/goodbye -X POST -H "X-Auth-Token: $token" -H
 "X-Object-Meta-Book: GoodbyeColumbus"
```

```
HTTP/1.1 202 Accepted
Content-Length: 76
Content-Type: text/html; charset=UTF-8
X-Trans-Id: txb5fb5c91ba1f4f37bb648-0052d84b3f
Date: Thu, 16 Jan 2014 21:12:31 GMT

<html><h1>Accepted</h1><p>The request is accepted for processing.</p></html>
```

- Update object metadata:

```
curl -i $publicURL/marktwain/goodbye -X POST -H "X-Auth-Token: $token" H "X-
Object-Meta-Book: GoodbyeOldFriend"
```

```
HTTP/1.1 202 Accepted
Content-Length: 76
Content-Type: text/html; charset=UTF-8
X-Trans-Id: tx5ec7ab81cdb34ced887c8-0052d84ca4
Date: Thu, 16 Jan 2014 21:18:28 GMT

<html><h1>Accepted</h1><p>The request is accepted for processing.</p></html>
```

Normal response codes: 202

8.5.6.1. Request

This table shows the header parameters for the create or update object metadata request:

Name	Type	Description
`X-Auth-Token`	String *(Optional)*	Authentication token. If you omit this header, your request fails unless the account owner has granted you access through an access control list (ACL).
`X-Object-Meta-name`	String *(Optional)*	The object metadata, where `{name}` is the name of the metadata item. You must specify an `X-Object-Meta-{name}` header for each metadata `{name}` item that you want to add or update.
`X-Delete-At`	Int *(Optional)*	The date and time in UNIX Epoch time stamp format when the system removes the object. For example, `1440619048` is equivalent to `Mon, Wed, 26 Aug 2015 19:57:28 GMT`.
`Content-Disposition`	String *(Optional)*	If set, specifies the override behavior for the browser. For example, this header might specify that the browser use a download program to save this file rather than show the file, which is the default.
`Content-Encoding`	String *(Optional)*	If set, the value of the `Content-Encoding` metadata.
`X-Delete-After`	Int *(Optional)*	The number of seconds after which the system removes the object. Internally, the Object Storage system stores this value in the `X-Delete-At` metadata item.
`Content-Type`	String *(Optional)*	Changes the MIME type for the object.
`X-Detect-Content-Type`	Boolean *(Optional)*	If set to `true`, Object Storage guesses the content type based on the file extension and ignores the value sent in the `Content-Type` header, if present.
`X-Trans-Id-Extra`	UUID *(Optional)*	Extra transaction information. Use the `X-Trans-Id-Extra` request header to include extra information to help you debug any errors that might occur with large object upload and other Object Storage transactions. Object Storage appends the first 32 characters of the `X-Trans-Id-Extra` request header value to the transaction ID value in the generated `X-Trans-Id` response header. You must UTF-8-encode and then URL-encode the extra transaction information before you include it in the `X-Trans-Id-Extra` request header. For example, you can include extra transaction information when you upload large objects such as images. When you upload each segment and the manifest, include the same value in the `X-Trans-Id-Extra` request header. If an error occurs, you can find all requests that are related to the large object upload in the Object Storage logs.

Name	Type	Description
		You can also use `X-Trans-Id-Extra` strings to help operators debug requests that fail to receive responses. The operator can search for the extra information in the logs.

This table shows the URI parameters for the create or update object metadata request:

Name	Type	Description
`{account}`	String	The unique name for the account. An account is also known as the project or tenant.
`{container}`	String	The unique name for the container. The container name must be from 1 to 256 characters long and can start with any character and contain any pattern. Character set must be UTF-8. The container name cannot contain a slash (`/`) character because this character delimits the container and object name. For example, `/account/container/object`.
`{object}`	String	The unique name for the object.

This operation does not accept a request body.

8.5.6.2. Response

This operation does not return a response body.

9. Orchestration API v1 (CURRENT)

Uses a template language to orchestrate OpenStack services.

Method	URI	Description
General API information		
API versions		
GET	/	Lists all Orchestration API versions.
Stacks		
POST	/v1/{tenant_id}/stacks	Creates a stack.
POST	/v1/{tenant_id}/stacks	Creates a stack from existing resources.
GET	/v1/{tenant_id}/stacks{?id, status,name,action,tenant, username,owner_id,limit,marker,show_deleted,show_nested, sort_keys,tags,tags_any, not_tags,not_tags_any,sort_dir, global_tenant,with_count}	Lists active stacks.
POST	/v1/{tenant_id}/stacks/preview	Previews a stack.
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/preview	Previews an update for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}	Finds the canonical URL for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/resources	Finds the canonical URL for a resource list of a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Shows details for a stack.
PUT	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Updates a stack.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}	Deletes a stack and its snapshots.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon	Deletes a stack but leaves its resources intact, and returns data that describes the stack and its resources.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Takes a snapshot of all resources in a stack. All snapshots are deleted when the stack is deleted.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots	Lists snapshots for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Shows details for a snapshot.
DELETE	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}	Deletes a stack snapshot.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}/restore	Restores a stack snapshot.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs	Lists outputs for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs/{output_key}	Shows details for a stack output.
Stack actions		

Method	URI	Description
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Suspends a stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Resumes a suspended stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Cancels a currently running update of a stack.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions	Checks whether the resources are in expected states for a stack.
Stack resources		
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources{?nested_depth,with_detail}	Lists resources in a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}	Shows data for a resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata	Shows metadata for a resource.
POST	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal	Sends a signal to a resource.
Stack events		
GET	/v1/{tenant_id}/stacks/{stack_name}/events	Finds the canonical URL for the event list of a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events{?resource_action,resource_status,resource_name,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events{?resource_action,resource_status,resource_type,limit,marker,sort_keys,sort_dir}	Lists events for a stack resource.
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}	Shows details for an event.
Templates		
GET	/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template	Gets a template for a stack.
GET	/v1/{tenant_id}/template_versions	Lists all available template versions.
POST	/v1/{tenant_id}/validate	Validates a template.
GET	/v1/{tenant_id}/resource_types/{type_name}/template{?template_type}	Shows the template representation for a resource type.
GET	/v1/{tenant_id}/resource_types/{type_name}	Shows the interface schema for a resource type.
GET	/v1/{tenant_id}/resource_types{?name,version,support_status}	Lists all supported template resource types.
Build info		
GET	/v1/{tenant_id}/build_info	Shows build information for an Orchestration deployment.
Software configuration		

Method	URI	Description
POST	`/v1/{tenant_id}/software_configs`	Creates a software configuration.
GET	`/v1/{tenant_id}/software_configs/{config_id}`	Shows details for a software configuration.
DELETE	`/v1/{tenant_id}/software_configs/{config_id}`	Deletes a software configuration.
GET	`/v1/{tenant_id}/software_deployments`	Lists all available software deployments.
POST	`/v1/{tenant_id}/software_deployments`	Creates a software deployment.
GET	`/v1/{tenant_id}/software_deployments/metadata/{server_id}`	Shows the deployment configuration metadata for a server.
GET	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Shows details for a software deployment.
PUT	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Updates a software deployment.
DELETE	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Deletes a software deployment.
Manage service		
GET	`/v1/{tenant_id}/services`	Enables administrative users to view details for all orchestration engines.

9.1. General API information

Authenticated calls that target a known URI but that use an HTTP method that the implementation does not support return a `405 Method Not Allowed` error code. In addition, the HTTP `OPTIONS` method is supported for each known URI. In both cases, the `Allow` response header indicates the HTTP methods that are supported for the resource.

9.2. API versions

Method	URI	Description
GET	/	Lists all Orchestration API versions.

9.2.1. List versions

Method	URI	Description
GET	/	Lists all Orchestration API versions.

Normal response codes: 200

9.2.1.1. Request

This operation does not accept a request body.

9.2.1.2. Response

Example 9.1. List versions: JSON response

```
{
    "versions": [
        {
            "status": "CURRENT",
            "id": "v1.0",
            "links": [
                {
                    "href": "http://23.253.228.211:8000/v1/",
                    "rel": "self"
                }
            ]
        }
    ]
}
```

9.3. Stacks

Method	URI	Description
POST	`/v1/{tenant_id}/stacks`	Creates a stack.
POST	`/v1/{tenant_id}/stacks`	Creates a stack from existing resources.
GET	`/v1/{tenant_id}/stacks{?id, status,name,action,tenant, username,owner_id,limit,marker,show_deleted,show_nested, sort_keys,tags,tags_any, not_tags,not_tags_any,sort_dir, global_tenant,with_count}`	Lists active stacks.
POST	`/v1/{tenant_id}/stacks/preview`	Previews a stack.
PUT	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/preview`	Previews an update for a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}`	Finds the canonical URL for a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/resources`	Finds the canonical URL for a resource list of a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}`	Shows details for a stack.
PUT	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}`	Updates a stack.

Method	URI	Description
DELETE	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}`	Deletes a stack and its snapshots.
DELETE	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon`	Deletes a stack but leaves its resources intact, and returns data that describes the stack and its resources.
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots`	Takes a snapshot of all resources in a stack. All snapshots are deleted when the stack is deleted.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots`	Lists snapshots for a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.
DELETE	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}`	Deletes a stack snapshot.
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}/restore`	Restores a stack snapshot.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs`	Lists outputs for a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs/{output_key}`	Shows details for a stack output.

9.3.1. Create stack

Method	URI	Description
POST	`/v1/{tenant_id}/stacks`	Creates a stack.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), conflict (409), internalServerError (500)

9.3.1.1. Request

This table shows the URI parameters for the create stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 9.2. Create stack: JSON request

```
{
    "files": {},
    "disable_rollback": true,
    "parameters": {
        "flavor": "m1.heat"
    },
    "stack_name": "teststack",
    "template": {
        "heat_template_version": "2013-05-23",
        "description": "Simple template to test heat commands",
        "parameters": {
            "flavor": {
                "default": "m1.tiny",
                "type": "string"
            }
        },
        "resources": {
            "hello_world": {
                "type": "OS::Nova::Server",
                "properties": {
                    "key_name": "heat_key",
                    "flavor": {
                        "get_param": "flavor"
                    },
                    "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
                    "user_data": "#!/bin/bash -xv\necho \"hello world\" &gt; /
root/hello-world.txt\n"
                }
            }
        }
    },
    "timeout_mins": 60
}
```

9.3.1.2. Response

Example 9.3. Create stack: JSON response

```
{
    "stack": {
        "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "links": [
            {
                "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/teststack/3095aefc-09fb-4bc7-b1f0-
f21a304e864c",
                "rel": "self"
            }
        ]
    }
}
```

9.3.2. Adopt stack

Method	URI	Description
POST	`/v1/{tenant_id}/stacks`	Creates a stack from existing resources.

Normal response codes: 201

Error response codes: badRequest (400), unauthorized (401), conflict (409), internalServerError (500)

9.3.2.1. Request

This table shows the URI parameters for the adopt stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 9.4. Adopt stack: JSON request

```
{
    "adopt_stack_data": {
        "action": "CREATE",
        "id": "bxxxxx4-0xx2-4xx1-axx6-exxxxxxxc",
        "name": "teststack",
        "resources": {
            "MyServer": {
                "action": "CREATE",
                "metadata": {},
                "name": "MyServer",
                "resource_data": {},
                "resource_id": "cxxxx3-dxx3-4xx-bxx2-3xxxxxxxxa",
                "status": "COMPLETE",
                "type": "OS::Trove::Instance"
            }
        },
        "status": "COMPLETE",
        "template": {
            "heat_template_version": "2013-05-23",
            "resources": {
                "MyServer": {
                    "type": "OS::Trove::Instance",
                    "properties": {
                        "flavor": "m1.small",
                        "size": 10
                    }
                }
            }
        }
    },
    "stack_name": "{stack_name}",
    "timeout_mins": "{timeout_mins}"
}
```

9.3.2.2. Response

Example 9.5. Adopt stack: JSON response

```
{
    "stack": {
        "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "links": [
            {
                "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/teststack/3095aefc-09fb-4bc7-b1f0-
f21a304e864c",
                "rel": "self"
            }
        ]
    }
}
```

9.3.3. List stack data

Method	URI	Description
GET	`/v1/{tenant_id}/stacks{?id, status,name,action,tenant, username,owner_id,limit,marker,show_deleted,show_nested, sort_keys,tags,tags_any, not_tags,not_tags_any,sort_dir, global_tenant,with_count}`	Lists active stacks.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), internalServerError (500)

9.3.3.1. Request

This table shows the URI parameters for the list stack data request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This table shows the query parameters for the list stack data request:

Name	Type	Description
`id`	UUID *(Optional)*	Filters the stack list by a stack ID. Use this filter multiple times to filter by multiple IDs.
`status`	String *(Optional)*	Filters the stack list by a status. Use this filter multiple times to filter by multiple statuses.
`name`	String *(Optional)*	Filters the stack list by a name. Use this filter multiple times to filter by multiple names.
`action`	String *(Optional)*	Filters the stack list by an action. Use this filter multiple times to filter by multiple actions.
`tenant`	String *(Optional)*	Filters the stack list by a tenant. Use this filter multiple times to filter by multiple tenants.
`username`	String *(Optional)*	Filters the stack list by a user name. Use this filter multiple times to filter by multiple user names.
`owner_id`	UUID *(Optional)*	Filters the stack list by an owner ID, which is the ID of the parent stack of listed stack. Use this filter multiple times to filter by multiple owner IDs.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

Name	Type	Description
show_deleted	Boolean *(Optional)*	Set to true to include deleted stacks in the list. Default is false, which excludes deleted stacks from the list.
show_nested	Boolean *(Optional)*	Set to true to include nested stacks in the list. Default is false, which excludes nested stacks from the list.
sort_keys	String *(Optional)*	Sorts the stack list by stack_name, stack_status, creation_time, or updated_time key.
tags	String *(Optional)*	Lists stacks that contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, tag1,tag2. The boolean AND expression is used to combine multiple tags.
tags_any	String *(Optional)*	Lists stacks that contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, tag1,tag2. The boolean OR expression is used to combine multiple tags.
not_tags	String *(Optional)*	Lists stacks that do not contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, tag1,tag2. The boolean AND expression is used to combine multiple tags.
not_tags_any	String *(Optional)*	Lists stacks that do not contain one or more simple string tags. To specify multiple tags, separate the tags with commas. For example, tag1,tag2. The boolean OR expression is used to combine multiple tags.
sort_dir	String *(Optional)*	The sort direction of the list. A valid value is asc (ascending) or desc (descending).
global_tenant	Boolean *(Optional)*	Set to true to include stacks from all tenants in the stack list. Specify policy requirements in the Orchestration policy.json file. Default is false.
with_count	Boolean *(Optional)*	Set to true to include a count key in the response. The count key value is the number of stacks that match the query criteria. Default is false.

This operation does not accept a request body.

9.3.3.2. Response

Example 9.6. List stack data: JSON response

```
{
    "stacks": [
        {
            "creation_time": "2014-06-03T20:59:46Z",
            "description": "sample stack",
            "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
            "links": [
                {
                    "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-
f21a304e864c",
                    "rel": "self"
                }
            ],
            "stack_name": "simple_stack",
            "stack_status": "CREATE_COMPLETE",
            "stack_status_reason": "Stack CREATE completed successfully",
            "updated_time": "",
            "tags": ""
```

```
        }
    ]
}
```

9.3.4. Preview stack

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/preview`	Previews a stack.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), conflict (409), internalServerError (500)

9.3.4.1. Request

This table shows the URI parameters for the preview stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 9.7. Preview stack: JSON request

```
{
    "files": {},
    "disable_rollback": true,
    "parameters": {
        "flavor": "m1.heat"
    },
    "stack_name": "teststack",
    "template": {
        "heat_template_version": "2013-05-23",
        "description": "Simple template to test heat commands",
        "parameters": {
            "flavor": {
                "default": "m1.tiny",
                "type": "string"
            }
        },
        "resources": {
            "hello_world": {
                "type": "OS::Nova::Server",
                "properties": {
                    "key_name": "heat_key",
                    "flavor": {
                        "get_param": "flavor"
                    },
                    "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
                    "user_data": "#!/bin/bash -xv\necho \"hello world\" &gt; /
root/hello-world.txt\n"
                }
            }
        }
    },
    "timeout_mins": 60
}
```

9.3.4.2. Response

Example 9.8. Preview stack: JSON response


```
{
    "stack": {
        "capabilities": [],
        "creation_time": "2015-01-31T15:12:36Z",
        "description": "HOT template for Nova Server resource.\n",
        "disable_rollback": true,
        "id": "None",
        "links": [
            {
                "href": "http://192.168.122.102:8004/v1/
6e18cc2bdbeb48a5basad2dc499f6804/stacks/test_stack/None",
                "rel": "self"
            }
        ],
        "notification_topics": [],
        "parameters": {
            "OS::project_id": "6e18cc2bdbeb48a5basad2dc499f6804",
            "OS::stack_id": "None",
            "OS::stack_name": "teststack",
            "admin_user": "cloud-user",
            "flavor": "m1.small",
            "image": "F20-cfg",
            "key_name": "heat_key",
            "server_name": "MyServer"
        },
        "parent": null,
        "resources": [
            {
                "attributes": {},
                "description": "",
                "metadata": {},
                "physical_resource_id": "",
                "properties": {
                    "description": "Ping and SSH",
                    "name": "the_sg",
                    "rules": [
                        {
                            "direction": "ingress",
                            "ethertype": "IPv4",
                            "port_range_max": null,
                            "port_range_min": null,
                            "protocol": "icmp",
                            "remote_group_id": null,
                            "remote_ip_prefix": null,
                            "remote_mode": "remote_ip_prefix"
                        },
                        {
                            "direction": "ingress",
                            "ethertype": "IPv4",
                            "port_range_max": 65535,
                            "port_range_min": 1,
                            "protocol": "tcp",
                            "remote_group_id": null,
                            "remote_ip_prefix": null,
                            "remote_mode": "remote_ip_prefix"
```

```
                    },
                    {
                        "direction": "ingress",
                        "ethertype": "IPv4",
                        "port_range_max": 65535,
                        "port_range_min": 1,
                        "protocol": "udp",
                        "remote_group_id": null,
                        "remote_ip_prefix": null,
                        "remote_mode": "remote_ip_prefix"
                    }
                ]
            },
            "required_by": [
                "server1"
            ],
            "resource_action": "INIT",
            "resource_identity": {
                "path": "/resources/the_sg_res",
                "stack_id": "None",
                "stack_name": "teststack",
                "tenant": "6e18cc2bdbeb48a5b3cad2dc499f6804"
            },
            "resource_name": "the_sg_res",
            "resource_status": "COMPLETE",
            "resource_status_reason": "",
            "resource_type": "OS::Neutron::SecurityGroup",
            "stack_identity": {
                "path": "",
                "stack_id": "None",
                "stack_name": "teststack",
                "tenant": "6e18cc2bdbeb48a5b3cad2dc499f6804"
            },
            "stack_name": "teststack",
            "updated_time": "2015-01-31T15:12:36Z"
        },
        {
            "attributes": {
                "accessIPv4": "",
                "accessIPv6": "",
                "addresses": "",
                "console_urls": "",
                "first_address": "",
                "instance_name": "",
                "name": "MyServer",
                "networks": "",
                "show": ""
            },
            "description": "",
            "metadata": {},
            "physical_resource_id": "",
            "properties": {
                "admin_pass": null,
                "admin_user": "cloud-user",
                "availability_zone": null,
                "block_device_mapping": null,
                "config_drive": null,
                "diskConfig": null,
                "flavor": "m1.small",
                "flavor_update_policy": "RESIZE",
```

```
                    "image": "F20-cfg",
                    "image_update_policy": "REPLACE",
                    "key_name": "heat_key",
                    "metadata": {
                        "ha_stack": "None"
                    },
                    "name": "MyServer",
                    "networks": [
                        {
                            "fixed_ip": null,
                            "network": "private",
                            "port": null,
                            "uuid": null
                        }
                    ],
                    "personality": {},
                    "reservation_id": null,
                    "scheduler_hints": null,
                    "security_groups": [
                        "None"
                    ],
                    "software_config_transport": "POLL_SERVER_CFN",
                    "user_data": "",
                    "user_data_format": "HEAT_CFNTOOLS"
                },
                "required_by": [],
                "resource_action": "INIT",
                "resource_identity": {
                    "path": "/resources/hello_world",
                    "stack_id": "None",
                    "stack_name": "teststack",
                    "tenant": "6e18cc2bdbeb48a3433cad2dc499sdf32234"
                },
                "resource_name": "hello_world",
                "resource_status": "COMPLETE",
                "resource_status_reason": "",
                "resource_type": "OS::Nova::Server",
                "stack_identity": {
                    "path": "",
                    "stack_id": "None",
                    "stack_name": "teststack",
                    "tenant": "6e18cc2bdbeb48a3433cad2dc499sdf32234"
                },
                "stack_name": "teststack",
                "updated_time": "2015-01-31T15:12:36Z"
            }
        ],
        "stack_name": "test_stack",
        "stack_owner": "admin",
        "template_description": "HOT template for Nova Server resource.\n",
        "timeout_mins": null,
        "updated_time": null
    }
}
```

9.3.5. Preview stack update

Method	URI	Description
PUT	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/preview`	Previews an update for a stack.

Normal response codes: 200

9.3.5.1. Request

This table shows the URI parameters for the preview stack update request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.9. Preview stack update: JSON request

```
{
    "template": {
        "heat_template_version": "2013-05-23",
        "description": "Create a simple stack",
        "parameters": {
            "flavor": {
                "default": "m1.tiny",
                "type": "string"
            }
        },
        "resources": {
            "hello_world": {
                "type": "OS::Nova::Server",
                "properties": {
                    "key_name": "heat_key",
                    "flavor": {
                        "get_param": "flavor"
                    },
                    "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
                    "user_data": "#!/bin/bash -xv\necho \"hello world\" > /
root/hello-world.txt\n"
                }
            }
        }
    },
    "parameters": {
        "flavor": "m1.small"
    }
}
```

9.3.5.2. Response

Example 9.10. Preview stack update: JSON response

```
{
    "unchanged": [
        {
```

```
            "updated_time": "datetime",
            "resource_name": "",
            "physical_resource_id": "{resource id or ''}",
            "resource_action": "CREATE",
            "resource_status": "COMPLETE",
            "resource_status_reason": "",
            "resource_type": "restype",
            "stack_identity": "{stack_id}",
            "stack_name": "{stack_name}"
        }
    ],
    "updated": [
        {
            "updated_time": "datetime",
            "resource_name": "",
            "physical_resource_id": "{resource id or ''}",
            "resource_action": "CREATE",
            "resource_status": "COMPLETE",
            "resource_status_reason": "",
            "resource_type": "restype",
            "stack_identity": "{stack_id}",
            "stack_name": "{stack_name}"
        }
    ],
    "replaced": [
        {
            "updated_time": "datetime",
            "resource_name": "",
            "physical_resource_id": "{resource id or ''}",
            "resource_action": "CREATE",
            "resource_status": "COMPLETE",
            "resource_status_reason": "",
            "resource_type": "restype",
            "stack_identity": "{stack_id}",
            "stack_name": "{stack_name}"
        }
    ],
    "added": [
        {
            "updated_time": "datetime",
            "resource_name": "",
            "physical_resource_id": "{resource id or ''}",
            "resource_action": "CREATE",
            "resource_status": "COMPLETE",
            "resource_status_reason": "",
            "resource_type": "restype",
            "stack_identity": "{stack_id}",
            "stack_name": "{stack_name}"
        }
    ],
    "deleted": [
        {
            "updated_time": "datetime",
            "resource_name": "",
            "physical_resource_id": "{resource id or ''}",
            "resource_action": "CREATE",
            "resource_status": "COMPLETE",
            "resource_status_reason": "",
            "resource_type": "restype",
            "stack_identity": "{stack_id}",
```

```
            "stack_name": "{stack_name}"
        }
    ]
}
```

9.3.6. Find stack

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}`	Finds the canonical URL for a stack.

Also works with verbs other than **GET**, so that you can perform **PUT** and **DELETE** operations on a current stack. Set your client to follow redirects. When redirecting, the request method should not change as defined in RFC2626. However, in many clients the default behavior is to change the method to **GET** when you receive a `302` response code because this behavior is ubiquitous in web browsers.

Normal response codes: 302

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.3.6.1. Request

This table shows the URI parameters for the find stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.

This operation does not accept a request body.

9.3.7. Find stack resources

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/resources`	Finds the canonical URL for a resource list of a stack.

The canonical URL is returned for only non-deleted stacks. To fetch the resource list for deleted stacks, use the following endpoint:

```
/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources
```

Normal response codes: 302

9.3.7.1. Request

This table shows the URI parameters for the find stack resources request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.

This operation does not accept a request body.

9.3.8. Show stack details

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}`	Shows details for a stack.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.3.8.1. Request

This table shows the URI parameters for the show stack details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This operation does not accept a request body.

9.3.8.2. Response

Example 9.11. Show stack details: JSON response

```
{
    "stack": {
        "capabilities": [],
        "creation_time": "2014-06-03T20:59:46Z",
        "description": "sample stack",
        "disable_rollback": true,
        "id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
        "links": [
            {
                "href": "http://192.168.123.200:8004/v1/
eb1c63a4f77141548385f113a28f0f52/stacks/simple_stack/3095aefc-09fb-4bc7-b1f0-
f21a304e864c",
                "rel": "self"
            }
        ],
        "notification_topics": [],
        "outputs": [],
        "parameters": {
            "OS::project_id": "3ab5b02f-a01f-4f95-afa1-e254afc4a435",
            "OS::stack_id": "3095aefc-09fb-4bc7-b1f0-f21a304e864c",
            "OS::stack_name": "simple_stack"
        },
        "stack_name": "simple_stack",
        "stack_owner": "simple_username",
        "stack_status": "CREATE_COMPLETE",
        "stack_status_reason": "Stack CREATE completed successfully",
        "template_description": "sample stack",
```

```
        "stack_user_project_id": "65728b74-cfe7-4f17-9c15-11d4f686e591",
        "timeout_mins": "",
        "updated_time": "",
        "parent": "",
        "tags": ""
    }
}
```

9.3.9. Update stack

Method	URI	Description
PUT	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}`	Updates a stack.

Normal response codes: 202

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.3.9.1. Request

This table shows the URI parameters for the update stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.12. Update stack: JSON request

```
{
    "template": {
        "heat_template_version": "2013-05-23",
        "description": "Create a simple stack",
        "parameters": {
            "flavor": {
                "default": "m1.tiny",
                "type": "string"
            }
        },
        "resources": {
            "hello_world": {
                "type": "OS::Nova::Server",
                "properties": {
                    "key_name": "heat_key",
                    "flavor": {
                        "get_param": "flavor"
                    },
                    "image": "40be8d1a-3eb9-40de-8abd-43237517384f",
                    "user_data": "#!/bin/bash -xv\necho \"hello world\" > /
root/hello-world.txt\n"
                }
            }
        }
    },
    "parameters": {
        "flavor": "m1.small"
    }
}
```

9.3.10. Delete stack

Method	URI	Description
DELETE	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}`	Deletes a stack and its snapshots.

Normal response codes: 204

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.3.10.1. Request

This table shows the URI parameters for the delete stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This operation does not accept a request body.

9.3.11. Abandon stack

Method	URI	Description
DELETE	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/abandon`	Deletes a stack but leaves its resources intact, and returns data that describes the stack and its resources.

This method can be disabled from the server side. If it is disabled, this call throws an exception.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.3.11.1. Request

This table shows the URI parameters for the abandon stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This operation does not accept a request body.

9.3.11.2. Response

Example 9.13. Abandon stack: JSON response

```
{
    "status": "COMPLETE",
    "name": "g",
    "dry_run": true,
    "template": {
        "outputs": {
            "instance_ip": {
                "value": {
                    "str_replace": {
                        "params": {
                            "username": "ec2-user",
                            "hostname": {
                                "get_attr": [
                                    "server",
                                    "first_address"
                                ]
                            }
                        },
                        "template": "ssh username@hostname"
                    }
                }
            }
        },
        "heat_template_version": "2013-05-23",
```

```
        "resources": {
            "server": {
                "type": "OS::Nova::Server",
                "properties": {
                    "key_name": {
                        "get_param": "key_name"
                    },
                    "image": {
                        "get_param": "image"
                    },
                    "flavor": {
                        "get_param": "flavor"
                    }
                }
            }
        },
        "parameters": {
            "key_name": {
                "default": "heat_key",
                "type": "string"
            },
            "image": {
                "default": "fedora-amd64",
                "type": "string"
            },
            "flavor": {
                "default": "m1.small",
                "type": "string"
            }
        }
    },
    "action": "CREATE",
    "id": "16934ca3-40e0-4fb2-a289-a700662ec05a",
    "resources": {
        "server": {
            "status": "COMPLETE",
            "name": "server",
            "resource_data": {},
            "resource_id": "39d5dad7-7d7a-4cc8-bd84-851e9e2ff4ea",
            "action": "CREATE",
            "type": "OS::Nova::Server",
            "metadata": {}
        }
    }
}
```

9.3.12. Snapshot stack

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots`	Takes a snapshot of all resources in a stack. All snapshots are deleted when the stack is deleted.

Normal response codes: 200

9.3.12.1. Request

This table shows the URI parameters for the snapshot stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.14. Snapshot stack: JSON request

```
{
    "name": "vol_snapshot"
}
```

9.3.12.2. Response

Example 9.15. Snapshot stack: JSON response

```
{
    "id": "13c3a4b5-0585-440e-85a4-6f96b20e7a78",
    "name": "vol_snapshot",
    "status": "IN_PROGRESS",
    "status_reason": null,
    "data": null,
    "creation_time": "2015-09-01T20:57:55Z"
}
```

9.3.13. List snapshots

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots`	Lists snapshots for a stack.

Normal response codes: 200

9.3.13.1. Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This operation does not accept a request body.

9.3.13.2. Response

Example 9.16. List snapshots: JSON response

```
{
    "snapshots": [
        {
            "id": "7c4e1ef4-bf1b-41ab-a0c8-ce01f4ffdfa1",
            "name": "vol_snapshot",
            "status": "IN_PROGRESS",
            "status_reason": null,
            "creation_time": "2015-08-04T20:57:55Z",
            "data": null
        }
    ]
}
```

9.3.14. Show snapshot

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}`	Shows details for a snapshot.

Normal response codes: 200

9.3.14.1. Request

This table shows the URI parameters for the show snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{snapshot_id}`	UUID	The UUID of the `snapshot`.

This operation does not accept a request body.

9.3.14.2. Response

Example 9.17. Show snapshot: JSON response

```
{
    "snapshot": {
        "id": "7c4e1ef4-bf1b-41ab-a0c8-ce01f4ffdfa1",
        "name": "vol_snapshot",
        "status": "COMPLETE",
        "status_reason": "Stack SNAPSHOT completed successfully",
        "creation_time": "2015-08-04T20:57:55Z",
        "data": {
            "status": "COMPLETE",
            "name": "stack_vol1",
            "stack_user_project_id": "fffa11067b1c48129ddfb78fba2bf09f",
            "environment": {
                "parameters": {},
                "resource_registry": {
                    "resources": {}
                }
            },
            "template": {
                "heat_template_version": "2013-05-23",
                "resources": {
                    "volume": {
                        "type": "OS::Cinder::Volume",
                        "properties": {
                            "size": 1
                        }
                    }
                }
            },
```

```
            "action": "SNAPSHOT",
            "project_id": "ecdb08032cd042179692a1b148f6565e",
            "id": "656452c2-e151-40da-8704-c844e69b485c",
            "resources": {
                "volume": {
                    "status": "COMPLETE",
                    "name": "volume",
                    "resource_data": {
                        "backup_id": "99108cf8-398f-461b-a043-bdceb7c9f572"
                    },
                    "resource_id": "3ab8cf79-807b-4c40-b743-0655f91e072f",
                    "action": "SNAPSHOT",
                    "type": "OS::Cinder::Volume",
                    "metadata": {}
                }
            }
        }
    }
}
```

9.3.15. Delete snapshot

Method	URI	Description
DELETE	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}`	Deletes a stack snapshot.

Normal response codes: 204

9.3.15.1. Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{snapshot_id}`	UUID	The UUID of the `snapshot`.

This operation does not accept a request body.

9.3.16. Restore snapshot

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/snapshots/{snapshot_id}/restore`	Restores a stack snapshot.

You can restore only active stacks from a snapshot. You must recreate deleted stacks.

Normal response codes: 202

9.3.16.1. Request

This table shows the URI parameters for the restore snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

9.3.17. List outputs

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs`	Lists outputs for a stack.

Normal response codes: 200

9.3.17.1. Request

This table shows the URI parameters for the list outputs request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This operation does not accept a request body.

9.3.17.2. Response

Example 9.18. List outputs: JSON response

```
{
    "outputs": [
        {
            "output_key": "output name",
            "output_value": "output value",
            "description": "output description",
            "output_error": null
        }
    ]
}
```

9.3.18. Show output

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/outputs/{output_key}`	Shows details for a stack output.

Normal response codes: 200

9.3.18.1. Request

This table shows the URI parameters for the show output request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{output_key}`	String	The key of a stack output.

This operation does not accept a request body.

9.3.18.2. Response

Example 9.19. Show output: JSON response

```
{
    "output": {
        "output_key": "output_name",
        "output_value": "output_value",
        "description": "output description",
        "output_error": null
    }
}
```

9.4. Stack actions

Performs non-lifecycle operations on the stack. Specify the action in the request body.

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Suspends a stack.
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Resumes a suspended stack.
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Cancels a currently running update of a stack.
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Checks whether the resources are in expected states for a stack.

9.4.1. Suspend stack

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Suspends a stack.

Normal response codes: 200

9.4.1.1. Request

This table shows the URI parameters for the suspend stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.20. Suspend stack: JSON request

```
{
    "suspend": null
}
```

9.4.2. Resume stack

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Resumes a suspended stack.

Normal response codes: 200

9.4.2.1. Request

This table shows the URI parameters for the resume stack request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.21. Resume stack: JSON request

```
{
    "resume": null
}
```

9.4.3. Cancel stack update

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Cancels a currently running update of a stack.

Normal response codes: 200

9.4.3.1. Request

This table shows the URI parameters for the cancel stack update request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.22. Cancel stack update: JSON request

```
{
    "cancel_update": null
}
```

9.4.4. Check stack resources

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/actions`	Checks whether the resources are in expected states for a stack.

Normal response codes: 200

9.4.4.1. Request

This table shows the URI parameters for the check stack resources request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

Example 9.23. Check stack resources: JSON request

```
{
    "check": null
}
```

9.5. Stack resources

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources{?nested_depth,with_detail}`	Lists resources in a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}`	Shows data for a resource.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata`	Shows metadata for a resource.
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal`	Sends a signal to a resource.

9.5.1. List resources

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources{?nested_depth,with_detail}`	Lists resources in a stack.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

9.5.1.1. Request

This table shows the URI parameters for the list resources request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This table shows the query parameters for the list resources request:

Name	Type	Description
`nested_depth`	Int *(Optional)*	Includes resources from nested stacks up to the `nested_depth` levels of recursion.
`with_detail`	Boolean *(Optional)*	Enables detailed resource information for each resource in list of resources.

This operation does not accept a request body.

9.5.1.2. Response

Example 9.24. List resources: JSON response

Lists resources in a stack, in JSON format.

```
{
    "resources": [
        {
            "creation_time": "2015-06-25T14:59:53",
            "links": [
                {
                    "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-
ac4f-4f63-b58d-f0d047b1ba4c/resources/random_key_name",
                    "rel": "self"
                },
                {
                    "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-
ac4f-4f63-b58d-f0d047b1ba4c",
                    "rel": "stack"
                }
```

```
            ],
            "logical_resource_id": "random_key_name",
            "physical_resource_id": "mystack-random_key_name-pmjmy5pks735",
            "required_by": [],
            "resource_name": "random_key_name",
            "resource_status": "CREATE_COMPLETE",
            "resource_status_reason": "state changed",
            "resource_type": "OS::Heat::RandomString",
            "updated_time": "2015-06-25T14:59:53"
        }
    ]
}
```

9.5.2. Show resource data

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}`	Shows data for a resource.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

9.5.2.1. Request

This table shows the URI parameters for the show resource data request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{resource_name}`	String	The name of a resource in the stack.

This operation does not accept a request body.

9.5.2.2. Response

Example 9.25. Show resource data: JSON response

Shows resource data for a resource, in JSON format.

```
{
    "resource": {
        "attributes": {
            "value": "I9S20uIp"
        },
        "creation_time": "2015-06-25T14:59:53",
        "description": "",
        "links": [
            {
                "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-
ac4f-4f63-b58d-f0d047b1ba4c/resources/random_key_name",
                "rel": "self"
            },
            {
                "href": "http://hostname/v1/1234/stacks/mystack/629a32d0-
ac4f-4f63-b58d-f0d047b1ba4c",
                "rel": "stack"
            }
        ],
        "logical_resource_id": "random_key_name",
        "physical_resource_id": "mystack-random_key_name-pmjmy5pks735",
        "required_by": [],
        "resource_name": "random_key_name",
        "resource_status": "CREATE_COMPLETE",
```

```
        "resource_status_reason": "state changed",
        "resource_type": "OS::Heat::RandomString",
        "updated_time": "2015-06-25T14:59:53"
    }
}
```

9.5.3. Show resource metadata

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/metadata`	Shows metadata for a resource.

Normal response codes: 200

9.5.3.1. Request

This table shows the URI parameters for the show resource metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{resource_name}`	String	The name of a resource in the stack.

This operation does not accept a request body.

9.5.3.2. Response

Example 9.26. Show resource metadata: JSON response

```
{
    "metadata": {
        "some_key": "some_value",
        "some_other_key": "some_other_value"
    }
}
```

9.5.4. Send a signal to a resource

Method	URI	Description
POST	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/signal`	Sends a signal to a resource.

The contents of the request body depends on the resource to which you send a signal.

Some resources cannot receive signals. If you send them a signal, they return a 400 error code.

Normal response codes: 200

9.5.4.1. Request

This table shows the URI parameters for the send a signal to a resource request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{resource_name}`	String	The name of a resource in the stack.

This operation does not accept a request body.

9.6. Stack events

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/events`	Finds the canonical URL for the event list of a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events{?resource_action,resource_status,resource_name,resource_type,limit,marker,sort_keys,sort_dir}`	Lists events for a stack.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events{?resource_action,resource_status,resource_type,limit,marker,sort_keys,sort_dir}`	Lists events for a stack resource.
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}`	Shows details for an event.

9.6.1. Find stack events

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/events`	Finds the canonical URL for the event list of a stack.

Normal response codes: 302

9.6.1.1. Request

This table shows the URI parameters for the find stack events request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.

This operation does not accept a request body.

9.6.2. List stack events

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/events{?resource_action,resource_status,resource_name,resource_type,limit,marker,sort_keys,sort_dir}`	Lists events for a stack.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.6.2.1. Request

This table shows the URI parameters for the list stack events request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This table shows the query parameters for the list stack events request:

Name	Type	Description
`resource_action`	String *(Optional)*	Filters the event list by a resource action. You can use this filter multiple times to filter by multiple resource actions. Valid resource actions are `ADOPT`, `CHECK`, `CREATE`, `DELETE`, `INIT`, `RESTORE`, `RESUME`, `ROLLBACK`, `SNAPSHOT`, `SUSPEND`, and `UPDATE`.
`resource_status`	String *(Optional)*	Filters the event list by a resource status. You can use this filter multiple times to filter by multiple resource statuses. Valid resource statuses are `COMPLETE`, `FAILED`, and `IN_PROGRESS`.
`resource_name`	String *(Optional)*	Filters the event list by a resource name. You can use this filter multiple times to filter by multiple resource names.
`resource_type`	String *(Optional)*	Filters the event list by a resource type. You can use this filter multiple times to filter by multiple resource types. Valid resource types include `OS::Cinder::Volume`, `OS::Nova::Server`, `OS::Neutron::Port`, and so on.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`sort_keys`	String *(Optional)*	Sorts the list by the `resource_type` or `created_at` key.
`sort_dir`	String *(Optional)*	The sort direction of the list. A valid value is `asc` (ascending) or `desc` (descending).

This operation does not accept a request body.

9.6.2.2. Response

Example 9.27. List stack events: JSON response


```
{
    "events": [
        {
            "resource_name": "port",
            "event_time": "2014-07-23T08:14:47Z",
            "links": [
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port/events/474bfdf0-a450-46ec-a78a-0c7faa404073",
                    "rel": "self"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port",
                    "rel": "resource"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5",
                    "rel": "stack"
                }
            ],
            "logical_resource_id": "port",
            "resource_status": "CREATE_FAILED",
            "resource_status_reason": "NotFound: Subnet
 f8a699d0-3537-429e-87a5-6b5a8d0c2bf0 could not be found",
            "physical_resource_id": null,
            "id": "474bfdf0-a450-46ec-a78a-0c7faa404073"
        },
        {
            "resource_name": "port",
            "event_time": "2014-07-23T08:14:47Z",
            "links": [
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port/events/66fa95b6-e6f8-4f05-b1af-e828f5aba04c",
                    "rel": "self"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port",
                    "rel": "resource"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5",
                    "rel": "stack"
```

```
                }
            ],
            "logical_resource_id": "port",
            "resource_status": "CREATE_IN_PROGRESS",
            "resource_status_reason": "state changed",
            "physical_resource_id": null,
            "id": "66fa95b6-e6f8-4f05-b1af-e828f5aba04c"
        }
    ]
}
```

9.6.3. List resource events

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events{?resource_action,resource_status,resource_type,limit,marker,sort_keys,sort_dir}`	Lists events for a stack resource.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

9.6.3.1. Request

This table shows the URI parameters for the list resource events request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{resource_name}`	String	The name of a resource in the stack.

This table shows the query parameters for the list resource events request:

Name	Type	Description
`resource_action`	String *(Optional)*	Filters the event list by a resource action. You can use this filter multiple times to filter by multiple resource actions. Valid resource actions are `ADOPT`, `CHECK`, `CREATE`, `DELETE`, `INIT`, `RESTORE`, `RESUME`, `ROLLBACK`, `SNAPSHOT`, `SUSPEND`, and `UPDATE`.
`resource_status`	String *(Optional)*	Filters the event list by a resource status. You can use this filter multiple times to filter by multiple resource statuses. Valid resource statuses are `COMPLETE`, `FAILED`, and `IN_PROGRESS`.
`resource_type`	String *(Optional)*	Filters the event list by a resource type. You can use this filter multiple times to filter by multiple resource types. Valid resource types include `OS::Cinder::Volume`, `OS::Nova::Server`, `OS::Neutron::Port`, and so on.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`marker`	String *(Optional)*	The ID of the last-seen item. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.
`sort_keys`	String *(Optional)*	Sorts the list by the `resource_type` or `created_at` key.
`sort_dir`	String *(Optional)*	The sort direction of the list. A valid value is `asc` (ascending) or `desc` (descending).

This operation does not accept a request body.

9.6.3.2. Response

Example 9.28. List resource events: JSON response


```
{
    "events": [
        {
            "resource_name": "port",
            "event_time": "2014-07-23T08:14:47Z",
            "links": [
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port/events/474bfdf0-a450-46ec-a78a-0c7faa404073",
                    "rel": "self"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port",
                    "rel": "resource"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5",
                    "rel": "stack"
                }
            ],
            "logical_resource_id": "port",
            "resource_status": "CREATE_FAILED",
            "resource_status_reason": "NotFound: Subnet
 f8a699d0-3537-429e-87a5-6b5a8d0c2bf0 could not be found",
            "physical_resource_id": null,
            "id": "474bfdf0-a450-46ec-a78a-0c7faa404073"
        },
        {
            "resource_name": "port",
            "event_time": "2014-07-23T08:14:47Z",
            "links": [
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port/events/66fa95b6-e6f8-4f05-b1af-e828f5aba04c",
                    "rel": "self"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5/resources/port",
                    "rel": "resource"
                },
                {
                    "href": "http://192.168.123.200:8004/v1/
dc4b074874244f7693dd65583733a758/stacks/aws_port/db467ed1-50b5-4a3e-
aeb1-396ff1d151c5",
                    "rel": "stack"
```

```
                }
            ],
            "logical_resource_id": "port",
            "resource_status": "CREATE_IN_PROGRESS",
            "resource_status_reason": "state changed",
            "physical_resource_id": null,
            "id": "66fa95b6-e6f8-4f05-b1af-e828f5aba04c"
        }
    ]
}
```

9.6.4. Show event details

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/resources/{resource_name}/events/{event_id}`	Shows details for an event.

Normal response codes: 200

9.6.4.1. Request

This table shows the URI parameters for the show event details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.
`{resource_name}`	String	The name of a resource in the stack.
`{event_id}`	UUID	The UUID of the event that is related to the resource in the stack.

This operation does not accept a request body.

9.6.4.2. Response

Example 9.29. Show event details: JSON response

```
{
    "event": {
        "event_time": "2015-06-25T14:59:53",
        "id": "8db23e2e-72b2-47a2-9ed9-b52417f56e50",
        "links": [
            {
                "href": "http://hostname/v1/1234/stacks/mystack/56789/
resources/random_key_name/events/8db23e2e-72b2-47a2-9ed9-b52417f56e50",
                "rel": "self"
            },
            {
                "href": "http://hostname/v1/1234/stacks/mystack/56789/
resources/random_key_name",
                "rel": "resource"
            },
            {
                "href": "http://hostname/v1/1234/stacks/mystack/56789",
                "rel": "stack"
            }
        ],
        "logical_resource_id": "random_key_name",
        "physical_resource_id": null,
        "resource_name": "random_key_name",
        "resource_properties": {
            "character_classes": null,
            "character_sequences": null,
```

```
            "length": 8,
            "salt": null,
            "sequence": null
        },
        "resource_status": "CREATE_IN_PROGRESS",
        "resource_status_reason": "state changed",
        "resource_type": "OS::Heat::RandomString"
    }
}
```

9.7. Templates

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template`	Gets a template for a stack.
GET	`/v1/{tenant_id}/template_versions`	Lists all available template versions.
POST	`/v1/{tenant_id}/validate`	Validates a template.
GET	`/v1/{tenant_id}/resource_types/{type_name}/template{?template_type}`	Shows the template representation for a resource type.
GET	`/v1/{tenant_id}/resource_types/{type_name}`	Shows the interface schema for a resource type.
GET	`/v1/{tenant_id}/resource_types{?name,version,support_status}`	Lists all supported template resource types.

9.7.1. Get stack template

Method	URI	Description
GET	`/v1/{tenant_id}/stacks/{stack_name}/{stack_id}/template`	Gets a template for a stack.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.7.1.1. Request

This table shows the URI parameters for the get stack template request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{stack_name}`	String	The name of a stack.
`{stack_id}`	UUID	The UUID of the stack.

This operation does not accept a request body.

9.7.1.2. Response

Example 9.30. Get stack template: JSON response

Returns the template for a stack.

```
{
    "description": "Hello world HOT template that just defines a single
 server. Contains just base features to verify base HOT support.\n",
    "heat_template_version": "2013-05-23",
    "outputs": {
        "foo": {
            "description": "Show foo parameter value",
            "value": {
                "get_param": "foo"
            }
        }
    },
    "parameters": {
        "foo": {
            "default": "secret",
            "description": "Name of an existing key pair to use for the
 server",
            "hidden": true,
            "type": "string"
        }
    },
    "resources": {
        "random_key_name": {
            "properties": {
                "length": 8
```

```
            },
            "type": "OS::Heat::RandomString"
        }
    }
}
```

9.7.2. List template versions

Method	URI	Description
GET	`/v1/{tenant_id}/template_versions`	Lists all available template versions.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.7.2.1. Request

This table shows the URI parameters for the list template versions request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

9.7.2.2. Response

Example 9.31. List template versions: JSON response

```
{
    "template_versions": [
        {
            "version": "heat_template_version.2014-10-16",
            "type": "hot"
        },
        {
            "version": "heat_template_version.2015-04-30",
            "type": "hot"
        },
        {
            "version": "HeatTemplateFormatVersion.2012-12-12",
            "type": "cfn"
        },
        {
            "version": "heat_template_version.2015-10-15",
            "type": "hot"
        },
        {
            "version": "AWSTemplateFormatVersion.2010-09-09",
            "type": "cfn"
        },
        {
            "version": "heat_template_version.2013-05-23",
            "type": "hot"
        },
        {
            "version": "heat_template_version.2016-04-08",
            "type": "hot"
        }
    ]
```

```
}
```

9.7.3. Validate template

Method	URI	Description
POST	`/v1/{tenant_id}/validate`	Validates a template.

Normal response codes: 200

9.7.3.1. Request

This table shows the URI parameters for the validate template request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 9.32. Validate template: JSON request

```
{
    "template_url": "/PATH_TO_HEAT_TEMPLATES/WordPress_Single_Instance.
template"
}
```

9.7.3.2. Response

Example 9.33. Validate template: JSON response

```
{
    "Description": "A template that provides a single server instance.",
    "Parameters": {
        "server-size": {
            "default": "1GB Standard Instance",
            "description": "Server size",
            "type": "String",
            "constraints": [
                {
                    "allowed_values": [
                        "512MB Standard Instance",
                        "1GB Standard Instance",
                        "4GB Standard Instance",
                        "8GB Standard Instance"
                    ],
                    "description": "Must be a valid server size."
                }
            ]
        },
        "key_name": {
            "description": "Keypair name for SSH access to the server",
            "required": true,
            "type": "String"
        },
        "server_name": {
            "default": "My server",
            "description": "My server",
            "type": "String"
        }
```

```
    },
    "ParameterGroups": [
        {
            "label": "Parameter groups",
            "description": "My parameter groups",
            "parameters": [
                "param_name-1",
                "param_name-2"
            ]
        }
    ]
}
```

9.7.4. Show resource template

Method	URI	Description
GET	`/v1/{tenant_id}/resource_types/{type_name}/template{?template_type}`	Shows the template representation for a resource type.

The returned template contains a single resource type. Each resource property is mapped to a template parameter and each resource attribute is mapped to a template output.

You can use these templates as a starting place for creating customized, template-based resources or as examples of using the particular resource in another template.

Use the `template_type` query parameter to specify the resource template type. Default type is `cfn`. The `hot` template type is supported. For example:

```
/v1/{tenant_id}/resource_types/{type_name}/template?template_type=cfn
```

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

9.7.4.1. Request

This table shows the URI parameters for the show resource template request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{type_name}`	String	The name of a resource type.

This operation does not accept a request body.

9.7.4.2. Response

Example 9.34. Show resource template: JSON response

```
{
    "HeatTemplateFormatVersion": "2012-12-12",
    "Outputs": {
        "private_key": {
            "Description": "The private key if it has been saved.",
            "Value": "{\"Fn::GetAtt\": [\"KeyPair\", \"private_key\"]}"
        },
        "public_key": {
            "Description": "The public key.",
            "Value": "{\"Fn::GetAtt\": [\"KeyPair\", \"public_key\"]}"
        }
    },
    "Parameters": {
        "name": {
            "Description": "The name of the key pair.",
            "Type": "String"
        },
```

```
        "public_key": {
            "Description": "The optional public key. This allows users to
 supply the public key from a pre-existing key pair. If not supplied, a new
 key pair will be generated.",
            "Type": "String"
        },
        "save_private_key": {
            "AllowedValues": [
                true,
                "true",
                false,
                "false"
            ],
            "Default": false,
            "Description": "true if the system should remember a generated
 private key; false otherwise.",
            "Type": "String"
        }
    },
    "Resources": {
        "KeyPair": {
            "Properties": {
                "name": {
                    "Ref": "name"
                },
                "public_key": {
                    "Ref": "public_key"
                },
                "save_private_key": {
                    "Ref": "save_private_key"
                }
            },
            "Type": "OS::Nova::KeyPair"
        }
    }
}
```

9.7.5. Show resource schema

Method	URI	Description
GET	`/v1/{tenant_id}/resource_types/{type_name}`	Shows the interface schema for a resource type.

A schema describes the properties that can be set on the resource, their types, constraints, descriptions, and default values. Additionally, the response shows the resource attributes and their descriptions.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401)

9.7.5.1. Request

This table shows the URI parameters for the show resource schema request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{type_name}`	String	The name of a resource type.

This operation does not accept a request body.

9.7.5.2. Response

Example 9.35. Show resource schema: JSON response

```
{
    "attributes": {
        "an_attribute": {
            "description": "A runtime value of the resource."
        }
    },
    "properties": {
        "a_property": {
            "constraints": [
                {
                    "description": "Must be between 1 and 255 characters",
                    "length": {
                        "max": 255,
                        "min": 1
                    }
                }
            ],
            "description": "A resource description.",
            "required": true,
            "type": "string",
            "update_allowed": false
        }
    },
    "resource_type": "OS::Heat::AResourceName",
    "support_status": {
```

```
        "message": "A status message",
        "status": "SUPPORTED",
        "version": "2014.1"
    }
}
```

9.7.6. List resource types

Method	URI	Description
GET	`/v1/{tenant_id}/resource_types{?name,version,support_status}`	Lists all supported template resource types.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401)

9.7.6.1. Request

This table shows the URI parameters for the list resource types request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This table shows the query parameters for the list resource types request:

Name	Type	Description
`name`	String *(Optional)*	Filters the response by a resource type name. You can set this value to a regular expression. Use this filter multiple times to filter by multiple names.
`version`	String *(Optional)*	Filters the response by resource types by heat version. For example, `2016.1` or `5.0.0`. Use this filter multiple times to filter by multiple versions.
`support_status`	String *(Optional)*	Filters the response by resource types by a support status. Valid support status values are `UNKNOWN`, `SUPPORTED`, `DEPRECATED`, `UNSUPPORTED`, and `HIDDEN`. Use this filter multiple times to filter by multiple support statuses.

This operation does not accept a request body.

9.7.6.2. Response

Example 9.36. List resource types: JSON response

```
{
    "resource_types": [
        "AWS::EC2::Instance",
        "OS::Heat::ScalingPolicy",
        "AWS::CloudFormation::Stack",
        "OS::Keystone::Group",
        "OS::Glance::Image",
        "AWS::EC2::Volume",
        "OS::Heat::SoftwareDeployment",
        "AWS::AutoScaling::ScalingPolicy",
        "AWS::EC2::InternetGateway",
        "OS::Heat::SoftwareDeployments",
        "AWS::EC2::VolumeAttachment",
        "AWS::CloudFormation::WaitConditionHandle",
        "OS::Cinder::VolumeAttachment",
        "OS::Cinder::EncryptedVolumeType",
```

```
        "OS::Heat::AutoScalingGroup",
        "OS::Nova::FloatingIP",
        "OS::Heat::HARestarter",
        "OS::Keystone::Project",
        "OS::Keystone::Endpoint",
        "OS::Heat::InstanceGroup",
        "AWS::CloudWatch::Alarm",
        "AWS::AutoScaling::AutoScalingGroup",
        "OS::Heat::CloudConfig",
        "OS::Heat::SoftwareComponent",
        "OS::Cinder::Volume",
        "OS::Keystone::Service",
        "OS::Heat::WaitConditionHandle",
        "OS::Heat::SoftwareConfig",
        "AWS::CloudFormation::WaitCondition",
        "OS::Heat::StructuredDeploymentGroup",
        "OS::Heat::RandomString",
        "OS::Heat::SoftwareDeploymentGroup",
        "OS::Nova::KeyPair",
        "OS::Heat::MultipartMime",
        "OS::Heat::UpdateWaitConditionHandle",
        "OS::Nova::Server",
        "AWS::IAM::AccessKey",
        "AWS::EC2::SecurityGroup",
        "AWS::EC2::EIPAssociation",
        "AWS::EC2::EIP",
        "OS::Heat::AccessPolicy",
        "AWS::IAM::User",
        "OS::Heat::WaitCondition",
        "OS::Heat::StructuredDeployment",
        "AWS::RDS::DBInstance",
        "AWS::AutoScaling::LaunchConfiguration",
        "OS::Heat::Stack",
        "OS::Nova::FloatingIPAssociation",
        "OS::Heat::ResourceGroup",
        "OS::Heat::StructuredConfig",
        "OS::Nova::ServerGroup",
        "OS::Heat::StructuredDeployments",
        "OS::Keystone::Role",
        "OS::Keystone::User",
        "AWS::ElasticLoadBalancing::LoadBalancer",
        "OS::Nova::Flavor",
        "OS::Cinder::VolumeType"
    ]
}
```

9.8. Build info

Method	URI	Description
GET	`/v1/{tenant_id}/build_info`	Shows build information for an Orchestration deployment.

9.8.1. Show build information

Method	URI	Description
GET	`/v1/{tenant_id}/build_info`	Shows build information for an Orchestration deployment.

Normal response codes: 200

9.8.1.1. Request

This table shows the URI parameters for the show build information request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

9.8.1.2. Response

Example 9.37. Show build information: JSON response

```
{
    "api": {
        "revision": "{api_build_revision}"
    },
    "engine": {
        "revision": "{engine_build_revision}"
    }
}
```

9.9. Software configuration

Method	URI	Description
POST	`/v1/{tenant_id}/software_configs`	Creates a software configuration.
GET	`/v1/{tenant_id}/software_configs/{config_id}`	Shows details for a software configuration.
DELETE	`/v1/{tenant_id}/software_configs/{config_id}`	Deletes a software configuration.
GET	`/v1/{tenant_id}/software_deployments`	Lists all available software deployments.
POST	`/v1/{tenant_id}/software_deployments`	Creates a software deployment.
GET	`/v1/{tenant_id}/software_deployments/metadata/{server_id}`	Shows the deployment configuration metadata for a server.
GET	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Shows details for a software deployment.
PUT	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Updates a software deployment.

Method	URI	Description
DELETE	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Deletes a software deployment.

9.9.1. Create configuration

Method	URI	Description
POST	`/v1/{tenant_id}/software_configs`	Creates a software configuration.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

9.9.1.1. Request

This table shows the URI parameters for the create configuration request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 9.38. Create configuration: JSON request

```
{
    "inputs": [
        {
            "default": null,
            "type": "String",
            "name": "foo",
            "description": null
        },
        {
            "default": null,
            "type": "String",
            "name": "bar",
            "description": null
        }
    ],
    "group": "script",
    "name": "a-config-we5zpvyu7b5o",
    "outputs": [
        {
            "type": "String",
            "name": "result",
            "error_output": false,
            "description": null
        }
    ],
    "config": "#!/bin/sh -x\necho \"Writing to /tmp/$bar\"\necho $foo > /
tmp/$bar\necho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
 $deploy_server_id during $deploy_action\" > $heat_outputs_path.result\necho \
"Written to /tmp/$bar\"\necho \"Output to stderr\" 1>&2",
    "options": null
}
```

9.9.1.2. Response

Example 9.39. Create configuration: JSON response

```
{
```

```
    "software_config": {
        "creation_time": "2015-01-31T15:12:36Z",
        "inputs": [
            {
                "default": null,
                "type": "String",
                "name": "foo",
                "description": null
            },
            {
                "default": null,
                "type": "String",
                "name": "bar",
                "description": null
            }
        ],
        "group": "script",
        "name": "a-config-we5zpvyu7b5o",
        "outputs": [
            {
                "type": "String",
                "name": "result",
                "error_output": false,
                "description": null
            }
        ],
        "options": null,
        "config": "#!/bin/sh -x\necho \"Writing to /tmp/$bar\"\necho $foo > /
tmp/$bar\necho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
 $deploy_server_id during $deploy_action\" > $heat_outputs_path.result\necho \
"Written to /tmp/$bar\"\necho \"Output to stderr\" 1>&2",
        "id": "ddee7aca-aa32-4335-8265-d436b20db4f1"
    }
}
```

9.9.2. Show configuration details

Method	URI	Description
GET	`/v1/{tenant_id}/software_configs/{config_id}`	Shows details for a software configuration.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404)

9.9.2.1. Request

This table shows the URI parameters for the show configuration details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{config_id}`	UUID	The UUID of the configuration.

This operation does not accept a request body.

9.9.2.2. Response

Example 9.40. Show configuration details: JSON response


```
{
    "software_config": {
        "inputs": [
            {
                "default": null,
                "type": "String",
                "name": "foo",
                "description": null
            },
            {
                "default": null,
                "type": "String",
                "name": "bar",
                "description": null
            }
        ],
        "group": "script",
        "name": "a-config-we5zpvyu7b5o",
        "outputs": [
            {
                "type": "String",
                "name": "result",
                "error_output": false,
                "description": null
            }
        ],
        "creation_time": "2015-01-31T15:12:36Z",
        "id": "ddee7aca-aa32-4335-8265-d436b20db4f1",
        "config": "#!/bin/sh -x\necho \"Writing to /tmp/$bar\"\necho $foo > /
tmp/$bar\necho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
```

```
 $deploy_server_id during $deploy_action\" > $heat_outputs_path.result\necho \
"Written to /tmp/$bar\"\necho \"Output to stderr\" 1>&2",
        "options": null
    }
}
```

9.9.3. Delete config

Method	URI	Description
DELETE	`/v1/{tenant_id}/software_configs/{config_id}`	Deletes a software configuration.

Normal response codes: 204

9.9.3.1. Request

This table shows the URI parameters for the delete config request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{config_id}`	UUID	The UUID of the configuration.

This operation does not accept a request body.

9.9.4. List deployments

Method	URI	Description
GET	`/v1/{tenant_id}/software_deployments`	Lists all available software deployments.

Normal response codes: 200

Error response codes: badRequest (400), unauthorized (401), itemNotFound (404), internalServerError (500)

9.9.4.1. Request

This table shows the URI parameters for the list deployments request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

9.9.4.2. Response

Example 9.41. List deployments: JSON response

```
{
    "software_deployments": [
        {
            "status": "COMPLETE",
            "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
            "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
            "output_values": {
                "deploy_stdout": "Writing to /tmp/barmy\nWritten to /tmp/barmy
\n",
                "deploy_stderr": "+ echo Writing to /tmp/barmy\n+ echo fu
\n+ cat /tmp/barmy\n+ echo -n The file /tmp/barmy contains fu for server
 ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\n+ echo Written to /tmp/
barmy\n+ echo Output to stderr\nOutput to stderr\n",
                "deploy_status_code": 0,
                "result": "The file /tmp/barmy contains fu for server
 ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
            },
            "input_values": null,
            "action": "CREATE",
            "status_reason": "Outputs received",
            "id": "ef422fa5-719a-419e-a10c-72e3a367b0b8",
            "creation_time": "2015-01-31T15:12:36Z",
            "updated_time": "2015-01-31T15:18:21Z"
        }
    ]
}
```

9.9.5. Create deployment

Method	URI	Description
POST	`/v1/{tenant_id}/software_deployments`	Creates a software deployment.

Normal response codes: 200

9.9.5.1. Request

This table shows the URI parameters for the create deployment request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

Example 9.42. Create deployment: JSON request

```
{
    "status": "IN_PROGRESS",
    "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
    "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
    "stack_user_project_id": "c024bfada67845ddb17d2b0c0be8cd79",
    "action": "CREATE",
    "status_reason": "Deploy data available"
}
```

9.9.5.2. Response

Example 9.43. Create deployment: JSON response

```
{
    "software_deployment": {
        "status": "IN_PROGRESS",
        "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
        "config_id": "8da95794-2ad9-4979-8ae5-739ce314c5cd",
        "output_values": null,
        "input_values": null,
        "action": "CREATE",
        "status_reason": "Deploy data available",
        "id": "ef422fa5-719a-419e-a10c-72e3a367b0b8",
        "creation_time": "2015-01-31T15:12:36Z",
        "updated_time": "2015-01-31T15:18:21Z"
    }
}
```

9.9.6. Show server configuration metadata

Method	URI	Description
GET	`/v1/{tenant_id}/software_deployments/metadata/{server_id}`	Shows the deployment configuration metadata for a server.

Use the `group` property to specify the configuration hook to which the pass the metadata item.

Normal response codes: 200

9.9.6.1. Request

This table shows the URI parameters for the show server configuration metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{server_id}`	UUID	The UUID of the server for which to fetch configuration metadata.

This operation does not accept a request body.

9.9.6.2. Response

Example 9.44. Show server configuration metadata: JSON response

```
{
    "metadata": [
        {
            "inputs": [
                {
                    "default": null,
                    "type": "String",
                    "name": "foo",
                    "value": "fooooo",
                    "description": null
                },
                {
                    "default": null,
                    "type": "String",
                    "name": "bar",
                    "value": "baaaaa",
                    "description": null
                },
                {
                    "type": "String",
                    "name": "deploy_server_id",
                    "value": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
                    "description": "ID of the server being deployed to"
                },
                {
                    "type": "String",
                    "name": "deploy_action",
                    "value": "CREATE",
                    "description": "Name of the current action being deployed"
                },
```

```
                {
                    "type": "String",
                    "name": "deploy_stack_id",
                    "value": "a/9bd57090-8954-48ab-bab9-adf9e1ac70fc",
                    "description": "ID of the stack this deployment belongs
 to"
                },
                {
                    "type": "String",
                    "name": "deploy_resource_name",
                    "value": "deployment",
                    "description": "Name of this deployment resource in the
 stack"
                },
                {
                    "type": "String",
                    "name": "deploy_signal_id",
                    "value": "http://192.168.20.103:8000/v1/signal/arn
%3Aopenstack%3Aheat%3A%3Ae2a84fbdaeb047ae8da4b503f3b69f1f%3Astacks%2Fa
%2F9bd57090-8954-48ab-bab9-adf9e1ac70fc%2Fresources%2Fdeployment?Timestamp=
2014-03-19T20%3A30%3A59Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=
ca3571413e4a49998d580215517b3685&SignatureVersion=2&Signature=w6Iu
%2BNbg86mqwSOUf1GLuKPO7KaD82PiGpL4ig9Q1l4%3D",
                    "description": "ID of signal to use for signalling output
 values"
                }
            ],
            "group": "script",
            "name": "a-config-we5zpvyu7b5o",
            "outputs": [
                {
                    "type": "String",
                    "name": "result",
                    "error_output": false,
                    "description": null
                }
            ],
            "options": null,
            "creation_time": "2015-01-31T15:12:36Z",
            "updated_time": "2015-01-31T15:18:21Z",
            "config": "#!/bin/sh -x\necho \"Writing to /tmp/$bar\"\necho $foo
 > /tmp/$bar\necho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
 $deploy_server_id during $deploy_action\" > $heat_outputs_path.result\necho \
"Written to /tmp/$bar\"\necho \"Output to stderr\" 1>&2",
            "id": "3d5ec2a8-7004-43b6-a7f6-542bdbe9d434"
        },
        {
            "inputs": [
                {
                    "default": null,
                    "type": "String",
                    "name": "foo",
                    "value": "fu",
                    "description": null
                },
                {
                    "default": null,
                    "type": "String",
                    "name": "bar",
                    "value": "barmy",
```

```
                    "description": null
                },
                {
                    "type": "String",
                    "name": "deploy_server_id",
                    "value": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
                    "description": "ID of the server being deployed to"
                },
                {
                    "type": "String",
                    "name": "deploy_action",
                    "value": "CREATE",
                    "description": "Name of the current action being deployed"
                },
                {
                    "type": "String",
                    "name": "deploy_stack_id",
                    "value": "a/9bd57090-8954-48ab-bab9-adf9e1ac70fc",
                    "description": "ID of the stack this deployment belongs
 to"
                },
                {
                    "type": "String",
                    "name": "deploy_resource_name",
                    "value": "other_deployment",
                    "description": "Name of this deployment resource in the
 stack"
                },
                {
                    "type": "String",
                    "name": "deploy_signal_id",
                    "value": "http://192.168.20.103:8000/v1/signal/arn
%3Aopenstack%3Aheat%3A%3Ae2a84fbdaeb047ae8da4b503f3b69f1f%3Astacks%2Fa
%2F9bd57090-8954-48ab-bab9-adf9e1ac70fc%2Fresources%2Fother_deployment?
Timestamp=2014-03-19T20%3A30%3A59Z&SignatureMethod=HmacSHA256&AWSAccessKeyId=
7b761482f8254946bcd3d5ccb36fe939&SignatureVersion=2&Signature=giMfv%2BhrAw6y
%2FCMKQIQz2IhO5PkAj5%2BfP5YsL6rul3o%3D",
                    "description": "ID of signal to use for signalling output
 values"
                }
            ],
            "group": "script",
            "name": "a-config-we5zpvyu7b5o",
            "outputs": [
                {
                    "type": "String",
                    "name": "result",
                    "error_output": false,
                    "description": null
                }
            ],
            "options": null,
            "creation_time": "2015-01-31T16:14:13Z",
            "updated_time": "2015-01-31T16:18:19Z",
            "config": "#!/bin/sh -x\necho \"Writing to /tmp/$bar\"\necho $foo
 > /tmp/$bar\necho -n \"The file /tmp/$bar contains `cat /tmp/$bar` for server
 $deploy_server_id during $deploy_action\" > $heat_outputs_path.result\necho \
"Written to /tmp/$bar\"\necho \"Output to stderr\" 1>&2",
            "id": "8da95794-2ad9-4979-8ae5-739ce314c5cd"
        }
```

```
    ]
}
```

9.9.7. Show deployment details

Method	URI	Description
GET	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Shows details for a software deployment.

Normal response codes: 200

9.9.7.1. Request

This table shows the URI parameters for the show deployment details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{deployment_id}`	UUID	The UUID of the deployment.

This operation does not accept a request body.

9.9.7.2. Response

Example 9.45. Show deployment details: JSON response

```
{
    "software_deployment": {
        "status": "IN_PROGRESS",
        "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
        "config_id": "3d5ec2a8-7004-43b6-a7f6-542bdbe9d434",
        "output_values": null,
        "input_values": null,
        "action": "CREATE",
        "status_reason": "Deploy data available",
        "id": "06e87bcc-33a2-4bce-aebd-533e698282d3",
        "creation_time": "2015-01-31T15:12:36Z",
        "updated_time": "2015-01-31T15:18:21Z"
    }
}
```

9.9.8. Update deployment

Method	URI	Description
PUT	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Updates a software deployment.

Normal response codes: 200

9.9.8.1. Request

This table shows the URI parameters for the update deployment request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{deployment_id}`	UUID	The UUID of the deployment.

Example 9.46. Update deployment: JSON request

```
{
    "status": "COMPLETE",
    "output_values": {
        "deploy_stdout": "Writing to /tmp/baaaaa\nWritten to /tmp/baaaaa\n",
        "deploy_stderr": "+ echo Writing to /tmp/baaaaa\n+ echo fooooo\n+
 cat /tmp/baaaaa\n+ echo -n The file /tmp/baaaaa contains fooooo for server
 ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\n+ echo Written to /tmp/
baaaaa\n+ echo Output to stderr\nOutput to stderr\n",
        "deploy_status_code": 0,
        "result": "The file /tmp/baaaaa contains fooooo for server
 ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
    },
    "status_reason": "Outputs received"
}
```

9.9.8.2. Response

Example 9.47. Update deployment: JSON response

```
{
    "software_deployment": {
        "status": "COMPLETE",
        "server_id": "ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5",
        "config_id": "3d5ec2a8-7004-43b6-a7f6-542bdbe9d434",
        "output_values": {
            "deploy_stdout": "Writing to /tmp/baaaaa\nWritten to /tmp/baaaaa\
n",
            "deploy_stderr": "+ echo Writing to /tmp/baaaaa\n+ echo fooooo\n
+ cat /tmp/baaaaa\n+ echo -n The file /tmp/baaaaa contains fooooo for server
 ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE\n+ echo Written to /tmp/
baaaaa\n+ echo Output to stderr\nOutput to stderr\n",
            "deploy_status_code": 0,
            "result": "The file /tmp/baaaaa contains fooooo for server
 ec14c864-096e-4e27-bb8a-2c2b4dc6f3f5 during CREATE"
        },
```

```
        "input_values": null,
        "action": "CREATE",
        "status_reason": "Outputs received",
        "id": "06e87bcc-33a2-4bce-aebd-533e698282d3",
        "creation_time": "2015-01-31T15:12:36Z",
        "updated_time": "2015-01-31T15:18:21Z"
    }
}
```

9.9.9. Delete deployment

Method	URI	Description
DELETE	`/v1/{tenant_id}/software_deployments/{deployment_id}`	Deletes a software deployment.

Normal response codes: 204

9.9.9.1. Request

This table shows the URI parameters for the delete deployment request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.
`{deployment_id}`	UUID	The UUID of the deployment.

This operation does not accept a request body.

9.10. Manage service

Method	URI	Description
GET	`/v1/{tenant_id}/services`	Enables administrative users to view details for all orchestration engines.

9.10.1. Show orchestration engine status

Method	URI	Description
GET	`/v1/{tenant_id}/services`	Enables administrative users to view details for all orchestration engines.

Orchestration engine details include `engine_id`, topic name, last updated time, health status, and host name.

Troubleshooting

- A `503` error code indicates that the heat engines are not operational. Run the **heat-manage service list** command or contact your cloud provider to determine why the heat engines are not operational.

Normal response codes: 200

Error response codes: 503, 403

9.10.1.1. Request

This table shows the URI parameters for the show orchestration engine status request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant. A tenant is also known as an account or project.

This operation does not accept a request body.

9.10.1.2. Response

Example 9.48. Show orchestration engine status: JSON response

```
{
    "services": [
        {
            "status": "up",
            "binary": "heat-engine",
            "report_interval": 60,
            "engine_id": "9d9242c3-4b9e-45e1-9e74-7615fbf20e5d",
            "created_at": "2015-02-03T05:55:59.000000",
            "hostname": "mrkanag",
            "updated_at": "2015-02-03T05:57:59.000000",
            "topic": "engine",
            "host": "engine-1",
            "deleted_at": null,
            "id": "e1908f44-42f9-483f-b778-bc814072c33d"
        },
        {
            "status": "down",
            "binary": "heat-engine",
            "report_interval": 60,
            "engine_id": "2d2434bf-adb6-4453-9c6b-b22fb8bd2306",
            "created_at": "2015-02-03T06:03:14.000000",
```

```
            "hostname": "mrkanag",
            "updated_at": "2015-02-03T06:09:55.000000",
            "topic": "engine",
            "host": "engine",
            "deleted_at": null,
            "id": "582b5657-6db7-48ad-8483-0096350faa21"
        }
    ]
}
```

10. Shared File Systems API v2 (CURRENT)

Provides coordinated access to shared or distributed file systems.

The Shared File Systems API uses Compute-style micro-versions. Use the HTTP `X-Openstack-Manila-Api-Version` request header to specify a valid micro-version. For example, `"X-Openstack-Manila-Api-Version: 2.15"`. If you omit this header, the default micro-version is 2.0.

Other than the switch to the micro-versions approach, the Shared File Systems API v2.0 is functionally identical to the Shared File Systems API v1.

Subsequent API v2 micro-versions, such as v2.1, differ from API v1. Look for notes that identify in which post-v2 micro-version a feature, method, or parameter was introduced.

Latest micro-version for `Liberty` release of Manila is 2.6

Latest micro-version for `Mitaka` release of Manila is 2.15

History of all Manila API changes.

Method	URI	Description
API versions		
	/	
API extensions		
GET	/v2/{tenant_id}/extensions	Lists all extensions.
Limits		
GET	/v2/{tenant_id}/limits	Lists share limits.
Shares		
POST	/v2/{tenant_id}/shares	Creates a share.
GET	/v2/{tenant_id}/shares{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares.
GET	/v2/{tenant_id}/shares/detail{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}	Lists all shares, with details.
POST	/v2/{tenant_id}/shares/manage	Configures Shared File Systems to manage a share. This API is available for micro-versions later than or equal to 2.7
GET	/v2/{tenant_id}/shares/{share_id}	Shows details for a share.
PUT	/v2/{tenant_id}/shares/{share_id}	Updates a share.
DELETE	/v2/{tenant_id}/shares/{share_id}{?consistency_group_id}	Deletes a share.
Share export locations (since API v2.9)		

Method	URI	Description
GET	/v2/{tenant_id}/shares/{share_id}/export_locations	Lists all export locations for a share.
GET	/v2/{tenant_id}/shares/{share_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share.
Share metadata		
GET	/v2/{tenant_id}/shares/{share_id}/metadata	Shows the metadata for a share.
PUT	/v2/{tenant_id}/shares/{share_id}/metadata	Updates the metadata for a share.
POST	/v2/{tenant_id}/shares/{share_id}/metadata	Sets the metadata on a share.
DELETE	/v2/{tenant_id}/shares/{share_id}/metadata/{key}	Unsets the metadata on a share.
Share actions		
POST	/v2/{tenant_id}/shares/{share_id}/action	Grants access to a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Revokes access from a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Lists access rules for a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Administrator only. Explicitly updates the state of a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Administrator only. Force-deletes a share in any state.
POST	/v2/{tenant_id}/shares/{share_id}/action	Increases the size of a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Shrinks the size of a share.
POST	/v2/{tenant_id}/shares/{share_id}/action	Configures Shared File Systems to stop managing a share.
Share snapshots		
POST	/v2/{tenant_id}/snapshots	Creates a snapshot from a share.
GET	/v2/{tenant_id}/snapshots	Lists all share snapshots.
POST	/v2/{tenant_id}/snapshots/manage	(Since API v2.12) Configures Shared File Systems to manage a share snapshot.
GET	/v2/{tenant_id}/snapshots/detail	Lists all share snapshots with details.
GET	/v2/{tenant_id}/snapshots/{snapshot_id}	Shows details for a share snapshot.
PUT	/v2/{tenant_id}/snapshots/{snapshot_id}	Updates a share snapshot.
DELETE	/v2/{tenant_id}/snapshots/{snapshot_id}	Deletes a share snapshot.
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	(Since API v2.12) Configures Shared File Systems to stop managing a share snapshot.
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	Administrator only. Explicitly updates the state of a share snapshot.
POST	/v2/{tenant_id}/snapshots/{snapshot_id}/action	Administrator only. Force-deletes a share snapshot in any state.
Share networks		
POST	/v2/{tenant_id}/share-networks	Creates a share network.

Method	URI	Description
GET	/v2/{tenant_id}/share-networks	Lists all share networks.
GET	/v2/{tenant_id}/share-networks/detail	Lists all share networks with details.
GET	/v2/{tenant_id}/share-networks/{share_network_id}	Shows details for a share network.
PUT	/v2/{tenant_id}/share-networks/{share_network_id}	Updates a share network.
DELETE	/v2/{tenant_id}/share-networks/{share_network_id}	Deletes a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Adds a security service to a share network.
POST	/v2/{tenant_id}/share-networks/{share_network_id}/action	Removes a security service from a share network.
Security services		
POST	/v2/{tenant_id}/security-services	Creates a security service.
GET	/v2/{tenant_id}/security-services	Lists all security services.
GET	/v2/{tenant_id}/security-services/detail	Lists all security services with details.
GET	/v2/{tenant_id}/security-services/{security_service_id}	Shows details for a security service.
PUT	/v2/{tenant_id}/security-services/{security_service_id}	Updates a security service.
DELETE	/v2/{tenant_id}/security-services/{security_service_id}	Deletes a security service.
Share servers		
GET	/v2/{tenant_id}/share-servers	Lists all share servers.
GET	/v2/{tenant_id}/share-servers/{share_server_id}	Shows details for a share server.
DELETE	/v2/{tenant_id}/share-servers/{share_server_id}	Deletes a share server.
Share instances (since API v2.3)		
GET	/v2/{tenant_id}/share_instances	Lists all share instances.
GET	/v2/{tenant_id}/share_instances/{share_instance_id}	Shows details for a share instance.
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Explicitly updates the state of a share instance.
POST	/v2/{tenant_id}/share_instances/{share_instance_id}/action	Administrator only. Force-deletes a share instance.
Share instance export locations (since API v2.9)		
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations	Lists all export locations for a share instance.
GET	/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations/{export_location_id}	Show details of an export location belonging to a share instance.
Share types		
POST	/v2/{tenant_id}/types	Creates a share type.
GET	/v2/{tenant_id}/types	Lists all share types.
GET	/v2/{tenant_id}/types/default	Lists default share types.

Method	URI	Description
DELETE	`/v2/{tenant_id}/types/{share_type_id}`	Deletes a share type.
GET	`/v2/{tenant_id}/types/{share_type_id}/extra_specs`	Lists the extra specifications for a share type.
POST	`/v2/{tenant_id}/types/{share_type_id}/extra_specs`	Sets an extra specification for the share type.
DELETE	`/v2/{tenant_id}/types/{share_type_id}/extra_specs/{key}`	Unsets an extra specification for the share type.
POST	`/v2/{tenant_id}/types/{share_type_id}/action`	Adds share type access for a project.
POST	`/v2/{tenant_id}/types/{share_type_id}/action`	Removes share type access from a project.
GET	`/v2/{tenant_id}/types/{share_type_id}/share_type_access`	Shows access details for a share type.
Back-end storage pools		
GET	`/v2/{tenant_id}/scheduler-stats/pools`	Lists all back-end storage pools.
GET	`/v2/{tenant_id}/scheduler-stats/pools/detail`	Lists all storage pools for a back end, with details.
Services		
GET	`/v2/{tenant_id}/services`	Lists all services.
PUT	`/v2/{tenant_id}/services/enable`	Enables a service.
PUT	`/v2/{tenant_id}/services/disable`	Disables a service.
Availability zones		
GET	`/v2/{tenant_id}/availability-zones`	Lists all availability zones.
Manage share		
POST	`/v2/{tenant_id}/os-share-manage`	Configures Shared File Systems to manage a share.
POST	`/v2/{tenant_id}/os-share-unmanage/{share_id}/unmanage`	Configures Shared File Systems to stop managing a share.
Quota sets		
GET	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Shows quotas for a tenant.
PUT	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Updates quotas for a tenant.
DELETE	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Deletes quotas for a tenant. The quota reverts to the default quota.
GET	`/v2/{tenant_id}/quota-sets/{tenant_id}/defaults`	Shows default quotas for a tenant.

10.1. API versions

Lists information for all Shared File Systems API versions.

Method	URI	Description
	`/`	

10.1.1.

Method	URI	Description
	/	

10.1.1.1. Request

This operation does not accept a request body.

10.2. API extensions

Lists available Shared File Systems API extensions.

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.

10.2.1. List extensions

Method	URI	Description
GET	`/v2/{tenant_id}/extensions`	Lists all extensions.

Normal response codes: 200

10.2.1.1. Request

This table shows the header parameters for the list extensions request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list extensions request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.2.1.2. Response

Example 10.1. List extensions: JSON response

```
{
    "extensions": [
        {
            "alias": "os-extended-quotas",
            "updated": "2013-06-09T00:00:00+00:00",
            "name": "ExtendedQuotas",
            "links": [],
            "description": "Extend quotas. Adds ability for admins to delete
 quota and optionally force the update Quota command."
        },
        {
            "alias": "os-quota-sets",
            "updated": "2011-08-08T00:00:00+00:00",
            "name": "Quotas",
            "links": [],
            "description": "Quotas management support."
        },
        {
            "alias": "os-quota-class-sets",
            "updated": "2012-03-12T00:00:00+00:00",
            "name": "QuotaClasses",
            "links": [],
            "description": "Quota classes management support."
        },
        {
            "alias": "os-share-unmanage",
            "updated": "2015-02-17T00:00:00+00:00",
            "name": "ShareUnmanage",
            "links": [],
            "description": "Enable share unmanage operation."
```

```
        },
        {
            "alias": "os-types-manage",
            "updated": "2011-08-24T00:00:00+00:00",
            "name": "TypesManage",
            "links": [],
            "description": "Types manage support."
        },
        {
            "alias": "share-actions",
            "updated": "2012-08-14T00:00:00+00:00",
            "name": "ShareActions",
            "links": [],
            "description": "Enable share actions."
        },
        {
            "alias": "os-availability-zone",
            "updated": "2015-07-28T00:00:00+00:00",
            "name": "AvailabilityZones",
            "links": [],
            "description": "Describe Availability Zones."
        },
        {
            "alias": "os-user-quotas",
            "updated": "2013-07-18T00:00:00+00:00",
            "name": "UserQuotas",
            "links": [],
            "description": "Project user quota support."
        },
        {
            "alias": "os-share-type-access",
            "updated": "2015-03-02T00:00:00Z",
            "name": "ShareTypeAccess",
            "links": [],
            "description": "share type access support."
        },
        {
            "alias": "os-types-extra-specs",
            "updated": "2011-08-24T00:00:00+00:00",
            "name": "TypesExtraSpecs",
            "links": [],
            "description": "Type extra specs support."
        },
        {
            "alias": "os-admin-actions",
            "updated": "2015-08-03T00:00:00+00:00",
            "name": "AdminActions",
            "links": [],
            "description": "Enable admin actions."
        },
        {
            "alias": "os-used-limits",
            "updated": "2014-03-27T00:00:00+00:00",
            "name": "UsedLimits",
            "links": [],
            "description": "Provide data on limited resources that are being
used."
        },
        {
            "alias": "os-services",
```

```
            "updated": "2012-10-28T00:00:00-00:00",
            "name": "Services",
            "links": [],
            "description": "Services support."
        },
        {
            "alias": "os-share-manage",
            "updated": "2015-02-17T00:00:00+00:00",
            "name": "ShareManage",
            "links": [],
            "description": "Allows existing share to be 'managed' by Manila."
        }
    ]
}
```

10.3. Limits

Limits are the resource limitations that are allowed for each tenant (project). An administrator can configure limits in the `manila.conf` file.

Users can query their rate and absolute limits. The absolute limits contain information about:

- Total maximum share memory, in GBs.
- Number of share-networks.
- Number of share-snapshots.
- Number of shares.
- Shares and total used memory, in GBs.
- Snapshots and total used memory, in GBs.

Rate limits control the frequency at which users can issue specific API requests. Administrators use rate limiting to configure limits on the type and number of API calls that can be made in a specific time interval. For example, a rate limit can control the number of **GET** requests that can be processed during a one-minute period.

Method	URI	Description
GET	`/v2/{tenant_id}/limits`	Lists share limits.

10.3.1. List share limits

Method	URI	Description
GET	`/v2/{tenant_id}/limits`	Lists share limits.

Normal response codes: 200

10.3.1.1. Request

This table shows the header parameters for the list share limits request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share limits request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.3.1.2. Response

Example 10.2. List share limits: JSON response

```
{
    "limits": {
        "rate": [],
        "absolute": {
            "totalShareNetworksUsed": 0,
            "maxTotalShareGigabytes": 1000,
            "maxTotalShareNetworks": 10,
            "totalSharesUsed": 0,
            "totalShareGigabytesUsed": 0,
            "totalShareSnapshotsUsed": 0,
            "maxTotalShares": 50,
            "totalSnapshotGigabytesUsed": 0,
            "maxTotalSnapshotGigabytes": 1000,
            "maxTotalShareSnapshots": 50
        }
    }
}
```

10.4. Shares

A share is a remote, mountable file system. You can mount a share to and access a share from several hosts by several users at a time.

You can create a share and associate it with a network, list shares, and show information for, update, and delete a share.

To create a share, specify one of these supported protocols:

- `NFS`. Network File System (NFS).
- `CIFS`. Common Internet File System (CIFS).
- `GLUSTERFS`. Gluster file system (GlusterFS).
- `HDFS`. Hadoop Distributed File System (HDFS).
- `CEPHFS`. Ceph File System (CephFS).

You can also create snapshots of shares. To create a snapshot, you specify the ID of the share that you want to snapshot.

A share has one of these status values:

Table 10.1. Share statuses

Status	Description
`creating`	The share is being created.
`deleting`	The share is being deleted.
`error`	An error occurred during share creation.
`error_deleting`	An error occurred during share deletion.
`available`	The share is ready to use.
`manage_starting`	Share manage started.
`manage_error`	Share manage failed.
`unmanage_starting`	Share unmanage started.
`unmanage_error`	Share cannot be unmanaged.
`unmanaged`	Share was unmanaged.
`extending`	The extend, or increase, share size request was issued successfully.
`extending_error`	Extend share failed.
`shrinking`	Share is being shrunk.
`shrinking_error`	Failed to update quota on share shrinking.
`shrinking_possible_data_loss_error`	Shrink share failed due to possible data loss.

Method	URI	Description
POST	`/v2/{tenant_id}/shares`	Creates a share.
GET	`/v2/{tenant_id}/shares{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}`	Lists all shares.
GET	`/v2/{tenant_id}/shares/detail{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}`	Lists all shares, with details.

Method	URI	Description
POST	`/v2/{tenant_id}/shares/manage`	Configures Shared File Systems to manage a share. This API is available for micro-versions later than or equal to 2.7
GET	`/v2/{tenant_id}/shares/{share_id}`	Shows details for a share.
PUT	`/v2/{tenant_id}/shares/{share_id}`	Updates a share.
DELETE	`/v2/{tenant_id}/shares/{share_id}{?consistency_group_id}`	Deletes a share.

10.4.1. Create share

Method	URI	Description
POST	`/v2/{tenant_id}/shares`	Creates a share.

Normal response codes: 200

10.4.1.1. Request

This table shows the header parameters for the create share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the create share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.3. Create share: JSON request

```
{
    "share": {
        "description": "My custom share London",
        "share_type": null,
        "share_proto": "nfs",
        "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
        "name": "share_London",
        "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
        "snapshot_id": null,
        "is_public": true,
        "size": 1,
        "metadata": {
            "project": "my_app",
            "aim": "doc"
        }
    }
}
```

10.4.1.2. Response

Example 10.4. Create share: JSON response

```
{
    "share": {
        "status": null,
        "share_server_id": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "name": "share_London",
        "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
        "share_type_name": "default",
        "availability_zone": null,
```

```
        "created_at": "2015-09-18T10:25:24.533287",
        "export_location": null,
        "links": [
            {
                "href": "http://172.18.198.54:8786/v1/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
                "rel": "bookmark"
            }
        ],
        "share_network_id": null,
        "export_locations": [],
        "share_proto": "NFS",
        "host": null,
        "access_rules_status": "active",
        "has_replicas": false,
        "replication_type": null,
        "task_state": null,
        "snapshot_support": true,
        "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
        "source_cgsnapshot_member_id": null,
        "volume_type": "default",
        "snapshot_id": null,
        "is_public": true,
        "metadata": {
            "project": "my_app",
            "aim": "doc"
        },
        "id": "011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "size": 1,
        "description": "My custom share London"
    }
}
```

10.4.2. List shares

Method	URI	Description
GET	`/v2/{tenant_id}/shares{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}`	Lists all shares.

Normal response codes: 200

10.4.2.1. Request

This table shows the header parameters for the list shares request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list shares request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list shares request:

Name	Type	Description
`all_tenants`	Boolean *(Optional)*	(Admin only). Defines whether to list shares for all tenants. Set to `1` to list shares for all tenants. Set to `0` to list shares only for the current tenant.
`name`	String *(Optional)*	The share name.
`status`	String *(Optional)*	Filters by a share status. A valid value is `creating`, `error`, `available`, `deleting`, `error_deleting`, `manage_starting`, `manage_error`, `unmanage_starting`, `unmanage_error`, `unmanaged`, `extending`, `extending_error`, `shrinking`, `shrinking_error`, or `shrinking_possible_data_loss_error`.
`share_server_id`	UUID *(Optional)*	The UUID of the share server.
`metadata`	Dict *(Optional)*	One or more metadata key-value pairs, as a dictionary of strings.
`extra_specs`	String *(Optional)*	The extra specifications as a set of one or more key-value pairs. In each pair, the key is the name of the extra specification and the value is the share type that was used to create the share.
`share_type_id`	UUID *(Optional)*	(Since API v2.6) The UUID of the share type.

Name	Type	Description
limit	Int *(Optional)*	The maximum number of shares to return.
offset	Int *(Optional)*	The offset to define start point of share listing.
sort_key	String *(Optional)*	The key to sort a list of shares. A valid value is id, status, size, host, share_proto, export_location, availability_zone, user_id, project_id, created_at, updated_at, display_name, name, share_type_id, share_type, share_network_id, share_network, snapshot_id, or snapshot.
sort_dir	String *(Optional)*	The direction to sort a list of shares. A valid value is asc, or desc.
snapshot_id	UUID *(Optional)*	The UUID of the snapshot that was used to create the share.
host	String *(Optional)*	The share host name.
share_network_id	UUID *(Optional)*	The UUID of the share network.
project_id	UUID *(Optional)*	The UUID of the project in which the share was created. Useful with all_tenants parameter.
consistency_group_id	UUID *(Optional)*	(Since API v2.4) The UUID of the consistency group where the share was created.

This operation does not accept a request body.

10.4.2.2. Response

Example 10.5. List shares: JSON response

```
{
    "shares": [
        {
            "id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
            "links": [
                {
                    "href": "http://172.18.198.54:8786/v1/
16e1ab15c35a457e9c2b2aa189f544e1/shares/d94a8548-2079-4be0-b21c-0a887acd31ca",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/d94a8548-2079-4be0-b21c-0a887acd31ca",
                    "rel": "bookmark"
                }
            ],
            "name": "My_share"
        },
        {
            "id": "406ea93b-32e9-4907-a117-148b3945749f",
            "links": [
                {
```

```
                    "href": "http://172.18.198.54:8786/v1/
16e1ab15c35a457e9c2b2aa189f544e1/shares/406ea93b-32e9-4907-a117-148b3945749f",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/406ea93b-32e9-4907-a117-148b3945749f",
                    "rel": "bookmark"
                }
            ],
            "name": "Share1"
        }
    ]
}
```

10.4.3. List shares with details

Method	URI	Description
GET	`/v2/{tenant_id}/shares/detail{?all_tenants,name,status,share_server_id,metadata,extra_specs,share_type_id,limit,offset,sort_key,sort_dir,snapshot_id,host,share_network_id,project_id,consistency_group_id}`	Lists all shares, with details.

Normal response codes: 200

10.4.3.1. Request

This table shows the header parameters for the list shares with details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list shares with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This table shows the query parameters for the list shares with details request:

Name	Type	Description
`all_tenants`	Boolean *(Optional)*	(Admin only). Defines whether to list shares for all tenants. Set to `1` to list shares for all tenants. Set to `0` to list shares only for the current tenant.
`name`	String *(Optional)*	The share name.
`status`	String *(Optional)*	Filters by a share status. A valid value is `creating`, `error`, `available`, `deleting`, `error_deleting`, `manage_starting`, `manage_error`, `unmanage_starting`, `unmanage_error`, `unmanaged`, `extending`, `extending_error`, `shrinking`, `shrinking_error`, or `shrinking_possible_data_loss_error`.
`share_server_id`	UUID *(Optional)*	The UUID of the share server.
`metadata`	Dict *(Optional)*	One or more metadata key-value pairs, as a dictionary of strings.
`extra_specs`	String *(Optional)*	The extra specifications as a set of one or more key-value pairs. In each pair, the key is the name of the extra specification and the value is the share type that was used to create the share.
`share_type_id`	UUID *(Optional)*	(Since API v2.6) The UUID of the share type.
`limit`	Int *(Optional)*	The maximum number of shares to return.
`offset`	Int	The offset to define start point of share listing.

Name	Type	Description
	(Optional)	
`sort_key`	String *(Optional)*	The key to sort a list of shares. A valid value is `id`, `status`, `size`, `host`, `share_proto`, `export_location`, `availability_zone`, `user_id`, `project_id`, `created_at`, `updated_at`, `display_name`, `name`, `share_type_id`, `share_type`, `share_network_id`, `share_network`, `snapshot_id`, or `snapshot`.
`sort_dir`	String *(Optional)*	The direction to sort a list of shares. A valid value is `asc`, or `desc`.
`snapshot_id`	UUID *(Optional)*	The UUID of the snapshot that was used to create the share.
`host`	String *(Optional)*	The share host name.
`share_network_id`	UUID *(Optional)*	The UUID of the share network.
`project_id`	UUID *(Optional)*	The UUID of the project in which the share was created. Useful with `all_tenants` parameter.
`consistency_group_id`	UUID *(Optional)*	(Since API v2.4) The UUID of the consistency group where the share was created.

This operation does not accept a request body.

10.4.3.2. Response

Example 10.6. List shares with details: JSON response

```
{
    "shares": [
        {
            "links": [
                {
                    "href": "http://172.18.198.54:8786/v2/
16e1ab15c35a457e9c2b2aa189f544e1/shares/f45cc5b2-d1bb-4a3e-ba5b-5c4125613adc",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/f45cc5b2-d1bb-4a3e-ba5b-5c4125613adc",
                    "rel": "bookmark"
                }
            ],
            "availability_zone": "nova",
            "share_network_id": "f9b2e754-ac01-4466-86e1-5c569424754e",
            "export_locations": [],
            "share_server_id": "87d8943a-f5da-47a4-b2f2-ddfa6794aa82",
            "snapshot_id": null,
            "id": "f45cc5b2-d1bb-4a3e-ba5b-5c4125613adc",
            "size": 1,
            "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
            "share_type_name": "default",
            "export_location": null,
            "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
```

```
            "metadata": {},
            "status": "error",
            "access_rules_status": "active",
            "description": "There is a share description.",
            "host": "manila2@generic1#GENERIC1",
            "task_state": null,
            "is_public": true,
            "snapshot_support": true,
            "name": "my_share4",
            "has_replicas": false,
            "replication_type": null,
            "created_at": "2015-09-16T18:19:50.000000",
            "share_proto": "NFS",
            "volume_type": "default",
            "source_cgsnapshot_member_id": null
        },
        {
            "links": [
                {
                    "href": "http://172.18.198.54:8786/v2/
16e1ab15c35a457e9c2b2aa189f544e1/shares/c4a2ced4-2c9f-4ae1-adaa-6171833e64df",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/c4a2ced4-2c9f-4ae1-adaa-6171833e64df",
                    "rel": "bookmark"
                }
            ],
            "availability_zone": "nova",
            "share_network_id": "f9b2e754-ac01-4466-86e1-5c569424754e",
            "export_locations": [
                "10.254.0.5:/shares/share-50ad5e7b-f6f1-4b78-
a651-0812cef2bb67"
            ],
            "share_server_id": "87d8943a-f5da-47a4-b2f2-ddfa6794aa82",
            "snapshot_id": null,
            "id": "c4a2ced4-2c9f-4ae1-adaa-6171833e64df",
            "size": 1,
            "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
            "share_type_name": "default",
            "export_location": "10.254.0.5:/shares/share-50ad5e7b-f6f1-4b78-
a651-0812cef2bb67",
            "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "metadata": {},
            "status": "available",
            "access_rules_status": "active",
            "description": "Changed description.",
            "host": "manila2@generic1#GENERIC1",
            "task_state": null,
            "is_public": true,
            "snapshot_support": true,
            "name": "my_share4",
            "has_replicas": false,
            "replication_type": null,
            "created_at": "2015-09-16T17:26:28.000000",
            "share_proto": "NFS",
            "volume_type": "default",
            "source_cgsnapshot_member_id": null
```

```
        }
    ]
}
```

10.4.4. Manage share

Method	URI	Description
POST	`/v2/{tenant_id}/shares/manage`	Configures Shared File Systems to manage a share. This API is available for micro-versions later than or equal to 2.7

Normal response codes: 200

10.4.4.1. Request

This table shows the header parameters for the manage share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the manage share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.7. Manage share: JSON request

```
{
    "share": {
        "protocol": "nfs",
        "name": "share_texas1",
        "share_type": "d",
        "is_public": false,
        "driver_options": {
            "opt1": "opt1",
            "opt2": "opt2"
        },
        "export_path": "10.254.0.5:/shares/share-42033c24-0261-424f-
abda-4fef2f6dbfd5",
        "service_host": "manila2@unmanage1#UNMANAGE1",
        "description": "Lets manage share."
    }
}
```

10.4.4.2. Response

Example 10.8. Manage share: JSON response

```
{
    "share": {
        "links": [
            {
                "href": "http://172.18.198.54:8786/v2/
16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
                "rel": "self"
            },
            {
```

```
                "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
                "rel": "bookmark"
            }
        ],
        "availability_zone": null,
        "share_network_id": null,
        "export_locations": [],
        "share_server_id": null,
        "snapshot_id": null,
        "id": "00137b40-ca06-4ae8-83a3-2c5989eebcce",
        "size": null,
        "share_type": "14747856-08e5-494f-ab40-a64b9d20d8f7",
        "share_type_name": "d",
        "export_location": "10.254.0.5:/shares/share-42033c24-0261-424f-
abda-4fef2f6dbfd5",
        "consistency_group_id": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "metadata": {},
        "status": "manage_starting",
        "description": "Lets manage share.",
        "host": "manila2@unmanage1#UNMANAGE1",
        "access_rules_status": "active",
        "has_replicas": false,
        "replication_type": null,
        "is_public": false,
        "snapshot_support": true,
        "name": "share_texas1",
        "created_at": "2015-09-17T16:21:12.000000",
        "share_proto": "NFS",
        "volume_type": "d",
        "source_cgsnapshot_member_id": null
    }
}
```

10.4.5. Show share details

Method	URI	Description
GET	`/v2/{tenant_id}/shares/{share_id}`	Shows details for a share.

Normal response codes: 200

10.4.5.1. Request

This table shows the header parameters for the show share details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

This operation does not accept a request body.

10.4.5.2. Response

Example 10.9. Show share details: JSON response

```
{
    "share": {
        "links": [
            {
                "href": "http://172.18.198.54:8786/v2/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "nova",
        "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
        "export_locations": [],
        "share_server_id": "e268f4aa-d571-43dd-9ab3-f49ad06ffaef",
        "snapshot_id": null,
        "id": "011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "size": 1,
        "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
        "share_type_name": "default",
        "export_location": null,
        "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
```

```
        "metadata": {
            "project": "my_app",
            "aim": "doc"
        },
        "status": "available",
        "description": "My custom share London",
        "host": "manila2@generic1#GENERIC1",
        "access_rules_status": "active",
        "has_replicas": false,
        "replication_type": null,
        "task_state": null,
        "is_public": true,
        "snapshot_support": true,
        "name": "share_London",
        "created_at": "2015-09-18T10:25:24.000000",
        "share_proto": "NFS",
        "volume_type": "default",
        "source_cgsnapshot_member_id": null
    }
}
```

10.4.6. Update share

Method	URI	Description
PUT	`/v2/{tenant_id}/shares/{share_id}`	Updates a share.

You can update these attributes:

- `display_name`, which also changes the `name` of the share.
- `display_description`, which also changes the `description` of the share.
- `is_public`. Changes the level of visibility.

If you try to update other attributes, they retain their previous values.

Normal response codes: 200

10.4.6.1. Request

This table shows the header parameters for the update share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the update share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.10. Update share: JSON request

```
{
    "share": {
        "is_public": true,
        "display_description": "Changing the share description."
    }
}
```

10.4.6.2. Response

Example 10.11. Update share: JSON response

```
{
    "share": {
        "links": [
            {
                "href": "http://172.18.198.54:8786/v2/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
                "rel": "self"
            },
```

```
            {
                "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/011d21e2-fbc3-4e4a-9993-9ea223f73264",
                "rel": "bookmark"
            }
        ],
        "availability_zone": "nova",
        "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
        "export_locations": [],
        "share_server_id": "e268f4aa-d571-43dd-9ab3-f49ad06ffaef",
        "snapshot_id": null,
        "id": "011d21e2-fbc3-4e4a-9993-9ea223f73264",
        "size": 1,
        "share_type": "25747776-08e5-494f-ab40-a64b9d20d8f7",
        "share_type_name": "default",
        "export_location": null,
        "consistency_group_id": "9397c191-8427-4661-a2e8-b23820dc01d4",
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "metadata": {
            "project": "my_app",
            "aim": "doc"
        },
        "status": "error",
        "description": "Changing the share description.",
        "host": "manila2@generic1#GENERIC1",
        "task_state": null,
        "is_public": true,
        "snapshot_support": true,
        "name": "share_London",
        "created_at": "2015-09-18T10:25:24.000000",
        "share_proto": "NFS",
        "volume_type": "default",
        "source_cgsnapshot_member_id": null
    }
}
```

10.4.7. Delete share

Method	URI	Description
DELETE	`/v2/{tenant_id}/shares/{share_id}{?consistency_group_id}`	Deletes a share.

Normal response codes: 202

10.4.7.1. Request

This table shows the header parameters for the delete share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

This table shows the query parameters for the delete share request:

Name	Type	Description
`consistency_group_id`	UUID *(Optional)*	(Since API v2.4) The UUID of the consistency group where the share was created. You can omit this parameter if the share was created without a consistency group.

This operation does not accept a request body.

10.5. Share export locations (since API v2.9)

Set of APIs used for viewing export locations of shares.

By default, these APIs are admin-only. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
GET	`/v2/{tenant_id}/shares/{share_id}/export_locations`	Lists all export locations for a share.
GET	`/v2/{tenant_id}/shares/{share_id}/export_locations/{export_location_id}`	Show details of an export location belonging to a share.

10.5.1. List export locations

Method	URI	Description
GET	`/v2/{tenant_id}/shares/{share_id}/export_locations`	Lists all export locations for a share.

Normal response codes: 200

10.5.1.1. Request

This table shows the header parameters for the list export locations request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list export locations request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

This operation does not accept a request body.

10.5.1.2. Response

Example 10.12. List export locations: JSON response

```
{
    "export_locations": [
        {
            "path": "10.254.0.3:/shares/share-e1c2d35e-fe67-4028-
ad7a-45f668732b1d",
            "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
            "is_admin_only": false,
            "id": "b6bd76ce-12a2-42a9-a30a-8a43b503867d",
            "preferred": false
        },
        {
            "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-
ad7a-45f668732b1d",
            "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
            "is_admin_only": true,
            "id": "6921e862-88bc-49a5-a2df-efeed9acd583",
            "preferred": false
        }
    ]
}
```

10.5.2. Show single export location

Method	URI	Description
GET	`/v2/{tenant_id}/shares/{share_id}/export_locations/{export_location_id}`	Show details of an export location belonging to a share.

Normal response codes: 200

10.5.2.1. Request

This table shows the header parameters for the show single export location request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show single export location request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.
`{export_location_id}`	UUID	The UUID of the export location.

This operation does not accept a request body.

10.5.2.2. Response

Example 10.13. Show single export location: JSON response

```
{
    "export_location": {
        "created_at": "2016-03-24T14:20:47.000000",
        "updated_at": "2016-03-24T14:20:47.000000",
        "preferred": false,
        "is_admin_only": true,
        "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
        "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
        "id": "6921e862-88bc-49a5-a2df-efeed9acd583"
    }
}
```

10.6. Share metadata

Shows, sets, updates, and unsets share metadata.

Method	URI	Description
GET	`/v2/{tenant_id}/shares/{share_id}/metadata`	Shows the metadata for a share.
PUT	`/v2/{tenant_id}/shares/{share_id}/metadata`	Updates the metadata for a share.

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/metadata`	Sets the metadata on a share.
DELETE	`/v2/{tenant_id}/shares/{share_id}/metadata/{key}`	Unsets the metadata on a share.

10.6.1. Show share metadata

Method	URI	Description
GET	`/v2/{tenant_id}/shares/{share_id}/metadata`	Shows the metadata for a share.

Normal response codes: 200

10.6.1.1. Request

This table shows the header parameters for the show share metadata request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

This operation does not accept a request body.

10.6.1.2. Response

Example 10.14. Show share metadata: JSON response

```
{
    "metadata": {
        "project": "my_app",
        "aim": "doc"
    }
}
```

10.6.2. Update share metadata

Method	URI	Description
PUT	`/v2/{tenant_id}/shares/{share_id}/metadata`	Updates the metadata for a share.

Normal response codes: 200

10.6.2.1. Request

This table shows the header parameters for the update share metadata request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the update share metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.15. Update share metadata: JSON request

```
{
    "metadata": {
        "aim": "changed_doc",
        "project": "my_app",
        "new_metadata_key": "new_information"
    }
}
```

10.6.2.2. Response

Example 10.16. Update share metadata: JSON response

```
{
    "metadata": {
        "aim": "changed_doc",
        "project": "my_app",
        "new_metadata_key": "new_information"
    }
}
```

10.6.3. Set share metadata

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/metadata`	Sets the metadata on a share.

Normal response codes: 200

10.6.3.1. Request

This table shows the header parameters for the set share metadata request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the set share metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.17. Set share metadata: JSON request

```
{
    "metadata": {
        "key1": "value1"
    }
}
```

10.6.3.2. Response

Example 10.18. Set share metadata: JSON response

```
{
    "metadata": {
        "aim": "changed_doc",
        "project": "my_app",
        "key1": "value1",
        "new_metadata_key": "new_information",
        "key": "value"
    }
}
```

10.6.4. Unset share metadata

Method	URI	Description
DELETE	`/v2/{tenant_id}/shares/{share_id}/metadata/{key}`	Unsets the metadata on a share.

To unset a metadata key value, specify only the key name in the URI.

Normal response codes: 200

10.6.4.1. Request

This table shows the header parameters for the unset share metadata request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the unset share metadata request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.
`{metadata_key}`	String	The metadata key.

This operation does not accept a request body.

10.7. Share actions

Grants or revokes share access, lists the permissions for a share, and explicitly updates the state of a share.

To grant or revoke share access, specify one of these supported share access levels:

- `rw`. Read and write (RW) access.
- `ro`. Read-only (RO) access.

You must also specify one of these supported authentication methods:

- `ip`. Authenticates an instance through its IP address. A valid format is `XX.XX.XX.XX` or `XX.XX.XX.XX/XX`. For example `0.0.0.0/0`.
- `cert`. Authenticates an instance through a TLS certificate. Specify the TLS identity as the IDENTKEY. A valid value is any string up to 64 characters long in the common name (CN) of the certificate. The meaning of a string depends on its interpretation.
- `user`. Authenticates by a user or group name. A valid value is an alphanumeric string that can contain some special characters and is from 4 to 32 characters long.

To verify that the access rules (ACL) were configured correctly for a share, you list permissions for a share.

As administrator, you can reset the state of a share and force-delete a share in any state. Use the `policy.json` file to grant permissions for this action to other roles.

You can set the state of a share to one of these supported states:

- `available`
- `error`
- `creating`
- `deleting`
- `error_deleting`

Caution

If micro-version 1.0-2.6 is used then all share actions, defined below, should include prefix `os-` in top element of request JSON's body.

For example: {"access_list": null} is valid for v2.7+. And {"os-access_list": null} is valid for v1.0-2.6

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Grants access to a share.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Revokes access from a share.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Lists access rules for a share.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Administrator only. Explicitly updates the state of a share.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Administrator only. Force-deletes a share in any state.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Increases the size of a share.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Shrinks the size of a share.
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Configures Shared File Systems to stop managing a share.

10.7.1. Grant access

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Grants access to a share.

Normal response codes: 200

10.7.1.1. Request

This table shows the header parameters for the grant access request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the grant access request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.19. Grant access: JSON request

```
{
    "allow_access": {
        "access_level": "rw",
        "access_type": "ip",
        "access_to": "0.0.0.0/0"
    }
}
```

10.7.1.2. Response

Example 10.20. Grant access: JSON response

```
{
    "access": {
        "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
        "created_at": "2015-09-07T09:14:48.000000",
        "updated_at": null,
        "access_type": "ip",
        "access_to": "0.0.0.0/0",
        "access_level": "rw",
        "id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452"
    }
}
```

10.7.2. Revoke access

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Revokes access from a share.

Normal response codes: 202

10.7.2.1. Request

This table shows the header parameters for the revoke access request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the revoke access request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.21. Revoke access: JSON request

```
{
    "deny_access": {
        "access_id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452"
    }
}
```

10.7.3. List access rules

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Lists access rules for a share.

Normal response codes: 200

10.7.3.1. Request

This table shows the header parameters for the list access rules request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list access rules request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.22. List access rules: JSON request

```
{
    "access_list": null
}
```

10.7.3.2. Response

Example 10.23. List access rules: JSON response

```
{
    "access_list": [
        {
            "access_level": "rw",
            "state": "error",
            "id": "507bf114-36f2-4f56-8cf4-857985ca87c1",
            "access_type": "cert",
            "access_to": "example.com"
        },
        {
            "access_level": "rw",
            "state": "active",
            "id": "a25b2df3-90bd-4add-afa6-5f0dbbd50452",
            "access_type": "ip",
            "access_to": "0.0.0.0/0"
        }
    ]
}
```

10.7.4. Reset share state

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Administrator only. Explicitly updates the state of a share.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

10.7.4.1. Request

This table shows the header parameters for the reset share state request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the reset share state request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.24. Reset share state: JSON request

```
{
    "reset_status": {
        "status": "error"
    }
}
```

10.7.5. Force-delete share

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Administrator only. Force-deletes a share in any state.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

10.7.5.1. Request

This table shows the header parameters for the force-delete share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the force-delete share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.25. Force-delete share: JSON request

```
{
    "force_delete": null
}
```

10.7.6. Extend share

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Increases the size of a share.

Normal response codes: 202

10.7.6.1. Request

This table shows the header parameters for the extend share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the extend share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.26. Extend share: JSON request

```
{
    "extend": {
        "new_size": 2
    }
}
```

10.7.7. Shrink share

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Shrinks the size of a share.

Normal response codes: 202

10.7.7.1. Request

This table shows the header parameters for the shrink share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the shrink share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.27. Shrink share: JSON request

```
{
    "shrink": {
        "new_size": 1
    }
}
```

10.7.8. Unmanage share

Method	URI	Description
POST	`/v2/{tenant_id}/shares/{share_id}/action`	Configures Shared File Systems to stop managing a share.

Normal response codes: 202

10.7.8.1. Request

This table shows the header parameters for the unmanage share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the unmanage share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

Example 10.28. Unmanage share: JSON request

```
{
    "unmanage": null
}
```

10.8. Share snapshots

Use the shared file service to make snapshots of shares. A share snapshot is a point-in-time, read-only copy of the data that is contained in a share. You can create, manage, update, and delete share snapshots. After you create or manage a share snapshot, you can create a share from it.

You can update a share snapshot to rename it, change its description, or update its state to one of these supported states:

- `available`
- `error`
- `creating`
- `deleting`
- `error_deleting`
- `manage_starting`
- `manage_error`

- `unmanage_starting`
- `unmanage_error`

As administrator, you can also reset the state of a snapshot and force-delete a share snapshot in any state. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots`	Creates a snapshot from a share.
GET	`/v2/{tenant_id}/snapshots`	Lists all share snapshots.
POST	`/v2/{tenant_id}/snapshots/manage`	(Since API v2.12) Configures Shared File Systems to manage a share snapshot.
GET	`/v2/{tenant_id}/snapshots/detail`	Lists all share snapshots with details.
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a share snapshot.
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Updates a share snapshot.
DELETE	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Deletes a share snapshot.
POST	`/v2/{tenant_id}/snapshots/{snapshot_id}/action`	(Since API v2.12) Configures Shared File Systems to stop managing a share snapshot.
POST	`/v2/{tenant_id}/snapshots/{snapshot_id}/action`	Administrator only. Explicitly updates the state of a share snapshot.
POST	`/v2/{tenant_id}/snapshots/{snapshot_id}/action`	Administrator only. Force-deletes a share snapshot in any state.

10.8.1. Create share snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots`	Creates a snapshot from a share.

Normal response codes: 200

10.8.1.1. Request

This table shows the header parameters for the create share snapshot request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the create share snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.29. Create share snapshot: JSON request

```
{
    "snapshot": {
        "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
        "force": "True",
        "name": "snapshot_share1",
        "description": "Here is a snapshot of share Share1"
    }
}
```

10.8.1.2. Response

Example 10.30. Create share snapshot: JSON response

```
{
    "snapshot": {
        "status": "creating",
        "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
        "name": "snapshot_share1",
        "links": [
            {
                "href": "http://172.18.198.54:8786/
v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                "rel": "bookmark"
            }
```

```
        ],
        "created_at": "2015-09-07T11:50:39.756808",
        "description": "Here is a snapshot of share Share1",
        "share_proto": "NFS",
        "share_size": 1,
        "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "size": 1
    }
}
```

10.8.2. List share snapshots

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots`	Lists all share snapshots.

Normal response codes: 200

10.8.2.1. Request

This table shows the header parameters for the list share snapshots request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share snapshots request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.8.2.2. Response

Example 10.31. List share snapshots: JSON response

```
{
    "snapshots": [
        {
            "id": "086a1aa6-c425-4ecd-9612-391a3b1b9375",
            "links": [
                {
                    "href": "http://172.18.198.54:8786/
v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-
c425-4ecd-9612-391a3b1b9375",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-
c425-4ecd-9612-391a3b1b9375",
                    "rel": "bookmark"
                }
            ],
            "name": "snapshot_My_share"
        },
        {
            "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
            "links": [
                {
                    "href": "http://172.18.198.
54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                    "rel": "self"
```

```
                },
                {
                    "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                    "rel": "bookmark"
                }
            ],
            "name": "snapshot_share1"
        }
    ]
}
```

10.8.3. Manage share snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots/manage`	(Since API v2.12) Configures Shared File Systems to manage a share snapshot.

Normal response codes: 200

10.8.3.1. Request

This table shows the header parameters for the manage share snapshot request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the manage share snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.32. Manage share snapshot: JSON request

```
{
    "snapshot": {
        "share_id": "dd6c5d35-9db1-4662-a7ae-8b52f880aeba",
        "provider_location": "4045fee5-4e0e-408e-97f3-15e25239dbc9",
        "name": "managed_snapshot",
        "description": "description_of_managed_snapshot",
        "driver_options": {
            "opt1": "opt1",
            "opt2": "opt2"
        }
    }
}
```

10.8.3.2. Response

Example 10.33. Manage share snapshot: JSON response

```
{
    "snapshot": {
        "id": "22de7000-3a32-4fe1-bd0c-38d03f93dec3",
        "share_id": "dd6c5d35-9db1-4662-a7ae-8b52f880aeba",
        "share_size": 1,
        "created_at": "2016-04-01T15:16:17.000000",
        "status": "manage_starting",
        "name": "managed_snapshot",
        "description": "description_of_managed_snapshot",
        "size": 1,
        "share_proto": "NFS",
        "links": [
            {
```

```
                "href": "http://127.0.0.1:8786/v2/
907004508ef4447397ce6741a8f037c1/snapshots/22de7000-3a32-4fe1-
bd0c-38d03f93dec3",
                "rel": "self"
            },
            {
                "href": "http://127.0.0.1:8786/
907004508ef4447397ce6741a8f037c1/snapshots/22de7000-3a32-4fe1-
bd0c-38d03f93dec3",
                "rel": "bookmark"
            }
        ],
        "provider_location": "4045fee5-4e0e-408e-97f3-15e25239dbc9"
    }
}
```

10.8.4. List share snapshots with details

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/detail`	Lists all share snapshots with details.

Normal response codes: 200

10.8.4.1. Request

This table shows the header parameters for the list share snapshots with details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share snapshots with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.8.4.2. Response

Example 10.34. List share snapshots with details: JSON response

```
{
    "snapshots": [
        {
            "status": "creating",
            "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
            "name": "snapshot_My_share",
            "links": [
                {
                    "href": "http://172.18.198.54:8786/
v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-
c425-4ecd-9612-391a3b1b9375",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/snapshots/086a1aa6-
c425-4ecd-9612-391a3b1b9375",
                    "rel": "bookmark"
                }
            ],
            "created_at": "2015-09-07T11:55:09.000000",
            "description": "Here is a snapshot of share My_share",
            "share_proto": "NFS",
            "share_size": 1,
            "id": "086a1aa6-c425-4ecd-9612-391a3b1b9375",
            "size": 1
        },
        {
```

```
            "status": "available",
            "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
            "name": "snapshot_share1",
            "links": [
                {
                    "href": "http://172.18.198.
54:8786/v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                    "rel": "self"
                },
                {
                    "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                    "rel": "bookmark"
                }
            ],
            "created_at": "2015-09-07T11:50:39.000000",
            "description": "Here is a snapshot of share Share1",
            "share_proto": "NFS",
            "share_size": 1,
            "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
            "size": 1
        }
    ]
}
```

10.8.5. Show share snapshot details

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Shows details for a share snapshot.

Normal response codes: 200

10.8.5.1. Request

This table shows the header parameters for the show share snapshot details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share snapshot details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

10.8.5.2. Response

Example 10.35. Show share snapshot details: JSON response

```
{
    "snapshot": {
        "status": "available",
        "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
        "name": "snapshot_share1",
        "links": [
            {
                "href": "http://172.18.198.54:8786/
v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                "rel": "bookmark"
            }
        ],
        "created_at": "2015-09-07T11:50:39.000000",
        "description": "Here is a snapshot of share Share1",
        "share_proto": "NFS",
        "share_size": 1,
        "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "size": 1
```

```
    }
}
```

10.8.6. Update share snapshot

Method	URI	Description
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Updates a share snapshot.

You can update these attributes:

- `display_name`, which also changes the `name` of the share snapshot.
- `display_description`, which also changes the `description` of the share snapshot.
- `is_public`. Changes the level of visibility.

If you try to update other attributes, they retain their previous values.

Normal response codes: 200

10.8.6.1. Request

This table shows the header parameters for the update share snapshot request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the update share snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 10.36. Update share snapshot: JSON request

```
{
    "snapshot": {
        "display_name": "snapshot_Share1",
        "display_description": "I am changing a description also. Here is a
 snapshot of share Share1"
    }
}
```

10.8.6.2. Response

Example 10.37. Update share snapshot: JSON response

```
{
    "snapshot": {
        "status": "available",
        "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
        "name": "snapshot_Share1",
        "links": [
```

```
            {
                "href": "http://172.18.198.54:8786/
v1/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.
54:8786/16e1ab15c35a457e9c2b2aa189f544e1/snapshots/
6d221c1d-0200-461e-8d20-24b4776b9ddb",
                "rel": "bookmark"
            }
        ],
        "created_at": "2015-09-07T11:50:39.000000",
        "description": "I am changing a description also. Here is a snapshot
 of share Share1",
        "share_proto": "NFS",
        "share_size": 1,
        "id": "6d221c1d-0200-461e-8d20-24b4776b9ddb",
        "size": 1
    }
}
```

10.8.7. Delete share snapshot

Method	URI	Description
DELETE	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Deletes a share snapshot.

Normal response codes: 202

10.8.7.1. Request

This table shows the header parameters for the delete share snapshot request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete share snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

This operation does not accept a request body.

10.8.8. Unmanage share snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots/{snapshot_id}/action`	(Since API v2.12) Configures Shared File Systems to stop managing a share snapshot.

Normal response codes: 202

10.8.8.1. Request

This table shows the header parameters for the unmanage share snapshot request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the unmanage share snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 10.38. Unmanage share snapshot: JSON request

```
{
    "unmanage": null
}
```

10.8.9. Reset share snapshot state

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots/{snapshot_id}/action`	Administrator only. Explicitly updates the state of a share snapshot.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

10.8.9.1. Request

This table shows the header parameters for the reset share snapshot state request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the reset share snapshot state request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 10.39. Reset share snapshot state: JSON request

```
{
    "reset_status": {
        "status": "error"
    }
}
```

10.8.10. Force-delete share snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots/{snapshot_id}/action`	Administrator only. Force-deletes a share snapshot in any state.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

10.8.10.1. Request

This table shows the header parameters for the force-delete share snapshot request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the force-delete share snapshot request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{snapshot_id}`	UUID	The UUID of the snapshot.

Example 10.40. Force-delete share snapshot: JSON request

```
{
    "force_delete": null
}
```

10.9. Share networks

A share network stores network information that share servers can use where shares are hosted. You can associate a share with a single share network. When you create a share, you can optionally specify the ID of a share network through which instances can access the share.

You can create, update, view, and delete a share network.

When you create a share network, you can specify only one type of network:

- Neutron network. Specify a network ID and subnet ID.
- Nova network. Specify a network ID.

For more information about supported plug-ins for share networks, see Manila Network Plugins.

A share network has these attributes:

- The IP block in Classless Inter-Domain Routing (CIDR) notation from which to allocate the network.

- The IP version of the network.
- The network type, which is `vlan`, `vxlan`, `gre`, or `flat`.
- If the network uses segmentation, a segmentation identifier. For example, VLAN, VXLAN, and GRE networks use segmentation.

Method	URI	Description
POST	`/v2/{tenant_id}/share-networks`	Creates a share network.
GET	`/v2/{tenant_id}/share-networks`	Lists all share networks.
GET	`/v2/{tenant_id}/share-networks/detail`	Lists all share networks with details.
GET	`/v2/{tenant_id}/share-networks/{share_network_id}`	Shows details for a share network.
PUT	`/v2/{tenant_id}/share-networks/{share_network_id}`	Updates a share network.
DELETE	`/v2/{tenant_id}/share-networks/{share_network_id}`	Deletes a share network.
POST	`/v2/{tenant_id}/share-networks/{share_network_id}/action`	Adds a security service to a share network.
POST	`/v2/{tenant_id}/share-networks/{share_network_id}/action`	Removes a security service from a share network.

10.9.1. Create share network

Method	URI	Description
POST	`/v2/{tenant_id}/share-networks`	Creates a share network.

Normal response codes: 200

10.9.1.1. Request

This table shows the header parameters for the create share network request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the create share network request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.41. Create share network: JSON request

```
{
    "share_network": {
        "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
        "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
        "name": "my_network",
        "description": "This is my share network"
    }
}
```

10.9.1.2. Response

Example 10.42. Create share network: JSON response

```
{
    "share_network": {
        "name": "my_network",
        "segmentation_id": null,
        "created_at": "2015-09-07T14:37:00.583656",
        "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
        "updated_at": null,
        "id": "77eb3421-4549-4789-ac39-0d5185d68c29",
        "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
        "ip_version": null,
        "nova_net_id": null,
        "cidr": null,
        "project_id": "e10a683c20da41248cfd5e1ab3d88c62",
        "network_type": null,
        "description": "This is my share network"
    }
}
```

10.9.2. List share networks

Method	URI	Description
GET	`/v2/{tenant_id}/share-networks`	Lists all share networks.

Normal response codes: 200

10.9.2.1. Request

This table shows the header parameters for the list share networks request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share networks request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.9.2.2. Response

Example 10.43. List share networks: JSON response

```
{
    "share_networks": [
        {
            "id": "32763294-e3d4-456a-998d-60047677c2fb",
            "name": "net_my1"
        },
        {
            "id": "713df749-aac0-4a54-af52-10f6c991e80c",
            "name": "net_my"
        },
        {
            "id": "fa158a3d-6d9f-4187-9ca5-abbb82646eb2",
            "name": null
        }
    ]
}
```

10.9.3. List share networks with details

Method	URI	Description
GET	`/v2/{tenant_id}/share-networks/detail`	Lists all share networks with details.

Normal response codes: 200

10.9.3.1. Request

This table shows the header parameters for the list share networks with details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share networks with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.9.3.2. Response

Example 10.44. List share networks with details: JSON response

```
{
    "share_networks": [
        {
            "name": "net_my1",
            "segmentation_id": null,
            "created_at": "2015-09-04T14:57:13.000000",
            "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
            "updated_at": null,
            "id": "32763294-e3d4-456a-998d-60047677c2fb",
            "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
            "ip_version": null,
            "nova_net_id": null,
            "cidr": null,
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "network_type": null,
            "description": "descr"
        },
        {
            "name": "net_my",
            "segmentation_id": null,
            "created_at": "2015-09-04T14:54:25.000000",
            "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
            "updated_at": null,
            "id": "713df749-aac0-4a54-af52-10f6c991e80c",
            "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
            "ip_version": null,
            "nova_net_id": null,
```

```
            "cidr": null,
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "network_type": null,
            "description": "desecr"
        },
        {
            "name": null,
            "segmentation_id": null,
            "created_at": "2015-09-04T14:51:41.000000",
            "neutron_subnet_id": null,
            "updated_at": null,
            "id": "fa158a3d-6d9f-4187-9ca5-abbb82646eb2",
            "neutron_net_id": null,
            "ip_version": null,
            "nova_net_id": null,
            "cidr": null,
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "network_type": null,
            "description": null
        }
    ]
}
```

10.9.4. Show share network details

Method	URI	Description
GET	`/v2/{tenant_id}/share-networks/{share_network_id}`	Shows details for a share network.

Normal response codes: 200

10.9.4.1. Request

This table shows the header parameters for the show share network details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share network details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_network_id}`	UUID	The UUID of the share network.

This operation does not accept a request body.

10.9.4.2. Response

Example 10.45. Show share network details: JSON response

```
{
    "share_network": {
        "name": "net_my1",
        "segmentation_id": null,
        "created_at": "2015-09-04T14:56:45.000000",
        "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
        "updated_at": null,
        "id": "7f950b52-6141-4a08-bbb5-bb7ffa3ea5fd",
        "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
        "ip_version": null,
        "nova_net_id": null,
        "cidr": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "network_type": null,
        "description": "descr"
    }
}
```

10.9.5. Update share network

Method	URI	Description
PUT	`/v2/{tenant_id}/share-networks/{share_network_id}`	Updates a share network.

Note that if the share network is used by any share server, you can update only the `name` and `description` attributes.

Normal response codes: 200

10.9.5.1. Request

This table shows the header parameters for the update share network request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the update share network request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_network_id}`	UUID	The UUID of the share network.

Example 10.46. Update share network: JSON request

```
{
    "share_network": {
        "description": "i'm adding a description"
    }
}
```

10.9.5.2. Response

Example 10.47. Update share network: JSON response

```
{
    "share_network": {
        "name": "net_my",
        "segmentation_id": null,
        "created_at": "2015-09-04T14:54:25.000000",
        "neutron_subnet_id": "53482b62-2c84-4a53-b6ab-30d9d9800d06",
        "updated_at": "2015-09-07T08:02:53.512184",
        "id": "713df749-aac0-4a54-af52-10f6c991e80c",
        "neutron_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
        "ip_version": "4",
        "nova_net_id": null,
        "cidr": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "network_type": null,
        "description": "i'm adding a description"
    }
```

```
}
```

10.9.6. Delete share network

Method	URI	Description
DELETE	`/v2/{tenant_id}/share-networks/{share_network_id}`	Deletes a share network.

Normal response codes: 202

10.9.6.1. Request

This table shows the header parameters for the delete share network request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete share network request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_network_id}`	UUID	The UUID of the share network.

This operation does not accept a request body.

10.9.7. Add security service to share network

Method	URI	Description
POST	`/v2/{tenant_id}/share-networks/{share_network_id}/action`	Adds a security service to a share network.

Normal response codes: 200

10.9.7.1. Request

This table shows the header parameters for the add security service to share network request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the add security service to share network request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_network_id}`	UUID	The UUID of the share network.

Example 10.48. Add security service to share network: JSON request

```
{
    "add_security_service": {
        "security_service_id": "3c829734-0679-4c17-9637-801da48c0d5f"
    }
}
```

10.9.7.2. Response

Example 10.49. Add security service to share network: JSON response

```
{
    "share_network": {
        "name": "net2",
        "segmentation_id": null,
        "created_at": "2015-09-07T12:31:12.000000",
        "neutron_subnet_id": null,
        "updated_at": null,
        "id": "d8ae6799-2567-4a89-aafb-fa4424350d2b",
        "neutron_net_id": null,
        "ip_version": null,
        "nova_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
        "cidr": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "network_type": null,
        "description": null
    }
}
```

10.9.8. Remove security service from share network

Method	URI	Description
POST	`/v2/{tenant_id}/share-networks/{share_network_id}/action`	Removes a security service from a share network.

Normal response codes: 200

10.9.8.1. Request

This table shows the header parameters for the remove security service from share network request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the remove security service from share network request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_network_id}`	UUID	The UUID of the share network.

Example 10.50. Remove security service from share network: JSON request

```
{
    "remove_security_service": {
        "security_service_id": "3c829734-0679-4c17-9637-801da48c0d5f"
    }
}
```

10.9.8.2. Response

Example 10.51. Remove security service from share network: JSON response

```
{
    "share_network": {
        "name": "net2",
        "segmentation_id": null,
        "created_at": "2015-09-07T12:31:12.000000",
        "neutron_subnet_id": null,
        "updated_at": null,
        "id": "d8ae6799-2567-4a89-aafb-fa4424350d2b",
        "neutron_net_id": null,
        "ip_version": null,
        "nova_net_id": "998b42ee-2cee-4d36-8b95-67b5ca1f2109",
        "cidr": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "network_type": null,
        "description": null
    }
}
```

10.10. Security services

You can create, update, view, and delete a security service. A security service stores configuration information for clients for authentication and authorization (AuthN/AuthZ). For example, a share server will be the client for an existing service such as LDAP, Kerberos, or Microsoft Active Directory.

You can associate a share with from one to three security service types:

- `ldap`. LDAP.
- `kerberos`. Kerberos.
- `active_directory`. Microsoft Active Directory.

You can configure a security service with these options:

- A DNS IP address.
- An IP address or host name.
- A domain.
- A user or group name.
- The password for the user, if you specify a user name.

Method	URI	Description
POST	`/v2/{tenant_id}/security-services`	Creates a security service.
GET	`/v2/{tenant_id}/security-services`	Lists all security services.
GET	`/v2/{tenant_id}/security-services/detail`	Lists all security services with details.
GET	`/v2/{tenant_id}/security-services/{security_service_id}`	Shows details for a security service.
PUT	`/v2/{tenant_id}/security-services/{security_service_id}`	Updates a security service.
DELETE	`/v2/{tenant_id}/security-services/{security_service_id}`	Deletes a security service.

10.10.1. Create security service

Method	URI	Description
POST	`/v2/{tenant_id}/security-services`	Creates a security service.

Normal response codes: 200

10.10.1.1. Request

This table shows the header parameters for the create security service request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the create security service request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.52. Create security service: JSON request

```
{
    "security_service": {
        "description": "Creating my first Security Service",
        "dns_ip": "10.0.0.0/24",
        "user": "demo",
        "password": "***",
        "type": "kerberos",
        "name": "SecServ1"
    }
}
```

10.10.1.2. Response

Example 10.53. Create security service: JSON response

```
{
    "security_service": {
        "status": "new",
        "domain": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "name": "SecServ1",
        "created_at": "2015-09-07T12:19:10.695211",
        "updated_at": null,
        "server": null,
        "dns_ip": "10.0.0.0/24",
        "user": "demo",
        "password": "supersecret",
        "type": "kerberos",
        "id": "3c829734-0679-4c17-9637-801da48c0d5f",
        "description": "Creating my first Security Service"
    }
```

```
}
```

10.10.2. List security services

Method	URI	Description
GET	`/v2/{tenant_id}/security-services`	Lists all security services.

Normal response codes: 200

10.10.2.1. Request

This table shows the header parameters for the list security services request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list security services request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.10.2.2. Response

Example 10.54. List security services: JSON response

```
{
    "security_services": [
        {
            "status": "new",
            "type": "kerberos",
            "id": "3c829734-0679-4c17-9637-801da48c0d5f",
            "name": "SecServ1"
        },
        {
            "status": "new",
            "type": "ldap",
            "id": "5a1d3a12-34a7-4087-8983-50e9ed03509a",
            "name": "SecServ2"
        }
    ]
}
```

10.10.3. List security services with details

Method	URI	Description
GET	`/v2/{tenant_id}/security-services/detail`	Lists all security services with details.

Normal response codes: 200

10.10.3.1. Request

This table shows the header parameters for the list security services with details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list security services with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.10.3.2. Response

Example 10.55. List security services with details: JSON response

```
{
    "security_services": [
        {
            "status": "new",
            "domain": null,
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "name": "SecServ1",
            "created_at": "2015-09-07T12:19:10.000000",
            "description": "Creating my first Security Service",
            "updated_at": null,
            "server": null,
            "dns_ip": "10.0.0.0/24",
            "user": "demo",
            "password": "supersecret",
            "type": "kerberos",
            "id": "3c829734-0679-4c17-9637-801da48c0d5f",
            "share_networks": []
        },
        {
            "status": "new",
            "domain": null,
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "name": "SecServ2",
            "created_at": "2015-09-07T12:25:03.000000",
            "description": "Creating my second Security Service",
            "updated_at": null,
            "server": null,
```

```
            "dns_ip": "10.0.0.0/24",
            "user": null,
            "password": null,
            "type": "ldap",
            "id": "5a1d3a12-34a7-4087-8983-50e9ed03509a",
            "share_networks": []
        }
    ]
}
```

10.10.4. Show security service details

Method	URI	Description
GET	`/v2/{tenant_id}/security-services/{security_service_id}`	Shows details for a security service.

Normal response codes: 200

10.10.4.1. Request

This table shows the header parameters for the show security service details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show security service details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_service_id}`	UUID	The UUID of the security service.

This operation does not accept a request body.

10.10.4.2. Response

Example 10.56. Show security service details: JSON response

```
{
    "security_service": {
        "status": "new",
        "domain": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "name": "SecServ1",
        "created_at": "2015-09-07T12:19:10.000000",
        "updated_at": null,
        "server": null,
        "dns_ip": "10.0.0.0/24",
        "user": "demo",
        "password": "supersecret",
        "type": "kerberos",
        "id": "3c829734-0679-4c17-9637-801da48c0d5f",
        "description": "Creating my first Security Service"
    }
}
```

10.10.5. Update security service

Method	URI	Description
PUT	`/v2/{tenant_id}/security-services/{security_service_id}`	Updates a security service.

If the security service is in `active` state, you can update only the `name` and `description` attributes. A security service in `active` state is attached to a share network with an associated share server.

Normal response codes: 200

10.10.5.1. Request

This table shows the header parameters for the update security service request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the update security service request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_service_id}`	UUID	The UUID of the security service.

Example 10.57. Update security service: JSON request

```
{
    "security_service": {
        "domain": "my_domain",
        "password": "***",
        "user": "new_user",
        "description": "Adding a description"
    }
}
```

10.10.5.2. Response

Example 10.58. Update security service: JSON response

```
{
    "security_service": {
        "status": "new",
        "domain": "my_domain",
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "name": "SecServ1",
        "created_at": "2015-09-07T12:19:10.000000",
        "updated_at": "2015-09-07T12:47:21.858737",
        "server": null,
        "dns_ip": "10.0.0.0/24",
        "user": "new_user",
        "password": "pass",
```

```
        "type": "kerberos",
        "id": "3c829734-0679-4c17-9637-801da48c0d5f",
        "description": "Adding a description"
    }
}
```

10.10.6. Delete security service

Method	URI	Description
DELETE	`/v2/{tenant_id}/security-services/{security_service_id}`	Deletes a security service.

Normal response codes: 202

10.10.6.1. Request

This table shows the header parameters for the delete security service request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete security service request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{security_service_id}`	UUID	The UUID of the security service.

This operation does not accept a request body.

10.11. Share servers

A share server is created by multi-tenant back-end drivers where shares are hosted. For example, with the `generic` driver, shares are hosted on Compute VMs. With the `cluster_mode` driver from NetApp, shares are hosted on virtual storage servers, also known as Vservers or SVMs.

Administrators can perform read and delete actions for share servers. An administrator can delete an active share server only if it contains no dependent shares. If an administrator deletes the share server, the Shared File Systems service creates a share server in response to a subsequent create share request.

An administrator can use the `policy.json` file to grant permissions for share server actions to other roles.

The status of a share server indicates its current state. After you successfully set up a share server, its status is `active`. If errors occur during set up such as when server data is not valid, its status is `error`.

The possible share servers statuses are:

Table 10.2. Share server statuses

Status	Description
`active`	Share server was successfully set up.
`error`	The set up or deletion of the share server failed.

Status	Description
`deleting`	The share server has no dependent shares and is being deleted.
`creating`	The share server is being created on the back end with data from the database.

Method	URI	Description
GET	`/v2/{tenant_id}/share-servers`	Lists all share servers.
GET	`/v2/{tenant_id}/share-servers/{share_server_id}`	Shows details for a share server.
DELETE	`/v2/{tenant_id}/share-servers/{share_server_id}`	Deletes a share server.

10.11.1. List share servers

Method	URI	Description
GET	`/v2/{tenant_id}/share-servers`	Lists all share servers.

Normal response codes: 200

10.11.1.1. Request

This table shows the header parameters for the list share servers request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share servers request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.11.1.2. Response

Example 10.59. List share servers: JSON response

```
{
    "share_servers": [
        {
            "status": "active",
            "updated_at": "2015-09-07T08:52:15.000000",
            "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
            "host": "manila2@generic1",
            "share_network_name": "net_my",
            "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
            "id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73"
        }
    ]
}
```

10.11.2. Show share server details

Method	URI	Description
GET	`/v2/{tenant_id}/share-servers/{share_server_id}`	Shows details for a share server.

Normal response codes: 200

10.11.2.1. Request

This table shows the header parameters for the show share server details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share server details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_server_id}`	UUID	The UUID of the share server.

This operation does not accept a request body.

10.11.2.2. Response

Example 10.60. Show share server details: JSON response

```
{
    "share_server": {
        "status": "active",
        "backend_details": {
            "username": "manila",
            "router_id": "4b62ce91-56c5-45c1-b0ef-8cbbe5dd34f4",
            "pk_path": "/opt/stack/.ssh/id_rsa",
            "subnet_id": "16e99ad6-5191-461c-9f34-ac84a39c3adb",
            "ip": "10.254.0.3",
            "instance_id": "75f2f282-af65-49ba-a7b1-525705b1bf1a",
            "public_address": "10.254.0.3",
            "service_port_id": "8ff21760-961e-4b83-a032-03fd559bb1d3"
        },
        "created_at": "2015-09-07T08:37:19.000000",
        "updated_at": "2015-09-07T08:52:15.000000",
        "share_network_name": "net_my",
        "host": "manila2@generic1",
        "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73"
    }
}
```

10.11.3. Delete share server

Method	URI	Description
DELETE	`/v2/{tenant_id}/share-servers/{share_server_id}`	Deletes a share server.

An administrator can delete an active share server only if it contains no dependent shares.

Normal response codes: 202

10.11.3.1. Request

This table shows the header parameters for the delete share server request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete share server request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_server_id}`	UUID	The UUID of the share server.

This operation does not accept a request body.

10.12. Share instances (since API v2.3)

Administrators can list, show information for, explicitly set the state of, and force-delete share instances. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
GET	`/v2/{tenant_id}/share_instances`	Lists all share instances.
GET	`/v2/{tenant_id}/share_instances/{share_instance_id}`	Shows details for a share instance.
POST	`/v2/{tenant_id}/share_instances/{share_instance_id}/action`	Administrator only. Explicitly updates the state of a share instance.
POST	`/v2/{tenant_id}/share_instances/{share_instance_id}/action`	Administrator only. Force-deletes a share instance.

10.12.1. List share instances

Method	URI	Description
GET	`/v2/{tenant_id}/share_instances`	Lists all share instances.

Normal response codes: 200

10.12.1.1. Request

This table shows the header parameters for the list share instances request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share instances request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.12.1.2. Response

Example 10.61. List share instances: JSON response

```
{
    "share_instances": [
        {
            "status": "error",
            "share_id": "406ea93b-32e9-4907-a117-148b3945749f",
            "availability_zone": "nova",
            "replica_state": null,
            "created_at": "2015-09-07T08:41:20.000000",
            "export_location": "10.254.0.3:/shares/share-081f7030-
c54f-42f5-98ee-93a37393e0f2",
            "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
            "export_locations": [
                "10.254.0.3:/shares/share-081f7030-
c54f-42f5-98ee-93a37393e0f2"
            ],
            "share_server_id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73",
            "host": "manila2@generic1#GENERIC1",
            "id": "081f7030-c54f-42f5-98ee-93a37393e0f2"
        },
        {
            "status": "available",
            "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
            "availability_zone": "nova",
            "replica_state": null,
            "created_at": "2015-09-07T08:51:34.000000",
            "export_location": "10.254.0.3:/shares/share-75559a8b-c90c-42a7-
bda2-edbe86acfb7b",
            "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
```

```
            "export_locations": [
                "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-
edbe86acfb7b"
            ],
            "share_server_id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73",
            "host": "manila2@generic1#GENERIC1",
            "id": "75559a8b-c90c-42a7-bda2-edbe86acfb7b"
        }
    ]
}
```

10.12.2. Show share instance details

Method	URI	Description
GET	`/v2/{tenant_id}/share_instances/{share_instance_id}`	Shows details for a share instance.

Normal response codes: 200

10.12.2.1. Request

This table shows the header parameters for the show share instance details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share instance details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_instance_id}`	UUID	The UUID of the share instance.

This operation does not accept a request body.

10.12.2.2. Response

Example 10.62. Show share instance details: JSON response

```
{
    "share_instance": {
        "status": "available",
        "share_id": "d94a8548-2079-4be0-b21c-0a887acd31ca",
        "availability_zone": "nova",
        "replica_state": null,
        "created_at": "2015-09-07T08:51:34.000000",
        "export_location": "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-
edbe86acfb7b",
        "share_network_id": "713df749-aac0-4a54-af52-10f6c991e80c",
        "export_locations": [
            "10.254.0.3:/shares/share-75559a8b-c90c-42a7-bda2-edbe86acfb7b"
        ],
        "share_server_id": "ba11930a-bf1a-4aa7-bae4-a8dfbaa3cc73",
        "host": "manila2@generic1#GENERIC1",
        "access_rules_status": "active",
        "id": "75559a8b-c90c-42a7-bda2-edbe86acfb7b"
    }
}
```

10.12.3. Reset share instance state

Method	URI	Description
POST	`/v2/{tenant_id}/share_instances/{share_instance_id}/action`	Administrator only. Explicitly updates the state of a share instance.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

10.12.3.1. Request

This table shows the header parameters for the reset share instance state request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the reset share instance state request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_instance_id}`	UUID	The UUID of the share instance.

Example 10.63. Reset share instance state: JSON request

```
{
    "reset_status": {
        "status": "available"
    }
}
```

10.12.4. Force-delete share instance

Method	URI	Description
POST	`/v2/{tenant_id}/share_instances/{share_instance_id}/action`	Administrator only. Force-deletes a share instance.

Use the `policy.json` file to grant permissions for this action to other roles.

Normal response codes: 202

10.12.4.1. Request

This table shows the header parameters for the force-delete share instance request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the force-delete share instance request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_instance_id}`	UUID	The UUID of the share instance.

Example 10.64. Force-delete share instance: JSON request

```
{
    "force_delete": null
}
```

10.13. Share instance export locations (since API v2.9)

Set of APIs used to view export locations of share instances.

By default, these APIs are admin-only. Use the `policy.json` file to grant permissions for these actions to other roles.

Method	URI	Description
GET	`/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations`	Lists all export locations for a share instance.
GET	`/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations/{export_location_id}`	Show details of an export location belonging to a share instance.

10.13.1. List export locations

Method	URI	Description
GET	`/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations`	Lists all export locations for a share instance.

Normal response codes: 200

10.13.1.1. Request

This table shows the header parameters for the list export locations request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list export locations request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_instance_id}`	UUID	The UUID of the share instance.

This operation does not accept a request body.

10.13.1.2. Response

Example 10.65. List export locations: JSON response

```
{
    "export_locations": [
        {
            "path": "10.254.0.3:/shares/share-e1c2d35e-fe67-4028-
ad7a-45f668732b1d",
            "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
            "is_admin_only": false,
            "id": "b6bd76ce-12a2-42a9-a30a-8a43b503867d",
            "preferred": false
        },
        {
            "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-
ad7a-45f668732b1d",
            "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
            "is_admin_only": true,
            "id": "6921e862-88bc-49a5-a2df-efeed9acd583",
            "preferred": false
        }
    ]
}
```

10.13.2. Show single export location

Method	URI	Description
GET	`/v2/{tenant_id}/share_instances/{share_instance_id}/export_locations/{export_location_id}`	Show details of an export location belonging to a share instance.

Normal response codes: 200

10.13.2.1. Request

This table shows the header parameters for the show single export location request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show single export location request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_instance_id}`	UUID	The UUID of the share instance.
`{export_location_id}`	UUID	The UUID of the export location.

This operation does not accept a request body.

10.13.2.2. Response

Example 10.66. Show single export location: JSON response

```
{
    "export_location": {
        "created_at": "2016-03-24T14:20:47.000000",
        "updated_at": "2016-03-24T14:20:47.000000",
        "preferred": false,
        "is_admin_only": true,
        "share_instance_id": "e1c2d35e-fe67-4028-ad7a-45f668732b1d",
        "path": "10.0.0.3:/shares/share-e1c2d35e-fe67-4028-ad7a-45f668732b1d",
        "id": "6921e862-88bc-49a5-a2df-efeed9acd583"
    }
}
```

10.14. Share types

A share type enables you to filter or choose back ends before you create a share. A share type behaves in the same way as a Block Storage volume type behaves.

You set a share type to private or public and manage the access to the private share types.

When you issue a create a share type request, you can submit a request body with either a `share_type` or `volume_type` object. The use of the `volume_type` object is deprecated but supported. It is recommended that you use the `share_type` object when you create a share type.

No matter which object type you include in the request, the API creates both a `volume_type` object and a `share_type` object. Both objects have the same ID. When you issue a list share types request, the response shows both `share_types` and `volume_types` objects.

You can set share types as either public or private. By default a share type is created as publicly accessible. Set `share_type_access:is_public` (`os-share-type-access:is_public` for micro-versions 1.0-2.6) to `False` to make the share type private.

You can manage the access to the private share types for the different projects. You can add access, remove access, and get information about access for a private share type.

Administrators can create share types with these extra specifications that are used to filter back ends:

- `driver_handles_share_servers`. Required. Defines the driver mode for share server, or storage, life cycle management. The Shared File Systems service creates a share server for the export of shares.

  Set to `True` when the share driver manages, or handles, the share server life cycle.

  Set to `False` when an administrator rather than a share driver manages the storage life cycle.

- `snapshot_support`. Filters back ends by whether they do or do not support share snapshots.

  Set to `True` to find back ends that support share snapshots.

  Set to `False` to find back ends that do not support share snapshots.

Administrators can also set additional extra specifications for a share type for the following purposes:

- Filter back ends. Specify these unqualified extra specifications in this format: `extra_spec=value`. For example, `netapp_raid_type=raid4`.

- Set data for the driver. Except for the special `capabilities` prefix, you specify these qualified extra specifications with its prefix followed by a colon: `vendor:extra_spec=value`. For example, `netapp:thin_provisioned=true`.

The scheduler uses the special `capabilities` prefix for filtering. The scheduler can only create a share on a back end that reports capabilities that match the unscoped extra-spec keys for the share type. For details, see Capabilities and Extra-Specs.

Each driver implementation determines which extra specification keys it uses. For details, see the documentation for the driver.

An administrator can use the `policy.json` file to grant permissions for share type creation with extra specifications to other roles.

Method	URI	Description
POST	`/v2/{tenant_id}/types`	Creates a share type.
GET	`/v2/{tenant_id}/types`	Lists all share types.
GET	`/v2/{tenant_id}/types/default`	Lists default share types.
DELETE	`/v2/{tenant_id}/types/{share_type_id}`	Deletes a share type.
GET	`/v2/{tenant_id}/types/{share_type_id}/extra_specs`	Lists the extra specifications for a share type.
POST	`/v2/{tenant_id}/types/{share_type_id}/extra_specs`	Sets an extra specification for the share type.
DELETE	`/v2/{tenant_id}/types/{share_type_id}/extra_specs/{key}`	Unsets an extra specification for the share type.
POST	`/v2/{tenant_id}/types/{share_type_id}/action`	Adds share type access for a project.
POST	`/v2/{tenant_id}/types/{share_type_id}/action`	Removes share type access from a project.
GET	`/v2/{tenant_id}/types/{share_type_id}/share_type_access`	Shows access details for a share type.

10.14.1. Create share type

Method	URI	Description
POST	`/v2/{tenant_id}/types`	Creates a share type.

Normal response codes: 200

10.14.1.1. Request

This table shows the header parameters for the create share type request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the create share type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.67. Create share type: JSON request

```
{
    "volume_type": {
        "os-share-type-access:is_public": true,
        "extra_specs": {
            "driver_handles_share_servers": true,
            "snapshot_support": true
        },
        "name": "my_new_volume_type"
    }
}
```

10.14.1.2. Response

Example 10.68. Create share type: JSON response

```
{
    "volume_type": {
        "os-share-type-access:is_public": true,
        "required_extra_specs": {
            "driver_handles_share_servers": true
        },
        "extra_specs": {
            "snapshot_support": "True",
            "driver_handles_share_servers": "True"
        },
        "name": "my_new_volume_type",
        "id": "1d600d02-26a7-4b23-af3d-7d51860fe858"
    },
    "share_type": {
        "os-share-type-access:is_public": true,
        "required_extra_specs": {
```

```
            "driver_handles_share_servers": true
        },
        "extra_specs": {
            "snapshot_support": "True",
            "driver_handles_share_servers": "True"
        },
        "name": "my_new_volume_type",
        "id": "1d600d02-26a7-4b23-af3d-7d51860fe858"
    }
}
```

10.14.2. List share types

Method	URI	Description
GET	`/v2/{tenant_id}/types`	Lists all share types.

Normal response codes: 200

10.14.2.1. Request

This table shows the header parameters for the list share types request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list share types request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.14.2.2. Response

Example 10.69. List share types: JSON response

```
{
    "volume_types": [
        {
            "os-share-type-access:is_public": true,
            "required_extra_specs": {
                "driver_handles_share_servers": "True"
            },
            "extra_specs": {
                "snapshot_support": "True",
                "driver_handles_share_servers": "True"
            },
            "name": "default",
            "id": "be27425c-f807-4500-a056-d00721db45cf"
        },
        {
            "os-share-type-access:is_public": true,
            "required_extra_specs": {
                "driver_handles_share_servers": "false"
            },
            "extra_specs": {
                "snapshot_support": "True",
                "driver_handles_share_servers": "false"
            },
            "name": "d",
            "id": "f015bebe-c38b-4c49-8832-00143b10253b"
        }
    ],
    "share_types": [
```

```
        {
            "os-share-type-access:is_public": true,
            "required_extra_specs": {
                "driver_handles_share_servers": "True"
            },
            "extra_specs": {
                "snapshot_support": "True",
                "driver_handles_share_servers": "True"
            },
            "name": "default",
            "id": "be27425c-f807-4500-a056-d00721db45cf"
        },
        {
            "os-share-type-access:is_public": true,
            "required_extra_specs": {
                "driver_handles_share_servers": "false"
            },
            "extra_specs": {
                "snapshot_support": "True",
                "driver_handles_share_servers": "false"
            },
            "name": "d",
            "id": "f015bebe-c38b-4c49-8832-00143b10253b"
        }
    ]
}
```

10.14.3. List default share types

Method	URI	Description
GET	`/v2/{tenant_id}/types/default`	Lists default share types.

Normal response codes: 200

10.14.3.1. Request

This table shows the header parameters for the list default share types request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list default share types request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.14.3.2. Response

Example 10.70. List default share types: JSON response

```
{
    "volume_type": {
        "required_extra_specs": null,
        "extra_specs": {
            "snapshot_support": "True",
            "driver_handles_share_servers": "True"
        },
        "name": "default",
        "id": "be27425c-f807-4500-a056-d00721db45cf"
    },
    "share_type": {
        "required_extra_specs": null,
        "extra_specs": {
            "snapshot_support": "True",
            "driver_handles_share_servers": "True"
        },
        "name": "default",
        "id": "be27425c-f807-4500-a056-d00721db45cf"
    }
}
```

10.14.4. Delete share type

Method	URI	Description
DELETE	`/v2/{tenant_id}/types/{share_type_id}`	Deletes a share type.

Normal response codes: 202

10.14.4.1. Request

This table shows the header parameters for the delete share type request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete share type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.

This operation does not accept a request body.

10.14.5. List extra specs

Method	URI	Description
GET	`/v2/{tenant_id}/types/{share_type_id}/extra_specs`	Lists the extra specifications for a share type.

Normal response codes: 200

10.14.5.1. Request

This table shows the header parameters for the list extra specs request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list extra specs request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.

This operation does not accept a request body.

10.14.5.2. Response

Example 10.71. List extra specs: JSON response

```
{
    "extra_specs": {
        "snapshot_support": "True",
        "driver_handles_share_servers": "True"
    }
}
```

10.14.6. Set extra spec for share type

Method	URI	Description
POST	`/v2/{tenant_id}/types/{share_type_id}/extra_specs`	Sets an extra specification for the share type.

Each driver implementation determines which extra specification keys it uses. For details, see Capabilities and Extra-Specs and documentation for your driver.

Normal response codes: 200

10.14.6.1. Request

This table shows the header parameters for the set extra spec for share type request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the set extra spec for share type request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.

Example 10.72. Set extra spec for share type: JSON request

```
{
    "extra_specs": {
        "my_key": "my_value"
    }
}
```

10.14.6.2. Response

Example 10.73. Set extra spec for share type: JSON response

```
{
    "extra_specs": {
        "my_key": "my_value"
    }
}
```

10.14.7. Unset an extra spec

Method	URI	Description
DELETE	`/v2/{tenant_id}/types/{share_type_id}/extra_specs/{key}`	Unsets an extra specification for the share type.

Normal response codes: 202

10.14.7.1. Request

This table shows the header parameters for the unset an extra spec request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the unset an extra spec request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.
`{extra-spec-key}`	String	The extra specification key.

This operation does not accept a request body.

10.14.8. Add share type access

Method	URI	Description
POST	`/v2/{tenant_id}/types/{share_type_id}/action`	Adds share type access for a project.

You can add access to private share types only.

Normal response codes: 202

10.14.8.1. Request

This table shows the header parameters for the add share type access request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the add share type access request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.

Example 10.74. Add share type access: JSON request

```
{
    "addProjectAccess": {
        "project": "e1284adea3ee4d2482af5ed214f3ad90"
    }
}
```

10.14.9. Remove share type access

Method	URI	Description
POST	`/v2/{tenant_id}/types/{share_type_id}/action`	Removes share type access from a project.

You can remove access from private share types only.

Normal response codes: 202

10.14.9.1. Request

This table shows the header parameters for the remove share type access request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the remove share type access request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.

Example 10.75. Remove share type access: JSON request

```
{
    "removeProjectAccess": {
        "project": "818a3f48dcd644909b3fa2e45a399a27"
    }
}
```

10.14.10. Show share type access details

Method	URI	Description
GET	`/v2/{tenant_id}/types/{share_type_id}/share_type_access`	Shows access details for a share type.

You can view access details for private share types only.

Normal response codes: 200

10.14.10.1. Request

This table shows the header parameters for the show share type access details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show share type access details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_type_id}`	UUID	The UUID of the share type.

This operation does not accept a request body.

10.14.10.2. Response

Example 10.76. Show share type access details: JSON response

```
{
    "share_type_access": [
        {
            "share_type_id": "1732f284-401d-41d9-a494-425451e8b4b8",
            "project_id": "818a3f48dcd644909b3fa2e45a399a27"
        },
        {
            "share_type_id": "1732f284-401d-41d9-a494-425451e8b4b8",
            "project_id": "e1284adea3ee4d2482af5ed214f3ad90"
        }
    ]
}
```

10.15. Back-end storage pools

An administrator can list all back-end storage pools that are known to the scheduler service.

Method	URI	Description
GET	`/v2/{tenant_id}/scheduler-stats/pools`	Lists all back-end storage pools.

Method	URI	Description
GET	`/v2/{tenant_id}/scheduler-stats/pools/detail`	Lists all storage pools for a back end, with details.

10.15.1. List back-end storage pools

Method	URI	Description
GET	`/v2/{tenant_id}/scheduler-stats/pools`	Lists all back-end storage pools.

Normal response codes: 200

10.15.1.1. Request

This table shows the header parameters for the list back-end storage pools request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list back-end storage pools request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.15.1.2. Response

Example 10.77. List back-end storage pools: JSON response

```
{
    "pools": [
        {
            "host": "manila2",
            "name": "manila2@generic1#GENERIC1",
            "pool": "GENERIC1",
            "backend": "generic1"
        },
        {
            "host": "manila2",
            "name": "manila2@unmanage1#UNMANAGE1",
            "pool": "UNMANAGE1",
            "backend": "unmanage1"
        },
        {
            "host": "manila2",
            "name": "manila2@ams_backend#AMS_BACKEND",
            "pool": "AMS_BACKEND",
            "backend": "ams_backend"
        }
    ]
}
```

10.15.2. List back-end storage pools with details

Method	URI	Description
GET	`/v2/{tenant_id}/scheduler-stats/pools/detail`	Lists all storage pools for a back end, with details.

Normal response codes: 200

10.15.2.1. Request

This table shows the header parameters for the list back-end storage pools with details request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list back-end storage pools with details request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.15.2.2. Response

Example 10.78. List back-end storage pools with details: JSON response

```
{
    "pools": [
        {
            "pool": "LONDON",
            "host": "nosb-devstack",
            "name": "nosb-devstack@london#LONDON",
            "capabilities": {
                "qos": false,
                "consistency_group_support": "pool",
                "timestamp": "2015-09-21T08:58:56.190856",
                "share_backend_name": "LONDON",
                "server_pools_mapping": {
                    "1320689d-80f4-49f6-8a70-0e2c1ed8ad90": [],
                    "3a4caac5-0880-4629-a334-6cdda88a0c0e": []
                },
                "driver_handles_share_servers": true,
                "driver_version": "1.0",
                "total_capacity_gb": "unknown",
                "reserved_percentage": 0,
                "pools": null,
                "vendor_name": "Open Source",
                "snapshot_support": true,
                "free_capacity_gb": "unknown",
                "storage_protocol": "NFS_CIFS"
```

```
            },
            "backend": "london"
        }
    ]
}
```

10.16. Services

Lists services. Services include `manila-share` and `manila-scheduler` and their binaries, hosts, availability zones, current statuses, and states (`up` or `down`).

Administrators can also enable or disable a service.

Caution

For old microversions 1.0-2.6 replace URL part `services` with `os-services`.

Method	URI	Description
GET	`/v2/{tenant_id}/services`	Lists all services.
PUT	`/v2/{tenant_id}/services/enable`	Enables a service.
PUT	`/v2/{tenant_id}/services/disable`	Disables a service.

10.16.1. List services

Method	URI	Description
GET	`/v2/{tenant_id}/services`	Lists all services.

Normal response codes: 200

10.16.1.1. Request

This table shows the header parameters for the list services request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list services request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.16.1.2. Response

Example 10.79. List services: JSON response

```
{
    "services": [
        {
            "status": "enabled",
            "binary": "manila-share",
            "zone": "nova",
            "host": "manila2@generic1",
            "updated_at": "2015-09-07T13:03:57.000000",
            "state": "up",
            "id": 1
        },
        {
            "status": "enabled",
            "binary": "manila-scheduler",
            "zone": "nova",
            "host": "manila2",
            "updated_at": "2015-09-07T13:03:57.000000",
            "state": "up",
            "id": 2
        }
    ]
}
```

10.16.2. Enable service

Method	URI	Description
PUT	`/v2/{tenant_id}/services/enable`	Enables a service.

Normal response codes: 200

10.16.2.1. Request

This table shows the header parameters for the enable service request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the enable service request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.80. Enable service: JSON request

```
{
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

10.16.2.2. Response

Example 10.81. Enable service: JSON response

```
{
    "disabled": false,
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

10.16.3. Disable service

Method	URI	Description
PUT	`/v2/{tenant_id}/services/disable`	Disables a service.

Normal response codes: 200

10.16.3.1. Request

This table shows the header parameters for the disable service request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the disable service request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.82. Disable service: JSON request

```
{
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

10.16.3.2. Response

Example 10.83. Disable service: JSON response

```
{
    "disabled": true,
    "binary": "manila-share",
    "host": "openstack@cmode"
}
```

10.17. Availability zones

Describes availability zones.

Caution

For old microversions 1.0-2.6 replace URL part `availability-zones` with `os-availability-zone`.

Method	URI	Description
GET	`/v2/{tenant_id}/availability-zones`	Lists all availability zones.

10.17.1. List availability zones

Method	URI	Description
GET	`/v2/{tenant_id}/availability-zones`	Lists all availability zones.

Normal response codes: 200

10.17.1.1. Request

This table shows the header parameters for the list availability zones request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the list availability zones request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

This operation does not accept a request body.

10.17.1.2. Response

Example 10.84. List availability zones: JSON response

```
{
    "availability_zones": [
        {
            "name": "nova",
            "created_at": "2015-09-18T09:50:55.000000",
            "updated_at": null,
            "id": "388c983d-258e-4a0e-b1ba-10da37d766db"
        }
    ]
}
```

10.18. Manage share

Configures Shared File Systems to manage or unmanage a share.

The share unmanage operation is not supported for shares that are created on top of share servers (created with share networks).

You can unmanage a share that has no dependent snapshots.

These APIs are available for micro-versions 1.0-2.6

Starting with micro-version 2.7 these APIs are available under 'shares' API group.

Method	URI	Description
POST	`/v2/{tenant_id}/os-share-manage`	Configures Shared File Systems to manage a share.

Method	URI	Description
POST	`/v2/{tenant_id}/os-share-unmanage/{share_id}/unmanage`	Configures Shared File Systems to stop managing a share.

10.18.1. Manage share

Method	URI	Description
POST	`/v2/{tenant_id}/os-share-manage`	Configures Shared File Systems to manage a share.

Normal response codes: 200

10.18.1.1. Request

This table shows the header parameters for the manage share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the manage share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.

Example 10.85. Manage share: JSON request

```
{
    "share": {
        "protocol": "nfs",
        "name": "share_texas1",
        "share_type": "d",
        "is_public": false,
        "driver_options": {
            "opt1": "opt1",
            "opt2": "opt2"
        },
        "export_path": "10.254.0.5:/shares/share-42033c24-0261-424f-
abda-4fef2f6dbfd5",
        "service_host": "manila2@unmanage1#UNMANAGE1",
        "description": "Lets manage share."
    }
}
```

10.18.1.2. Response

Example 10.86. Manage share: JSON response

```
{
    "share": {
        "links": [
            {
                "href": "http://172.18.198.54:8786/v2/
16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
                "rel": "self"
            },
            {
                "href": "http://172.18.198.54:8786/
16e1ab15c35a457e9c2b2aa189f544e1/shares/00137b40-ca06-4ae8-83a3-2c5989eebcce",
```

```
                "rel": "bookmark"
            }
        ],
        "availability_zone": null,
        "share_network_id": null,
        "export_locations": [],
        "share_server_id": null,
        "snapshot_id": null,
        "id": "00137b40-ca06-4ae8-83a3-2c5989eebcce",
        "size": null,
        "share_type": "14747856-08e5-494f-ab40-a64b9d20d8f7",
        "share_type_name": "d",
        "export_location": "10.254.0.5:/shares/share-42033c24-0261-424f-
abda-4fef2f6dbfd5",
        "consistency_group_id": null,
        "project_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "metadata": {},
        "status": "manage_starting",
        "description": "Lets manage share.",
        "host": "manila2@unmanage1#UNMANAGE1",
        "access_rules_status": "active",
        "has_replicas": false,
        "replication_type": null,
        "is_public": false,
        "snapshot_support": true,
        "name": "share_texas1",
        "created_at": "2015-09-17T16:21:12.000000",
        "share_proto": "NFS",
        "volume_type": "d",
        "source_cgsnapshot_member_id": null
    }
}
```

10.18.2. Unmanage share

Method	URI	Description
POST	`/v2/{tenant_id}/os-share-unmanage/{share_id}/unmanage`	Configures Shared File Systems to stop managing a share.

Normal response codes: 202

10.18.2.1. Request

This table shows the header parameters for the unmanage share request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the unmanage share request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{share_id}`	UUID	The UUID of the share.

This operation does not accept a request body.

10.19. Quota sets

Provides quotas management support.

Caution

For old microversions 1.0-2.6 replace URL part `quota-sets` with `os-quota-sets`.

Method	URI	Description
GET	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Shows quotas for a tenant.
PUT	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Updates quotas for a tenant.
DELETE	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Deletes quotas for a tenant. The quota reverts to the default quota.
GET	`/v2/{tenant_id}/quota-sets/{tenant_id}/defaults`	Shows default quotas for a tenant.

10.19.1. Show quotas

Method	URI	Description
GET	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Shows quotas for a tenant.

If you specify the optional `user_id` query parameter, you get the quotas for this user in the tenant. If you omit this parameter, you get the quotas for the project.

Normal response codes: 200

10.19.1.1. Request

This table shows the header parameters for the show quotas request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{tenant_id}`	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

This operation does not accept a request body.

10.19.1.2. Response

Example 10.87. Show quotas: JSON response

```
{
    "quota_set": {
        "gigabytes": 1000,
        "shares": 50,
        "snapshot_gigabytes": 1000,
        "snapshots": 50,
        "id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "share_networks": 10
    }
}
```

10.19.2. Update quotas

Method	URI	Description
PUT	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Updates quotas for a tenant.

If you specify the optional `user_id` query parameter, you update the quotas for this user in the tenant. If you omit this parameter, you update the quotas for the project.

Normal response codes: 200

10.19.2.1. Request

This table shows the header parameters for the update quotas request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the update quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{tenant_id}`	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

Example 10.88. Update quotas: JSON request

```
{
    "quota_set": {
        "tenant_id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "snapshot_gigabytes": 999,
        "snapshots": 49,
        "share_networks": 9
    }
}
```

10.19.2.2. Response

Example 10.89. Update quotas: JSON response

```
{
    "quota_set": {
        "gigabytes": 1000,
        "snapshot_gigabytes": 999,
        "shares": 50,
        "snapshots": 49,
        "share_networks": 9
    }
}
```

10.19.3. Delete quotas

Method	URI	Description
DELETE	`/v2/{tenant_id}/quota-sets/{tenant_id}{?user_id}`	Deletes quotas for a tenant. The quota reverts to the default quota.

If you specify the optional `user_id` query parameter, you delete the quotas for this user in the tenant. If you omit this parameter, you delete the quotas for the project.

Normal response codes: 202

10.19.3.1. Request

This table shows the header parameters for the delete quotas request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the delete quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{tenant_id}`	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

This operation does not accept a request body.

10.19.4. Show default quotas

Method	URI	Description
GET	`/v2/{tenant_id}/quota-sets/{tenant_id}/defaults`	Shows default quotas for a tenant.

Normal response codes: 200

10.19.4.1. Request

This table shows the header parameters for the show default quotas request:

Name	Type	Description
`X-Openstack-Manila-Api-Version`	String *(Optional)*	The HTTP header to specify a valid Shared File Systems API microversion. For example, `"X-Openstack-Manila-Api-Version: 2.6"`. If you omit this header, the default microversion is 2.0.

This table shows the URI parameters for the show default quotas request:

Name	Type	Description
`{tenant_id}`	UUID	The UUID of the tenant in a multi-tenancy cloud.
`{tenant_id}`	UUID	The UUID for the tenant for which you want to show, update, or delete quotas. This ID is different from the first tenant ID that you specify in the URI: That ID is for the administrative tenant.

This operation does not accept a request body.

10.19.4.2. Response

Example 10.90. Show default quotas: JSON response

```
{
    "quota_set": {
        "gigabytes": 1000,
        "shares": 50,
        "snapshot_gigabytes": 1000,
        "snapshots": 50,
        "id": "16e1ab15c35a457e9c2b2aa189f544e1",
        "share_networks": 10
    }
}
```

11. Shared File Systems API v1 (SUPPORTED)

Provides coordinated access to shared or distributed file systems.

The Shared File Systems API v1 is functionally identical to the Shared File Systems API v2. Subsequent API v2 micro-versions, such as v2.1, differ from API v1.

(Since API v2.0) The API uses Compute-style micro-versions.

12. Telemetry API v2 (CURRENT)

Manages alarms, meters, samples, resources, and capabilities through a set of services:

- Aodh. An alarm service.
- Ceilometer . A data collection service.
- Gnocchi. A time-series database and resource indexing service.

If Gnocchi is enabled, meters, samples, and resources API operations return the `410` response code.

Method	URI	Description
	Alarms	
GET	`/v2/alarms{?q}`	Lists alarms, based on a query.
POST	`/v2/alarms{?data}`	Creates an alarm.
GET	`/v2/alarms/{alarm_id}`	Shows details for an alarm, by alarm ID.
PUT	`/v2/alarms/{alarm_id}{?data}`	Updates an alarm.
DELETE	`/v2/alarms/{alarm_id}`	Deletes an alarm, by alarm ID.
PUT	`/v2/alarms/{alarm_id}/state{?state}`	Sets the state of an alarm.
GET	`/v2/alarms/{alarm_id}/state`	Shows the state for an alarm, by alarm ID.
GET	`/v2/alarms/{alarm_id}/history{?q}`	Assembles and shows the history for an alarm, by alarm ID.
	Meters	
GET	`/v2/meters{?q,limit}`	Lists meters, based on the data recorded so far.
POST	`/v2/meters/{meter_name}{?direct,samples}`	Adds samples to a meter, by meter name.
GET	`/v2/meters/{meter_name}{?q,limit}`	Lists samples for a meter, by meter name.
GET	`/v2/meters/{meter_name}/statistics{?q,groupby,period,aggregate,limit}`	Computes and lists statistics for samples in a time range.
	Samples	
GET	`/v2/samples{?q,limit}`	Lists all known samples, based on the data recorded so far.
GET	`/v2/samples/{sample_id}`	Shows details for a sample, by sample ID.
	Resources	
GET	`/v2/resources{?q,meter_links}`	Lists definitions for all resources.
GET	`/v2/resources/{resource_id}`	Shows details for a resource, by resource ID.
	Capabilities	
GET	`/v2/capabilities`	A representation of the API and storage capabilities. Usually, the storage driver imposes constraints.
	Events	
GET	`/v2/events{?q,limit}`	Lists all events.
GET	`/v2/events/{message_id}`	Shows details for an event.

12.1. Alarms

Lists, creates, gets details for, updates, and deletes alarms.

Method	URI	Description
GET	`/v2/alarms{?q}`	Lists alarms, based on a query.
POST	`/v2/alarms{?data}`	Creates an alarm.
GET	`/v2/alarms/{alarm_id}`	Shows details for an alarm, by alarm ID.
PUT	`/v2/alarms/{alarm_id}{?data}`	Updates an alarm.
DELETE	`/v2/alarms/{alarm_id}`	Deletes an alarm, by alarm ID.
PUT	`/v2/alarms/{alarm_id}/state{?state}`	Sets the state of an alarm.
GET	`/v2/alarms/{alarm_id}/state`	Shows the state for an alarm, by alarm ID.
GET	`/v2/alarms/{alarm_id}/history{?q}`	Assembles and shows the history for an alarm, by alarm ID.

12.1.1. List alarms

Method	URI	Description
GET	`/v2/alarms{?q}`	Lists alarms, based on a query.

Normal response codes: 200

12.1.1.1. Request

This table shows the query parameters for the list alarms request:

Name	Type	Description
q	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.

This operation does not accept a request body.

12.1.1.2. Response

Example 12.1. List alarms: JSON response

```
[
    {
        "alarm_actions": [
            "http://site:8000/alarm"
        ],
        "alarm_id": null,
        "combination_rule": null,
        "description": "An alarm",
        "enabled": true,
        "insufficient_data_actions": [
            "http://site:8000/nodata"
        ],
        "name": "SwiftObjectAlarm",
        "ok_actions": [
            "http://site:8000/ok"
        ],
        "project_id": "c96c887c216949acbdfbd8b494863567",
        "repeat_actions": false,
        "state": "ok",
        "state_timestamp": "2013-11-21T12:33:08.486228",
        "threshold_rule": null,
        "timestamp": "2013-11-21T12:33:08.486221",
        "type": "threshold",
        "user_id": "c96c887c216949acbdfbd8b494863567"
    }
]
```

Example 12.2. List alarms: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
    <value>
        <alarm_actions>
```

```
            <item>http://site:8000/alarm</item>
        </alarm_actions>
        <alarm_id nil="true" />
        <combination_rule nil="true" />
        <description>An alarm</description>
        <enabled>true</enabled>
        <insufficient_data_actions>
            <item>http://site:8000/nodata</item>
        </insufficient_data_actions>
        <name>SwiftObjectAlarm</name>
        <ok_actions>
            <item>http://site:8000/ok</item>
        </ok_actions>
        <project_id>c96c887c216949acbdfbd8b494863567</project_id>
        <repeat_actions>false</repeat_actions>
        <state>ok</state>
        <state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
        <threshold_rule nil="true" />
        <timestamp>2013-11-21T12:33:08.486221</timestamp>
        <type>threshold</type>
        <user_id>c96c887c216949acbdfbd8b494863567</user_id>
    </value>
</values>
```

This operation does not return a response body.

12.1.2. Create alarm

Method	URI	Description
POST	`/v2/alarms{?data}`	Creates an alarm.

Normal response codes: 200

12.1.2.1. Request

This table shows the query parameters for the create alarm request:

Name	Type	Description
`data`	Alarm *(Optional)*	An alarm within the request body.

This operation does not accept a request body.

12.1.2.2. Response

Example 12.3. Create alarm: JSON response

```
{
    "alarm_actions": [
        "http://site:8000/alarm"
    ],
    "alarm_id": null,
    "combination_rule": null,
    "description": "An alarm",
    "enabled": true,
    "insufficient_data_actions": [
        "http://site:8000/nodata"
    ],
    "name": "SwiftObjectAlarm",
    "ok_actions": [
        "http://site:8000/ok"
    ],
    "project_id": "c96c887c216949acbdfbd8b494863567",
    "repeat_actions": false,
    "state": "ok",
    "state_timestamp": "2013-11-21T12:33:08.486228",
    "threshold_rule": null,
    "timestamp": "2013-11-21T12:33:08.486221",
    "type": "threshold",
    "user_id": "c96c887c216949acbdfbd8b494863567"
}
```

Example 12.4. Create alarm: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
    <alarm_actions>
        <item>http://site:8000/alarm</item>
    </alarm_actions>
    <alarm_id nil="true" />
```

```
    <combination_rule nil="true" />
    <description>An alarm</description>
    <enabled>true</enabled>
    <insufficient_data_actions>
        <item>http://site:8000/nodata</item>
    </insufficient_data_actions>
    <name>SwiftObjectAlarm</name>
    <ok_actions>
        <item>http://site:8000/ok</item>
    </ok_actions>
    <project_id>c96c887c216949acbdfbd8b494863567</project_id>
    <repeat_actions>false</repeat_actions>
    <state>ok</state>
    <state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
    <threshold_rule nil="true" />
    <timestamp>2013-11-21T12:33:08.486221</timestamp>
    <type>threshold</type>
    <user_id>c96c887c216949acbdfbd8b494863567</user_id>
</value>
```

This operation does not return a response body.

12.1.3. Show alarm details

Method	URI	Description
GET	`/v2/alarms/{alarm_id}`	Shows details for an alarm, by alarm ID.

Normal response codes: 200

12.1.3.1. Request

This table shows the URI parameters for the show alarm details request:

Name	Type	Description
`{alarm_id}`	UUID	The UUID of the alarm.

This operation does not accept a request body.

12.1.3.2. Response

Example 12.5. Show alarm details: JSON response

```
{
    "alarm_actions": [
        "http://site:8000/alarm"
    ],
    "alarm_id": null,
    "combination_rule": null,
    "description": "An alarm",
    "enabled": true,
    "insufficient_data_actions": [
        "http://site:8000/nodata"
    ],
    "name": "SwiftObjectAlarm",
    "ok_actions": [
        "http://site:8000/ok"
    ],
    "project_id": "c96c887c216949acbdfbd8b494863567",
    "repeat_actions": false,
    "state": "ok",
    "state_timestamp": "2013-11-21T12:33:08.486228",
    "threshold_rule": null,
    "timestamp": "2013-11-21T12:33:08.486221",
    "type": "threshold",
    "user_id": "c96c887c216949acbdfbd8b494863567"
}
```

Example 12.6. Show alarm details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
    <alarm_actions>
        <item>http://site:8000/alarm</item>
    </alarm_actions>
    <alarm_id nil="true" />
    <combination_rule nil="true" />
    <description>An alarm</description>
```

```
    <enabled>true</enabled>
    <insufficient_data_actions>
        <item>http://site:8000/nodata</item>
    </insufficient_data_actions>
    <name>SwiftObjectAlarm</name>
    <ok_actions>
        <item>http://site:8000/ok</item>
    </ok_actions>
    <project_id>c96c887c216949acbdfbd8b494863567</project_id>
    <repeat_actions>false</repeat_actions>
    <state>ok</state>
    <state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
    <threshold_rule nil="true" />
    <timestamp>2013-11-21T12:33:08.486221</timestamp>
    <type>threshold</type>
    <user_id>c96c887c216949acbdfbd8b494863567</user_id>
</value>
```

This operation does not return a response body.

12.1.4. Update alarm

Method	URI	Description
PUT	`/v2/alarms/{alarm_id}{?data}`	Updates an alarm.

Normal response codes: 200

12.1.4.1. Request

This table shows the URI parameters for the update alarm request:

Name	Type	Description
`{alarm_id}`	UUID	The UUID of the alarm.

This table shows the query parameters for the update alarm request:

Name	Type	Description
`data`	Alarm *(Optional)*	An alarm within the request body.

This operation does not accept a request body.

12.1.4.2. Response

Example 12.7. Update alarm: JSON response

```
{
    "alarm_actions": [
        "http://site:8000/alarm"
    ],
    "alarm_id": null,
    "combination_rule": null,
    "description": "An alarm",
    "enabled": true,
    "insufficient_data_actions": [
        "http://site:8000/nodata"
    ],
    "name": "SwiftObjectAlarm",
    "ok_actions": [
        "http://site:8000/ok"
    ],
    "project_id": "c96c887c216949acbdfbd8b494863567",
    "repeat_actions": false,
    "state": "ok",
    "state_timestamp": "2013-11-21T12:33:08.486228",
    "threshold_rule": null,
    "timestamp": "2013-11-21T12:33:08.486221",
    "type": "threshold",
    "user_id": "c96c887c216949acbdfbd8b494863567"
}
```

Example 12.8. Update alarm: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<value>
    <alarm_actions>
        <item>http://site:8000/alarm</item>
    </alarm_actions>
    <alarm_id nil="true" />
    <combination_rule nil="true" />
    <description>An alarm</description>
    <enabled>true</enabled>
    <insufficient_data_actions>
        <item>http://site:8000/nodata</item>
    </insufficient_data_actions>
    <name>SwiftObjectAlarm</name>
    <ok_actions>
        <item>http://site:8000/ok</item>
    </ok_actions>
    <project_id>c96c887c216949acbdfbd8b494863567</project_id>
    <repeat_actions>false</repeat_actions>
    <state>ok</state>
    <state_timestamp>2013-11-21T12:33:08.486228</state_timestamp>
    <threshold_rule nil="true" />
    <timestamp>2013-11-21T12:33:08.486221</timestamp>
    <type>threshold</type>
    <user_id>c96c887c216949acbdfbd8b494863567</user_id>
</value>
```

This operation does not return a response body.

12.1.5. Delete alarm

Method	URI	Description
DELETE	`/v2/alarms/{alarm_id}`	Deletes an alarm, by alarm ID.

Normal response codes: 204

12.1.5.1. Request

This table shows the URI parameters for the delete alarm request:

Name	Type	Description
`{alarm_id}`	UUID	The UUID of the alarm.

This operation does not accept a request body.

12.1.6. Update alarm state

Method	URI	Description
PUT	`/v2/alarms/{alarm_id}/state{?state}`	Sets the state of an alarm.

Normal response codes: 200

12.1.6.1. Request

This table shows the URI parameters for the update alarm state request:

Name	Type	Description
`{alarm_id}`	UUID	The UUID of the alarm.

This table shows the query parameters for the update alarm state request:

Name	Type	Description
`state`	String *(Required)*	The alarm state. A valid value is `ok`, `alarm`, or `insufficient data`.

This operation does not accept a request body.

12.1.7. Show alarm state

Method	URI	Description
GET	`/v2/alarms/{alarm_id}/state`	Shows the state for an alarm, by alarm ID.

Normal response codes: 200

12.1.7.1. Request

This table shows the URI parameters for the show alarm state request:

Name	Type	Description
`{alarm_id}`	UUID	The UUID of the alarm.

12.1.8. Show alarm history

Method	URI	Description
GET	`/v2/alarms/{alarm_id}/history{?q}`	Assembles and shows the history for an alarm, by alarm ID.

Normal response codes: 200

12.1.8.1. Request

This table shows the URI parameters for the show alarm history request:

Name	Type	Description
`{alarm_id}`	UUID	The UUID of the alarm.

This table shows the query parameters for the show alarm history request:

Name	Type	Description
q	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.

This operation does not accept a request body.

12.2. Meters

Lists all meters, adds samples to meters, and lists samples for meters. For list operations, if you do not explicitly set the `limit` query parameter, a default limit is applied. The default limit is the `default_api_return_limit` configuration option value.

Also, computes and lists statistics for samples in a time range. You can use the `aggregate` query parameter in the `statistics` URI to explicitly select the `stddev`, `cardinality`, or any other standard function. For example:

```
GET /v2/meters/METER_NAME/statistics?aggregate.func=NAME&aggregate.param=VALUE
```

The `aggregate.param` parameter value is optional for all functions except the `cardinality` function.

The API silently ignores any duplicate aggregate function and parameter pairs.

The API accepts and storage drivers support duplicate functions with different parameter values. In this example, the `cardinality` function is accepted twice with two different parameter values:

```
GET /v2/meters/METER_NAME/statistics?aggregate.func=cardinality&aggregate.
param=resource_id
        &aggregate.func=cardinality&aggregate.param=project_id
```

Examples:

Use the `stddev` function to request the standard deviation of CPU utilization:

```
GET /v2/meters/cpu_util/statistics?aggregate.func=stddev
```

The response looks like this:

```
[
    {
        "aggregate": {
            "stddev": 0.6858829
        },
        "duration_start": "2014-01-30T11:13:23",
        "duration_end": "2014-01-31T16:07:13",
        "duration": 104030,
        "period": 0,
        "period_start": "2014-01-30T11:13:23",
        "period_end": "2014-01-31T16:07:13",
        "groupby": null,
        "unit": "%"
    }
]
```

Use the `cardinality` function with the project ID to return the number of distinct tenants with images:

```
GET /v2/meters/image/statistics?aggregate.func=cardinality&aggregate.param=
project_id
```

The following, more complex, example determines:

- The number of distinct instances (`cardinality`)
- The total number of instance samples (`count`) for a tenant in 15-minute intervals (`period` and `groupby` options)

```
GET /v2/meters/instance/statistics?aggregate.func=cardinality&aggregate.param=
resource_id
        &aggregate.func=count&groupby=project_id&period=900
```

The response looks like this:

```
[
    {
        "count": 19,
        "aggregate": {
            "count": 19,
            "cardinality/resource_id": 3
        },
        "duration": 328.47803,
        "duration_start": "2014-01-31T10:00:41.823919",
        "duration_end": "2014-01-31T10:06:10.301948",
        "period": 900,
        "period_start": "2014-01-31T10:00:00",
        "period_end": "2014-01-31T10:15:00",
        "groupby": {
            "project_id": "061a5c91811e4044b7dc86c6136c4f99"
        },
        "unit": "instance"
    },
    {
        "count": 22,
        "aggregate": {
```

```
            "count": 22,
            "cardinality/resource_id": 4
        },
        "duration": 808.00385,
        "duration_start": "2014-01-31T10:15:15",
        "duration_end": "2014-01-31T10:28:43.003840",
        "period": 900,
        "period_start": "2014-01-31T10:15:00",
        "period_end": "2014-01-31T10:30:00",
        "groupby": {
            "project_id": "061a5c91811e4044b7dc86c6136c4f99"
        },
        "unit": "instance"
    },
    {
        "count": 2,
        "aggregate": {
            "count": 2,
            "cardinality/resource_id": 2
        },
        "duration": 0,
        "duration_start": "2014-01-31T10:35:15",
        "duration_end": "2014-01-31T10:35:15",
        "period": 900,
        "period_start": "2014-01-31T10:30:00",
        "period_end": "2014-01-31T10:45:00",
        "groupby": {
            "project_id": "061a5c91811e4044b7dc86c6136c4f99"
        },
        "unit": "instance"
    }
]
```


Method	URI	Description
GET	`/v2/meters{?q,limit}`	Lists meters, based on the data recorded so far.
POST	`/v2/meters/{meter_name}{?direct,samples}`	Adds samples to a meter, by meter name.
GET	`/v2/meters/{meter_name}{?q,limit}`	Lists samples for a meter, by meter name.
GET	`/v2/meters/{meter_name}/statistics{?q,groupby,period,aggregate,limit}`	Computes and lists statistics for samples in a time range.

12.2.1. List meters

Method	URI	Description
GET	`/v2/meters{?q,limit}`	Lists meters, based on the data recorded so far.

Normal response codes: 200

12.2.1.1. Request

This table shows the query parameters for the list meters request:

Name	Type	Description
`q`	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

12.2.1.2. Response

Example 12.9. List meters: JSON response

```
[
    {
        "meter_id":
 "YmQ5NDMxYzEtOGQ2OS00YWQzLTgwM2EtOGQ0YTZiODlmZDM2K2luc3RhbmNl",
        "name": "instance",
        "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
        "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
        "source": "openstack",
        "type": "gauge",
        "unit": "instance",
        "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
    }
]
```

Example 12.10. List meters: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
   <value>
      <name>instance</name>
      <type>gauge</type>
      <unit>instance</unit>
      <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
      <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
      <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
      <source>openstack</source>
      <meter_id>YmQ5NDMxYzEtOGQ2OS00YWQzLTgwM2EtOGQ0YTZiODlmZDM2K2luc3RhbmNl</
meter_id>
   </value>
```

```
</values>
```

This operation does not return a response body.

12.2.2. Add samples to meter

Method	URI	Description
POST	`/v2/meters/{meter_name}{?direct, samples}`	Adds samples to a meter, by meter name.

If you attempt to add a sample that is not supported, this call returns the `409` response code.

Normal response codes: 200

Error response codes: conflict (409)

12.2.2.1. Request

This table shows the URI parameters for the add samples to meter request:

Name	Type	Description
`{meter_name}`	String	The name of the meter.

This table shows the query parameters for the add samples to meter request:

Name	Type	Description
`direct`	String *(Optional)*	Indicates whether the samples are **POST**ed directly to storage. Set `?direct=True` to **POST** the samples directly to storage.
`samples`	List *(Optional)*	A list of samples.

Example 12.11. Add samples to meter: JSON request

```
{
    "id": "8db08c68-bc70-11e4-a8c4-fa163e1d1a9b",
    "metadata": {
        "name1": "value1",
        "name2": "value2"
    },
    "meter": "instance",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "recorded_at": "2015-02-24T22:00:32.747930",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "timestamp": "2015-02-24T22:00:32.747930",
    "type": "gauge",
    "unit": "instance",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
    "volume": 1.0
}
```

Example 12.12. Add samples to meter: XML request

```
<value>
  <id>8db08c68-bc70-11e4-a8c4-fa163e1d1a9b</id>
  <meter>instance</meter>
```

```
  <type>gauge</type>
  <unit>instance</unit>
  <volume>1.0</volume>
  <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
  <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
  <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
  <source>openstack</source>
  <timestamp>2015-02-24T22:00:32.747930</timestamp>
  <recorded_at>2015-02-24T22:00:32.747930</recorded_at>
  <metadata>
    <item>
      <key>name2</key>
      <value>value2</value>
    </item>
    <item>
      <key>name1</key>
      <value>value1</value>
    </item>
  </metadata>
</value>
```

This operation does not accept a request body.

12.2.2.2. Response

Example 12.13. Add samples to meter: JSON response

```
{
    "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
    "metadata": {
        "name1": "value1",
        "name2": "value2"
    },
    "meter": "instance",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "recorded_at": "2015-09-22T14:52:54.850725",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "timestamp": "2015-09-22T14:52:54.850718",
    "type": "gauge",
    "unit": "instance",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
    "volume": 1
}
```

Example 12.14. Add samples to meter: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
   <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
   <meter>instance</meter>
   <type>gauge</type>
   <unit>instance</unit>
   <volume>1.0</volume>
   <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
   <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
   <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
   <source>openstack</source>
   <timestamp>2015-09-22T14:52:54.850718</timestamp>
```

```
   <recorded_at>2015-09-22T14:52:54.850725</recorded_at>
   <metadata>
      <item>
         <key>name2</key>
         <value>value2</value>
      </item>
      <item>
         <key>name1</key>
         <value>value1</value>
      </item>
   </metadata>
</value>
```

This operation does not return a response body.

12.2.3. List samples for meter

Method	URI	Description
GET	`/v2/meters/{meter_name}{?q,limit}`	Lists samples for a meter, by meter name.

Normal response codes: 200

12.2.3.1. Request

This table shows the URI parameters for the list samples for meter request:

Name	Type	Description
`{meter_name}`	String	The name of the meter.

This table shows the query parameters for the list samples for meter request:

Name	Type	Description
`q`	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

12.2.3.2. Response

Example 12.15. List samples for meter: JSON response

```
[
    {
        "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
        "metadata": {
            "name1": "value1",
            "name2": "value2"
        },
        "meter": "instance",
        "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
        "recorded_at": "2015-09-22T14:52:54.850725",
        "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
        "source": "openstack",
        "timestamp": "2015-09-22T14:52:54.850718",
        "type": "gauge",
        "unit": "instance",
        "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
        "volume": 1
    }
]
```

Example 12.16. List samples for meter: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<values>
   <value>
      <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
      <meter>instance</meter>
      <type>gauge</type>
      <unit>instance</unit>
      <volume>1.0</volume>
      <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
      <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
      <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
      <source>openstack</source>
      <timestamp>2015-09-22T14:52:54.850718</timestamp>
      <recorded_at>2015-09-22T14:52:54.850725</recorded_at>
      <metadata>
         <item>
            <key>name2</key>
            <value>value2</value>
         </item>
         <item>
            <key>name1</key>
            <value>value1</value>
         </item>
      </metadata>
   </value>
</values>
```

This operation does not return a response body.

12.2.4. Show meter statistics

Method	URI	Description
GET	`/v2/meters/{meter_name}/statistics{?q,groupby,period,aggregate,limit}`	Computes and lists statistics for samples in a time range.

Normal response codes: 200

12.2.4.1. Request

This table shows the URI parameters for the show meter statistics request:

Name	Type	Description
`{meter_name}`	String	The name of the meter.

This table shows the query parameters for the show meter statistics request:

Name	Type	Description
`q`	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.
`groupby`	Dict *(Optional)*	Fields for group by aggregation.
`period`	Int *(Optional)*	The period, in seconds, for which you want statistics.
`aggregate`	Dict *(Optional)*	A list of selectable aggregation functions to apply. For example: `GET /v2/meters/METER_NAME/statistics?aggregate.func=cardinality&aggregate.param=resource_id&aggregate.func=cardinality&aggregate.param=project_id`
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

12.2.4.2. Response

Example 12.17. Show meter statistics: JSON response

```
[
    {
        "avg": 4.5,
        "count": 10,
        "duration": 300,
        "duration_end": "2013-01-04T16:47:00",
        "duration_start": "2013-01-04T16:42:00",
        "max": 9,
        "min": 1,
        "period": 7200,
```

```
        "period_end": "2013-01-04T18:00:00",
        "period_start": "2013-01-04T16:00:00",
        "sum": 45,
        "unit": "GiB"
    }
]
```

Example 12.18. Show meter statistics: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
  <value>
    <avg>4.5</avg>
    <count>10</count>
    <duration>300.0</duration>
    <duration_end>2013-01-04T16:47:00</duration_end>
    <duration_start>2013-01-04T16:42:00</duration_start>
    <max>9.0</max>
    <min>1.0</min>
    <period>7200</period>
    <period_end>2013-01-04T18:00:00</period_end>
    <period_start>2013-01-04T16:00:00</period_start>
    <sum>45.0</sum>
    <unit>GiB</unit>
  </value>
</values>
```

This operation does not return a response body.

12.3. Samples

Lists all samples and gets information for a sample.

For list operations, if you do not explicitly set the `limit` query parameter, a default limit is applied. The default limit is the `default_api_return_limit` configuration option value.

Method	URI	Description
GET	`/v2/samples{?q,limit}`	Lists all known samples, based on the data recorded so far.
GET	`/v2/samples/{sample_id}`	Shows details for a sample, by sample ID.

12.3.1. List samples

Method	URI	Description
GET	`/v2/samples{?q,limit}`	Lists all known samples, based on the data recorded so far.

Normal response codes: 200

12.3.1.1. Request

This table shows the query parameters for the list samples request:

Name	Type	Description
`q`	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.
`limit`	Int *(Optional)*	Requests a page size of items. Returns a number of items up to a limit value. Use the `limit` parameter to make an initial limited request and use the ID of the last-seen item from the response as the `marker` parameter value in a subsequent limited request.

This operation does not accept a request body.

12.3.1.2. Response

Example 12.19. List samples: JSON response

```
[
    {
        "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
        "metadata": {
            "name1": "value1",
            "name2": "value2"
        },
        "meter": "instance",
        "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
        "recorded_at": "2015-09-22T14:52:54.850725",
        "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
        "source": "openstack",
        "timestamp": "2015-09-22T14:52:54.850718",
        "type": "gauge",
        "unit": "instance",
        "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
        "volume": 1
    }
]
```

Example 12.20. List samples: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
   <value>
      <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
      <meter>instance</meter>
      <type>gauge</type>
      <unit>instance</unit>
```

```
      <volume>1.0</volume>
      <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
      <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
      <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
      <source>openstack</source>
      <timestamp>2015-09-22T14:52:54.850718</timestamp>
      <recorded_at>2015-09-22T14:52:54.850725</recorded_at>
      <metadata>
         <item>
            <key>name2</key>
            <value>value2</value>
         </item>
         <item>
            <key>name1</key>
            <value>value1</value>
         </item>
      </metadata>
   </value>
</values>
```

This operation does not return a response body.

12.3.2. Show sample details

Method	URI	Description
GET	`/v2/samples/{sample_id}`	Shows details for a sample, by sample ID.

Normal response codes: 200

12.3.2.1. Request

This table shows the URI parameters for the show sample details request:

Name	Type	Description
`{sample_id}`	UUID	The UUID of the sample.

This operation does not accept a request body.

12.3.2.2. Response

Example 12.21. Show sample details: JSON response

```
{
    "id": "9b23b398-6139-11e5-97e9-bc764e045bf6",
    "metadata": {
        "name1": "value1",
        "name2": "value2"
    },
    "meter": "instance",
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "recorded_at": "2015-09-22T14:52:54.850725",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "timestamp": "2015-09-22T14:52:54.850718",
    "type": "gauge",
    "unit": "instance",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff",
    "volume": 1
}
```

Example 12.22. Show sample details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
   <id>9b23b398-6139-11e5-97e9-bc764e045bf6</id>
   <meter>instance</meter>
   <type>gauge</type>
   <unit>instance</unit>
   <volume>1.0</volume>
   <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
   <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
   <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
   <source>openstack</source>
   <timestamp>2015-09-22T14:52:54.850718</timestamp>
   <recorded_at>2015-09-22T14:52:54.850725</recorded_at>
   <metadata>
      <item>
```

```
         <key>name2</key>
         <value>value2</value>
      </item>
      <item>
         <key>name1</key>
         <value>value1</value>
      </item>
   </metadata>
</value>
```

This operation does not return a response body.

12.4. Resources

Lists all and gets information for resources.

Method	URI	Description
GET	`/v2/resources{?q,meter_links}`	Lists definitions for all resources.
GET	`/v2/resources/{resource_id}`	Shows details for a resource, by resource ID.

12.4.1. List resources

Method	URI	Description
GET	`/v2/resources{?q,meter_links}`	Lists definitions for all resources.

Normal response codes: 200

12.4.1.1. Request

This table shows the query parameters for the list resources request:

Name	Type	Description
`q`	List *(Optional)*	Filters the response by one or more arguments. For example: `?q.field=Foo&q.value=my_text`.
`meter_links`	Int *(Optional)*	Set `?meter_links=1` to return a self link and related meter links.

This operation does not accept a request body.

12.4.1.2. Response

Example 12.23. List resources: JSON response

```
[
    {
        "links": [
            {
                "href": "http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
                "rel": "self"
            },
            {
                "href": "http://localhost:8777/v2/meters/volume?q.field=
resource_id&q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
                "rel": "volume"
            }
        ],
        "metadata": {
            "name1": "value1",
            "name2": "value2"
        },
        "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
        "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
        "source": "openstack",
        "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
    }
]
```

This table shows the body parameters for the list resources response:

Name	Type	Description
`links`	List	A list that contains a self link and associated meter links.

Name	Type	Description
	(Required)	
`metadata`	Dict *(Required)*	A set of one or more arbitrary metadata key and value pairs that are associated with the resource.
`project_id`	UUID *(Required)*	The UUID of the owning project or tenant.
`resource_id`	UUID *(Required)*	The UUID of the resource.
`source`	String *(Required)*	The name of the source from which the resource came.
`user_id`	UUID *(Required)*	The UUID of the user who either created or last updated the resource.

Example 12.24. List resources: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<values>
   <value>
      <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
      <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
      <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
      <metadata>
         <item>
            <key>name2</key>
            <value>value2</value>
         </item>
         <item>
            <key>name1</key>
            <value>value1</value>
         </item>
      </metadata>
      <links>
         <item>
            <href>http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
            <rel>self</rel>
         </item>
         <item>
            <href>http://localhost:8777/v2/meters/volume?q.field=resource_id&
amp;q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
            <rel>volume</rel>
         </item>
      </links>
      <source>openstack</source>
   </value>
</values>
```

This operation does not return a response body.

12.4.2. Show resource details

Method	URI	Description
GET	`/v2/resources/{resource_id}`	Shows details for a resource, by resource ID.

Normal response codes: 200

12.4.2.1. Request

This table shows the URI parameters for the show resource details request:

Name	Type	Description
`{resource_id}`	UUID	The UUID of the resource.

This operation does not accept a request body.

12.4.2.2. Response

Example 12.25. Show resource details: JSON response

```
{
    "links": [
        {
            "href": "http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
            "rel": "self"
        },
        {
            "href": "http://localhost:8777/v2/meters/volume?q.field=
resource_id&q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
            "rel": "volume"
        }
    ],
    "metadata": {
        "name1": "value1",
        "name2": "value2"
    },
    "project_id": "35b17138-b364-4e6a-a131-8f3099c5be68",
    "resource_id": "bd9431c1-8d69-4ad3-803a-8d4a6b89fd36",
    "source": "openstack",
    "user_id": "efd87807-12d2-4b38-9c70-5f5c2ac427ff"
}
```

Example 12.26. Show resource details: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
   <resource_id>bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</resource_id>
   <project_id>35b17138-b364-4e6a-a131-8f3099c5be68</project_id>
   <user_id>efd87807-12d2-4b38-9c70-5f5c2ac427ff</user_id>
   <metadata>
      <item>
         <key>name2</key>
         <value>value2</value>
      </item>
```

```
      <item>
         <key>name1</key>
         <value>value1</value>
      </item>
   </metadata>
   <links>
      <item>
         <href>http://localhost:8777/v2/resources/
bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
         <rel>self</rel>
      </item>
      <item>
         <href>http://localhost:8777/v2/meters/volume?q.field=resource_id&
amp;q.value=bd9431c1-8d69-4ad3-803a-8d4a6b89fd36</href>
         <rel>volume</rel>
      </item>
   </links>
   <source>openstack</source>
</value>
```

This operation does not return a response body.

12.5. Capabilities

Gets information for API and storage capabilities.

The Telemetry service enables you to store samples, events, and alarm definitions in supported database back ends. The `capabilities` resource enables you to list the capabilities that a database supports.

The `capabilities` resource returns a flattened dictionary of capability properties, each with an associated boolean value. A value of `true` indicates that the corresponding capability is available in the back end.

You can optionally configure separate database back ends for samples, events, and alarms definitions. The `capabilities` response shows a value of `true` to indicate that the definitions database for samples, events, or alarms is ready to use in a production environment.

Method	URI	Description
GET	`/v2/capabilities`	A representation of the API and storage capabilities. Usually, the storage driver imposes constraints.

12.5.1. List capabilities

Method	URI	Description
GET	`/v2/capabilities`	A representation of the API and storage capabilities. Usually, the storage driver imposes constraints.

Normal response codes: 200

12.5.1.1. Response

Example 12.27. List capabilities: JSON response

```
{
    "alarm_storage": {
        "storage:production_ready": true
    },
    "api": {
        "alarms:history:query:complex": true,
        "alarms:history:query:simple": true,
        "alarms:query:complex": true,
        "alarms:query:simple": true,
        "events:query:simple": true,
        "meters:query:complex": false,
        "meters:query:metadata": true,
        "meters:query:simple": true,
        "resources:query:complex": false,
        "resources:query:metadata": true,
        "resources:query:simple": true,
        "samples:query:complex": true,
        "samples:query:metadata": true,
        "samples:query:simple": true,
        "statistics:aggregation:selectable:avg": true,
        "statistics:aggregation:selectable:cardinality": true,
        "statistics:aggregation:selectable:count": true,
        "statistics:aggregation:selectable:max": true,
        "statistics:aggregation:selectable:min": true,
        "statistics:aggregation:selectable:quartile": false,
        "statistics:aggregation:selectable:stddev": true,
        "statistics:aggregation:selectable:sum": true,
        "statistics:aggregation:standard": true,
        "statistics:groupby": true,
        "statistics:query:complex": false,
        "statistics:query:metadata": true,
        "statistics:query:simple": true
    },
    "event_storage": {
        "storage:production_ready": true
    },
    "storage": {
        "storage:production_ready": true
    }
}
```

This table shows the body parameters for the list capabilities response:

Name	Type	Description
`api`	Dict	A set of key and value pairs that contain the API capabilities for the configured storage driver.

Name	Type	Description
	(Required)	
`alarms:history:query:complex`	Boolean *(Required)*	If `true`, the complex query capability for alarm history is available for the configured database back end.
`alarms:history:query:simple`	Boolean *(Required)*	If `true`, the simple query capability for alarm history is available for the configured database back end.
`alarms:query:complex`	Boolean *(Required)*	If `true`, the complex query capability for alarm definitions is available for the configured database back end.
`alarms:query:simple`	Boolean *(Required)*	If `true`, the simple query capability for alarm definitions is available for the configured database back end.
`events:query:simple`	Boolean *(Required)*	If `true`, the simple query capability for events is available for the configured database back end.
`meters:query:complex`	Boolean *(Required)*	If `true`, the complex query capability for meters is available for the configured database back end.
`meters:query:metadata`	Boolean *(Required)*	If `true`, the simple query capability for the metadata of meters is available for the configured database back end.
`meters:query:simple`	Boolean *(Required)*	If `true`, the simple query capability for meters is available for the configured database back end.
`resources:query:complex`	Boolean *(Required)*	If `true`, the complex query capability for resources is available for the configured database back end.
`resources:query:metadata`	Boolean *(Required)*	If `true`, the simple query capability for the metadata of resources is available for the configured database back end.
`resources:query:simple`	Boolean *(Required)*	If `true`, the simple query capability for resources is available for the configured database back end.
`samples:query:complex`	Boolean *(Required)*	If `true`, the complex query capability for samples is available for the configured database back end.
`samples:query:metadata`	Boolean *(Required)*	If `true`, the simple query capability for the metadata of samples is available for the configured database back end.
`samples:query:simple`	Boolean *(Required)*	If `true`, the simple query capability for samples is available for the configured database back end.
`statistics:aggregation:selectable:avg`	Boolean *(Required)*	If `true`, the `avg` capability is available for the configured database back end. Use the `avg` capability to get average values for samples.
`statistics:aggregation:selectable:cardinality`	Boolean *(Required)*	If `true`, the `cardinality` capability is available for the configured database back end. Use the `cardinality` capability to get cardinality for samples.
`statistics:aggregation:selectable:count`	Boolean *(Required)*	If `true`, the `count` capability is available for the configured database back end. Use the `count` capability to calculate the number of samples for a query.
`statistics:aggregation:selectable:max`	Boolean *(Required)*	If `true`, the `max` capability is available for the configured database back end. . Use the `max` capability to calculate the maximum value for a query.
`statistics:aggregation:selectable:min`	Boolean *(Required)*	If `true`, the `min` capability is available for the configured database back end. Use the `min` capability to calculate the minimum value for a query.

Name	Type	Description
statistics:aggregation:selectable:quartile	Boolean (Required)	If true, the quartile capability is available for the configured database back end. Use the quartile capability to calculate the quartile of sample volumes for a query.
statistics:aggregation:selectable:stddev	Boolean (Required)	If true, the stddev capability is available for the configured database back end. Use the stddev capability to calculate the standard deviation of sample volumes for a query.
statistics:aggregation:selectable:sum	Boolean (Required)	If true, the sum capability is available for the configured database back end. Use the sum capability to calculate the sum of sample volumes for a query.
statistics:aggregation:standard	Boolean (Required)	If true, the standard set of aggregation capability is available for the configured database back end.
statistics:groupby	Boolean (Required)	If true, the groupby capability is available for calculating statistics for the configured database back end.
statistics:query:complex	Boolean (Required)	If true, the complex query capability for statistics is available for the configured database back end.
statistics:query:metadata	Boolean (Required)	If true, the simple query capability for the sample metadata that is used to calculate statistics is available for the configured database back end.
statistics:query:simple	Boolean (Required)	If true, the simple query capability for statistics is available for the configured database back end.
alarm_storage	Dict (Required)	Defines the capabilities for the storage that stores persisting alarm definitions. A value of true indicates that the capability is available.
event_storage	Dict (Required)	If true, the capabilities for the storage that stores persisting events is available.
storage	Dict (Required)	If true, the capabilities for the storage that stores persisting samples is available.
storage:production_ready	Boolean (Required)	If true, the database back end is ready to use in a production environment.

Example 12.28. List capabilities: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<value>
   <api>
      <item>
         <key>statistics:query:complex</key>
         <value>false</value>
      </item>
      <item>
         <key>alarms:history:query:simple</key>
         <value>true</value>
      </item>
      <item>
         <key>meters:query:metadata</key>
         <value>true</value>
      </item>
      <item>
         <key>alarms:query:simple</key>
         <value>true</value>
      </item>
      <item>
```

```
        <key>resources:query:simple</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:quartile</key>
        <value>false</value>
    </item>
    <item>
        <key>statistics:query:simple</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:count</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:min</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:aggregation:selectable:sum</key>
        <value>true</value>
    </item>
    <item>
        <key>alarms:query:complex</key>
        <value>true</value>
    </item>
    <item>
        <key>meters:query:complex</key>
        <value>false</value>
    </item>
    <item>
        <key>statistics:groupby</key>
        <value>true</value>
    </item>
    <item>
        <key>alarms:history:query:complex</key>
        <value>true</value>
    </item>
    <item>
        <key>meters:query:simple</key>
        <value>true</value>
    </item>
    <item>
        <key>samples:query:metadata</key>
        <value>true</value>
    </item>
    <item>
        <key>statistics:query:metadata</key>
        <value>true</value>
    </item>
    <item>
        <key>samples:query:simple</key>
        <value>true</value>
    </item>
    <item>
        <key>resources:query:metadata</key>
        <value>true</value>
    </item>
    <item>
```

```
            <key>statistics:aggregation:selectable:max</key>
            <value>true</value>
         </item>
         <item>
            <key>samples:query:complex</key>
            <value>true</value>
         </item>
         <item>
            <key>statistics:aggregation:standard</key>
            <value>true</value>
         </item>
         <item>
            <key>events:query:simple</key>
            <value>true</value>
         </item>
         <item>
            <key>statistics:aggregation:selectable:stddev</key>
            <value>true</value>
         </item>
         <item>
            <key>statistics:aggregation:selectable:avg</key>
            <value>true</value>
         </item>
         <item>
            <key>statistics:aggregation:selectable:cardinality</key>
            <value>true</value>
         </item>
         <item>
            <key>resources:query:complex</key>
            <value>false</value>
         </item>
      </api>
      <storage>
         <item>
            <key>storage:production_ready</key>
            <value>true</value>
         </item>
      </storage>
      <alarm_storage>
         <item>
            <key>storage:production_ready</key>
            <value>true</value>
         </item>
      </alarm_storage>
      <event_storage>
         <item>
            <key>storage:production_ready</key>
            <value>true</value>
         </item>
      </event_storage>
   </value>
```

This operation does not return a response body.

12.6. Events

Lists all events and shows details for an event.

Method	URI	Description
GET	`/v2/events{?q,limit}`	Lists all events.
GET	`/v2/events/{message_id}`	Shows details for an event.

12.6.1. List events

Method	URI	Description
GET	`/v2/events{?q,limit}`	Lists all events.

Normal response codes: 200

12.6.1.1. Request

This table shows the query parameters for the list events request:

Name	Type	Description
`q`	List *(Optional)*	Filters the response by one or more event arguments. For example: `GET /v2/events?q.field=Foo&q.value=my_text`
`limit`	Int *(Optional)*	Limits the maximum number of samples that the response returns. For example: `GET /v2/events?limit=1000`

This operation does not accept a request body.

12.6.1.2. Response

Example 12.29. List events: JSON response

```
[
    {
        "raw": {},
        "traits": [
            {
                "type": "string",
                "name": "action",
                "value": "read"
            },
            {
                "type": "string",
                "name": "eventTime",
                "value": "2015-10-28T20:26:58.545477+0000"
            }
        ],
        "generated": "2015-10-28T20:26:58.546933",
        "message_id": "bae43de6-e9fa-44ad-8c15-40a852584444",
        "event_type": "http.request"
    }
]
```

12.6.2. Show event details

Method	URI	Description
GET	`/v2/events/{message_id}`	Shows details for an event.

Normal response codes: 200

12.6.2.1. Request

This table shows the URI parameters for the show event details request:

Name	Type	Description
`{message_id}`	UUID	The UUID of the message.

12.6.2.2. Response

Example 12.30. Show event details: JSON response

```
{
    "raw": {},
    "traits": [
        {
            "type": "string",
            "name": "action",
            "value": "read"
        },
        {
            "type": "string",
            "name": "eventTime",
            "value": "2015-10-28T20:26:58.545477+0000"
        }
    ],
    "generated": "2015-10-28T20:26:58.546933",
    "message_id": "bae43de6-e9fa-44ad-8c15-40a852584444",
    "event_type": "http.request"
}
```