

# Atlas OVFT API

## Cloud Execution Environment

### INTERWORK DESCRIPTION

**Copyright**

© Ericsson AB 2016–2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

**Disclaimer**

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

**Trademark List**

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Summary of Supported Operations</b>	<b>2</b>
<b>3</b>	<b>API Operations</b>	<b>4</b>
3.1	Create cApp	4
3.2	List cApp	8
3.3	Show cApp	9
3.4	Update cApp	10
3.5	Delete cApp	11
3.6	Show cApp Template	12
3.7	List cApp File	16
3.8	Show cApp File	17
3.9	Update cApp File	17
3.10	Cleanup cApps	19
3.11	Personalize cApp Template	20
3.12	Export Application	24
3.13	Scale out Stack	25
3.14	Scale in Stack	26
3.15	Stack Scaling List	26





# 1 Introduction

This document describes the Atlas Open Virtualization Format Translator (OVFT) Application Programming Interface (API) used in the Cloud Execution Environment (CEE).

OVFT is used to generate Heat Orchestration Templates (HOTs) from Open Virtual Appliance/Alliance (OVA) and Topology and Orchestration Specification for Cloud Applications (TOSCA) files. Then it stores the converted cloud application (cApp) in an application catalog. The application template serves for launching Virtual Machines (VMs) using the Heat Orchestration API. Additionally, OVFT assigns metadata to applications.

The Atlas OVFT API can perform the following functions:

- Create a new application
- List accessible applications
- Describe a specific application
- Update a specific application
- Delete specified applications
- Describe a specific application template
- Describe a specific application env file
- Update a specific application env file
- Delete applications in error state
- Personalizes the application template
- Export an application
- Scale out an accessible stack
- Scale in an accessible stack
- List the accessible scaling groups and servers of a stack

The response information is JSON formatted.



## 2 Summary of Supported Operations

A summary of the supported operations is shown in Table 1.

Table 1 Summary of Supported Operations

Operation Name	Description	HTTP Operation	Uniform Resource Identifier (URI)
capp-create	Create a new application.	POST	/capps
capp-list	List accessible applications.	GET	/capps
capp-show	Describe a specific application.	GET	/capps/<capp_id>
capp-update	Update a specific application.	PUT	/capps/<capp_id>
capp-delete	Delete specified applications.	DELETE	/capps/<capp_id>
capp-template-show	Describe a specific application template.	GET	/capps/template/<capp_id>
capp-file-list	Describe a specific application env file.	GET	/capps/files/<capp_id>
capp-cleanup	Delete applications in error state.	DELETE	/capps
capp-template-personalize	Personalizes the application template.	POST	/capps/personalize/<id>
capp-file-show	Describe a specific file of an application.	GET	/capps/file/detail/<file_id>
capp-file-update	Update a specific file of an application.	POST	/capps/file/<file_id>
capp-export	Export an accessible application.	GET	/capps/source/<capp_id>
stack-scale --method out	Scale out an accessible stack.	POST	/scale_out/stack



Operation Name	Description	HTTP Operation	Uniform Resource Identifier (URI)
stack-scale --method in	Scale in an accessible stack.	POST	/scale_in/stack
stack-scaling-list	List the accessible scaling groups and servers of a stack.	GET	/stack/scaling_list/stack



## 3 API Operations

This section describes the operations supported by the API.

**Note:** Throughout this document, port 8888 is used by default for all operations, and the term “cApp” is used for cloud application.

### 3.1 Create cApp

This section describes how to create a new application.

#### 3.1.1 Create cApp with Type OVF Headers

This section describes how to create a new application with OVF headers.

```
x-capp-meta-size: 25180160
Content-Length: 25180160
x-capp-meta-name: demo
User-Agent: python-ovftclient
X-Auth-Token: 99fd7426dc6646869e57d995f8f795c1
Content-Type: application/octet-stream
x-capp-meta-type: ovf
```

#### Request

The metadata about the OVFT cApp is sent to the HTTP headers, with headers content-type:

```
application/octet-stream content-length: size of the body.
```

The data is sent as a JSON-encoded mapping in the following format:

```
{
  'ovft_capps' : [
    {'name': 'Example',
     'is_public': true,
     'is_protected': false,
     'description': 'test'}]
}
```

The body of the HTTP request to the API is a MIME-encoded disk application data.

#### Response

The data is returned as a JSON-encoded mapping in the following format:





```
{
  'ovft_capps' : [
    {'id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'name': 'Example',
     'type': 'ovf',
     'owner': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'created_at': '2010-02-03 09:34:01',
     'updated_at': '2010-02-03 09:34:01',
     'is_public': true,
     'is_protected': false,
     'status': 'active',
     'description': 'test'}]
}
```

### Example of capp-create

```
curl -i -X POST -H 'x-capp-meta-size: 26357760' -H
'Content-Length: 26357760' -H 'x-capp-meta-name: test' -H
'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt
-H 'X-Auth-Token: 566bb071092b4d6b9824dd41bfaed111' -H
'Content-Type: application/octet-stream' -H 'x-capp-meta-type:
ovf' --data-binary '@external_stack_package.ova' https://public.at
las.local:8888/v1/bc2f8c86fbe7434eb58523230af21575/capps
```

```
HTTP/1.1 201 Created
date: Thu, 10 Sep 2015 10:16:49 GMT
content-length: 327
content-type: application/json
```

```
{"ovft_capps": {"status": "Creating", "user_id": "927
c06847e9346078bad06fb0353cc67", "description": "",
"fault": "", "created_at": "2015-09-10T10:15:10.000000",
"updated_at": "2015-09-10T10:16:49.000000", "image_ids":
"", "is_protected": false, "is_public": false, "id":
"39c944ee-70bc-4780-8a58-0f0fefb9ea9f", "name": "test", "type":
"ovf"}}
```

## 3.1.2 Create cApp with Type HOT Headers

This section describes how to create a new application with HOT headers.

```
x-capp-meta-size: 25180160
Content-Length: 25180160
x-capp-meta-name: demo
User-Agent: python-ovftclient
X-Auth-Token: 99fd7426dc6646869e57d995f8f795c1
Content-Type: application/octet-stream
x-capp-meta-type: hot
```



## Request

The metadata about the OVFT application is sent to the HTTP headers, with headers content-type:

application/octet-stream content-length: size of the body.

The data is sent as a JSON-encoded mapping in the following format:

```
{
  'ovft_capps' : [
    {'name': 'Example',
     'is_public': true,
     'is_protected': false,
     'description': 'test'}]
}
```

The body of the HTTP request to the API is a MIME-encoded disk application data.

## Response

The data is returned as a JSON-encoded mapping in the following format:

```
{
  'ovft_capps' : [
    {'id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'name': 'Example',
     'type': 'hot',
     'owner': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'created_at': '2010-02-03 09:34:01',
     'updated_at': '2010-02-03 09:34:01',
     'is_public': true,
     'is_protected': false,
     'status': 'active',
     'description': 'test'}]
}
```

## Example of capp-create

```
curl -i -X POST -H 'x-capp-meta-size: 1397' -H 'Content-Length: 1397' -H 'x-capp-meta-name: example' -H 'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt -H 'X-Auth-Token: d354a70bd45e43ff9d5043ec525bd676' -H 'Content-Type: application/octet-stream' -H 'x-capp-meta-type: hot' --data-binary '@image_with_attach_vol.yaml' https://public.atlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d66/capps
```

HTTP/1.1 201 Created



```
date: Mon, 14 Sep 2015 12:14:50 GMT
content-length: 330
content-type: application/json
```

```
{"ovft_capps": {"status": "Creating", "user_id": "35d87e7d
cba2421baccee70a693e1ef0", "description": "", "fault": "",
"created_at": "2015-09-14T12:14:50.000000", "updated_at":
"2015-09-14T12:14:50.000000", "image_ids": "", "is_protected":
false, "is_public": false, "id": "10c8da9e-e304-41a9-81b9-63b646
2b5e22", "name": "example", "type": "hot"}}
```

### 3.1.3 Create cApp with Type TOSCA Headers

This section describes how to create a new application with TOSCA headers.

```
x-capp-meta-size: 25180160
Content-Length: 25180160
x-capp-meta-name: demo
User-Agent: python-ovftclient
X-Auth-Token: 99fd7426dc6646869e57d995f8f795c1
Content-Type: application/octet-stream
x-capp-meta-type: tosca
```

#### Request

The metadata about the OVFT application is sent to the HTTP headers, with headers content-type:

```
application/octet-stream content-length: size of the body.
```

The data is sent as a JSON-encoded mapping in the following format:

```
{
  'ovft_capps' : [
    {'name': 'Example',
     'is_public': true,
     'is_protected': false,
     'description': 'test'}]
}
```

The body of the HTTP request to the API is a MIME-encoded disk application data.

#### Response

The data is returned as a JSON-encoded mapping in the following format:



```
{
  'ovft_capps' : [
    {'id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'name': 'Example',
     'type': 'tosca',
     'owner': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'created_at': '2010-02-03 09:34:01',
     'updated_at': '2010-02-03 09:34:01',
     'is_public': true,
     'is_protected': false,
     'status': 'active',
     'description': 'test'}]
}
```

### Example of capp-create

```
curl -i -X POST -H 'x-capp-meta-size: 1397' -H 'Content-Length:
1397' -H 'x-capp-meta-name: example' -H 'User-Agent:
python-ovftclient' --cacert /etc/ssl/certs/ca.crt -H
'X-Auth-Token: fdf7d8f5ad844905b31bf3ad26ac543e' -H
'Content-Type: application/octet-stream' -H 'x-capp-meta-type:
tosca' --data-binary '@image_with_attach_vol.yaml' https://public.
atlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d66/capps
```

```
HTTP/1.1 201 Created
date: Mon, 14 Sep 2015 12:20:46 GMT
content-length: 330
content-type: application/json
```

```
{"ovft_capps": {"status": "Creating", "user_id": "35d87e7d
cba2421baccee70a693e1ef0", "description": "", "fault": "",
"created_at": "2015-09-14T12:20:45.000000", "updated_at":
"2015-09-14T12:20:45.000000", "image_ids": "", "is_protected":
false, "is_public": false, "id": "aac19301-135d-4798-aeb6-725ba3
364e4a", "name": "example", "type": "tosca"}}
```

## 3.2 List cApp

This section describes how to list accessible applications.

### Headers

```
Auth-Token: bb526f4a55d0404994dfec5c8f889679
Content-Type: application/json
```



## Request

No Body

## Response

The data is returned as a JSON-encoded mapping in the following format:

```
{
  'ovft_capp': [
    {id: '71c675ab-d94f-49cd-a114-e12490b328d9',
      'name': 'Example',
      'type': 'ovf',
      'owner': '71c675ab-d94f-49cd-a114-e12490b328d9',
      'created_at': '2010-02-03 09:34:01',
      'updated_at': '2010-02-03 09:34:01',
      'is_public': true,
      'is_protected': false,
      'status': 'active',
      'description': 'test'}]
}
```

## Example of capp-list

```
curl -i -X GET -H 'X-Auth-Token: 1ec9cdb0efc148f58bbad8b56c
69ac30' -H 'Content-Type: application/json' -H 'User-Agent:
python-ovftclient' --cacert /etc/ssl/certs/ca.crt https://public.a
tlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d66/capps
```

```
HTTP/1.1 200 OK
date: Tue, 15 Sep 2015 07:18:49 GMT
content-length: 155
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": [{"status": "active", "created_at": "2015-09-14T
06:36:25.000000", "id": "d43af622-f45c-46ca-8e21-4dfc221b60d8",
"name": "volume_attached", "type": "ovf"}]}
```

## 3.3

## Show cApp

This section describes how to get information about a specific application.

## Headers

```
Auth-Token: 243bd64f895a45459ec4fcaa0099b574
Content-Type: application/json
```



## Request

Encode the ID of the application into the request URI to get the details of a particular application.

## Response

The data is returned as a JSON-encoded mapping in the following format:

```
{
  'ovft_catalogs' : [
    {'catalog_id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'name': 'Example',
     'type': 'ovf',
     'owner': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'created_at': '2010-02-03 09:34:01',
     'updated_at': '2010-02-03 09:34:01',
     'is_public': true,
     'is_protected': false,
     'status': 'active',
     'images': ['71c675ab-d94f-49cd-a114-e12490b328d9'],
     'description': 'test'}]
}
```

## Example of capp-show

```
curl -i -X GET -H 'X-Auth-Token: 2099fa4b67e64529a088
424c0e3da706' -H 'Content-Type: application/json' -H
'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt
https://public.atlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d6
6/capps/volume_attached
```

```
HTTP/1.1 200 OK
date: Tue, 15 Sep 2015 07:22:42 GMT
content-length: 372
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": {"status": "active", "user_id": "35d87e7dc
ba2421baccee70a693e1ef0", "description": "", "fault": "",
"created_at": "2015-09-14T06:36:25.000000", "updated_at":
"2015-09-14T06:36:31.000000", "image_ids": "d3993e92-582c-4
c4a-a44d-e05bb76098a2", "is_protected": false, "is_public":
false, "id": "d43af622-f45c-46ca-8e21-4dfc221b60d8", "name":
"volume_attached", "type": "ovf"}}
```

## 3.4 Update cApp

This section describes how to update a specific application.



## Headers

Auth-Token: c7df2fbb82174a19aca0765404e99a73  
Content-Type: application/json

## Request

The data is sent as a JSON-encoded mapping in the following format:

```
{
  'ovft_catalogs' : [
    {'name': 'Example',
     'is_public': true,
     'is_protected': false,
     'description': 'test'}]
}
```

## Response

On success, the PUT request will return the application metadata encoded as HTTP headers. Only the new values are updated, keeping the previous values as they were.

## Example of capp-update

```
curl -i -X PUT -H 'X-Auth-Token: 3b519070002747e1b98e70fcf4
af0278' -H 'Content-Type: application/json' -H 'User-Agent:
python-ovftclient' --cacert /etc/ssl/certs/ca.crt -d '{"name":
"volume_attach"}' https://public.atlas.local:8888/v1/f99b4db83ab44
19caafe0a5f268c2d66/capps/volume_attached
```

```
HTTP/1.1 200 OK
date: Tue, 15 Sep 2015 07:26:10 GMT
content-length: 370
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": {"status": "active", "user_id": "35d87e7dc
ba2421baccee70a693e1ef0", "description": "", "fault": "",
"created_at": "2015-09-14T06:36:25.000000", "updated_at":
"2015-09-15T07:26:10.000000", "image_ids": "d3993e92-582c-4
c4a-a44d-e05bb76098a2", "is_protected": false, "is_public":
false, "id": "d43af622-f45c-46ca-8e21-4dfc221b60d8", "name":
"volume_attach", "type": "ovf"}}
```

## 3.5 Delete cApp

This section describes how to delete specified applications.

**Headers**

Auth-Token: a98f57e1095145c5978cca6b574fa975  
Content-type: application/json

**Request**

Encode the ID of the application into the request URI. Request body is ignored.

**Response**

Response is empty.

**Example of capp-delete**

```
curl -i -X DELETE -H 'X-Auth-Token: a98f57e1095145c5978cca6b574fa975' -H 'Content-Type: application/octet-stream' -H 'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt https://public.atlas.local:8888/v1/d9def1247d114b0b8a6b5b598af1322c/capps/18731bd4-df1e-425c-aa5e-86947299bc4e
```

```
HTTP/1.1 200 OK
date: Fri, 21 Nov 2014 05:19:42 GMT
content-length: 16
content-type: application/json; charset=UTF-8

{"delete": true}
```

## 3.6 Show cApp Template

This section describes how to view a specific application template.

**Headers**

Auth-Token: bad83b9c147b4f82907b6dce0da8f  
Content-Type: application/json

**Request**

Encode the ID of the application into the request URI to get the HOT template.

**Response**

The data is returned as a JSON-encoded mapping in the following format:





```
{
  ovft_capp': [
    {id': '71c675ab-d94f-49cd-a114-e12490b328d9',
      'name': 'Example',
      'template': 'template data',
      'files': 'files data'}}
  ]
}
```

### Example of capp-template-show

```
curl -i -X GET -H 'X-Auth-Token: bad83b9c147b4f82907b
6dce0da8fdc7' -H 'Content-Type: application/json' -H
'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt
https://public.atlas.local:8888/v1/d9def1247d114b0b8a6b5b598af1322
c/capps/template/sample
```

```
HTTP/1.1 200 OK
date: Fri, 21 Nov 2014 04:58:50 GMT
content-length: 9211
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": {"files": [], "id": "9235621c-da24-4f4b-8afc-aad
7d9b3ae8e", "template": "description: stack template generated
from OVF file\nheat_template_version: '2013-05-23'\nparameters:\n
Layer3_Network1_bgw_port_binding_host:\n constraints:\n
- allowed_values:\n - p1-sr0-sl1\n - p1-sr0-sl3\n -
p1-sr0-sl5\n - p1-sr0-sl7\n - p1-sr0-sl9\n - BGW-1\n - BGW-2\n
description: Binding host id to create BGW port\n label:
Layer3_Network1_bgw_port Binding Host ID\n type: string\n
Layer3_Network2_bgw_port_binding_host:\n constraints:\n
- allowed_values:\n - p1-sr0-sl1\n - p1-sr0-sl3\n -
p1-sr0-sl5\n - p1-sr0-sl7\n - p1-sr0-sl9\n - BGW-1\n - BGW-2\n
description: Binding host id to create BGW port\n label:
Layer3_Network2_bgw_port Binding Host ID\n type: string\n
param_1:\n description: Device owner attachedto the port, use
baremetal:bgw for port used\n for external connectivity(via BGW)\n
label: Layer3_Network1_bgw_port Device Owner\n type: string\n
param_2:\n description: Device owner attachedto the port, use
baremetal:bgw for port used\n for external connectivity(via
BGW)\n label: Layer3_Network2_bgw_port Device Owner\n type:
string\nresources:\n Internal_Network_1:\n properties:\n
name: Internal_Network_1\n type: OS::Neutron::Net\n
Internal_Network_1_subnet:\n properties:\n cidr: 192.16.0.0/24\n
enable_dhcp: true\n name: Internal_Network_1_subnet\n
network_id:\n get_resource: Internal_Network_1\n type:
OS::Neutron::Subnet\n Internal_Network_2:\n properties:\n
name: Internal_Network_2\n type: OS::Neutron::Net\n
Internal_Network_2_subnet:\n properties:\n cidr: 172.16.0.0/24\n
enable_dhcp: true\n name: Internal_Network_2_subnet\n
network_id:\n get_resource: Internal_Network_2\n type:
OS::Neutron::Subnet\n Layer3_Network1:\n properties:\n name:
```



```
Layer3_Network1\n value_specs:\n provider:network_type: vlan\n provider:physical_network: default\n provider:segmentation_id:\n '3982'\n router:external: true\n type: OS::Neutron::Net\n Layer3_Network1_bgw_port:\n properties:\n binding:\n host_id:\n get_param: Layer3_Network1_bgw_port_binding_host\n device_owner:\n get_param: param_1\n network_id:\n get_resource: Layer3_Network1\n type: Ericsson::Neutron::Port\n Layer3_Network1_subnet:\n properties:\n cidr: 12.41.0.16/29\n enable_dhcp: true\n name: Layer3_Network1_subnet\n network_id:\n get_resource: Layer3_Network1\n type:\n OS::Neutron::Subnet\n Layer3_Network2:\n properties:\n name:\n Layer3_Network2\n value_specs:\n provider:network_type: vlan\n provider:physical_network: default\n provider:segmentation_id:\n '3983'\n router:external: true\n type: OS::Neutron::Net\n Layer3_Network2_bgw_port:\n properties:\n binding:\n host_id:\n get_param: Layer3_Network2_bgw_port_binding_host\n device_owner:\n get_param: param_2\n network_id:\n get_resource:\n Layer3_Network2\n type: Ericsson::Neutron::Port\n Layer3_Network2_subnet:\n properties:\n cidr: 12.41.0.24/29\n enable_dhcp:\n true\n name: Layer3_Network2_subnet\n network_id:\n get_resource:\n Layer3_Network2\n type: OS::Neutron::Subnet\n VRF1:\n depends_\n on:\n - Layer3_Network1_bgw_port\n - Layer3_Network2_bgw_port\n properties:\n name: VRF1\n type: OS::Neutron::Router\n VRF_VNIC1:\n properties:\n router_id:\n get_resource: VRF1\n subnet_id:\n get_resource: Internal_Network_1_subnet\n type:\n OS::Neutron::RouterInterface\n VRF_VNIC2:\n properties:\n router_id:\n get_resource: VRF1\n subnet_id:\n get_resource:\n Internal_Network_2_subnet\n type: OS::Neutron::RouterInterface\n VRF_VNIC3:\n properties:\n router_id:\n get_resource: VRF1\n subnet_id:\n get_resource: Layer3_Network1_subnet\n type:\n OS::Neutron::RouterInterface\n VRF_VNIC4:\n properties:\n router_id:\n get_resource: VRF1\n subnet_id:\n get_resource:\n Layer3_Network2_subnet\n type: OS::Neutron::RouterInterface\n VirtualMachine_1:\n properties:\n config_drive: 'True'\n flavor:\n get_resource: flavor_VirtualMachine_1\n image:\n 43871b95-beec-42d3-85e3-52c4c74bb22d\n name: VirtualMachine_1\n networks:\n - network:\n get_resource: Internal_Network_1\n personality:\n /ovffiles/ovf-env.xml: \n<Environment\n xmlns:ovfenv=\\\"http://schemas.dmtf.org/ovf/environment/1\\\"\\\"\\n\n \\ xmlns=\\\"http://schemas.dmtf.org/ovf/envelope/1\\\"\\\">\n<PlatformSection>\n\\n\\n \\ <Info>Product Information</Info>\n<Product>Ericsson Cloud</Product>\n\\n\\n \\ <Version>1.0</Vers\nion>\n<Vendor>Ericsson</Vendor>\n</PlatformSection>\n\\n\\n\n \\ <PropertySection>\n<Property ovfenv:key=\\\"g3.subnet\nwork\\\" ovfenv:value=\\\"\\n Germany\\\"/>\n<Property\novfenv:key=\\\"io.oam.subnet\\\" ovfenv:value=\\\"\\n\n10.63.69.96/27\\\"/>\n<Property ovfenv:key=\\\"oam.vip\\\"\\n\novfenv:value=\\\"\\n 10.63.69.164\\\"/>\n<Property\novfenv:key=\\\"application.type\\\" ovfenv:value=\\\"\\n\nCSCF\\\"/>\n<Property ovfenv:key=\\\"icscf.traffic.vip\\\"\\n\novfenv:value=\\\"\\n 10.63.69.181\\\"/>\n<Property
```



```

ovfenv:key=\\\\"pcscf.traffic.vip\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.180\\\\"/>\\\\" <Property ovfenv:key=\\\\"time.server\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.224.254\\\\"/>\\\\" <Property
ovfenv:key=\\\\"scscf.traffic.vip\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.182\\\\"/>\\\\" <Property ovfenv:key=\\\\"time.zone.city\\\\"
ovfenv:value=\\\\"\\\\"n Berlin\\\\"/>\\\\" <Property ovfenv:key=\\\\"e
cscf.traffic.vip\\\\" ovfenv:value=\\\\"\\\\"n 10.63.69.183\\\\"/>\\\\"
<Property ovfenv:key=\\\\"io.oam.sc1\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.100\\\\"/>\\\\" <Property ovfenv:key=\\\\"io.oam.sc2\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.69.101\\\\"/>\\\\" <Property
ovfenv:key=\\\\"domain.name\\\\" ovfenv:value=\\\\"\\\\"n
ims.ericcloud\\\\"/>\\\\" <Property ovfenv:key=\\\\"io.oam.gateway\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.69.97\\\\"/>\\\\" <Property
ovfenv:key=\\\\"time.zone.region\\\\" ovfenv:value=\\\\"\\\\"n
Europe\\\\"/>\\\\" </PropertySection>\\\\"</Environment>\\\\"\\\\"n type:
OS::Nova::Server\\\\"n VirtualMachine_2:\\\\"n properties:\\\\"n config_drive:
'True'\\\\"n flavor:\\\\"n get_resource: flavor_VirtualMachine_2\\\\"n image:
b12b45bc-feb7-4093-9309-c7e7827bb8d5\\\\"n name: VirtualMachine_2\\\\"n
networks:\\\\"n - network:\\\\"n get_resource: Internal_Network_2\\\\"n
personality:\\\\"n /ovffiles//ovf-env.xml: \\\\"<Environment
xmlns:ovfenv=\\\\"http://schemas.dmtf.org/ovf/environment/1\\\\"\\\\"n
\\\\" xmlns=\\\\"http://schemas.dmtf.org/ovf/envelope/1\\\\">\\\\"n
<PlatformSection>\\\\"n\\\\"n \\\\" <Info>Product Information</Info>\\\\"n
<Product>Ericsson Cloud</Product>\\\\"n\\\\"n \\\\" <Version>1.0</Vers
ion>\\\\"n <Vendor>Ericsson</Vendor>\\\\"n </PlatformSection>\\\\"n\\\\"n
\\\\" <PropertySection>\\\\"n <Property ovfenv:key=\\\\"g3.subnet
work\\\\" ovfenv:value=\\\\"\\\\"n Germany\\\\"/>\\\\" <Property
ovfenv:key=\\\\"io.oam.subnet\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.96/27\\\\"/>\\\\" <Property ovfenv:key=\\\\"oam.vip\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.69.164\\\\"/>\\\\" <Property
ovfenv:key=\\\\"application.type\\\\" ovfenv:value=\\\\"\\\\"n
CSCF\\\\"/>\\\\" <Property ovfenv:key=\\\\"icscf.traffic.vip\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.69.181\\\\"/>\\\\" <Property
ovfenv:key=\\\\"pcscf.traffic.vip\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.180\\\\"/>\\\\" <Property ovfenv:key=\\\\"time.server\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.224.254\\\\"/>\\\\" <Property
ovfenv:key=\\\\"scscf.traffic.vip\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.182\\\\"/>\\\\" <Property ovfenv:key=\\\\"time.zone.city\\\\"
ovfenv:value=\\\\"\\\\"n Berlin\\\\"/>\\\\" <Property ovfenv:key=\\\\"e
cscf.traffic.vip\\\\" ovfenv:value=\\\\"\\\\"n 10.63.69.183\\\\"/>\\\\"
<Property ovfenv:key=\\\\"io.oam.sc1\\\\" ovfenv:value=\\\\"\\\\"n
10.63.69.100\\\\"/>\\\\" <Property ovfenv:key=\\\\"io.oam.sc2\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.69.101\\\\"/>\\\\" <Property
ovfenv:key=\\\\"domain.name\\\\" ovfenv:value=\\\\"\\\\"n
ims.ericcloud\\\\"/>\\\\" <Property ovfenv:key=\\\\"io.oam.gateway\\\\"
ovfenv:value=\\\\"\\\\"n 10.63.69.97\\\\"/>\\\\" <Property
ovfenv:key=\\\\"time.zone.region\\\\" ovfenv:value=\\\\"\\\\"n
Europe\\\\"/>\\\\" </PropertySection>\\\\"</Environment>\\\\"\\\\"n type:
OS::Nova::Server\\\\"n flavor_VirtualMachine_1:\\\\"n properties:\\\\"n disk:
1\\\\"n ram: 1024\\\\"n vcpus: 1\\\\"n type: Ericsson::Nova::Flavor\\\\"n

```



```
flavor_VirtualMachine_2:\n properties:\n disk: 1\n ram: 1024\n vcpus: 1\n type: Ericsson::Nova::Flavor\n}}
```

## 3.7 List cApp File

This section describes how to view details about a specific application env file.

### Headers

```
Auth-Token: 2dbfc8d528ed444fa1a507c5bbe2a697
Content-type: application/json
```

### Request

Encode the ID of the application into the request URI to get the details of a particular application.

### Response

The data is returned as a JSON-encoded mapping in the following format:

```
{
  'ovft_capp_files' : [
    {'file_id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'file_name': 'file1'},
    {'file_id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'file_name': 'file2'},...]
}
```

### Example of capp-file-list

```
curl -i -X GET -H 'X-Auth-Token: 2dbfc8d528ed444fa1a5
07c5bbe2a697' -H 'Content-Type: application/json' -H
'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt
https://public.atlas.local:8888/v1/d9def1247d114b0b8a6b5b598af13
22c/capps/files/withenv
```

```
HTTP/1.1 200 OK
date: Fri, 21 Nov 2014 06:15:49 GMT
content-length: 355
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": [{"file_content": "env\\nfile\\nexample\\n",
"file_name": "config1.xml", "file_id": "06ee9d05-1980-418e-aa9
3-56511ae5b488", "id": "0855c096-bb79-419a-b732-d9ebcd26e467"},
{"file_content": "sample\\nenv\\nfile\\n", "file_name": "MME-GP-2.3
.xml", "file_id": "64852b02-ce25-438a-8b08-06927f03d7d2", "id":
"0855c096-bb79-419a-b732-d9ebcd26e467"}]}
```



## 3.8 Show cApp File

This section describes how to view information about a specific file in an application.

### Headers

Auth-Token: bf006373ea8f4e1fb4984fee75430f3d  
Content-type: application/json

### Request

Encode the file ID of the file into the request URI to get the HOT template.

### Response

The data is returned as a JSON-encoded mapping in the following format:

```
{
  'ovft_capps_files':[
    {'file_id': '71c675ab-d94f-49cd-a114-e12490b328d9',
     'file_name': 'file1',
     'file_content': 'file data'}}
}
```

### Example of capp-file-show

```
curl -i -X GET -H 'X-Auth-Token: bf006373ea8f4e1fb498
4fee75430f3d' -H 'Content-Type: application/json' -H
'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt
https://public.atlas.local:8888/v1/d9def1247d114b0b8a6b5b598af1322
c/capps/file/detail/06ee9d05-1980-418e-aa93-56511ae5b488
```

```
HTTP/1.1 200 OK
date: Fri, 21 Nov 2014 06:16:59 GMT
content-length: 138
content-type: application/json; charset=UTF-8
```

```
{"ovft_s": {"file_content": "env\nfile\nexample\n", "file_name":
"config1.xml", "file_id": "06ee9d05-1980-418e-aa93-56511ae5b48
8"}}
```

## 3.9 Update cApp File

This section describes how to update information about a specific file in an application.

## Headers

```
Auth-Token: bf006373ea8f4e1fb4984fee75430f3d
Content-Type: application/json
```

## Request

Encode the content of the file into the request URI to update the content of the specified file using `file_id`.

## Response

The data is returned as a JSON-encoded mapping in the following format:

```
{"ovft_capps": {"file_content": "heat_template_version: 2015-04-30
```

```
parameters:
  image_name:
    type: string
    description: Name of a image

resources:
  server:
    type: OS::Nova::Server
    properties:
      flavor: m1.medium
      image: {get_param: image_name}
      networks:
        - network: provider_51
      user_data: {get_file: ../user-data}
  my_server_new:
    type: two_nova.yaml
    properties:
      net_name: provider_51

", "file_name": "new/tripled_nest.yaml", "file_id": =>
"f9dc69cf-5056-4bdb-98e4-600499f6eef9"]}]}
```

### Example of capp-file-update

```
curl -i -X POST -H 'X-Auth-Token: gAAAAABZibHjEFs7zo1rwSM0g
PJo2Rz2MUXVLgnoyDUt7AdqkqN-nq20QL20VQtoP_6fAHgGdCzBUJSus_ZH
bqn1SjxffsMz3gk7yzAzEzQzQpHwW8Jx-Vc1q2gYV7K3kx0KAAuirfKYAiJ
495vXe4HM7I8086kclANNr03NPz9fb4P_Nrocnuk' -H 'Content-Type:
application/json' -H 'User-Agent: python-ovftclient'
--cacert /etc/ssl/certs/ca.crt -d '{"file_content":
"heat_template_version: 2015-04-30\n\nparameters:\n
image_name:\n type: string\n description: Name of a
image\n\nresources:\n server:\n type: OS::Nova::Server\n
properties:\n flavor: m1.medium\n image: {get param:
```



```
image_name}\n networks:\n - network: provider_51\n
user_data: {get_file: ../user-data}\n my_server_new:\n type:
two_nova.yaml\n properties:\n net_name: provider_51\n\n",
"file_id": "f9dc69cf-5056-4bdb-98e4-600499f6eef9"}'
https://public.atlas.local:8888/v1/e0d98cdf6d574c29923d4e81390817b
b/capps/file/f9dc69cf-5056-4bdb-98e4-600499f6eef9
```

```
HTTP/1.1 200 OK
date: Tue, 08 Aug 2017 12:43:16 GMT
content-length: 556
content-type: application/json; charset=UTF-8
connection: close
```

```
{"ovft_capps": {"file_content": "heat_template_version:
2015-04-30\n\nparameters:\n image_name:\n type: string\n
description: Name of a image\n\nresources:\n server:\n
type: OS::Nova::Server\n properties:\n flavor: m1.medium\n
image: {get_param: image_name}\n networks:\n - network:
provider_51\n user_data: {get_file: ../user-data}\n
my_server_new:\n type: two_nova.yaml\n properties:\n net_name:
provider_51\n\n", "file_name": "new/tripled_nest.yaml",
"file_id": "f9dc69cf-5056-4bdb-98e4-600499f6eef9"}}
```

## 3.10 Cleanup cApps

This section describes how to delete erroneous applications.

### Headers

```
Auth-Token: 032d46b9f7ba42b8ba2db4b3da4b2038
Content-Type: application/json
```

### Request

No Body

### Response

Response is empty.

### Example of capps-cleanup

```
curl -i -X DELETE -H 'X-Auth-Token: 032d46b9f7ba42b8ba2db4b3d
a4b2038' -H 'Content-Type: application/json' -H 'User-Agent:
python-ovftclient' --cacert /etc/ssl/certs/ca.crt https://public.a
tlas.local:8888/v1/d9def1247d114b0b8a6b5b598af1322c/capps
```

```
HTTP/1.1 200 OK
```



```
date: Fri, 21 Nov 2014 06:24:31 GMT
content-length: 16
content-type: application/json; charset=UTF-8

{"delete": true}
```

## 3.11 Personalize cApp Template

This section describes how to personalize an application template.

The following options are used to personalize an application template:

- `-e` Path to the environment file input
- `-o` Output path to save the updated HOT template
- `-s` If true, updates the cAPP template in database with updated HOT template
- `-i` Path to input HOT template to replace Database cAPP template.

### 3.11.1 Personalize cApp Template for Adding Properties

This section describes how to update the following HOT resource properties of cAPP template, using `-e` and `-o` options described in Section 3.11 on page 20:

- User data
- Metadata
- File injection
- Availability zone
- Extra specs for flavor from template, or use of existing flavor

#### Headers

```
Auth-Token: 93906540af16420d8e0b306ddec71cb5
Content-Type: application/json
```

#### Request

The data is sent as a JSON-encoded mapping in the following format:





```
{
  "ovft_capps": {
    "flavor_Controller_node1": {
      "extra_specs": {
        "quota:disk_read_bytes_sec": "10240000"},
      "type": "Ericsson::Nova::Flavor"},
    "Controller_node1": {
      "metadata": {
        "app_ip": "IP of the application"},
      "type": "OS::Nova::Server",
      "user_data": {
        "get_file": "user-data"},
      "availability_zone": "nova"},
    "store": "False"}
}
```

## Response

The HOT Template, with the injected files, is redirected to a file.

## Example of capp-template-personalize for Adding Properties

```
curl -i -X POST -H 'X-Auth-Token: 31fcc9b885d04b33983f38aaa680d8f2' -H 'Content-Type: application/json' -H 'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt -d '{"ovft_capps": {"flavor_Controller_node1": {"extra_specs": {"quota:disk_read_bytes_sec": "10240000"}, "type": "Ericsson::Nova::Flavor"}, "Controller_node1": {"metadata": {"app_ip": "IP of the application"}, "type": "OS::Nova::Server", "user_data": {"get_file": "user-data"}, "availability_zone": "nova"}, "store": "False"}}}' https://public.atlas.local:8888/v1/9dc4edc5ed584ecfa6ac7dc91bca0493/capps/personalize/demo_pkg
```

```
HTTP/1.1 200 OK
date: Thu, 01 Sep 2016 13:35:14 GMT
content-length: 2466
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": {"id": "4fdf86ed-9e2f-4a0d-b77e-785e114c4715",
"template": "description: Stack template generated by OVFT
0.3.0 from demo_package.ovf file\nheat_template_version:
'2013-05-23'\nparameters:\n param_1: {description: IP subnet,
label: demo subnet, type: string}\n param_2: {description:
Gateway, label: demo gateway, type: string}\nresources:\n
Controller_node1:\n properties:\n availability_zone: nova\n
config_drive: 'True'\n flavor: {get_resource: flavor_Contr
oller_node1}\n image: a1db5a95-40ed-4a65-9380-6eff9a21d7cc\n
metadata: {app_ip: IP of the application}\n name: Controller
node1\n networks:\n - port: {get_resource: port_1}\n\n
```



```

personality: {setup/cfg.xml: \ "<instance name=\\\\"cirros\\\\">\\n
<image name=\\\\"cirros_img\\\\">\\n >cirros_img</image>\\n
<flavor name=\\\\"m1.tiny\\\\">m1.tiny</flavor>\\n <property\\n
\\ key='username' value='$username'/>\\n</instance>\\n",
setup/resources.xml: \ "<instance\\n \\ name=\\\\"cirros\\\\">\\n
<image name=\\\\"cirros_img\\\\">cirros_img</image>\\n <flavor\\n
\\ name=\\\\"m1.tiny\\\\">m1.tiny</flavor>\\n <property
key='username' value='$username'/>\\n\\n</instance>\\n"}\\n
user_data: {get_file: user-data}\\n type: OS::Nova::Server\\n
Payload_node1:\\n properties:\\n config_drive: 'True'\\n
flavor: {get_resource: flavor_Payload_node1}\\n image:
fc626805-cc86-4828-bc2c-10d4a552719a\\n metadata: {}\\n
name: Payload node1\\n networks:\\n - port: {get_resource:
port_2}\\n type: OS::Nova::Server\\n demo:\\n properties:\\n
name: demo\\n value_specs: {'router:external': true}\\n
type: OS::Neutron::Net\\n demo_subnet:\\n properties:\\n cidr:
{get_param: param_1}\\n enable_dhcp: true\\n gateway_ip:
{get_param: param_2}\\n ip_version: '4'\\n name: demo_subnet\\n
network_id: {get_resource: demo}\\n type: OS::Neutron::Subnet\\n
flavor_Controller_node1:\\n properties:\\n disk: 1\\n extra_specs:
{'quota:disk_read_bytes_sec': '10240000'}\\n ram: 1024\\n vcpus:
1\\n type: Ericsson::Nova::Flavor\\n flavor_Payload_node1:\\n
properties: {disk: 1, ram: 1024, vcpus: 1}\\n type:
Ericsson::Nova::Flavor\\n port_1:\\n properties:\\n network_id:
{get_resource: demo}\\n type: OS::Neutron::Port\\n port_2:\\n
properties:\\n network_id: {get_resource: demo}\\n type:
OS::Neutron::Port\\n"}

```

### 3.11.2 Personalize cApp Template for Updating Main Template

This section describes how to replace the cAPP template stored in the database with a new HOT template, using the -i option described in Section 3.11 on page 20.

#### Headers

```

Auth-Token: bf006373ea8f4e1fb4984fee75430f3d
Content-Type: application/json

```

#### Request

The data is sent as a JSON-encoded mapping in the following format:



```
{
  "ovft_capps": {
    "input": "heat_template_version: 2015-04-30"
  },
  "resources": {
    "server111": {
      "type": "OS::Nova::Server",
      "properties": {
        "flavor": "m1.medium",
        "image": "620b69d2-02cc-412c-a717-6a277215a015",
        "networks": [
          {
            "network": "provider_51"
          }
        ],
        "user_data": {
          "get_file": "user-data"
        }
      }
    },
    "my_server": {
      "type": "new/tripled_nest.yaml",
      "properties": {
        "image_name": "cirros-0.3.2-x86_64-disk"
      }
    },
    "my_server111": {
      "type": "new/three_nova.yaml",
      "properties": {
        "net_name": "provider_50"
      }
    }
  }
}
```

## Response

The data is returned as a JSON-encoded mapping in the following format:

```
{
  "ovft_capps": {
    "input": "heat_template_version: 2015-04-30"
  },
  "resources": {
    "server111": {
      "type": "OS::Nova::Server",
      "properties": {
        "flavor": "m1.small",
        "image": "620b69d2-02cc-412c-a717-6a277215a015",
        "networks": [
          {
            "network": "provider_51"
          }
        ],
        "user_data": {
          "get_file": "user-data"
        }
      }
    },
    "my_server": {
      "type": "new/tripled_nest.yaml",
      "properties": {
        "image_name": "cirros-0.3.2-x86_64-disk"
      }
    },
    "my_server111": {
      "type": "new/three_nova.yaml",
      "properties": {
        "net_name": "provider_50"
      }
    }
  }
}
```

## Example of capp-template-personalize with Option -i

```
curl -i -X POST -H 'X-Auth-Token: gAAAAABZipGwGPtKAMaYDyQF4
T5DGAhAy8G9bYjIEq0tvG3ND2kJZs8cGwlwLF8kheoTVa-GBQ0_y7A8M_QK
FUJLrgDilixVHYGL4H5hhV9Dq0ilkeC2SxioJdb28S3eKg-sau0WDUmZ6nc
2D0bZSIJUM4QDwWVoMEkfBKu2VRB0neI90PhKC1c' -H 'Content-Type:
```



```
application/json' -H 'User-Agent: python-ovftclient'
--cacert /etc/ssl/certs/ca.crt -d '{"ovft_capps": {"input":
"heat_template_version: 2015-04-30\nresources:\n server111:\n
type: OS::Nova::Server\n properties:\n flavor: m1.medium\n
image: 620b69d2-02cc-412c-a717-6a277215a015\n networks:\n -
network: provider_51\n user_data: {get_file: user-data}\n
my_server:\n type: new/tripled_nest.yaml\n properties:\n
image_name: cirros-0.3.2-x86_64-disk\n my_server111:\n type:
new/three_nova.yaml\n properties:\n net_name: provider_50\n"}}'
https://public.atlas.local:8888/v1/e0d98cdf6d574c29923d4e81390817b
b/capps/personalize/sample
```

```
HTTP/1.1 200 OK
date: Wed, 09 Aug 2017 04:38:08 GMT
content-length: 549
content-type: application/json; charset=UTF-8
connection: close
```

```
{"ovft_capps": {"id": "eacade0d-d89b-4c82-b1d1-3727b3319dbf",
"template": "heat_template_version: 2015-04-30\nresources:\n
server111:\n type: OS::Nova::Server\n properties:\n flavor:
m1.medium\n image: 620b69d2-02cc-412c-a717-6a277215a015\n
networks:\n - network: provider_51\n user_data: {get_file:
user-data}\n my_server:\n type: new/tripled_nest.yaml\n
properties:\n image_name: cirros-0.3.2-x86_64-disk\n
my_server111:\n type: new/three_nova.yaml\n properties:\n
net_name: provider_50\n"}}}
```

## 3.12 Export Application

This section describes how to export an accessible application.

### Headers

```
'X-Auth-Token: c9048c6cc24b43d98758d3aad6f53336'
'Content-Type: application/json'
'User-Agent: python-ovftclient'
```

### Request

Encode the ID of the application into the request URI to export the particular application.

### Response

```
{"ovft_capps": {"capp_source": "file.ova", "capp_type": =>
```



```
"ovf"}}}
```

### Example of capp-export

```
curl -i -X GET -H 'X-Auth-Token: 9f388dd203bb49e292f68a2d4b36d52a' -H 'Content-Type: application/json' -H 'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt -d '{"id": "test", "file": "check_file.ova"}' https://public.atlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d66/capps/source/test
```

```
HTTP/1.1 200 OK
date: Mon, 02 Nov 2015 09:52:54 GMT
content-length: 73
content-type: application/json; charset=UTF-8
```

```
{"ovft_capps": {"capp_source": "/tmp/test/test.ova",
"capp_type": "ovf"}}
```

## 3.13 Scale out Stack

This section describes how to scale out an accessible stack.

### Headers

```
'X-Auth-Token: 2d6a449ef63d49b79c84325d73f8643c'
'Content-Type: application/json'
'User-Agent: python-ovftclient'
```

### Request

Encode the stack ID into the request URL to scale out the stack.

### Response

```
{"scaled_info": {"resources": ["new_vm1_scaled"]}}
```

### Example of stack-scale-method-out

```
curl -i -X POST -H 'X-Auth-Token: 2d6a449ef63d49b79c84325d73f8643c' -H 'Content-Type: application/json' -H 'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt -d '{"scaling_groups": {"resources": [{"source": "vm1", "target": "new_vm1_scaled", "personality": [{"path": "/etc/script.sh", "contents": "Something"}]}]}, "id": "stack"}' https://public.atlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d66/scale_out/stack
```

```
HTTP/1.1 200 OK
```



```
date: Mon, 28 Sep 2015 06:50:57 GMT
content-length: 50
content-type: application/json; charset=UTF-8
```

```
{"scaled_info": {"resources": ["new_vm1_scaled"]}}
```

## 3.14 Scale in Stack

This section describes how to scale in an accessible stack.

### Headers

```
'X-Auth-Token: eb3e75e662814d8c868936cd1d840087'
'Content-Type: application/json'
'User-Agent: python-ovftclient'
```

### Request

Encode the stack ID into the request URL to scale in the stack.

### Response

```
{"scaled_info": "OK"}
```

### Example of stack-scale-method-in

```
curl -i -X POST -H 'X-Auth-Token: eb3e75e662814d8c868936cd1d
840087' -H 'Content-Type: application/json' -H 'User-Agent:
python-ovftclient' --cacert /etc/ssl/certs/ca.crt -d '{"id":
"stack", "groups": ["vm1_scaled"]}' https://public.atlas.local:88
88/v1/f99b4db83ab4419caafe0a5f268c2d66/scale_in/stack
```

```
HTTP/1.1 200 OK
date: Mon, 02 Nov 2015 10:39:08 GMT
content-length: 21
content-type: application/json; charset=UTF-8
```

```
{"scaled_info": "OK"}
```

## 3.15 Stack Scaling List

This section describes how to list accessible scaling groups and servers of a stack.



## Headers

```
'X-Auth-Token: 632d8cb8a78c47e7a133a24a59990ffd'  
'Content-Type: application/json'  
'User-Agent: python-ovftclient'
```

## Request

No Body

## Response

```
{"stack_resources": {"scaling groups": ["vm1_scaled"],  
"servers": ["vm1"]}}
```

## Example of stack-scaling-list

```
curl -i -X GET -H 'X-Auth-Token: 632d8cb8a78c47e7a133  
a24a59990ffd' -H 'Content-Type: application/json' -H  
'User-Agent: python-ovftclient' --cacert /etc/ssl/certs/ca.crt  
https://public.atlas.local:8888/v1/f99b4db83ab4419caafe0a5f268c2d6  
6/stack/scaling_list/stack
```

```
HTTP/1.1 200 OK  
date: Mon, 02 Nov 2015 10:46:20 GMT  
content-length: 75  
content-type: application/json; charset=UTF-8
```

```
{"stack_resources": {"scaling groups": ["vm1_scaled"],  
"servers": ["vm1"]}}
```