ERICSSON

OpenStack Compute API in CEE

Cloud Execution Environment

INTERWORK DESCRIPTION

1/155 19-CRA 119 1861/5 Uen E

Contents

1 Introduction

This document serves as an introduction to the use of the Application Programming Interface (API) of the OpenStack component “Compute” in the Cloud Execution Environment (CEE).

While the main aim of the document is to present the Compute API in CEE, it also contains descriptive information about the features of CEE Compute.

1.1 API Version

By default, the external Virtualized Network Function Manager (VNFM) and Network Function Virtualization Orchestrator (NFVO) use the CEE Compute API microversion based on OpenStack Compute API v2.1.

Note: At deployment, OpenStack Compute API v2 can also be selected (however, v2.1 is recommended). In this case, some functions — including `forcemove` — are available as extensions.

1.2 Document References

This section contains the official OpenStack API reference.

1.2.1 API Design Base Reference

For the description of the API operations and extensions of Compute, refer to section “OpenStack Compute API v2.1” in the *OpenStack API Complete Reference*.

This is a stored copy of the OpenStack API Reference document version that was the base for the development of this version of CEE.

2 Supported Operations

The following sections contain the API operations and API extensions that are supported in the CEE.

2.1 Basic OpenStack Operations

For the detailed description of basic Compute API operations, refer to section “OpenStack Compute API v2.1” in the *OpenStack API Complete Reference*.

CEE Compute supports basic Compute API operations, with the limitations listed in Section 4 on page 11.

2.2 OpenStack Extensions

Not applicable for OpenStack Compute API v2.1.

3 Ericsson Extended Functions

This section presents the extended API functions that are specific to the CEE.

3.1 forcemove

`forcemove` is an API function that supports the migration and evacuation of Virtual Machines (VMs) from a compute host.

`forcemove` uses Nova rescheduling, honoring `same_host`, `different_host`, `server_groups` affinity filters, and High Availability (HA) policy.

Note: The HA policy are configured in the **metadata** field of the VM.

3.1.1 API Operation

In CEE, the standard OpenStack API is extended with a call for `forcemove`:

POST v2/*{tenant_id}*/servers/*{server_id}*/action

Request body:

```
{
  "forcemove": {
      "ignore_hints": false,
      "ignore_broken_dependencies": false,
      "block_migrate": false,
      "disk_over_commit": false
  }
}
```

Note: It is required to pass all four `forcemove` parameters, even though, in most cases, these are false (OpenStack default).

If the request succeeds, the response body is the following:

```
{
  "needs_start": false
}
```

In case of an error, the response body is the following:

```
{
  "badRequest": {
       "message": "No valid host was found.",
```

```
        "code": 507
    }
}
```

Note: The message and the code can vary.

3.1.2 VM Migration and Evacuation

CEE allows the configuration of HA policy for each VM. This policy is honored by the `forcemove` function.

CEE 15B policies VM Migration and VM Evacuation are removed from CEE R6 and replaced by ha-policies. For more information, see Section 3.1.2.1 on page 4.

3.1.2.1 HA Policy

The user can define the level of High Availability (HA) needed on a specific VM. Three levels of HA can be achieved by defining a HA Policy on the VM.

The possible configuration values for `ha-policy` are the following:

unmanaged "Unmanaged" means that CEE does not try to manage the VM after it was started. No action is performed by CEE on this VM. (This is the default, if no HA policy is provided.)

managed-on-host "Managed on host" means that the VM starts up with the host and shuts down with it. In case of failure, the VM is not moved to another host, but it is restarted, when the node is restarted. On `forcemove`, the VM is shut down.

ha-offline "High Availability with offline migration" means that the VM is evacuated in case of failure, moved to another host on `forcemove`.

Note:

- `ha-policy` is case-sensitive.

3.1.2.2 Policy Configuration

During the creation of a VM, configure policies in the `metadata` field in the following way:

```
{
  "server":{
     "flavorRef":"http://openstack.example.com/openstack⇒
/flavors/1",
     "imageRef":"http://openstack.example.com/openstack/images⇒
/70a599e0-31e7-49b7-b260-868f441e862b",
     "metadata":{
        "ha-policy":"managed-on-host"
     },
     "name":"new-server-test"
  }
}
```

Here, the migration policy is set to `managed-on-host`.

3.1.3 Scheduling

The default OpenStack migration call contain a parameter for the destination Compute host to where the VM is moved.

For evacuation *<host>* is optional:

```
root@cic-1:~# nova evacuate
```

Syntax:

```
nova evacuate [--password <password>] ⇒
[--on-shared-storage] <server> [<host>]
```

In CEE, a VM can be moved simply from one compute host to another compute host in the cluster using Compute rescheduling to choose the destination compute host.

This process is supported by the `forcemove` operation.

3.1.4 Hints for VM Affinity and Antiaffinity

In order to have high availability and redundancy, the application layer needs to guarantee that some VMs are on different compute hosts.

To achieve this, OpenStack provides a feature called “scheduler hints”. For example, if `vm1` is booted, and, after that, `vm2` is booted using `--hint different_host=vm1`, then OpenStack guarantees that `vm2` will be on a different node than `vm1`.

The `forcemove` functionality of CEE adds a patch to OpenStack to save hints used during boot.

3.1.4.1 Supported Hints

In CEE, `server_group` hint is supported and recommended , refer to the section “Server groups (os-server-groups)” in the *OpenStack API Complete Reference*.

The hints `same_host` and `different_host` are also supported, but deprecated.

3.1.4.2 Prerequisites

In order to use scheduler hints, `nova.conf` must add them through `scheduler_default_filters`.

Note: This prerequisite is automatically met during installation, and does not require further action.

3.1.4.3 Recommended Setup

The following scheduling filters are configured with CEE:

- In multi-server deployments: `RetryFilter`, `AvailabilityZoneFilter`, `RamFilter`, `CoreFilter`, `DiskFilter`, `ComputeFilter`, `ComputeCapabilitiesFilter`, `ImagePropertiesFilter`, `AggregateInstanceExtraSpecsFilter`, `SameHostFilter`, `DifferentHostFilter`, `ServerGroupAntiAffinityFilter`, `ServerGroupAffinityFilter`, `PciPassthroughFilter`, `NUMATopologyFilter`, `AggregateMultiTenancyIsolation`.

- In single server deployments: `RamFilter`, `CoreFilter`, `DiskFilter`, `ComputeFilter`, `ImagePropertiesFilter`.

3.1.4.4 Hint Configuration

During the boot of a VM, configure scheduler hints in the following way:

```
{
  "os:scheduler_hints": {
    "different_host": "f2e31dcd-927b-4231-a652-3ceb42c9182e"
  },
  "server": {
    "name": "test-server",
    "imageRef": "e5bb056f-af7e-4d10-9b85-fca2519a74a0",
    "flavorRef": "1",
    "max_count": 1,
    "min_count": 1,
    "networks": [{"uuid": "d67ccfaf-0de5-4ae6-9cbb-765882d1c895"}]
  }
}
```

3.1.4.5 Broken Dependencies

`forcemove` always tries to take into consideration the saved scheduler hints, but, in certain cases, it may not be able to follow the saved hints.

```
$ nova boot vm1 --image ... --flavor ... # this vm got id 8088e8b6-fd1a-4bf2-bff5-e2debb668a3a
$ nova boot vm2 --image ... --flavor ... --hint same_host=8088e8b6-fd1a-4bf2-bff5-e2debb668a3a
$ nova delete vm1
$ nova forcemove vm2
+--------------------------------------+---------------+---------------------------+-------------+
| Server UUID                          | Move accepted | Error Message             | Needs Start |
+--------------------------------------+---------------+---------------------------+-------------+
| 59961962-fc06-4d90-82e4-366aaa3a9b38 | False         | No valid host was found.  | False       |
+--------------------------------------+---------------+---------------------------+-------------+
```

Example 1 Broken Dependencies 1

Here, `forcemove` tries to use the saved hint, but cannot succeed, since `vm2` has been deleted.

However, deleted VMs can be ignored during the handling of hints:

```
$ nova forcemove vm2 --ignore-broken-dependencies
+--------------------------------------+---------------+---------------+-------------+
| Server UUID                          | Move accepted | Error Message | Needs Start |
+--------------------------------------+---------------+---------------+-------------+
| 59961962-fc06-4d90-82e4-366aaa3a9b38 | True          |               | False       |
+--------------------------------------+---------------+---------------+-------------+
```

Example 2 Broken Dependencies 2

Here, the hint `same_host=8088e8b6-fd1a-4bf2-bff5-e2debb668a3a` is ignored.

Note: If the `server_groups` affinity filter is used, hints can be ignored, but `forcemove` does not check the deleted VMs, so broken dependencies have no effect on `forcemove`.

3.1.4.6 Ignore Hints

`forcemove` can be configured to ignore all hints:

```
$ nova forcemove vm1 --ignore-hints
+--------------------------------------+---------------+---------------+-------------+
| Server UUID                          | Move accepted | Error Message | Needs Start |
+--------------------------------------+---------------+---------------+-------------+
| 59961962-fc06-4d90-82e4-366aaa3a9b38 | True          |               | False       |
+--------------------------------------+---------------+---------------+-------------+
```

Example 3 Ignore Hints

3.2 Bandwidth Based Scheduling

Bandwidth based scheduling is an extension that enables VM scheduling based on free network bandwidth. The user requests the required bandwidth using attributes in flavor. Nova scheduler uses these attributes when scheduling

VMs with that specific flavor. Nova keeps track of the reserved bandwidth on each network interface controller (NIC), on every host. Nova ensures that no NIC will be overprovisioned. Both bit rate and packet rate capacity are taken into consideration.

3.2.1 API Operation

This extension extends the `extra_specs` attribute in flavor. Two new attributes are added to `extra_specs`:

- `bandwidth:vif_inbound_average`
- `bandwidth:vif_outbound_average`

The values for the new attributes are in JSON format and contain requested bandwidth and average packet size. Rate is requested bandwidth in kbyte per sec. Size is average packet size in bytes of the VM frames on this interface. The size is used together with the byte rate, to calculate the requested frames per second.

Format:

```
{
'<device name of sriov hwnic>' : [<minimum-bandwidth-vf1>, ⇒
<minimum-bandwidth-vf2>,... ],⇒
'<device name of neutron network hwnic>' : { "rate": ⇒
[<rate-vnic1>, <rate-vnic2>, ...], "size": ⇒
[<avg-packet-size-vnic1> , <avg-packet-size-vnic2>,...] }
...
}
```

The physical interfaces can be both Neutron physical networks and SR-IOV NICs. Currently only one Neutron physical network is supported. This network has the name `'default'`. SR-IOV NICs have names defined as `'pool_'` + *<PCI-bus-address of device>*.

```
{
   'pool_41_00_0': [ 20000, 10000 ],
   'pool_41_00_1': [ 10000 ],
   'default':
      {
         'rate': [ 10000, 20000, 30000 ],
         'size': [ 512, 1024, 1024 ]
      }
}
```

Example 4 bandwidth:vif_*bound_average Attribute

```
{
“extra_specs”:{
“bandwidth:vif_inbound_average: ”{ ‘pool_41_00_0’: ⇒
[20000, 10000], ‘pool_41_00_1’: [10000], ‘default’: { ‘rate’: ⇒
[ 10000, 20000, 30000 ], ‘size’: [ 512, 1024, 1024] } }”,
“bandwidth:vif_outbound_average: ”{ ‘pool_41_00_0’: ⇒
[20000, 10000], ‘pool_41_00_1’: [10000], ‘default’: { ‘rate’: ⇒
[ 10000, 20000, 30000 ], ‘size’: [ 512, 1024, 1024] } }”,
"pci_passthrough:alias": "pool_41_00_0:2, pool_41_00_1:1"

}
}
```

Example 5 Complete extra_specs Definition

The vNICs specified in the flavor have to match the vNICs specified on ‘‘nova boot’’. To use this feature user has to specify bandwidth on all NICs on the VM, including SR-IOV interfaces. For example, it is not allowed to specify one bandwidth in flavor and then boot VM with two vNICs. It is accepted to not use the extension by not specifying any ‘‘bandwidth:*’’ attributes in flavor.

3.2.2 VMs with and without Bandwidth Requirements

This section provides information about handling VMs with and without bandwidth requirements in the same CEE Region.

VMs with unspecified bandwidth can consume all bandwidth. VMs with unspecified bandwidth cannot be scheduled on the same hosts as VMs with specified bandwidth. This can lead to fragmentation issues.

In CEE, the default `ram_weight_multiplier` is set to 1. This makes the scheduler spread VMs on hosts. In a worst-case scenario, this results in all hosts having VMs without specified bandwidth. This makes it impossible to schedule VMs with specified bandwidth as illustrated in Figure 1.

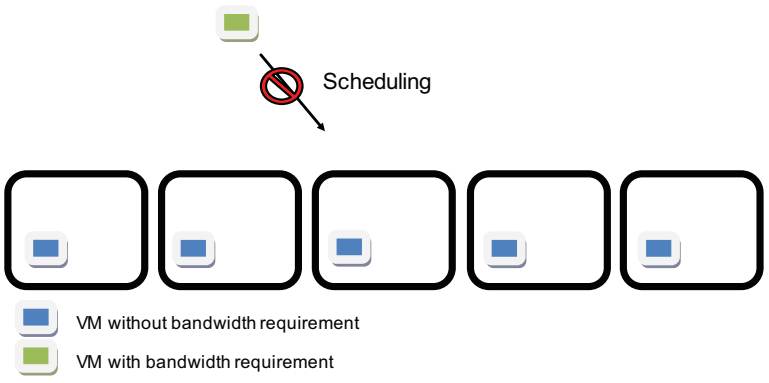

Figure 1 Bandwidth Based Scheduling Fails

To avoid this issue, divide the compute hosts into two groups: VMs with bandwidth requirements, and VMs without bandwidth requirements as shown in Figure 2. Use host aggregates. When a flavor is created, the host aggregate must be set accordingly.

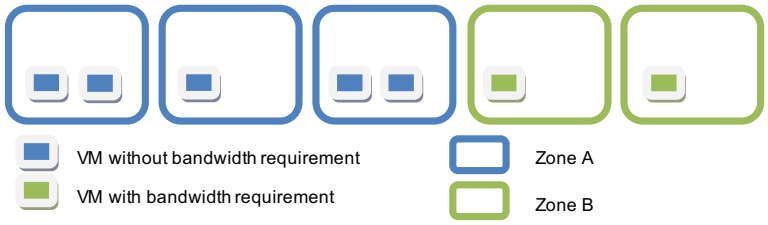

Figure 2 Using Host Aggregates to Manage Bandwidth Requirements

4 Limitations

Note: In addition to the limitations listed in this section, also refer to Section 4.2 on page 13 for limitations specific to **CEE in multi-server deployment**.

In addition to the limitations listed in this section, also refer to Section 4.3 on page 14 for limitations specific to **CEE in single server deployment**.

The following limitations exist in CEE Compute:

- The following operations are not supported:
  - `nova suspend`
  - `nova resume`
  - `nova pause`
  - `nova unpause`
  - `nova shelve`
  - `nova unshelve`
- `nova rebuild` is not supported for volume booted VMs.
- `metadata service` is not supported.
- As the maximum body size for HTTP requests is set to 112 KB in the default settings of the Nova API, larger environment files cannot be injected, even if the `injected_file_content_bytes` quota value is changed for the specific tenant.
- To refer to another VM within a hint, only the ID of the VM can be used instead of the name of the VM.

  ```
  nova boot ... --hint same_host=cbd9fa13-f51e-4ab0-b2⇒
  06-31526a236bc9
  ```

- Only VM offline migration is supported. Live migration is not supported in this release.
- Connecting VMs to additional networks on the fly with `nova interface-attach` is not supported.
- The following default flavors are configured during installation:

ID	Name	Memory_MB	Disk	Ephemeral Swap	VCPUs	RXTX_Factor	Is_Publics
2	m1.small	2048	20	0	1	1.0	True
3	m1.medium	4096	40	0	2	1.0	True
4	m1.large	8192	80	0	4	1.0	True
5	m1.xlarge	16384	160	0	8	1.0	True

- Only flavors with the following RAM and CPU values are supported:

  Note: Nova API does not reject any unused or unimplemented keys. It can result in command getting executed but with no visible impact or failed instantiation of the VM.

  – RAM allocated in steps of 1024 MB, matching the size of the hugepages used by the instances (1024, 2048, 3072 MB…)

  The following `extra_specs` need to be defined when creating the flavor:

  – `- hw:mem_page_size=`**`1048576`**

    - This will enable the hugepage backing of the instances by 1 GB hugepages.

- A VM booted with a flavor created with NUMA topology cannot be resized with another flavor with different NUMA topology. The new NUMA topology will not be applied to the resized VM. In multi-server deployment, this is for example when `hw:numa_nodes` is used in the extra specs of the flavor.

- Only the below flavor keys are supported:

  Note: This limitation also applies to setting the `hw_watchdog_action` key as a Glance image property.

  – `hw:watchdog_action` is supported with the following values:

    - `disabled`
    - `poweroff`
    - `reset`
    - `none`

    Example:

    `hw:watchdog_action=poweroff`

- Live snapshot is supported. If the VM does not support live snapshots (for example, `qemu-agent` is not installed in the VM), the VM needs to be stopped before creating a snapshot, by following these steps:

  1 Stop the VM you want to create a snapshot from:

nova stop *<VM-ID>*

2 Create the snapshot:

nova image-create *<VM-ID> <Snapshot-name>*

3 Restart the VM:

nova start *<VM-ID>*

- The `host-describe` and `hypervisor-show` commands of Nova are not reliable regarding memory usage. The used memory does not consider different page sizes and does not consider the reservations for vCIC and vFuel. For proper values check the `/proc/meminfo` file on the Compute host. This limitation is only about the shown values, the actual reservation is taken into account during scheduling.

- For the admin only API call `--availability-zone` no scheduling and resource checking happens, if the host name is used after the availability zone.

  Using the call in the format **--availability-zone nova:*<host_name>*.domain.tld** forces Nova to start the VM on the specified compute, regardless of the available resources. If you remove the compute name from the command, Nova selects a proper host with sufficient resources. If no such host exists, your request fails.

4.1 Volumes of a Deleted VM Not Removed if nova-compute is Not Running

If a VM is removed while `nova-compute` is not running, the attached volumes of the VM are not detached and removed. There are no errors indicating that the volumes are not removed.

Workaround

To remove the attached volumes, execute the following commands:

```
cinder reset-state --state available $<volume_uuid>
cinder reset-state --attach-status detached $<volume_uuid>
cinder delete $<volume_uuid>
```

4.2 Limitations for Multi-Server Deployment

In addition to the limitations described in Section 4 on page 11, the following limitations apply to CEE in multi-server deployment:

- Only flavors with the following RAM and CPU values are supported:

  - An even number of CPUs
  - The following `extra_specs` need to be defined when creating the flavor:

    - `hw:cpu_policy=`**`dedicated`**
      - This setting ensures that the instances get dedicated CPU cores from the host.

- Only the below flavor keys are supported:
  - The `hw:numa_nodes` flavor key is supported with following restrictions:

    Note: There is no visible impact between `hw:numa_nodes=1` and not defining the key at all.

    Usage of `hw:numa_nodes` with the value equal or greater than two forces symmetric splitting of vCPU and RAM resource allocations across all NUMA nodes. Therefore, following restrictions are in place to prevent incorrect VM placement or failure to instantiate a VM.

    - Only the value `hw:numa_nodes=2` is supported.
    - Use of `hw:numa_nodes` is supported only with hardware architecture where multiple NUMA nodes exist. (That is, HP or Dell hardware with 2 CPU sockets.)
    - Use of `hw:numa_nodes` key is not supported on BSP hardware.
    - Use of `hw:numa_nodes` flavor key is supported only on flavors where the requested number of vCPU resources is a multiple of 4 (4, 8, 12, 16, 20, and so on).
    - Use of `hw:numa_nodes` flavor key is supported only on flavors where the requested amount of RAM resources is a multiple of 2048 MB (2048, 4096, 6144, 8192, 10240, and so on).

- The below flavor keys are not supported:
  - `hw_video` and `hw_rng` is not supported
  - Instance resource `quota` is not supported

4.3 Limitations for Single Server Deployment

In addition to the limitations described in Section 4 on page 11, the following limitations apply to CEE in single server deployments:

Single server installation requires special flavor metadata (`extra specs`) settings.

- `hw:cpu_list` specifies the number of physical CPUs regarding the physical NUMA node, on which the virtual CPUs are pinned.

  For example, the string `node1:{9,15}/node1:{11}/node0:{10,16}/node1:{3,9}` means that there are 4 vCPUs in the flavor, which are selected by the next sequence:

  – The first vCPU is selected from NUMA node 1, physical CPUs 9 or 15

  – The second vCPU is selected from NUMA node 1, physical CPU 11

  – The third vCPU is selected from NUMA node 0, physical CPUs 1 or 16

  – The fourth vCPU is selected from NUMA node 1, physical CPUs 3 or 9

  Note: Assigning syntax does not support negation (^). For example, {2-9,^8} is not a valid definition.

  Use `lscpu` on the designated compute node to get the NUMA topology available for pinning.

  Note: Make sure to allocate the physical CPUs enlisted with owner `vm` in the `config.yaml` file as pinned vCPUs for the VM. Physical CPUs enlisted with owner `host` can be used to allocate floating vCPUs for the VM.

- `hw:numa_memory` allows defining and assigning memory of host NUMA nodes to VMs.

  For example, the string `node0:1048576/node1:1048576` means that 1 GB memory from host NUMA node 1 and 1 GB memory from second host NUMA node is used. (In this case altogether 2 GB is needed for the flavor.)