Manuals

Release 2015.1.0 (94352b9)

OPNFV

May 25, 2016

CONTENTS

CHAPTER

ONE

OPENSTACK NOVA API FOR MARKING HOST DOWN.

1.1 Related Blueprints:

https://blueprints.launchpad.net/nova/+spec/mark-host-down https://blueprints.launchpad.net/python-novaclient/+spec/support-force-down-service

1.2 What the API is for

This API will give external fault monitoring system a possibility of telling OpenStack Nova fast that compute host is down. This will immediately enable calling of evacuation of any VM on host and further enabling faster HA actions.

1.3 What this API does

In OpenStack the nova-compute service state can represent the compute host state and this new API is used to force this service down. It is assumed that the one calling this API has made sure the host is also fenced or powered down. This is important, so there is no chance same VM instance will appear twice in case evacuated to new compute host. When host is recovered by any means, the external system is responsible of calling the API again to disable forced_down flag and let the host nova-compute service report again host being up. If network fenced host come up again it should not boot VMs it had if figuring out they are evacuated to other compute host. The decision of deleting or booting VMs there used to be on host should be enhanced later to be more reliable by Nova blueprint: https://blueprints.launchpad.net/nova/+spec/robustify-evacuate

1.4 REST API for forcing down:

Parameter explanations: tenant_id: Identifier of the tenant. binary: Compute service binary name. host: Compute host name. forced_down: Compute service forced down flag. token: Token received after successful authentication. service_host_ip: Serving controller node ip.

request: PUT /v2.1/{tenant_id}/os-services/force-down { “binary”: “nova-compute”, “host”: “compute1”, “forced_down”: true }

response: 200 OK { “service”: { “host”: “compute1”, “binary”: “nova-compute”, “forced_down”: true } }

Example: curl -g -i -X PUT http://{service_host_ip}:8774/v2.1/{tenant_id}/os-services /force-down -H “Content-Type: application/json” -H “Accept: application/json ” -H “X-OpenStack-Nova-API-Version: 2.11” -H “X-Auth-Token: {token}” -d ‘{“b inary”: “nova-compute”, “host”: “compute1”, “forced_down”: true}’

1.5 CLI for forcing down:

nova service-force-down <hostname> nova-compute

Example: nova service-force-down compute1 nova-compute

1.6 REST API for disabling forced down:

Parameter explanations: tenant_id: Identifier of the tenant. binary: Compute service binary name. host: Compute host name. forced_down: Compute service forced down flag. token: Token received after successful authentication. service_host_ip: Serving controller node ip.

request: PUT /v2.1/{tenant_id}/os-services/force-down { “binary”: “nova-compute”, “host”: “compute1”, “forced_down”: false }

response: 200 OK { “service”: { “host”: “compute1”, “binary”: “nova-compute”, “forced_down”: false } }

Example: curl -g -i -X PUT http://{service_host_ip}:8774/v2.1/{tenant_id}/os-services /force-down -H “Content-Type: application/json” -H “Accept: application/json ” -H “X-OpenStack-Nova-API-Version: 2.11” -H “X-Auth-Token: {token}” -d ‘{“b inary”: “nova-compute”, “host”: “compute1”, “forced_down”: false}’

1.7 CLI for disabling forced down:

nova service-force-down –unset <hostname> nova-compute

Example: nova service-force-down –unset compute1 nova-compute

CHAPTER

TWO

GET VALID SERVER STATE

2.1 Related Blueprints:

https://blueprints.launchpad.net/nova/+spec/get-valid-server-state

2.2 Problem description

Previously when the owner of a VM has queried his VMs, he has not received enough state information, states have not changed fast enough in the VIM and they have not been accurate in some scenarios. With this change this gap is now closed.

A typical case is that, in case of a fault of a host, the user of a high availability service running on top of that host, needs to make an immediate switch over from the faulty host to an active standby host. Now, if the compute host is forced down [1] as a result of that fault, the user has to be notified about this state change such that the user can react accordingly. Similarly, a change of the host state to "maintenance" should also be notified to the users.

2.3 What is changed

A new `host_status` parameter is added to the `/servers/{server_id}` and `/servers/detail` endpoints in microversion 2.16. By this new parameter user can get additional state information about the host.

`host_status` possible values where next value in list can override the previous:

- `UP` if nova-compute is up.
- `UNKNOWN` if nova-compute status was not reported by servicegroup driver within configured time period. Default is within 60 seconds, but can be changed with `service_down_time` in nova.conf.
- `DOWN` if nova-compute was forced down.
- `MAINTENANCE` if nova-compute was disabled. MAINTENANCE in API directly means nova-compute service is disabled. Different wording is used to avoid the impression that the whole host is down, as only scheduling of new VMs is disabled.
- Empty string indicates there is no host for server.

`host_status` is returned in the response in case the policy permits. By default the policy is for admin only in Nova policy.json:

```
"os_compute_api:servers:show:host_status": "rule:admin_api"
```

For an NFV use case this has to also be enabled for the owner of the VM:

```
"os_compute_api:servers:show:host_status": "rule:admin_or_owner"
```

2.4 REST API examples:

Case where nova-compute is enabled and reporting normally:

```
GET /v2.1/{tenant_id}/servers/{server_id}

200 OK
{
  "server": {
    "host_status": "UP",
    ...
  }
}
```

Case where nova-compute is enabled, but not reporting normally:

```
GET /v2.1/{tenant_id}/servers/{server_id}

200 OK
{
  "server": {
    "host_status": "UNKNOWN",
    ...
  }
}
```

Case where nova-compute is enabled, but forced_down:

```
GET /v2.1/{tenant_id}/servers/{server_id}

200 OK
{
  "server": {
    "host_status": "DOWN",
    ...
  }
}
```

Case where nova-compute is disabled:

```
GET /v2.1/{tenant_id}/servers/{server_id}

200 OK
{
  "server": {
    "host_status": "MAINTENANCE",
    ...
  }
}
```

Host Status is also visible in python-novaclient:

```
+-------+------+--------+------------+-------------+----------+-------------+
| ID    | Name | Status | Task State | Power State | Networks | Host Status |
+-------+------+--------+------------+-------------+----------+-------------+
```

```
| 9a... | vm1  | ACTIVE | -          | RUNNING     | xnet=... | UP          |
+-------+------+--------+------------+-------------+----------+-------------+
```

2.5 Links:

[1] Manual for OpenStack NOVA API for marking host down http://artifacts.opnfv.org/doctor/brahmaputra/docs/manuals/mark-host-down_manual.html

[2] OpenStack compute manual page http://developer.openstack.org/api-ref-compute-v2.1.html#compute-v2.1