**System Host Management**

Introduce system host management for neutron.

/\* TO BE FILLED: business requirements \*/

This spec proposes an host extension which adds host management to the neutron API. It is expected that another entity (e.g., VIM or mtce) is sending host update notifications to let neutron know when hosts are added, deleted or when they change state. It also allows for that external entity to tell neutron which provider networks are associated to which host. This enables neutron to make scheduling decisions about where dhcp and router services can be scheduled. This also enables neutron to generate, report host-level issues when something goes wrong.

**Problem Description**

/\* TO BE FILLED: business use cases or user stories \*/

UseCase I.

UseCase II.

**Proposed Change**

**Data Model Impact**

The following new tables are added as part of the system host management

Feature::

 CREATE TABLE hosts (

 id CHAR(36) NOT NULL PRI KEY,

 name VARCHAR(255) NOT NULL UNIQUE KEY,

 availability ENUM(constants.HOST\_UP, constants.HOST\_DOWN) NOT NULL,

 created\_at DateTime NOT NULL,

 updated\_at DateTime

 );

 CREATE TABLE hostinterfaces (

 id CHAR(36) NOT NULL PRI KEY,

 host\_id CHAR(36),

 mtu INT NOT NULL,

 network\_type VARCHAR(255) NOT NULL,

 vlans VARCHAR(255) NOT NULL,

 created\_at DateTime NOT NULL,

 updated\_at DateTime

 CONSTRAINT `host\_id\_constraint` FOREIGN KEY (
 `host\_id`) REFERENCES `hosts` (`id`) ON DELETE CASCADE,

 );

 CREATE TABLE hostinterfaceprovidernetbindings (

 providernet\_id CHAR(36) NOT NULL,

 interface\_id CHAR(36) NOT NULL,

 CONSTRAINT `providernet\_id\_constraint` FOREIGN KEY (
 `providernet\_id`) REFERENCES `'providernets` (`id`) ON DELETE CASCADE,

 CONSTRAINT `interface\_id\_constraint` FOREIGN KEY (
 `interface\_id`) REFERENCES `hostinterfaces` (`id`) ON DELETE CASCADE,

 CONSTRAINT `providernet\_id\_interface\_id\_constraint` PRIMARY KEY (

 providernet\_id, interface\_id),

 );

 CREATE TABLE hostprovidernetbindings (

 providernet\_id CHAR(36) NOT NULL,

 host\_id CHAR(36) NOT NULL,

 CONSTRAINT `providernet\_id\_constraint` FOREIGN KEY (
 `providernet\_id`) REFERENCES `'providernets` (`id`) ON DELETE CASCADE,

 CONSTRAINT `host\_id\_constraint` FOREIGN KEY (
 `interface\_id`) REFERENCES `hosts` (`id`) ON DELETE CASCADE,

 CONSTRAINT `providernet\_id\_host\_id\_constraint` PRIMARY KEY (

 providernet\_id, host\_id),

 );

**Resource Extension**

The following new resources are being introduced and their attributes maps would be like:

RESOURCE\_ATTRIBUTE\_MAP = {
 ‘hosts': {
 'id': {'allow\_post': True, 'allow\_put': False,
 'is\_visible': True,
 'validate': {'type:uuid': None}},
 'name': {'allow\_post': True, 'allow\_put': False,
 'is\_visible': True},
 'availability': {'allow\_post': True, 'allow\_put': True,
 'validate': {'type:values': ['UP', 'DOWN',
 'up', 'down']},
 'is\_visible': True},
 'created\_at': {'allow\_post': False, 'allow\_put': False,
 'is\_visible': True},
 'updated\_at': {'allow\_post': False, 'allow\_put': False,
 'is\_visible': True},
 'tenant\_id': {'allow\_post': True, 'allow\_put': False,
 'is\_visible': False},
 'agents': {'allow\_post': False, 'allow\_put': False,
 'is\_visible': True},
 'subnets': {'allow\_post': False, 'allow\_put': False,
 'is\_visible': True},
 'routers': {'allow\_post': False, 'allow\_put': False,
 'is\_visible': True},
 'ports': {'allow\_post': False, 'allow\_put': False,
 'is\_visible': True},
 },
}

In addition, neutron routers will be extended with an attribute host which exposes the host assigned to the routers resource.

EXTENDED\_ATTRIBUTES\_2\_0 = {

 'routers': {

 ‘host’: {'allow\_post': False, 'allow\_put': False,

 'enforce\_policy': True,

 'is\_visible': True},

 },

}

**REST API Impact**

The following system host management Rest APIs will be provided in line with the new resources previously introduced:

Create host
POST /v2.0/hosts/<host-uuid>

Delete host

DELETE /v2.0/hosts/<host-uuid>

Update host

Example use cases could be:

\* Updates the availability status of a host.

\* Bind an interface with a set of provider networks while also providing system information about

an interface.

\* Unbind an interface to clear all provider network mappings.
PUT /v2.0/hosts/<host-uuid>

List hosts

GET /v2.0/hosts

List host

GET /v2.0/hosts/<host-uuid>

**Command Line Client Impact**

Openstack Client would add system host management related CLIs, for instance:

 openstack host list

 openstack host show

 openstack host create

 openstack host update

 openstack host delete

 openstack host bind-interface

 openstack host unbind-interface

**Other Impact**

\* Agent-side changes for DHCP, L3 agent etc. are expected to accommodate the server-side implementation.

\* A host driver will be needed to handle the underlying operations.

\* Current network dhcp and router scheduling/rescheduling will have to refer to the system host information this feature provides.

\* Necessary adjustments to the existing ML2 plugins are expected.