



Reference Information

This chapter contains the following sections:

- [Configuring Hierarchical Port Binding, on page 1](#)

Configuring Hierarchical Port Binding

This section describes configuring the Single Root I/O Virtualization (SR-IOV) and other vlan based m12 mechanism agents to work with OpFlex plugin. This is accomplished by using the Hierarchical Port Binding (HPB) and should work without any special modification to the configuration. Here are the basic steps needed to configure opflex with SR-IOV.

When using HPB, datapath connectivity in ACI is accomplished by creating static vlan bindings to the EPGs for networks created by OpenStack. There maybe other configuration required for datapath, for example, setting up vlan on SRIOV NIC or configuring ovs (or a load balancer in case of LBaaS). This is done by the third party agent or mechanism driver (e.g. sriovnicswitch).

How to create these assets:

Before you begin

In order to configure the datapath using static vlan bindings. The plugin requires following assets:

- A physical domain (physdom) with the correct vlan pool.
- Host-link information (which compute node fabric ethernet interface is connected to which leaf switch port)
- Host-link-network-label information (describing which fabric ethernet interface on compute node is used to serve which physnet)

This information is only needed if the deployment uses multiple physnets.

-
- Step 1** Before deploying Canonical OpenStack with ACI, make sure you have a physical domain (physdom) created with the correct vlan pool attached. The name of the physdom is passed through the "aci-physdom-id" parameter in configuration of the "neutron-api-plugin-aci" charm.
- Step 2** The host-link information usually comes from the LLDP auto discovery. It can also be provided statically using the "aci-connection-json" parameter in the "neutron-api-plugin-aci" charm deployment configuration as shown in step 3 of the example.

Step 3 In case of multiple physnet deployment you need to set `host-link-network-label` mapping in the "neutron-api-plugin-aci" charm configuration using the "aci-physnet-host-mapping" parameter.

You also need to set `NeutronNetworkVLANRanges`, enable the third party mechanism drivers using the `aci-mechanism-drivers` parameter, and make sure that the `apic_aim` parameter is the last mechanism in the list.

Example:

```
neutron-api-plugin-aci:
  aci-repo: 'deb http://10.104.1.1/ubuntu ocata31 main'
  aci-apic-hosts: 172.31.218.136
  aci-apic-username: admin
  aci-apic-password: cisco123
  ....
  ....
  aci-mechanism-drivers: openvswitch,sriovnicswitch,apic_aim
  neutron-vlan-ranges: physnet1:104:109,physnet2:1001:1100,physnet2:1101:1109
  aci-physnet-host-mapping: '{ "physnet1": "srv1:bond0,srv2:bond0,srv3:bond0",
  "physnet2": "srv1:bond1,srv2:bond1,srv4:bond1" }'
  aci-physdom-id: pdom_physnet
  aci-connection-json: '{"101": ["srv1|bond0:vpc-1-25/101-102-1-25",
  "srv2|bond0:vpc-1-26/101-102-1-26", "srv3|bond0:vpc-1-27/101-102-1-27",
  "srv1|bond1:vpc-1-28/101-102-1-28", "srv2|bond1:vpc-1-29/101-102-1-29",
  "srv4|bond1:vpc-1-30/101-102-1-30"], "102": ["srv1|bond0:vpc-1-25/101-102-1-25",
  "srv2|bond0:vpc-1-26/101-102-1-26", "srv3|bond0:vpc-1-27/101-102-1-27",
  "srv1|bond1:vpc-1-28/101-102-1-28", "srv2|bond1:vpc-1-29/101-102-1-29",
  "srv4|bond1:vpc-1-30/101-102-1-30"]}'
  aci-vpc-pairs: '101:102'
```
