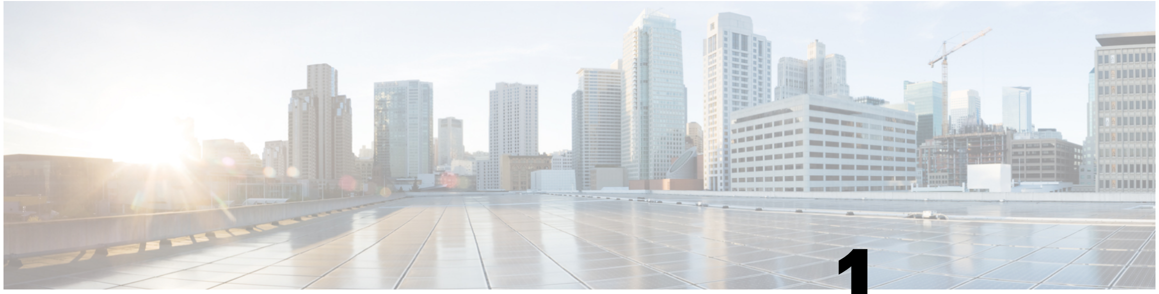
Cisco ACI Installation Guide for Red Hat OpenStack Using the OpenStack Platform 17.1 Director

First Published: 2024-02-20

Last Modified: 2025-04-10

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS



CHAPTER 1

New and Changed Information

New and Changed Information

The following table provides an overview of the significant changes to this guide for this current release. The table does not provide an exhaustive list of all changes to the guide or of the new features up to this release.

OpenStack ACI Unified plug-in	Feature	Description	Where Documented
6.0(3)	Cisco Application Centric Infrastructure (ACI) support for OpenStack 17.1.	This guide was released to document how to install Red Hat OpenStack Platform 17.1 using the Cisco ACI OpenStack plug-in.	This guide.

CHAPTER 2

Installation

Cisco ACI with OpenStack Using the OpenStack Platform 17.1 Director

The Cisco Application Centric Infrastructure (ACI) is a comprehensive policy-based architecture that provides an intelligent, controller-based network switching fabric. This fabric is designed to be programmatically managed through an API interface that can be directly integrated into multiple orchestration, automation, and management tools, including OpenStack. Integrating Cisco ACI with OpenStack allows dynamic creation of networking constructs to be driven directly from OpenStack requirements, while providing extra visibility within the Cisco Application Policy Infrastructure Controller (APIC) down to the level of the individual virtual machine (VM) instance.

OpenStack defines a flexible software architecture for creating cloud-computing environments. The reference software-based implementation of OpenStack allows for multiple Layer 2 transports including VLAN, GRE, and VXLAN. The Neutron project within OpenStack can also provide software-based Layer 3 forwarding. When used with Cisco ACI and the ACI OpenStack Unified ML2 plug-in provides an integrated Layer 2 and Layer 3 VXLAN-based overlay networking capability. This architecture provides the flexibility of software overlay networking along with the performance and operational benefits of hardware-based networking.

The Cisco ACI OpenStack plug-in can be used in either ML2 or GBP mode. In Modular Layer 2 (ML2) mode, a standard Neutron API is used to create networks. This is the traditional way of deploying VMs and services in OpenStack. In Group Based Policy (GBP) mode, a new API is provided to describe, create, and deploy applications as policy groups without worrying about network-specific details. Keep in mind that mixing GBP and Neutron APIs in a single OpenStack project is not supported. For more information, see the *OpenStack Group-Based Policy User Guide* at:

http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/openstack/b_OpenStack_Group-Based_Policy_User_Guide.html

Requirements and Prerequisites for Cisco ACI with OpenStack Using OSP Director

- Target audience: You must have working knowledge of Linux, Red Hat OpenStack distribution, the Cisco Application Centric Infrastructure (ACI) policy model and Cisco Application Policy Infrastructure Controller (APIC) configuration. You must also be familiar with OpenStack architecture and deployment.
- Cisco ACI fabric: You must have a Cisco ACI fabric that is installed and initialized with the minimum supported version that is documented in the Cisco ACI Virtualization Compatibility Matrix.

Note For communication between multiple leaf pairs, the fabric must have a BGP route reflector that is enabled to use an OpenStack external network.

- When using bonded fabric interface with a virtual port channel (vPC), adding the `ovs_bond` for the fabric interface is not supported. That is because it must be added as a single interface to the Open vSwitch (OVS) bridge. You must set the `type` to `linux_bond` for aggregating the fabric interfaces. Here is a rough example of how the fabric interface must be created in the `nic-config` templates:

```
type: linux_bond
    name: bond1
    use_dhcp: false
    mtu: 8000
    bonding_options: mode=802.3ad miimon=100 lacp_rate=slow xmit_hash_policy=layer2
    members:
    - type: interface
      name: nic2
      primary: true
      mtu: 8000
    - type: interface
      name: nic3
      mtu: 8000
```

- When using bonding, only 802.3ad is supported.
- When deploying with UCS-B series, only dual vNICs with bonding is supported for the fabric interface for redundancy.

Note Do not use a single vNIC with hardware failover.

- In the Cisco APIC GUI, disable the OpFlex authentication in the fabric. Make sure "To enforce Opflex client certificate authentication for GOLF and Linux." is not checked in **System** > **System Settings** > **Fabric Wide Setting** > **Fabric Wide Setting Policy** pane.
- When you delete the Overcloud Heat stack, the Overcloud nodes are freed, but the virtual machine manager (VMM) domain remains present in Cisco APIC. The VMM appears in Cisco APIC as a stale VMM domain along with the tenant unless you delete the VMM domain manually.

Before you delete the VMM domain, verify that the stack has been deleted from the undercloud, and check that any hypervisors appearing under the VMM domain are no longer in the connected state. Once both of these conditions are met, then you can safely delete the VMM domain from Cisco APIC.

Related Documentation

For more information, see the relevant version of the *Director Installation and Usage, Red Hat OpenStack Platform* documentation on the Red Hat website.

Deploying OpFlex

This section describes how to install and configure the Cisco Application Centric Infrastructure (ACI) OpenStack Plug-in on a Red Hat OpenStack distribution.

These example steps were validated on OpenStack Platform 17.1 releases of Red Hat OpenStack. OpenStack systems can vary widely in how they are installed. Therefore, the examples provided may be used as a basis to be adapted to the specifics of your installation.

Follow the Red Hat OpenStack Platform Director installation document to prepare the OpenStack Platform Director and create the correct deployment and resource files.

For more information, see Related Documentation, on page 5 in this guide.

Preparing Cisco ACI for OpenStack Installation

Setting Up the Cisco APIC and the Network

This section describes how to set up the Cisco Application Policy Infrastructure Controller (APIC) and the network.

Refer to the Network Planning section of the OpenStack Platform Director documentation for network layout such as the one shown in the figure below. For more information, see *Installing and Managing Red Hat OpenStack Platform with Director* documentation on the Red Hat website.

Figure 1: Typical OpenStack platform topology

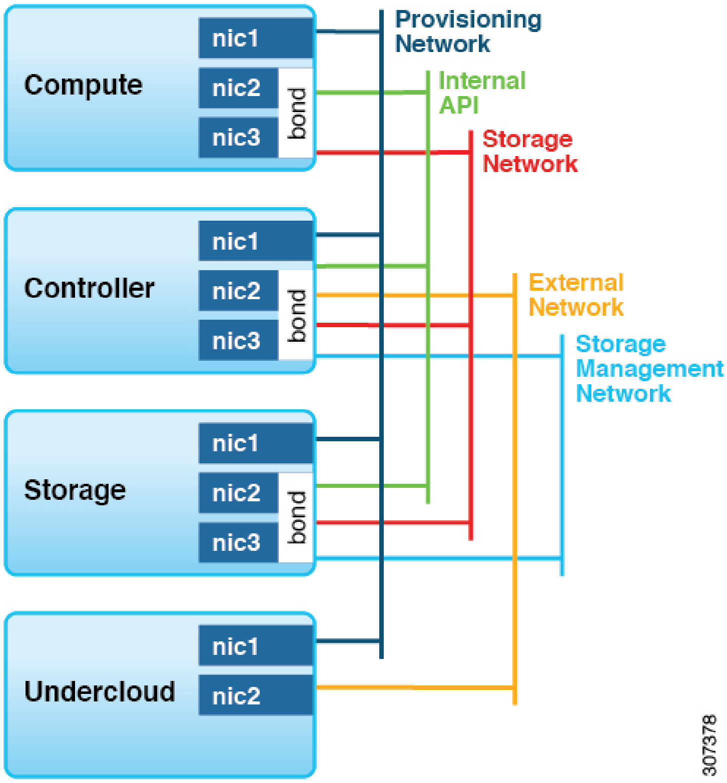
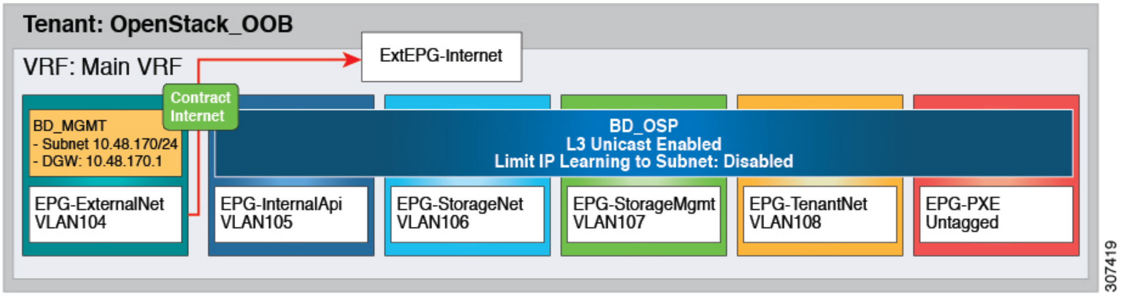


Figure 2: Typical topology for installation of Red Hat OpenStack Platform with the Cisco ACI plug-in



- The PXE network must use a native VLAN. Because native VLAN typically is defined as a dedicated NIC on the OpenStack nodes, you can connect PXE network interfaces either to the Cisco Application Centric Infrastructure (ACI) fabric or to a different switching fabric.
- All OpenStack Platform (OSP) networks except for PXE are in-band (IB) through Cisco ACI. The following VLANs are examples:
  - API: VLAN 10
  - Storage: VLAN 11
  - StorageMgmt: VLAN 12
  - Tenant: VLAN 13

  - External: VLAN 14
  - Cisco ACI Infra: VLAN 4093

- ExtEPG-Internet used in this example is the L3Out external EPG that allows connectivity to Internet for the OpenStack External Network. You also may need to provide external connectivity for the Internal API OpenStack network, according to your requirements.

To prepare Cisco ACI for in-band configuration you can use the physical domain and the static binding to the EPGs created for these networks. This involves creating the required physical domain and attachable access entity profile (AEP). Note that the infra VLAN should be enabled for the AEP. For more details, see the knowledge base article *Creating Domains, Attach Entity Profiles, and VLANs to Deploy an EPG on a Specific Port*.

Procedure

Step 1 Log in to the Cisco APIC GUI and create a VLAN pool for the VLANs required for OpenStack Platform installation.

a) On the menu bar, choose **Fabric** > **Access Policies** > **Pools** and right-click **VLAN** to create a VLAN pool.
b) In the **Name** field, enter the VLAN range namespace policy name. (For example, OSP17.1-infra.)
c) (Optional) In the **Description** field, enter the description of the VLAN range namespace policy.
d) In the **Encap Blocks** section, click on the **+** icon to enter the encap block range.
e) Click **Submit**.

Step 2 Create an attachable entity profile and assign the above PhysDom to it. Also make sure **Enable Infra VLAN** is selected:

a) On the menu bar, choose **Fabric** > **Access Policies** > **Global Policies** and right-click **Attachable Access Entity Profile** to create an attachable access entity profile.
b) In the **Name** field, enter the name of the attachable access entity profile. (For example, OSP17.1-AEP.)
c) (Optional) In the **Description** field, enter the description of the attachable access entity profile.
d) Check the **Enable Infrastructure VLAN** check box to enable the infrastructure VLAN.
e) In the **Domains (VMM, Physical or External) To Be Associated To Interfaces:** section, click on the + icon, from the drop-down list, choose the domain profile and click **Update**.
f) Click **Next**.
g) Click **Finish**.

Step 3 Create a Physical Domain (PhysDom) and assign the VLAN pool to it.

a) On the menu bar, choose **Fabric** > **Access Policies** > **Physical and External Domains** and right-click **Physical Domains** to create a Physical Domain.
b) In the **Name** field, enter the name of the physical domain. (OSP17.1-phys).
c) In the **Associated Attachable Entity Profile** field, choose an associated attachable entity profile.
d) In the **VLAN Pool** field, choose a VLAN pool ([OSP17.1-infra-dynamic]).

   If VLAN is used as the encapsulation method between the OpenStack nodes and the Cisco ACI leaf switches, you need to choose a VLAN pool range according to the pool used from OpenStack Neutron networks.

e) Click **Submit**.

Step 4 In a separate tenant, you can also use Common to create an application profile. (For example, OSP-17.1.) Create the EPGs, bridge domains, and a VRF for the OSP Networks. If the PXE network is also going through Cisco ACI then also create EPG and BD for PXE (This is not shown in this example).

Step 5 Add static bindings (Paths) for the required VLANs. You have to expand the EPG to see the ":Static Binding Paths".

a) Make sure the physical domain you created is attached to this EPG. You can add the physical domain using **Application Profiles** > **EPG** > **EPG_name** > **Domains**.

b) On the menu bar, choose **Tenants** > **Tenant common** > **Application Profiles** > ***ACI-OSP17.1*** > **Application EPGs** > ***EPG API*** > **Static Binding Paths**.

Step 6 Make sure the PhysDom is attached to the EPG.

Note

Cisco ACI needs to be provisioned for networks mentioned above except for Tenant, External and Floating IP network. This involves creating the required phys-doms and attached entity profile. Important thing to note is that Infra VLAN should be enabled for the attached entity profile.

Cisco ACI should now be ready for OpenStack deployment.

Setting up Overcloud

You must follow the *Director Installation and Usage, Red Hat OpenStack Platform* document to prepare the OpenStack Platform Director (ensure you are referring to the correct document version), and create the correct deployment and resource files.

For more information, see the document on the Red Hat website. When following Chapter 5—"Configuring a Container Image Source"—note the registry address. You might need to prepare the custom NIC templates as required following the Red Hat documentation.

After you set up the OpenStack Plaform Director, you must install the Cisco Application Centric Infrastructure (ACI) TripleO orchestration before proceeding with deployment.

Prepare Undercloud for Cisco ACI with OpFlex Orchestration

This section describes how to install the integration package for Cisco Application Centric Infrastructure (ACI) with OpFlex Orchestration.

Procedure

Step 1 Log in to undercloud as user `stack`.

Step 2 Download the Cisco ACI OSP (tripleo-ciscoaci-17.1) RPM 5.1.3 or later and the corresponding plug-in tarball (openstack-ciscorpms-repo-17.1) from Cisco.com and place them on the OpenStack Platform Director.

Step 3 Install the RPM. This action installs the dependencies.

If the RPM is installed using the `rpm` command, some dependency may need to be manually installed

Example:

```
$ sudo yum --nogpgcheck localinstall <rpm file>
```

Step 4 Create the Cisco ACI containers by completing the following steps:

a) Run the following command: **`sudo podman login registry.connect.redhat.com`**
b) When prompted, use your Red Hat credentials to enter the `redhat` username and password.
c) After you log in, run the following script as root to create the Cisco ACI containers:

`/opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py`

d) Point the script to the downloaded plug-in tarball.

Example:

```
sudo /opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py -z
/home/stack/openstack-ciscorpms-repo-17.1-778.tar.gz --image-tag 17.1 --pull
```

The command pulls the upstream Red Hat Certified ACI container images and pushes them to the local container repository. It creates an environment file named `/home/stack/templates/ciscoaci_containers.yaml`, which should be included as a template during Overcloud deployment. You can use the `-o` option to override the output filename. Verify that the output file was created as you specified.

Note

- To build the containers locally you can omit the '-p' option. However, those containers may not be Red Hat certified. You will need to login to **registry.redhat.io** before building the containers.
- During execution of the local container-creation command, you may see an error that is generated by the command **`/bin/gbp-db-manage`**. You can safely ignore this error, which should not cause the execution of the script to fail.
- OpenStack Director 17.1 deployments support configuration of a Docker registry. Users have the following choices for the registry:
    - Upstream registry (allows for using a local satellite server – currently the Red Hat registry)
    - Downstream registry address/port/URI (currently the underlay controller, 8787, /rhosp17.1)

The Docker registry is configured using the `build_openstack_aci_containers.py` script:

```
usage: build_openstack_aci_containers.py [-h] [-u UCLOUD_IP] [-o OUTPUT_FILE]
                                         [-c CONTAINERS_TB]
                                         [-s UPSTREAM_REGISTRY]
                                         [-d DESTINATION_REGISTRY]
                                         [-r REGSEPARATOR] [-i RELEASE_TAG]
                                         [-t TAG]
                                         [-a [ADDITIONAL_REPOS [ADDITIONAL_REPOS
...]]]
                                         [--force] [-p] (-f FILE | -z FILE)

Build containers for ACI Plugin

optional arguments:
  -h, --help            show this help message and exit
  -u UCLOUD_IP, --ucloud_ip UCLOUD_IP
                        Undercloud ip address
  -o OUTPUT_FILE, --output_file OUTPUT_FILE
                        Environment file to create, default is
                        /home/stack/templates/ciscoaci_containers.yaml
  -c CONTAINERS_TB, --container CONTAINERS_TB
                        Containers to build, comma separated, default is all
  -s UPSTREAM_REGISTRY, --upstream UPSTREAM_REGISTRY
                        Upstream registry to pull base images from, eg.
                        registry.access.redhat.com/rhosp13, defaults to
                        registry.access.redhat.com/rhosp13
  -d DESTINATION_REGISTRY, --destregistry DESTINATION_REGISTRY
                        Destination registry to push to, eg:
                        1.100.1.1:8787/rhosp13
  -r REGSEPARATOR, --regseparator REGSEPARATOR
                        Upstream registry separator for images, eg. '/' for
                        normal upstream registrys (default). Will be added
                        between upstream registry name and container name. Use
                        '_' for satellite based registries.
  -i RELEASE_TAG, --image-tag RELEASE_TAG
                        Upstream release tag for images, defaults to 17.1
  -t TAG, --tag TAG     tag for images, defaults to current timestamp
  -a [ADDITIONAL_REPOS [ADDITIONAL_REPOS ...]], --additional-repos [ADDITIONAL_REPOS
[ADDITIONAL_REPOS ...]]
                        Additional repos to use when building containers
                        (defaults to empty list). Use with
                        'rhel-8-for-x86_64-baseos-eus-rpms
                        rhel-8-for-x86_64-appstream-eus-rpms' when using
                        satellite.
  --force               Override check for md5sum mismatch
  -p, --pull            Pull upstream containers instead of building locally
  -f FILE, --aci_repo_file FILE
                        Path to yum repoistory file, which describes the
                        repository which provides ACI plugin rpm files. If you
                        want this script to create a repository on undercloud,
                        please use the -z option to provide path to openstack-
                        aci-rpms-repo tar file downloaded from cisco website
  -z FILE, --aci_rpm_repo_tar_file FILE
                        Path to openstack-aci-rpms-repo tar file. This will be
                        use to create a local yum repository on undercloud
```

Install Overcloud

This section describes how to install Overcloud.

Procedure

Step 1 Copy the `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` file to a private location.

Example:

```
cp /usr/share/openstack-tripleo-heat-templates/roles_data.yaml
/home/stack/templates/custom_roles_data.yaml
```

Step 2 Edit the local copy of `roles_data.yaml(custom_roles_data.yaml)` to add CiscoAciAIM and CiscoAciLldp service to the controller role and CiscoAciLldp service to the compute role.

a) Under the controller role, add the following lines:

```
- OS::TripleO::Services::CiscoAciAIM
- OS::TripleO::Services::CiscoAciLldp
- OS::TripleO::Services::CiscoAciOpflexAgent
```

b) Under the compute role, add the following line:

```
  - OS::TripleO::Services::CiscoAciLldp
  - OS::TripleO::Services::CiscoAciOpflexAgent
```

An ansible playbook is provided, that modifies the upstream `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` file to add these roles. You can skip the above step and run the playbook instead using the `ansible-playbook -i ~/inventory.yaml /opt/ciscoaci-tripleo-heat-templates/tools/generate_ciscoaci_role_data.yaml` command.

Step 3 Follow the OpenStack Director instructions and provision the network, VIPs and nodes.

Step 4 Declare resources for Cisco Application Centric Infrastructure (ACI) environment.

Define Cisco ACI resources in a `.yaml` template file to include with deployment. For example, `/home/stack/templates/aci_cs.yaml`. This step describes the resource declaration for an OpFlex agent use case.

Note

- For an example of a full resources declaration, see the section "Example of Resources Declaration" in the appendix of this guide.
- For a list of parameters that are required for the Cisco ACI environment, see the section "Parameters for the Cisco ACI Environment" in the appendix of this guide.

Example:

The following example shows resources for deploying OSP with opflex:

```
A Heat environment file which can be used to enable a Neutron Cisco ACI backend on the
controller, configured via puppet resource_registry:

#controller
  OS::TripleO::ControllerExtraConfigPre: /opt/ciscoaci-tripleo-heat-templates//nodepre.yaml

  OS::TripleO::Services::NeutronOvsAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron_opflex/neutron-opflex-agent-container-puppet.yaml

  OS::TripleO::Services::CiscoAciOpflexAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/opflex/opflex-agent-container-puppet.yaml
  OS::TripleO::Services::NeutronMl2PluginBase:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron/neutron-ml2-ciscoaci.yaml
  OS::TripleO::Services::CiscoAciAIM:
/opt/ciscoaci-tripleo-heat-templates/deployment/aciaim/cisco-aciaim-container-puppet.yaml
  OS::TripleO::Services::NeutronMetadataAgent:
/usr/share/openstack-tripleo-heat-templates/deployment/neutron/neutron-metadata-container-puppet.yaml

  OS::TripleO::Services::NeutronDhcpAgent:
/usr/share/openstack-tripleo-heat-templates/deployment/neutron/neutron-dhcp-container-puppet.yaml

  #compute
  OS::TripleO::ComputeExtraConfigPre: /opt/ciscoaci-tripleo-heat-templates//nodepre.yaml
  OS::TripleO::Services::ComputeNeutronOvsAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron_opflex/neutron-opflex-agent-container-puppet.yaml

  OS::TripleO::Services::ComputeCiscoAciOpflexAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/opflex/opflex-agent-container-puppet.yaml
  OS::TripleO::Services::ComputeNeutronMetadataAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/compute_neutron_metadata/compute-neutron-metadata.yaml


  OS::TripleO::Services::CiscoAciLldp:
```

```
/opt/ciscoaci-tripleo-heat-templates/deployment/lldp/cisco_lldp.yaml
  OS::TripleO::NodeUserData:
/usr/share/openstack-tripleo-heat-templates/firstboot/userdata_root_password.yaml

  OS::TripleO::Services::OVNDBs: OS::Heat::None
  OS::TripleO::Services::OVNController: OS::Heat::None
  OS::TripleO::Services::OVNMetadataAgent: OS::Heat::None
  OS::TripleO::Services::ComputeNeutronL3Agent: OS::Heat::None
  OS::TripleO::Services::NeutronL3Agent: OS::Heat::None

parameter_defaults:

  DockerInsecureRegistryAddress: ["fab205-ucloud-17.ctlplane.localdomain:8787",
"1.100.1.1:8787", "172.28.184.248"]
  NeutronCorePlugin: 'ml2plus'
  NeutronServicePlugins: 'group_policy,ncp,apic_aim_l3'
  NeutronEnableIsolatedMetadata: true
  NeutronEnableForceMetadata: true
  NeutronPluginExtensions: apic_aim,port_security,dns
  NeutronPhysicalDevMappings: physnet1:eth1,physnet2:eth2
  EnablePackageInstall: true
  ACIScopeNames: true
  ACIApicHosts: 10.30.120.148
  ACIApicUsername: admin
  ACIApicPassword: noir0123
  ACIApicSystemId: fab205
  ACIMechanismDrivers: 'apic_aim'
  ACIApicEntityProfile: sauto_fab205_aep
  ACIApicInfraVlan: 4093
  ACIApicInfraSubnetGateway: 10.0.0.30
  ACIApicInfraAnycastAddr: 10.0.0.32
  ACIOpflexUplinkInterface: bond1
  ACIOpflexEncapMode: vxlan
  NeutronNetworkVLANRanges: physnet1:1701:1750
  ACIOpflexVlanRange: 701:750
  HeatEnginePluginDirs:
/usr/lib64/heat,/usr/lib/heat,/usr/local/lib/heat,/usr/local/lib64/heat,/usr/lib/python2.7/site-packages/gbpautomation/heat


  ACIVpcPairs: 101:102
  NeutronPluginMl2PuppetTags: 'neutron_plugin_ml2,neutron_plugin_cisco_aci'

  AciVmmMcastRanges: 225.5.1.1:225.5.255.255
  AciVmmMulticastAddress: 225.5.10.3
  ACIYumRepo: http://1.100.1.1:8787/v2/__acirepo
```

Step 5 To use Cisco ACI certificate-based authentication, create a local user with an X.509 certificate and specify the certificate and key in the Cisco ACI resources file using the parameters ACIApicPrivateKey and ACIApicCertName.

See the section "Creating a Local User and Adding a User Certificate" in the *Cisco APIC Security Configuration Guide*.

Note
When you use certificate-based authentication, make sure that you do not specify the parameter `ACIApicPassword`.

Step 6 Deploy Overcloud.

When deploying Overcloud, include the custom roles data file created using the `-r` option. Also include the Cisco ACI environment file and Cisco ACI containers YAML file in the environment list in addition to site-specific environment files.

Example:

```
openstack overcloud deploy --templates -n /home/stack/templates/network-environment.yaml
-r /home/stack/templates/custom_roles_data.yaml -e
/home/stack/templates/overcloud-baremetal-deployed.yaml -e
/home/stack/templates/overcloud-networks-deployed.yaml -e
/home/stack/templates/overcloud-vip-deployed.yaml -e
/home/stack/containers-prepare-parameter.yaml -e /home/stack/templates/ciscoaci-config.yaml
 -e /home/stack/templates/ciscoaci_containers.yaml
```

The preceding example illustrates the use of Cisco ACI templates and roles. Other templates may differ depending on your installation configuration. Follow the Red Hat guidelines for the creation of custom template(s).

References

- Director installation and usage for the Red Hat OpenStack platform on the Red Hat website.
- The knowledge base article *Creating Domains, Attach Entity Profiles, and VLANs to Deploy an EPG on a Specific Port* on Cisco.com.
- Also see, Cisco ACI Fabric Initialization Example.

CHAPTER 3

Upgrading Cisco ACI and OSP

Guidelines for Upgrading Cisco APIC and OSP

The OpenStack plug-in is released with Cisco Application Policy Infrastructure Controller APIC releases, and therefore uses the same semantic version as Cisco APIC. For example, the 5.1(1) plug-in is provided with the Cisco APIC 5.1(1) release. Generally, the OpenStack plug-in releases are tested against the matching Cisco APIC release, as well as the previous Long Term Support (LTS) Cisco APIC release. However, a given plug-in release may be compatible with additional Cisco APIC releases. See the *Cisco ACI Virtualization Compatibility Matrix* to verify that the version of the plug-in used is compatible with the version of Cisco APIC.

See the *Cisco ACI Virtualization Compatibility Matrix* for information about compatible Cisco APIC and Red Hat OSP releases.

Pre-upgrade Guidelines

Upgrade the Cisco Application Centric Infrastructure (ACI) plug-in.

For more information about the compatibility of the plug-in with various OpenStack versions, see the *Cisco ACI Virtualization Compatibility Matrix*.

Upgrade Guidelines

The Cisco Application Centric Infrastructure (ACI) fabric can be upgraded following the information in the *Cisco APIC Installation, Upgrade, and Downgrade Guide*.

Optionally, you can upgrade the Cisco ACI fabric without upgrading the plug-in, as long as the Cisco ACI plug-in and Cisco ACI fabric release combination is supported. For more information, see the *Cisco ACI Virtualization Compatibility Matrix*.

Post-upgrade Guidelines

After you upgrade the Cisco Application Centric Infrastructure (ACI) fabric, you can optionally upgrade the OpenStack Cisco ACI packages to a version which is equal or lower than the Cisco ACI fabric code you have upgraded to. You should also refer to the OpenStack Cisco ACI plug-in Release Notes on Cisco.com for specific information.

For more information on how to upgrade the OpenStack Cisco ACI plug-in, see Upgrade the Cisco ACI Packages, on page 16 in this guide.

Upgrade the Cisco ACI Packages

The following procedure updates fully deployed Overcloud with the new version of the Cisco Application Centric Infrastructure (ACI) plug-in. The upgrade can be live.

Note Follow the Red Hat Director documentation to upgrade the plug-in in step 4.

Procedure

Step 1 Copy the updated version of the `tripeo-ciscoaci-`*`version`* RPM and corresponding plug-in tarball (`openstack-ciscorpms-repo`) from Cisco.com to the OSP Director.

Step 2 Update the `tripeo-ciscoaci-`*`version`* package using `yum`: **`yum update tripleo-ciscoaci-`**

Step 3 Create the Cisco ACI containers by running the command by first logging into **registry.connect.redhat.com** using **sudo podman login registry.connect.redhat.com** and then running `/opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py` and then pointing it to the downloaded plug-in tarball. For example:

```
opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py -p
-z/home/stack/openstack-ciscorpms-repo-163.0-848.tar.gz
```

The command updates the Docker images and updates the `/home/stack/templates/cisco_containers.yaml` file.

Note
If you are upgrading a system using Satellite or any other server, you must ensure that the undercloud and/or Satellite has the `ciscoaci-puppet` RPM from the new release. The overcloud nodes use the `nodepre.yaml` tripleo template to update this RPM in the overcloud nodes during upgrades. This RPM must also be used to rebuild the new containers for the upgrade.

Step 4 Follow the Red Hat upgrade procedure on the Red Hat Customer Portal.

See Chapter 4, "Updating the Overcloud" in the article *Keeping Red Hat OpenStack Platform Updated*. Go to **Products & Services** > **Product Documentation** > **Red Hat OpenStack Platform** > **17.1**.

CHAPTER 4

Adding an Openstack External Network

Add an OpenStack External Network

This section describes how to add an OpenStack external network.

Note Execute the commands in this procedure sourcing the keystone file for the project where you want to create the network constructs and the instance.

Before you begin

You must have done the following before adding an OpenStack external network:

- Created a Layer 3 outside connection (L3Out) in Cisco Application Centric Infrastructure (ACI).

  The L3Out can be in the OpenStack-created tenant (dedicated L3out for the OpenStack tenant) or in the Common tenant (Shared L3out across multiple OpenStack tenants). This procedure assumes that a dedicated L3out called *l3out1* is configured in the OpenStack tenant.

- Specified the following in the L3Out:
  - Interfaces and their IP address information.
  - Dynamic routing, if used.
  - An external endpoint group (EPG).

    This procedure uses an external EPG named *extEpg*.

Note Do not add any contracts; the plug-in adds them automatically.

Important If you require Source Network Address Translation (SNAT) or a floating IP (FIP) address, you must define the L3Out in a different VRF from the one created by OpenStack.

Procedure

Step 1 Create the Neutron external network and provide the distinguished name of the L3Out.

Example:

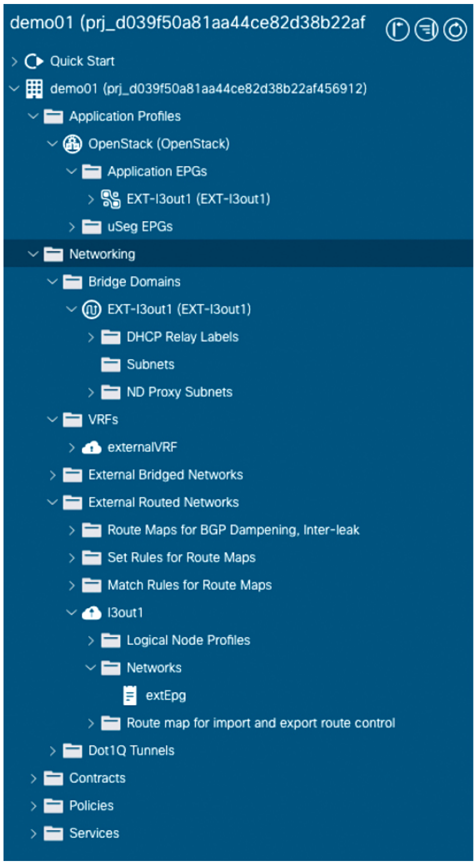
```
neutron net-create network_name --router:external --apic:distinguished_names type=dict
ExternalNetwork=uni/tn-ACI_tenant_name/out-ACI_L3out_name/instP-ACI_externalEPG_name
(--apic:nat_type "")
```

`--apic:nat_type ""` is optional. Use it only if you do not use NAT for the specific external Neutron network.

The following shows an example of the creation of the external network with NAT enabled:

```
neutron net-create external-net-dedicated --router:external --apic:distinguished_names
type=dict ExternalNetwork=uni/tn-prj_$demo01/out-l3out1/instP-extEpg
Created a new network:
+-------------------------------------+----------------------------------------------------+
| Field                                 | Value
    |
+-------------------------------------+----------------------------------------------------+
| admin_state_up                        | True
    |
| apic:bgp_asn                          | 0
    |
| apic:bgp_enable                       | False
    |
| apic:bgp_type                         | default_export
    |
| apic:distinguished_names              | {"EndpointGroup":
"uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/ap-OpenStack/epg-EXT-l3out1",
"ExternalNetwork": "uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/out-l3out1/instP-extEpg",
"VRF": "uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/ctx-externalVRF", "BridgeDomain":
"uni/tn-prj_cdeda9c674a94394a09e86a2fea498c2/BD-EXT-l3out1"}
    |
| apic:external_cidrs                   | 0.0.0.0/0
    |
| apic:nat_type                         | distributed
    |
| apic:nested_domain_allowed_vlans      |
    |
| apic:nested_domain_infra_vlan         |
    |
| apic:nested_domain_name               |
    |
| apic:nested_domain_node_network_vlan |
    |
| apic:nested_domain_service_vlan       |
    |
| apic:nested_domain_type               |
    |
| apic:svi                              | False
    |
| apic:synchronization_state            | build
    |
| availability_zone_hints               |
    |
| availability_zones                    |
    |
| created_at                            | 2019-05-22T13:38:32Z
    |
| description                           |
    |
| id                                    | 635623ed-5dba-42ec-b3f8-3cff18f925c6
    |
| ipv4_address_scope                    |
    |
| ipv6_address_scope                    |
    |
| is_default                            | False
    |
| mtu                                   | 9000
    |
| name                                  | external-net-dedicated
    |
| port_security_enabled                 | True
    |
| project_id                            | cdeda9c674a94394a09e86a2fea498c2
    |
| provider:network_type                 | opflex
    |
| provider:physical_network             | physnet1
    |
| provider:segmentation_id              |
    |
| revision_number                       | 6
    |
| router:external                       | True
    |
| shared                                | False
    |
| status                                | ACTIVE
    |
| subnets                               |
    |
| tags                                  |
    |
| tenant_id                             | cdeda9c674a94394a09e86a2fea498c2
    |
| updated_at                            | 2019-05-22T13:38:33Z
    |
+------------------------------------+---------------------------------------------------+
```

In Cisco ACI, the command creates a new EPG—*EXT-l3out1*— and a new bridge domain—*EXT-l3out1*, as shown in the following screen capture of the Cisco Application Policy Infrastructure Controller (APIC) GUI:



Step 2 Create a Neutron subnet that will be used for SNAT and the floating IP address.

This step is not required if you used `--apic:nat_type ""` when you created the Neutron external network (because NAT is disabled).

Example:

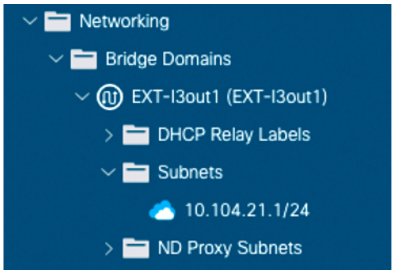
```
neutron subnet-create net_name subnet/mask --name subnet_name --disable-dhcp --gateway
gateway_ip --apic:snat_host_pool True
```

The OpFlex agent automatically assigns one IP address for every compute node from the subnet. Virtual machines (VMs) connecting to the external use this IP address (one-to-many NAT) unless they have been assigned with a floating IP address.

The following shows an example of the creation of the external network with NAT enabled

```
neutron subnet-create external-net-dedicated 10.104.21.0/24 --name ext-subnet --disable-dhcp
 --gateway 10.104.21.1 --apic:snat_host_pool True
Created a new subnet:
+----------------------------+--------------------------------------------------+
| Field                      | Value                                            |
+----------------------------+--------------------------------------------------+
| allocation_pools           | {"start": "10.104.21.2", "end": "10.104.21.254"} |
| apic:distinguished_names   | {}                                               |
| apic:snat_host_pool        | True                                             |
| apic:synchronization_state | N/A                                              |
| cidr                       | 10.104.21.0/24                                   |
| created_at                 | 2019-05-22T13:38:35Z                             |
| description                |                                                  |
| dns_nameservers            |                                                  |
| enable_dhcp                 | False                                           |
| gateway_ip                  | 10.104.21.1                                     |
| host_routes                 |                                                 |
| id                          | 238aa55d-1537-4f01-86c9-5f6fc4bde625            |
| ip_version                  | 4                                               |
| ipv6_address_mode           |                                                 |
| ipv6_ra_mode                |                                                 |
| name                        | ext-subnet                                      |
| network_id                  | 635623ed-5dba-42ec-b3f8-3cff18f925c6            |
| project_id                  | cdeda9c674a94394a09e86a2fea498c2                |
| revision_number             | 0                                               |
| service_types               |                                                 |
| subnetpool_id               |                                                 |
| tags                        |                                                 |
| tenant_id                   | cdeda9c674a94394a09e86a2fea498c2                |
| updated_at                  | 2019-05-22T13:38:35Z                            |
+-----------------------------+-------------------------------------------------+
```

Creating a SNAT subnet generates a new subnet under the bridge domain, as shown in the following screen capture of the Cisco APIC GUI:



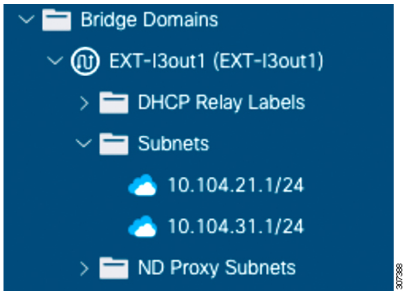
Step 3 (Optional) Assign one or more floating subnets to the external Neutron network:

Example:

```
neutron subnet-create net_name fip_subnet/mask --name subnet_name --allocation-pool
start=start_ip,end=end_ip --disable-dhcp --gateway gateway_ip
```

The following output and screen capture in the Cisco APIC GUI show an example of the creation of a floating IP subnet:

```
neutron subnet-create external-net-dedicated 10.104.31.0/24 --name ext-subnet-FIP
--allocation-pool start=10.104.31.10,end=10.104.31.100 --disable-dhcp --gateway 10.104.31.1
Created a new subnet:
+-----------------------------+--------------------------------------------------+
| Field                       | Value                                            |
+-----------------------------+--------------------------------------------------+
| allocation_pools            | {"start": "10.104.31.10", "end": "10.104.31.100"} |
| apic:distinguished_names    | {}                                               |
| apic:snat_host_pool         | False                                            |
| apic:synchronization_state  | N/A                                              |
| cidr                        | 10.104.31.0/24                                   |
| created_at                  | 2019-05-22T13:38:38Z                             |
| description                 |                                                  |
| dns_nameservers             |                                                  |
| enable_dhcp                 | False                                            |
| gateway_ip                  | 10.104.31.1                                      |
| host_routes                 |                                                  |
| id                          | 107c2714-2ace-44a7-9cb0-1a7f40ba2833             |
| ip_version                  | 4                                                |
| ipv6_address_mode           |                                                  |
| ipv6_ra_mode                |                                                  |
| name                        | ext-subnet-FIP                                   |
| network_id                  | 635623ed-5dba-42ec-b3f8-3cff18f925c6             |
| project_id                 | cdeda9c674a94394a09e86a2fea498c2                   |
| revision_number            | 0                                                  |
| service_types              |                                                    |
| subnetpool_id              |                                                    |
| tags                       |                                                    |
| tenant_id                  | cdeda9c674a94394a09e86a2fea498c2                   |
| updated_at                 | 2019-05-22T13:38:38Z                               |
+----------------------------+----------------------------------------------------+
```

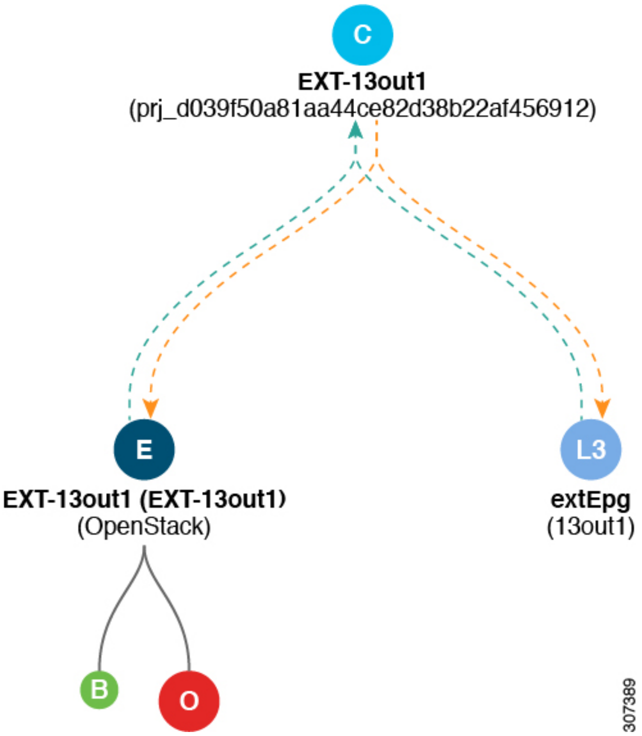


Step 4 Attach the Neutron external network to one OpenStack router as a gateway.

Example:

```
openstack router set --external-gateway external_net_name router_name
```

The command creates a contract that allows external connectivity for tenant networks attached to the OpenStack router of the external Neutron network, as shown in the following image:



CHAPTER 5

In-place Upgrades

In-place upgrades from Red Hat OSP16.2 to OSP17.1 are supported starting from Cisco ACI OpenStack Plug-in 5.2(1). The upgrade process is related to, and largely based on the procedures discussed in the FRAMEWORK FOR UPGRADES (16.2 to 17.1) Red Hat guide.

Each procedure discussed in this chapter is mandatory and required for the upgrade. The associated procedure from the FRAMEWORK FOR UPGRADES (16.2 to 17.1) guide is indicated.

Removing Custom ACI Repository

Use this procedure to remove the custom ACI repository.

Procedure

Step 1 Create the following ansible playbook to remove the ACI repo from the current yum repos on overcloud nodes.

```
---
- name: Remove ACI Repo
  hosts: overcloud
  become: yes
  tasks:
    - name: remove_acirepo
      ansible.builtin.file:
        path: /etc/yum.repos.d/ciscoaci.repo
        state: absent
```

Step 2 Run the playbook from the undercloud.

```
ansible-playbook -i ~/inventory.yaml <name of playbook file>
```

Customizing Roles

Use this procedure for customizing roles.

In the *Updating composable services in custom roles_data files* section in the *FRAMEWORK FOR UPGRADES (16.2 to 17.1)* guide, use a custom roles template to add the Cisco composable services. An ansible playbook is provided that modifies the upstream `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` file to add these roles.

Procedure

Step 1 Copy the `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` file to a private location.

Example:

```
cp /usr/share/openstack-tripleo-heat-templates/roles_data.yaml
/home/stack/templates/custom_roles_data.yaml
```

Step 2 Edit the local copy of `roles_data.yaml(custom_roles_data.yaml)` to add CiscoAciAIM and CiscoAciLldp service to the controller role and CiscoAciLldp service to the compute role.

a) Under the controller role, add the following lines:

```
- OS::TripleO::Services::CiscoAciAIM
- OS::TripleO::Services::CiscoAciLldp
- OS::TripleO::Services::CiscoAciOpflexAgent
```

b) Under the compute role, add the following line:

```
- OS::TripleO::Services::CiscoAciLldp
- OS::TripleO::Services::CiscoAciOpflexAgent
```

Note

RedHat may also instruct to add/change some roles like `OS::TripleO::Services::NovaLibvirt` for OpenStack upgrade. The above changes are in addition to all other changes that are required for FFU.

An ansible playbook is provided, that modifies the upstream `/usr/share/openstack-tripleo-heat-templates/roles_data.yaml` file to add these roles. You can skip the above step and run the playbook instead using the `sudo ansible-playbook -i ~/inventory.yaml /opt/ciscoaci-tripleo-heat-templates/tools/generate_ciscoaci_role_data.yaml` command.

Building Cisco Containers

Use this procedure for building Cisco containers.

Before proceeding to the *Upgrading a standard overcloud* section of the *FRAMEWORK FOR UPGRADES (16.2 to 17.1)* Red Hat guide , you need to build Cisco-specific containers.

Procedure

Step 1 Delete the earlier Cisco tripleo package from the undercloud, and install the new OSP17.1 RPM.

Example: For installing `tripleo-ciscoaci-17.1-1054.noarch.rpm`, use the following commands:

```
sudo yum remove tripleo-ciscoaci
sudo yum install ./tripleo-ciscoaci-17.1-1054.noarch.rpm
```

Step 2 Log in to an upstream container registry. For example, if you are using the upstream container registry from Red Hat, run the following command:

```
sudo podman login registry.connect.redhat.com
```

Step 3 After logging in to the upstream container registry, run the ACI containers build script for OSP16 using:

```
sudo /opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py -z
openstack-ciscorpms-repo-17.1-1006.tar.gz --image-tag 17.1 --pull
```

This script creates the `/home/stack/templates/ciscoaci_containers.yaml` file, which provides the mapping for the Cisco-specific or modified upstream services to their container images.

Preparing Upstream Containers

It is mandatory to create the `container-prepare-parameter.yaml` file for Overcloud adoption. For creating the `container-prepare-parameter.yaml` file, you will need multiple strategies listing for pulling containers for different openstack services during the upgrade process.

Cisco ACI neutron plugin provides the neutron-api, horizon and heat containers and it is necessary to make sure that these containers are excluded from all pull strategies, so the upgrade process does not pull the upstream versions of these containers. Ensure these containers are excluded from all pull strategies in the `container-prepare-parameter.yaml` file. Here is an example of the file:

```
parameter_defaults:
  ContainerImageRegistryCredentials:
    registry.redhat.io:
      mcohen2@cisco.com: 'XXXXXXXX'
  ContainerImagePrepare:
  - tag_from_label: '{version}-{release}'
    set:
      namespace: registry.redhat.io/rhosp-rhel9
      name_prefix: openstack-
      name_suffix: ''
      tag: '17.1'
      rhel_containers: false
      neutron_driver: ovn
      ceph_namespace: registry.redhat.io/rhceph
      ceph_image: rhceph-6-rhel9
      ceph_tag: latest
      ceph_prometheus_namespace: registry.redhat.io/openshift4
      ceph_prometheus_image: ose-prometheus
      ceph_prometheus_tag: v4.6
      ceph_alertmanager_namespace: registry.redhat.io/openshift4
      ceph_alertmanager_image: ose-prometheus-alertmanager
      ceph_alertmanager_tag: v4.6
      ceph_node_exporter_namespace: registry.redhat.io/openshift4
      ceph_node_exporter_image: ose-prometheus-node-exporter
      ceph_node_exporter_tag: v4.6
      ceph_grafana_namespace: registry.redhat.io/rhceph
      ceph_grafana_image: rhceph-6-dashboard-rhel9
      ceph_grafana_tag: latest
    push_destination: true
  MultiRhelRoleContainerImagePrepare: &id001
  - tag_from_label: '{version}-{release}'
    set:
      namespace: registry.redhat.io/rhosp-rhel9
      name_prefix: openstack-
      name_suffix: ''
      tag: '17.1'
      rhel_containers: false
      neutron_driver: ovn
      ceph_namespace: registry.redhat.io/rhceph
      ceph_image: rhceph-6-rhel9
      ceph_tag: latest
      ceph_prometheus_namespace: registry.redhat.io/openshift4
      ceph_prometheus_image: ose-prometheus
      ceph_prometheus_tag: v4.6
      ceph_alertmanager_namespace: registry.redhat.io/openshift4
      ceph_alertmanager_image: ose-prometheus-alertmanager
      ceph_alertmanager_tag: v4.6
      ceph_node_exporter_namespace: registry.redhat.io/openshift4
      ceph_node_exporter_image: ose-prometheus-node-exporter
      ceph_node_exporter_tag: v4.6
      ceph_grafana_namespace: registry.redhat.io/rhceph
      ceph_grafana_image: rhceph-6-dashboard-rhel9
      ceph_grafana_tag: latest
    push_destination: true
    excludes:
    - collectd
    - nova-libvirt
    - horizon
    - heat-engine
    - neutron-server
  - tag_from_label: '{version}-{release}'
    set:
      namespace: registry.redhat.io/rhosp-rhel8
      name_prefix: openstack-
      name_suffix: ''
      tag: '17.1'
      rhel_containers: false
      neutron_driver: ovn
      ceph_namespace: registry.redhat.io/rhceph
      ceph_image: rhceph-6-rhel9
      ceph_tag: latest
      ceph_prometheus_namespace: registry.redhat.io/openshift4
      ceph_prometheus_image: ose-prometheus
      ceph_prometheus_tag: v4.6
      ceph_alertmanager_namespace: registry.redhat.io/openshift4
      ceph_alertmanager_image: ose-prometheus-alertmanager
      ceph_alertmanager_tag: v4.6
      ceph_node_exporter_namespace: registry.redhat.io/openshift4
      ceph_node_exporter_image: ose-prometheus-node-exporter
      ceph_node_exporter_tag: v4.6
      ceph_grafana_namespace: registry.redhat.io/rhceph
      ceph_grafana_image: rhceph-6-dashboard-rhel9
      ceph_grafana_tag: latest
    push_destination: true
    includes:
    - collectd
    - nova-libvirt
  ComputeContainerImagePrepare: *id001
  ControllerContainerImagePrepare: *id001
```

Cisco ACI containers are compatible with both UBI8 and UBI9 and the same version can be deployed in multi-RHELenvironment(s).

For more details, see the *Red Hat 16.2-17.1 FFU Guide*.

OpenStack Upgrade

Upgrade the undercloud as described in the *FRAMEWORK FOR UPGRADES (16.2 TO 17.1) Red Hat guide*. Follow the Overcloud upgrade instructions till Chapter 5, in the above-mentioned guide.

For creating the **`overcloud_upgrade_prepare.sh`** file, ensure to include the following files, which are specific for Cisco ACI integration.

- Custom roles file as stated in the Customizing Roles, on page 24 section.
- Cisco ACI OSP17.1 containers mapping environment file `/home/stack/templates/ciscoaci_containers.yaml`
- Cisco ACI specific configuration environment file `ciscoaci-config.yaml`. Use compatible resources as discussed in the Install Overcloud, on page 10 procedure (step 4).

Following is an example of the `overcloud_upgrade_prepare.sh` file with Cisco specific templates:

```
openstack overcloud upgrade prepare --yes \
  --timeout 460 \
  --templates /usr/share/openstack-tripleo-heat-templates \
  --ntp-server 172.28.184.8 \
  --stack overcloud \
  -r /home/stack/templates/custom_roles_data.yaml \
  -e /home/stack/templates/upgrades-environment.yaml \
  -e /home/stack/templates/ciscoaci_containers.yaml \
  -e /home/stack/templates/ciscoaci-config.yaml \
  -e /home/stack/overcloud-deploy/overcloud/overcloud-network-environment.yaml \
  -e /home/stack/overcloud_adopt/baremetal-deployment.yaml \
  -e /home/stack/overcloud_adopt/generated-networks-deployed.yaml \
  -e /home/stack/overcloud_adopt/generated-vip-deployed.yaml \
  -e
/usr/share/openstack-tripleo-heat-templates/environments/nova-hw-machine-type-upgrade.yaml
 \
  -e /home/stack/skip_rhel_release.yaml \
  -e ~/containers-prepare-parameter.yaml
```

Proceed with the openstack upgrade process as in *FRAMEWORK FOR UPGRADES (16.2 TO 17.1) Red Hat guide.*

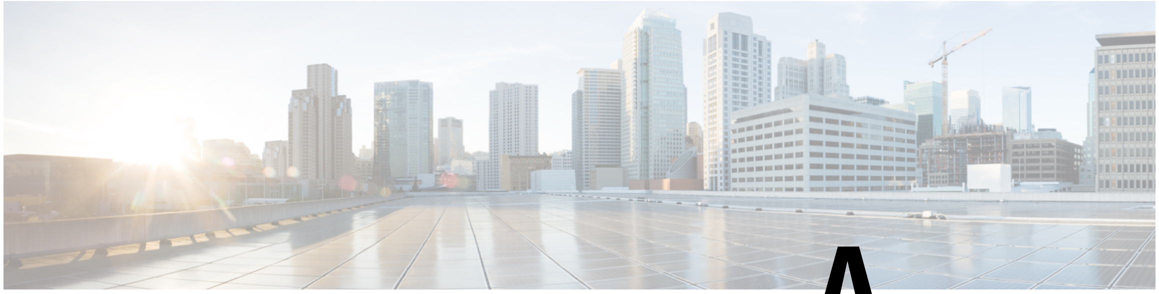
OS Upgrade

For an OS upgrade to RHEL-9, follow the instructions for OS upgrade as detailed in the *FRAMEWORK FOR UPGRADES (16.2 TO 17.1) Red Hat guide*. When creating the `overcloud_upgrade_prepare.sh` file for overcloud OS upgrade, ensure to include the ACI specific templates as discussed above.

- Custom roles file
- Cisco ACI OSP17.1 containers mapping environment file `/home/stack/templates/ciscoaci_containers.yaml`
- Cisco ACI specific configuration environment file `ciscoaci-config.yaml`
- `container-prepare-parameter.yaml` file

  The preparation of the `container-prepare-parameter.yaml` is similar to the one in the Openstack upgrade process. Ensure to exclude neutron-api, horizon and heat containers from all pull strategies in the `container-prepare-parameter.yaml` file. Here is an example of the file used during LEAPP upgrade:

```
parameter_defaults:
  ContainerImageRegistryCredentials:
    registry.redhat.io:
      mcohen2@cisco.com: 'enter password'
  ContainerImagePrepare:
  - tag_from_label: '{version}-{release}'
    set:
      namespace: registry.redhat.io/rhosp-rhel9
      name_prefix: openstack-
      name_suffix: ''
      tag: '17.1'
      rhel_containers: false
      neutron_driver: ovn
      ceph_namespace: registry.redhat.io/rhceph
      ceph_image: rhceph-6-rhel9
      ceph_tag: latest
      ceph_prometheus_namespace: registry.redhat.io/openshift4
      ceph_prometheus_image: ose-prometheus
      ceph_prometheus_tag: v4.6
      ceph_alertmanager_namespace: registry.redhat.io/openshift4
      ceph_alertmanager_image: ose-prometheus-alertmanager
      ceph_alertmanager_tag: v4.6
      ceph_node_exporter_namespace: registry.redhat.io/openshift4
      ceph_node_exporter_image: ose-prometheus-node-exporter
      ceph_node_exporter_tag: v4.6
      ceph_grafana_namespace: registry.redhat.io/rhceph
      ceph_grafana_image: rhceph-6-dashboard-rhel9
      ceph_grafana_tag: latest
    push_destination: true
    excludes:
    - horizon
    - heat-engine
    - neutron-server
```



APPENDIX A

Advanced Configuration

Advanced Configuration

For large installations using VXLAN encapsulation for Virtual Machine Manager (VMM) domains, you may need to make extra configurations.

You can configure the number of multicast groups to match the maximum number of endpoint groups (EPGs) for the host. You can also increase the maximum auxiliary memory for sockets and add extra kernel boot parameters to the compute or controller nodes.

Configure Multicast Groups and Increase Memory for Sockets

Procedure

Configure multicast groups and increase memory for sockets adding the following parameters to parameter_defaults in the deployment template:

Example:

```
parameter_defaults:
  ControllerParameters:
      ExtraSysctlSettings:
         net.ipv4.igmp_max_memberships:
             value: 4096
         net.core.optmem_max:
             value: 1310720
   ComputeParameters:
      ExtraSysctlSettings:
         net.ipv4.igmp_max_memberships:
             value: 1024
```

The IGMP maximum memberships value should be greater than or equal to the number of Neutron networks that the host has Neutron ports on. For example, if a compute host has 100 instances, and each instance is on a different Neutron network, then you must set this number to at least 100. Controller hosts running the `neutron-dhcp-agent` must set this value to match the number of Neutron networks managed by that agent. This means that this number probably must be higher on controller hosts than compute hosts.

Add Extra Kernel Boot Parameters

You add extra kernel boot parameters to the compute or controller nodes by modifying the `resource_registry` and the `parameter_defaults` files.

Procedure

Step 1 Modify the `resource_registry` file.

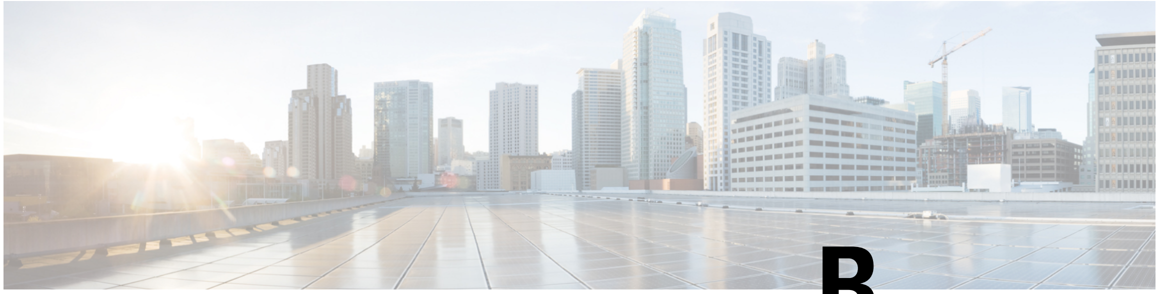
Example:

```
resource_registry:
     OS::TripleO::Compute::PreNetworkConfig:
/usr/share/openstack-tripleo-heat-templates/extraconfig/pre_network/host_config_and_reboot.yaml
```

Step 2 Modify the `parameter_defaults` file.

Example:

```
parameter_defaults:
    ComputeParameters:
        KernelArgs: "intel_iommu=on iommu=pt default_hugepagesz=1GB hugepagesz=1G
hugepages=60"
```



APPENDIX B

Reference Information

Configure Hierarchical Port Binding

This section describes configuring the Single Root I/O Virtualization (SR-IOV) and other VLAN-based ml2 mechanism agents to work with OpFlex plug-in. The configuration is accomplished by using Hierarchical Port Binding (HPB) and should work without any special modification to the configuration. Here are the basic steps that you need to configure OpFlex with SR-IOV.

When using HPB, data path connectivity in Cisco Application Centric Infrastructure (ACI) is accomplished by creating static VLAN bindings to the EPGs for networks created by OpenStack. There maybe other configuration required for data path, for example, setting up VLAN on SR-IOV NIC or configuring OVS (or a load balancer in case of LBaaS). This is done by the third-party agent or mechanism driver (for example, sriovnicswitch).

How to create these assets:

Before you begin

To configure the data path using static VLAN bindings, ensure that the plug-in 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)
- You need this information only if the deployment uses multiple physnets.

Procedure

Before deploying OpenStack Platform Overcloud, make sure you have one Physical Domain (physdom) created per each physnet required. Add pdom_ prefix to the name of physical domain created. For example for physnet1 create pdom_physnet1, and attach the right VLAN pool.

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

Example:

```
NeutronPhysicalDevMappings: physnet1:ens11,physnet2:ens7,physnet3:ens9
NeutronNetworkVLANRanges:physnet1:1200:1250,physnet2:1251:1300,physnet3:1301:1350
ACIMechanismDrivers: 'sriovnicswitch,apic_aim'
ACIHostLinks: '{"101": [{"host01|ens11": "1/14"}], "102": [{"host02|ens9": "1/14"}]}'
```

Parameters for the Cisco ACI Environment

The following table provides information about parameters that are required to configure the Cisco Application Centric Infrastructure (ACI) environment.

Parameter	Details
NeutronCorePlugin	• **Value:** 'ml2plus' • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None
NeutronServicePlugins	• **Value:** 'group_policy,ncp,apic_aim_l3' • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None
NeutronEnableIsolatedMetadata	• **Value:** true • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** Must be set to true

Parameter	Details
NeutronEnableForceMetadata	• **Value:** true • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** Must be set to true
ACIYumRepo	• **Value:**: http://undercloud pxe network ipaddress:8787/v2/__acirepo • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None
ACIApicHosts	• **Value:** Cisco Application Policy Infrastructure Controller (APIC) name and addresses • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None
ACIApicUsername	• **Value:** Username with administrative privileges • **Default:** admin • **Mandatory or Optional:** Optional • **Comments:** None
ACIApicPassword	• **Value:** Password • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None **Note** Do not provide this parameter if certificate-based authentication is used.
ACIMechanismDrivers	• **Value:** 'apic_aim' • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** Add extra drivers—for example, for Open vSwitch when using neutron ovs agent or for sriovnicswitch when using sriov

Parameter	Details
ACIApicEntityProfile	• **Value:** The Cisco ACI entity profile that has been preprovisioned on Cisco ACI • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None
ACIApicInfraVlan	• **Value:** The Cisco ACI fabric infra VLAN • **Default:** 4093 • **Mandatory or Optional:** Optional • **Comments:** Contact the Cisco ACI administrator for the correct value
ACIApicInfraSubnetGateway	• **Value:** The Cisco ACI infra subnet gateway • **Default:** 10.0.0.30 • **Mandatory or Optional:** Optional • **Comments:** Contact the Cisco ACI administrator for the correct value
ACIApicInfraAnycastAddr	• **Value:** The Cisco ACI anycast address • **Default:** 10.0.0.32 • **Mandatory or Optional:** Optional • **Comments:** Contact the Cisco ACI administrator for the correct value
ACIUseLldp	• **Value:** true or false • **Default:** true • **Mandatory or Optional:** Optional • **Comments:** If set to false, set CiscoAciLldp service to `OS::Hat:None`
ACIOpflexUplinkInterface	• **Value:** Interface name that is connected to the Cisco ACI leaf switch • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** Actual interface name—for example, enp8s0

Parameter	Details
ACIOpflexEncapMode	• **Value:** vxlan or vlan • **Default:** vxlan • **Mandatory or Optional:** Optional • **Comments:** None
ACIOpflexVlanRange	• **Value:** *starting_vlan*:*ending_vlan* • **Default:** None • **Mandatory or Optional:** Mandatory if ACIOpflexEncapMode is set to vlan • **Comments:** None
ACIOpflexInterfaceType	• **Value:** linux or ovs • **Default:** linux • **Mandatory or Optional:** Optional • **Comments:** Set this value to 'ovs' when planning to deploy an "OpenShift on OpenStack" nested installation. The setting causes the OpFlex interface to be created on the ovs switch.
ACIOpflexInterfaceMTU	• **Value:** Intended MTU size • **Default:** 1500 • **Mandatory or Optional:** Optional • **Comments:** Use this parameter to set the MTU for the OpFlex interface. This must be set to 8000 for installing OpenShift on OpenStack.
NeutronPluginMl2PuppetTags	• **Value:** 'neutron_plugin_ml2,neutron_plugin_cisco_aci' • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** None
NeutronNetworkVLANRanges	• **Value:** *physnet*:*starting vlan*:*ending vlan* (For example, physnet1:1100:1150,physnet2:1201:1211) • **Default:** None • **Mandatory or Optional:** Mandatory when using neutron ovs agent, recommended for use of VLAN type networks and HPB. • **Comments:** None

Parameter	Details
AciPhysDomMappings	• **Value**: For example: 'physnet0:my_pdom0, physnet1:my_pdom1' • **Default**: None • **Mandatory or Optional**: Optional • **Comments**: List of *physical_network*:*ACI Physdom*. By default each physnet maps to a precreated ACI physdom with **pdom**_*physnet_name*. For example **physnet0** will map to physdom named **pdom_phynet0**. This parameter allows the user to override the mapping.
NeutronBridgeMappings	• **Value:** For example: 'physnet1:br-ex,physnet2:br-ex' • **Default:** None • **Mandatory or Optional:** Mandatory when using neutron ovs agent • **Comments:** Physnets should match as provided in NeutronBridgeMappings
AciTenantNetworkType	• **Value:** vlan • **Default:** None • **Mandatory or Optional:** Mandatory when using neutron ovs agent • **Comments:** None
AciOpenvswitch	• **Value:** true or false • **Default:** false • **Mandatory or Optional:** Set to true when using neutron ovs agent • **Comments:** None
NeutronOVSFirewallDriver	• **Value:** 'neutron.agent.linux.iptables_firewall.OVSHybridIptablesFirewallDriver' • **Default:** None • **Mandatory or Optional:** Mandatory • **Comments:** Set to the value shown when using neutron ovs agent

Parameter	Details
ACIHostLinks	• **Value:** For example: '{"101":{"ha.dom":"1/1", "hb.dom":"1/2"}, "102":{"hc.dom":"1/1"} }' • **Default:** None • **Mandatory or Optional:** Mandatory when using neutron ovs agent and not using lldp agent • **Comments:** Describes the host connections to switches in JSON format. In the example, The ha.dom host is connected to port 1/1 of switch ID 101, the hb.dom host is connected to port 1/2 of switch ID 101, and the hc.dom is connected to port 1/1 of switch ID 102.
NeutronPhysicalDevMappings	• **Value:** • **Default:** None • **Mandatory or Optional:** Optional • **Comments:**
NeutronPhysicalDevMappings	• **Value:** For example: physnet1:eth1,physnet2:eth2 • **Default:** None • **Mandatory or Optional:** Optional • **Comments:** You must set this parameter when you want to map a particular interface to a specific physnet
ACIApicCertName	• **Value:** Name of the Cisco APIC cert User (used for certificate-based authentication) • **Type:** String • **Default:** None • **Mandatory or Optional:** Optional
ACIApicPrivateKey	• **Value:** Private key for the cert User • **Type:** String • **Default:** None • **Mandatory or Optional:** Optional

Parameter	Details
AciKeystoneNotificationPurge	• **Value:** True or False • **Type:** Boolean • **Default:** False • **Comment:** Enables the automatic purge of Cisco APIC tenants when the project is deleted in OpenStack.
NeutronPluginExtensions	• **Value:** Comma-separated list of enabled extension plugins. • **Default:** apic_aim, port_security • **Mandatory or Optional:** Optional • **Comment:**Recommended values when parameter is explicitly configured are - apic_aim, port_security, qos.
NeutronMechanismDrivers	• **Value:** 'apic_aim' • **Default:** ovn • **Mandatory or Optional:** Mandatory (when Octavia is deployed) • **Comment:**When deploying Octavia, by default **ovn** provider is deployed. In Octavia deployment, NeutronMechanismDrivers is taken into consideration to define provider. In Amphora deployment, **ovn** must be explicitly set.
ACIVpcPairs	• **Value**: For example “101:102,103:104” • **Default**: N/A • **Mandatory or Optional**: Required only when interface bonding/VPC is used • **Comment**:When deploying using VPC/bonding, it is a comma-separated list of switch pairs used for VPC.
AciVmmMcastRanges	• **Value**: For example: “225.5.1.1:225.5.255.255” • **Default**: None • **Mandatory or Optional**: Mandatory • **Comment**: Should be unique for each OpenStack cluster sharing the same ACI fabric.

Parameter	Details
AciVmmMulticastAddress	• **Value**: For example “225.5.10.3” • **Default**: None • **Mandatory or Optional**: Mandatory • **Comment**: This is a per-VMM domain multicast address for BUM traffic (one per VMM domain). This should not overlap with other OpenStack installations in the same fabric.
ACIScopeNames	• **Value**: True or False • **Default**: False • **Mandatory or Optional**: Optional • **Comment**: This is needed to ensure that each OpenStack installation on the same fabric does not create resources with conflicting names. Resources are scoped by this name to avoid namespace conflicts.

Example of Resources Declaration

The following is a full example of the Cisco Application Centric Infrastructure (ACI) resources declaration (`ciscoaci-env.yaml`):

The following is a full example of the Cisco Application Centric Infrastructure (ACI) resources declaration (`ciscoaci-env.yaml`). The example provided is for Cisco ACI Release 5.2(1) or later. You declare resources for the Cisco ACI environment when you install the Overcloud. See the procedure Install Overcloud, on page 10 in this guide.

```
# A Heat environment file which can be used to enable a Neutron Cisco Aci backend on the
controller, configured via puppet
resource_registry:

  #controller
  OS::TripleO::ControllerExtraConfigPre: /opt/ciscoaci-tripleo-heat-templates//nodepre.yaml

  OS::TripleO::Services::NeutronOvsAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron_opflex/neutron-opflex-agent-container-puppet.yaml

  OS::TripleO::Services::CiscoAciOpflexAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/opflex/opflex-agent-container-puppet.yaml
  OS::TripleO::Services::NeutronMl2PluginBase:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron/neutron-ml2-ciscoaci.yaml
  OS::TripleO::Services::CiscoAciAIM:
/opt/ciscoaci-tripleo-heat-templates/deployment/aciaim/cisco-aciaim-container-puppet.yaml
  OS::TripleO::Services::NeutronMetadataAgent:
/usr/share/openstack-tripleo-heat-templates/deployment/neutron/neutron-metadata-container-puppet.yaml

  OS::TripleO::Services::NeutronDhcpAgent:
/usr/share/openstack-tripleo-heat-templates/deployment/neutron/neutron-dhcp-container-puppet.yaml

  #compute
  OS::TripleO::ComputeExtraConfigPre: /opt/ciscoaci-tripleo-heat-templates//nodepre.yaml
  OS::TripleO::Services::ComputeNeutronOvsAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/neutron_opflex/neutron-opflex-agent-container-puppet.yaml

  OS::TripleO::Services::ComputeCiscoAciOpflexAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/opflex/opflex-agent-container-puppet.yaml
  OS::TripleO::Services::ComputeNeutronMetadataAgent:
/opt/ciscoaci-tripleo-heat-templates/deployment/compute_neutron_metadata/compute-neutron-metadata.yaml


  OS::TripleO::Services::CiscoAciLldp:
/opt/ciscoaci-tripleo-heat-templates/deployment/lldp/cisco_lldp.yaml
  OS::TripleO::NodeUserData:
/usr/share/openstack-tripleo-heat-templates/firstboot/userdata_root_password.yaml

  OS::TripleO::Services::OVNDBs: OS::Heat::None
  OS::TripleO::Services::OVNController: OS::Heat::None
  OS::TripleO::Services::OVNMetadataAgent: OS::Heat::None
  OS::TripleO::Services::ComputeNeutronL3Agent: OS::Heat::None
  OS::TripleO::Services::NeutronL3Agent: OS::Heat::None

parameter_defaults:

  DockerInsecureRegistryAddress: ["fab205-ucloud-17.ctlplane.localdomain:8787",
"1.100.1.1:8787", "172.28.184.248"]
  NeutronCorePlugin: 'ml2plus'
  NeutronServicePlugins: 'group_policy,ncp,apic_aim_l3'
  NeutronEnableIsolatedMetadata: true
  NeutronEnableForceMetadata: true
  NeutronPluginExtensions: apic_aim,port_security,dns
  NeutronPhysicalDevMappings: physnet1:eth1,physnet2:eth2
  EnablePackageInstall: true
  ACIScopeNames: true
  ACIApicHosts: 10.30.120.148
  ACIApicUsername: admin
  ACIApicPassword: noir0123
  ACIApicSystemId: fab205
  ACIMechanismDrivers: 'apic_aim'
  ACIApicEntityProfile: sauto_fab205_aep
  ACIApicInfraVlan: 4093
  ACIApicInfraSubnetGateway: 10.0.0.30
  ACIApicInfraAnycastAddr: 10.0.0.32
  ACIOpflexUplinkInterface: bond1
  ACIOpflexEncapMode: vxlan
  NeutronNetworkVLANRanges: physnet1:1701:1750
  ACIOpflexVlanRange: 701:750
  HeatEnginePluginDirs:
/usr/lib64/heat,/usr/lib/heat,/usr/local/lib/heat,/usr/local/lib64/heat,/usr/lib/python2.7/site-packages/gbpautomation/heat


  ACIVpcPairs: 101:102
  NeutronPluginMl2PuppetTags: 'neutron_plugin_ml2,neutron_plugin_cisco_aci'

  AciVmmMcastRanges: 225.5.1.1:225.5.255.255
  AciVmmMulticastAddress: 225.5.10.3
  ACIYumRepo: http://1.100.1.1:8787/v2/__acirepo
```

Note If you are deploying a release prior to Cisco ACI Release 5.2(1), you need to make the following changes in the above example:

- Remove the definition for `OS::TripleO::Services::CiscoAciOpflexAgent`.
- Change the `OS::TripleO::Services::NeutronOvsAgent` and `OS::TripleO::Services::ComputeNeutronOvsAgent` to reference the `/opt/ciscoaci-tripleo-heat-templates/deployment/opflex/opflex-agent-container-puppet.yaml` template.

Examples of Creating Host Reports

During troubleshooting, you might need to collect host reports from the OpenStack cluster. You do it using the provided playbook `/opt/ciscoaci-tripleo-heat-templates/tools/report.yaml`. This section provides example of using the host report playbook.

-
  ```
  ansible-playbook -i
  overcloud-deploy/overcloud/tripleo-ansible-inventory.yaml
  /opt/ciscoaci-tripleo-heat-templates/tools/report.yaml
  ```

  This example collects data from all nodes and creates the file `/home/stack/overcloud_aci_report.tgz`.

-
  ```
  ansible-playbook -i
  overcloud-deploy/overcloud/tripleo-ansible-inventory.yaml
  /opt/ciscoaci-tripleo-heat-templates/tools/report.yaml -e
  '{"limit_flavors":['control'], "dest_file":/tmp/abc}
  ```

  This example limits the repot to controllers and changes the default output file.

-
  ```
  ansible-playbook -i
  overcloud-deploy/overcloud/tripleo-ansible-inventory.yaml
  /opt/ciscoaci-tripleo-heat-templates/tools/report.yaml -e
  '{"limit_hosts":[overcloud-controller-0, overcloud-controller-2]}'
  ```

  This example limits the report collection to the hosts specified. You can club `"limit_flavors"` and `"limit_hosts"` to further filter the nodes from which to collect data.

Deploying with TLS

Deploying Red Hat OpenStack 17.1 with Transport Layer Security (TLS) is a supported configuration. To enable TLS on OpenStack endpoints, follow the instructions in *Advanced Overcloud Customization* on the Red Hat website.

To enable TLS between AIM and Cisco Application Policy Infrastructure Controller (APIC), follow the cert base authentication procedure described in step 4 of Install Overcloud, on page 10 in this guide.

Cleaning up Cisco ACI Container Images

If you run the cisco container image generation `/opt/ciscoaci-tripleo-heat-templates/tools/build_openstack_aci_containers.py` script multiple times (even for minor updates), you will continue to have old Cisco ACI container images in your repository. Use this procedure to clear old container images.

Procedure

Step 1 Find the tag for the latest built images.

You can check the file `ciscoaci_containers.yaml` to find out the latest image tags. In the below example, the tag is 1614292118.

```
[stack@director16 ~]$ cat templates/ciscoaci_containers.yaml
parameter_defaults:
ContainerHorizonImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-horizon-ciscoaci:1614292118
ContainerHeatEngineImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-heat-engine-ciscoaci:1614292118
ContainerNeutronApiImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-neutron-server-ciscoaci:1614292118
ContainerNeutronConfigImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-neutron-server-ciscoaci:1614292118
ContainerCiscoLldpImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-ciscoaci-lldp:1614292118
ContainerCiscoAciAimImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-ciscoaci-aim:1614292118
ContainerCiscoAciAimConfigImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-ciscoaci-aim:1614292118
ContainerOpflexAgentImage:
director16.ctlplane.localdomain:8787/ciscoaci/openstack-ciscoaci-opflex:1614292118
[stack@director16 ~]$
```

Step 2 To identify old images in your repository, use the **sudo openstack tripleo container image list** command.

Example below displays the new and older images. With reference to the example in Step 1, the new images are indicated with 1614292118, the others are older images.

```
[stack@director16 ~]$ sudo openstack tripleo container image list|grep cisco
| docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-horizon-ciscoaci:1613593371
  |
| docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-horizon-ciscoaci:1613614575
  |
| docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-horizon-ciscoaci:1614292118
  |
|
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-heat-engine-ciscoaci:1613593371
  |
|
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-heat-engine-ciscoaci:1613614575
  |
|
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-heat-engine-ciscoaci:1614292118
  |
|
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-neutron-server-ciscoaci:1613593371
 |
|
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-neutron-server-ciscoaci:1613614575
 |
|
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-neutron-server-ciscoaci:1614292118
```

With reference to the example in Step 1, the new images are indicated with 1614292118, the others are older images.

Step 3 To delete old images from your repository, use the **sudo openstack tripleo container image delete** command.

```
[stack@director16 ~]$ sudo openstack tripleo container image delete \
docker://director16.ctlplane.localdomain:8787/ciscoaci/openstack-horizon-ciscoaci:1613593371
```