Cisco ACI with OpenStack Deployment Guide for Ubuntu Using Juju Charms

First Published: 2018-07-19

Last Modified: 2019-02-21

CONTENTS

Preface

This preface includes the following sections:

-
-
-
-

Audience

This guide is intended for network and systems administrators who configure and maintain the Application Centric Infrastructure fabric.

Document Conventions

Command descriptions use the following conventions:

Convention	Description
bold	Bold text indicates the commands and keywords that you enter literally as shown.
Italic	Italic text indicates arguments for which the user supplies the values.
[x]	Square brackets enclose an optional element (keyword or argument).
[x | y]	Square brackets enclosing keywords or arguments separated by a vertical bar indicate an optional choice.
{x | y}	Braces enclosing keywords or arguments separated by a vertical bar indicate a required choice.
[x {y | z}]	Nested set of square brackets or braces indicate optional or required choices within optional or required elements. Braces and a vertical bar within square brackets indicate a required choice within an optional element.

Convention	Description
variable	Indicates a variable for which you supply values, in context where italics cannot be used.
string	A nonquoted set of characters. Do not use quotation marks around the string or the string will include the quotation marks.

Examples use the following conventions:

Convention	Description
screen font	Terminal sessions and information the switch displays are in screen font.
boldface screen font	Information you must enter is in boldface screen font.
italic screen font	Arguments for which you supply values are in italic screen font.
< >	Nonprinting characters, such as passwords, are in angle brackets.
[ ]	Default responses to system prompts are in square brackets.
!, #	An exclamation point (!) or a pound sign (#) at the beginning of a line of code indicates a comment line.

This document uses the following conventions:

Note Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.

Caution Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.

Warning IMPORTANT SAFETY INSTRUCTIONS

This warning symbol means danger. You are in a situation that could cause bodily injury. Before you work on any equipment, be aware of the hazards involved with electrical circuitry and be familiar with standard practices for preventing accidents. Use the statement number provided at the end of each warning to locate its translation in the translated safety warnings that accompanied this device.

SAVE THESE INSTRUCTIONS

Related Documentation

To learn more about the features and operation of the APIC, see the *Cisco Application Centric Infrastructure Fundamentals*.

Documentation Feedback

To provide technical feedback on this document, or to report an error or omission, please send your comments to apic-docfeedback@cisco.com. We appreciate your feedback.

New and Changed Information

This chapter contains the following sections:

New and Changed Information

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

Table 1: New Features and Changed Behavior

Cisco APIC Release Version	Feature	Description	Where Documented
3.1(2)	--	This guide was released.	--
3.1(2)	--	Added the configuring Hierarchical Port Binding (HPB) section.	For more information, see the Configuring Hierarchical Port Binding, on page 11.

Install OpFlex Plugin on Ubuntu with Juju

This chapter contains the following sections:

Prerequisites

This section describes the prerequisites for installing opflex plugin on Ubuntu with Juju.

- Working knowledge of Ubuntu Linux

- Experience in deployment of OpenStack using MAAS and Juju

- Some experience in APIC UI maybe required

- Working ACI Fabric

- MAAS setup on Ubuntu Xenial with Juju2

- Minimum of 4 servers with at least one fabric interface

Figure 1: Typical Network Topology
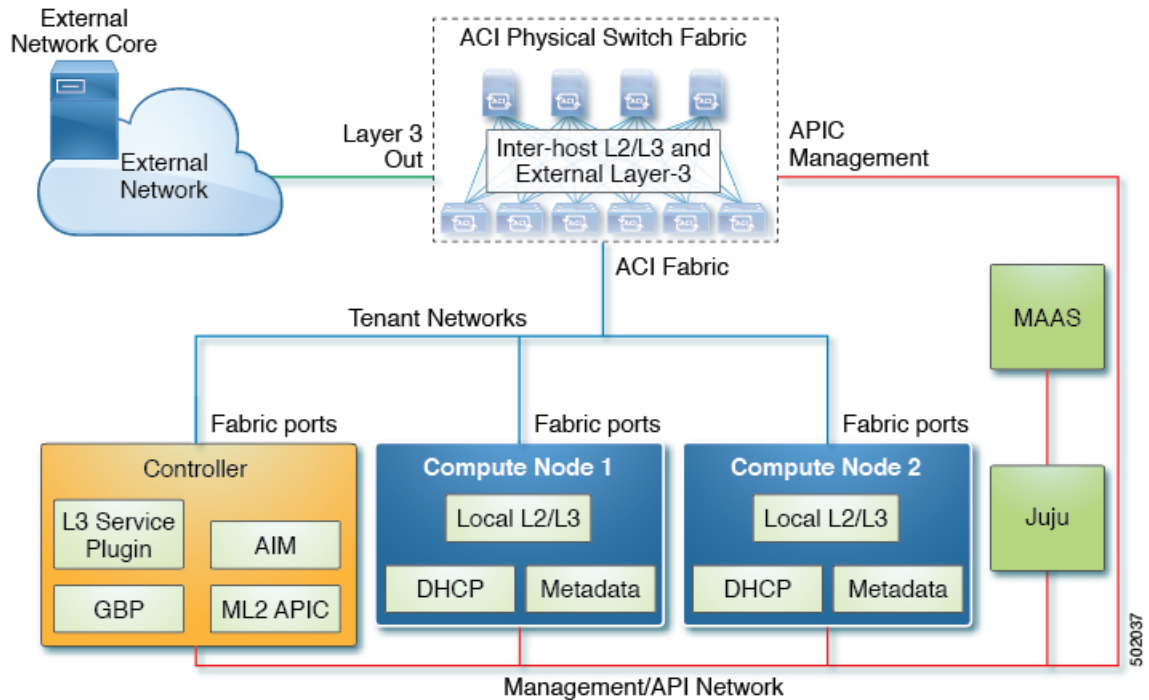
Server Preparation

This section describes the server preparation.

- The OpenStack nodes need to be connected to the leaf switches either in direct mode or VPC/bonded mode.

- The MTU of the interfaces should be greater than 1600.

- The bonding should be setup using 802.3ad

Site-Preparation

This section describes how to prepare the site by downloading opflex released archive file from the software download link and setting up an apt repository on the repo server. Charms support both signed and un-signed repositories.

Step 1   Take note of the following:

- APIC IP addresses

- APIC credentials

- AAEP created for deployment

For example, my-aep in the .

- Infra VLAN ID

- `opflex_peer_ip` and `opflex_remote_ip` parameters

  The `opflex_remote_ip` address matches the anycast ip address assigned to interface Loopback 1023 on the leaf switches. The `opflex_peer_ip` address for the OpFlex peer is the anycast IP address assigned to the SVI of the infra VLAN on the leaf switches.

- An ID string to identify this OpenStack instance (APIC_SYSTEM_ID)

- Server interface name that is connected to fabric.

  For example, `'eth2'` or `'bond0'`.

Step 2   Install, configure MAAS and Juju.

For more information, see the Ubuntu documentation.

Step 3   Download the **Debian packages for Openstack** *<Release_Name>* **ACI plugins** gzip file matching your OpenStack, ACI release and create a Debian/Ubuntu repo.

a) Go to the Software Download link:

   https://software.cisco.com/download/home/285968390/type

b) Click **APIC OpenStack and Container Plugins**.

c) Choose the Debian packages for Openstack *<Release_Name>* ACI plugins gzip file matching your OpenStack, ACI release and click the download icon.

Step 4   Untar and extract the files.

Step 5   Take note of the repo URL. The "`deb http://1.11.1.1/ubuntu ocata32 main`" repo URL is used as an example in the .

Requirements for the config `juju-aci-ocata.yaml` file:

- Nodes are connected to leaf in a vPC/Linux bonding configuration. The bonded device name is "`bond0`". If it is a single interface, replace it with the interface name. For example, "`eth1`", "`ens9`".

- APIC is configured with a default VLAN 3901 (infra VLAN). Replace this number with whatever VLAN is configured on APIC.

- The AAEP created for deployment is `my-aep`.

- The URL for the created package repo is "`deb http://1.11.1.1/ubuntu ocata32 main`".

- APIC_SYSTEM_ID is "`juju2-ostack`".

Installation

This section describes how to install the opflex plugin on Ubuntu with Juju charm by downloading from the software download link, untar and extract the files on the Juju server under a directory.

Step 1 Download the **Juju charms for ACI plugins for Openstack** *<Release_Name>* gzip file matching your OpenStack and ACI release.

    a) Go to the Software Download link:

       https://software.cisco.com/download/home/285968390/type

    b) Click **APIC OpenStack and Container Plugins**.

    c) Choose the **Juju charms for ACI plugins for Openstack** *<Release_Name>* matching your OpenStack, ACI release and click the download icon.

Step 2 Untar and extract the files into a directory on the Juju server. For example, `/home/ubuntu/charms/xenial`.

    Note To support Openstack deployment on ACI fabric using Juju, the following charms from upstream had to be modified:

        • charm-neutron-gateway

        • charm-neutron-api

    In addition, the following new charms were developed:

        • charm-neutron-api-plugin-aci (runs as sub-ordinate to neutron-api)

        • charm-neutron-aci-opflex (runs as sub-ordinate to nova-compute and charm-neutron-gateway)

        • charm-openstack-dashboard-plugin-gbp (runs as sub-ordinate to openstack-dashboard)

    Additionally, when using the ACI plugin the security groups are implemented in vswitch. For the best performance, nova should be configured with NoopFirewallDriver as below. Currently, Ubuntu Juju charm does not allow that to be configured and adds an extra bridge as it uses the OVSHybridFirewallDriver and has some performance impact. You can ignore the extra bridge as it does not create or correct any issues.

Step 3 Create a configuration juju-aci-ocata.yaml file and place it in a directory. For example, `/home/ubuntu`.

    For more information, see Example of the Configuration juju-aci-ocata.yaml File

Step 4 Deploy the charms and add relations. Below is a sample script for deployment of the ACI charms:

```
juju bootstrap maas maas-controller --config=config.yaml
juju add-machine -n 4

CONFIG=/home/ubuntu/juju-aci-ocata.yaml

juju deploy --config=$CONFIG cs:trusty/mysql --to lxd:1
juju config mysql max-connections=1300
juju deploy --config=$CONFIG rabbitmq-server --to lxd:1


sleep 120

juju deploy --config=$CONFIG keystone --to lxd:1
juju add-relation keystone:shared-db mysql:shared-db

juju deploy --config=$CONFIG glance --to lxd:1
juju add-relation glance:identity-service keystone:identity-service
juju add-relation glance:shared-db mysql:shared-db

juju deploy /home/noiro/charms/xenial/charm-neutron-api --series xenial
--config=$CONFIG neutron-api --to 1
```

```
juju add-relation neutron-api:amqp rabbitmq-server:amqp
juju add-relation neutron-api:identity-service keystone:identity-service
juju add-relation neutron-api:shared-db mysql:shared-db

juju deploy /home/noiro/charms/xenial/charm-neutron-api-plugin-aci --series xenial
--config=$CONFIG neutron-api-plugin-aci
juju add-relation neutron-api-plugin-aci neutron-api
juju add-relation neutron-api-plugin-aci:amqp rabbitmq-server:amqp
juju add-relation neutron-api-plugin-aci:shared-db mysql:shared-db


juju deploy /home/noiro/charms/xenial/charm-neutron-gateway --series xenial
--config=$CONFIG neutron-gateway --to 3
juju add-relation neutron-gateway:amqp rabbitmq-server:amqp
juju add-relation neutron-gateway:neutron-plugin-api neutron-api:neutron-plugin-api

juju deploy --config=$CONFIG nova-cloud-controller --to lxd:1
juju add-relation nova-cloud-controller:amqp rabbitmq-server:amqp
juju add-relation nova-cloud-controller:identity-service keystone:identity-service
juju add-relation nova-cloud-controller:image-service glance:image-service
juju add-relation nova-cloud-controller:neutron-api neutron-api:neutron-api
juju add-relation nova-cloud-controller:shared-db mysql:shared-db
juju add-relation nova-cloud-controller:quantum-network-service
neutron-gateway:quantum-network-service

juju deploy --config=$CONFIG nova-compute --to 2
juju add-relation nova-compute:amqp rabbitmq-server:amqp
juju add-relation nova-compute:cloud-compute nova-cloud-controller:cloud-compute
juju add-relation nova-compute:image-service glance:image-service
juju add-relation nova-compute:shared-db mysql:shared-db

juju deploy /home/noiro/charms/xenial/charm-neutron-aci-opflex --config=$CONFIG neutron-aci-opflex
juju add-relation neutron-aci-opflex:neutron-plugin-api neutron-api:neutron-plugin-api
juju add-relation neutron-aci-opflex:neutron-plugin nova-compute:neutron-plugin
juju add-relation neutron-aci-opflex:amqp rabbitmq-server:amqp
juju add-relation neutron-aci-opflex:quantum-network-service neutron-gateway:quantum-network-service

juju deploy --config=$CONFIG openstack-dashboard --to 0
juju add-relation openstack-dashboard:identity-service keystone:identity-service

juju deploy /home/noiro/charms/xenial/charm-openstack-dashboard-plugin-gbp --series xenial
--config=$CONFIG openstack-dashboard-plugin-gbp

juju add-relation openstack-dashboard-plugin-gbp openstack-dashboard
```

Example of the Configuration juju-aci-ocata.yaml File

This is an example of the configuration `juju-aci-ocata.yaml` file.

```
mysql:
  max-connections: 1300
keystone:
  openstack-origin: 'cloud:xenial-ocata'
  admin-password: 'noir0123'
  log-level: 'DEBUG'
  verbose: true
  debug: true
nova-cloud-controller:
  openstack-origin: 'cloud:xenial-ocata'
```

```
      network-manager: Neutron
      console-access-protocol: novnc
   neutron-gateway:
     plugin: aci
     openstack-origin: 'cloud:xenial-ocata'
     data-port: 'br-data:bond0'
     vlan-ranges: physnet1:250:300
     enable-isolated-metadata: true
     enable-metadata-network: true
     aci-repo: 'deb http://1.11.1.1/ubuntu ocata32 main'
     aci-apic-system-id: juju2-ostack
     aci-encap: vxlan
     aci-uplink-interface: bond0
     aci-infra-vlan: 3901
   neutron-api:
     openstack-origin: 'cloud:xenial-ocata'
     neutron-plugin: aci
     neutron-security-groups: true
     #vlan-ranges: physnet1:250:300
   neutron-aci-opflex:
     aci-repo: 'deb http://1.11.1.1/ubuntu ocata32 main'
     aci-apic-system-id: juju2-ostack
     aci-encap: vxlan
     aci-uplink-interface: bond0
     aci-infra-vlan: 3901
   rabbitmq-server:
   ## Cinder is deployed in two parts: one for the API and scheduler
   ## (which can live in a container), one for the volume service (which
   ## cannot, at least not for the LVM/iSCSI backend)
   cinder-api:
     openstack-origin: 'cloud:xenial-ocata'
     enabled-services: api,scheduler
   cinder-volume:
     openstack-origin: 'cloud:xenial-ocata'
     enabled-services: volume
     # Adjust this to match the block device on your volume host
     block-device: vdb
   glance:
     openstack-origin: 'cloud:xenial-ocata'
   heat:
     openstack-origin: 'cloud:xenial-ocata'
   mysql:
   openstack-dashboard:
     openstack-origin: 'cloud:xenial-ocata'
     webroot: /
     vip_iface: bond0
   nova-compute:
     openstack-origin: 'cloud:xenial-ocata'
     virt-type: kvm
   ##
   ## ACI
   ##
   neutron-api-plugin-aci:
     aci-repo: 'deb http://1.11.1.1/ubuntu ocata32 main'
     aci-apic-hosts: 10.3.12.1
     aci-apic-username: admin
     aci-apic-password: k%jiou^
     aci-apic-entity-profile: my-aep
     aci-apic-system-id: juju2-ostack
     aci-encap: vxlan
     aci-vlan-ranges: 250:300
     aci-connection-json: '{"101": ["srv1:vpc-1-25/101-102-1-25", "srv2:vpc-1-26/101-102-1-26",

     "srv3:vpc-1-27/101-102-1-27", "srv4:vpc-1-28/101-102-1-28", "srv5:vpc-1-29/101-102-1-29"],
```

```
"102": ["srv1:vpc-1-25/101-102-1-25", "srv2:vpc-1-26/101-102-1-26",
"srv3:vpc-1-27/101-102-1-27",
"srv4:vpc-1-28/101-102-1-28", "srv5:vpc-1-29/101-102-1-29"]}'
  aci-vpc-pairs: '101:102'
#
openstack-dashboard-plugin-gbp:
  aci-repo: 'deb http://1.11.1.1/ubuntu ocata32 main'
```

Adding an OpenStack External Network

This section describes how to add an OpenStack external network.

As opposed to the Classical mode, there is no need to specify any L3Out or host_pool_cidr config parameters anywhere in the ostack controller configuration.

There is a change in the command for configuring external-network (new extensions have been added) in the OpenStack to correspond to the above pre-existing L3Out and its EPG. The configuration requires the DN of the EPG for the specific the L3Out in APIC. It can be done in two ways:

Before you begin

- An L3Out on the APIC must be pre-created along with its endpoint group (EPG).

Step 1 Go to the APIC and get the DN of the external network using the API-Inspector:
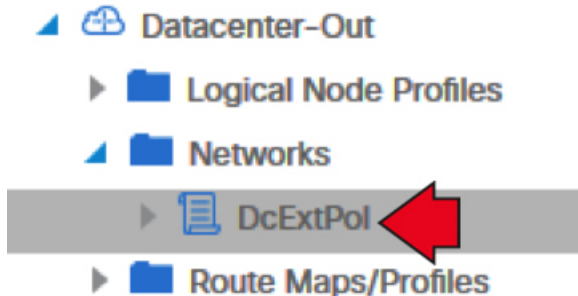
Step 2 Use the **aimctl manager** command on the Openstack-Controller to get the DN as shown below:

Example:

```
aimctl manager external-network-find
+--------------+---------------+-----------+
| tenant_name  | l3out_name    | name      |
|--------------+---------------+-----------|
| common       | Datacenter-Out | DcExtPol  |
| common       | Management-Out | MgmtExtPol |
| common       | default       | MgmtNet   |
+--------------+---------------+-----------+


 aimctl manager external-network-get common Datacenter-Out DcExtPol
+------------------------+--------------------------------------------------+
|Property                |Value                                             |
|------------------------+--------------------------------------------------
| tenant_name            | common                                           |
```

```
| l3out_name             | Datacenter-Out                                |
| name                   | DcExtPol                                      |
| display_name           |                                               |
| nat_epg_dn             |                                               |
| provided_contract_names | []                                           |
| consumed_contract_names | []                                           |
| monitored              | True                                          |
| dn                     | uni/tn-common/out-Datacenter-Out/instP-DcExtPol |
+--------------------------------------------------------------------------+
```

Note   See the example for L3Out: Datacenter-Out.

Now the DN from above can be used to create the external network.

For Distributed NAT:

Example:

```
neutron net-create Datacenter-Out --router:external True --shared --apic:distinguished_names \
type=dict ExternalNetwork=uni/tn-common/out-Datacenter-Out/instP-DcExtPol
```

For No NAT:

Example:

```
neutron net-create Datacenter-Out --router:external True --shared --apic:distinguished_names \
type=dict ExternalNetwork=uni/tn-common/out-Datacenter-Out/instP-DcExtPol --apic:nat_type ""
```

For Floating IPs, add a subnet to external network with the desired floating IP pool cidr.

For SNAT, add a subnet to the external network as below:

Example:

```
neutron subnet-create Datacenter-Out 10.104.21.0/24 --name ext-subnet --disable-dhcp --gateway \
10.104.21.1 --apic:snat_host_pool True
```

For GBP external connectivity, first create the external segment using the SNAT subnet:

Example:

```
gbp external-segment-create Datacenter-Out --subnet-id <SNAT Subnet>
```

Now this segment can be used as usual to create the network service policy.

Example:

```
gbp nat-pool-create nat-pool-0 --ip-pool 10.104.31.0/24 \
--external-segment Datacenter-Out gbp network-service-policy-create --network-service-params \
type=ip_pool,name=nat-pool-0,value=nat_pool net-svc-nat-0
```

Reference Information

This chapter contains the following sections:

Configuring 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 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'
```