White Paper

OpenStack Director and Cisco ACI Unified Plug-in Support for Multi-Pod

First Published: March 2018

Contents

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (https://www.openssl.org/) This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: http:// https://www.cisco.com/go/trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

Introduction

This document explains the procedure for deploying an OpenStack installation based on Red Hat OpenStack Platform (OSP) Director with nodes spanning multiple Cisco® Application Centric Infrastructure (Cisco ACI™) fabric pods.

Readers should have a general understanding of Cisco ACI and OpenStack Red Hat OSP Director.

OpenStack VMM domain

Cisco ACI virtual machine networking supports hypervisors from multiple vendors. It provides the hypervisors with programmable, automated access to a high-performance, scalable, virtualized data center infrastructure.

Programmability and automation are critical features of scalable data center virtualization infrastructure. The Cisco ACI open representational state transfer (REST) API enables virtual machine integration with, and orchestration of, the policy model-based Cisco ACI fabric. Cisco ACI virtual machine networking enables consistent enforcement of policies across both virtual and physical workloads managed by hypervisors from multiple vendors.

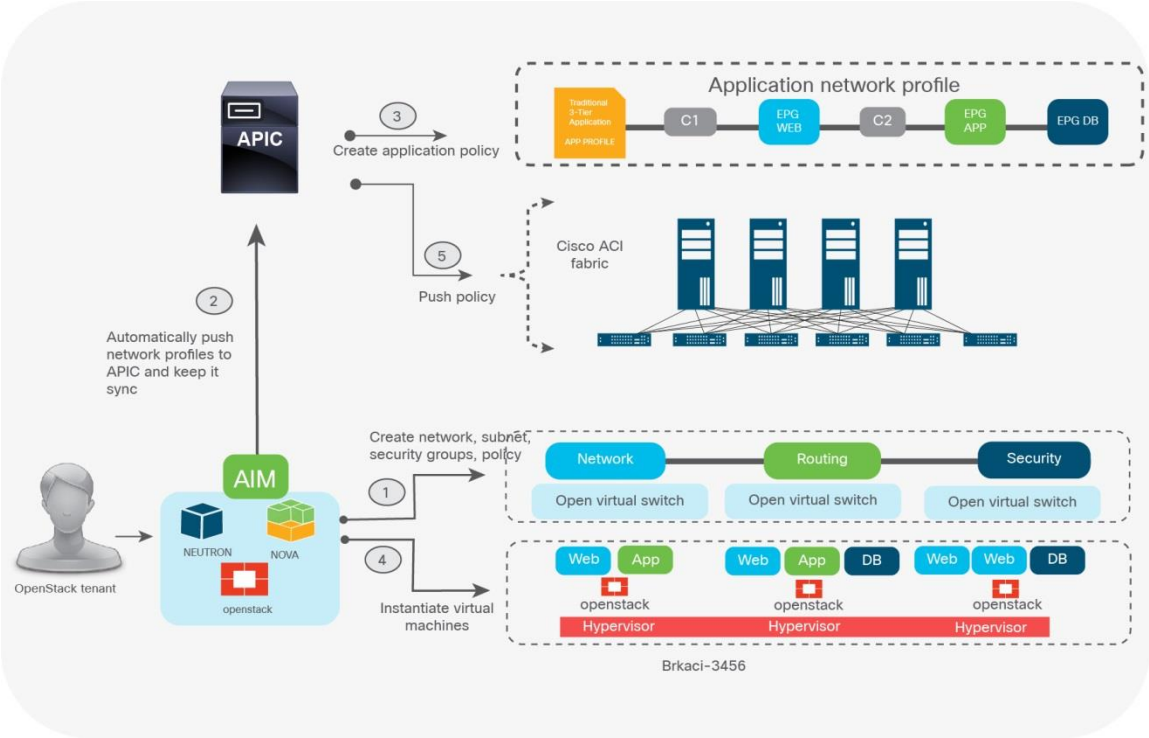
A Cisco ACI Virtual Machine Manager (VMM) domain reduces or eliminates manual configuration and manual errors. Such a domain enables virtualized data centers to support larger numbers of virtual machines reliably and cost effectively. (See the Cisco ACI Virtualization Compatibility Matrix for the most current list of verified interoperable products.)

In the context of OpenStack, the Cisco ACI Unified Plug-in enables a tight integration with this cloud platform. There are several benefits to running this integration:

- **Distributed and scalable virtual networking:** The Cisco ACI Unified Plug-in for Neutron fully replaces the vanilla Neutron Node data path and enables completely distributed functionalities such as Layer 2, distributed routing, Dynamic Host Configuration Protocol (DHCP), metadata, distributed Network Address Translation (NAT), and floating IP.
- **Hardware-accelerated performance:** Users can decide to use VLAN or Virtual Extensible LAN (VXLAN) encapsulation from the compute node. In both scenarios, Cisco ACI Top-of-Rack (ToR) switches will provide line-rate automatic VXLAN tunnels. Optional use of Single Root I/O Virtualization (SR-IOV) is also supported on the compute nodes.
- **Integrated overlay and underlay:** The Cisco Application Policy Infrastructure Controller (APIC) offers a fully managed underlay and overlay network, enabling your to connect physical servers and multiple hypervisors with a penalty-free switching fabric. Using Cisco ACI you can connect any server to OpenStack Neutron-created networks while leveraging line-rate hardware encapsulation translation on the ToR switches.
- **Operations and telemetry:** With ACI Unified Plug-in for OpenStack, troubleshooting becomes much easier. The APIC controller offers complete visibility of the OpenStack physical and overlay topology, combining the health scores and capacity planning per tenant network.

Figure 1 shows the high-level workflow of the Unified Plug-in integration with OpenStack. The OpenStack tenant will leverage the OpenStack Controller to operate the cloud platform. Through the Cisco ACI Integration Module (AIM), ACI objects will be synchronized on the APIC controller, and application policies will be propagated through the OpFlex protocol down to the switches and the compute node open virtual switch rules.

Figure 1. Cisco ACI Unified Plug-in workflow

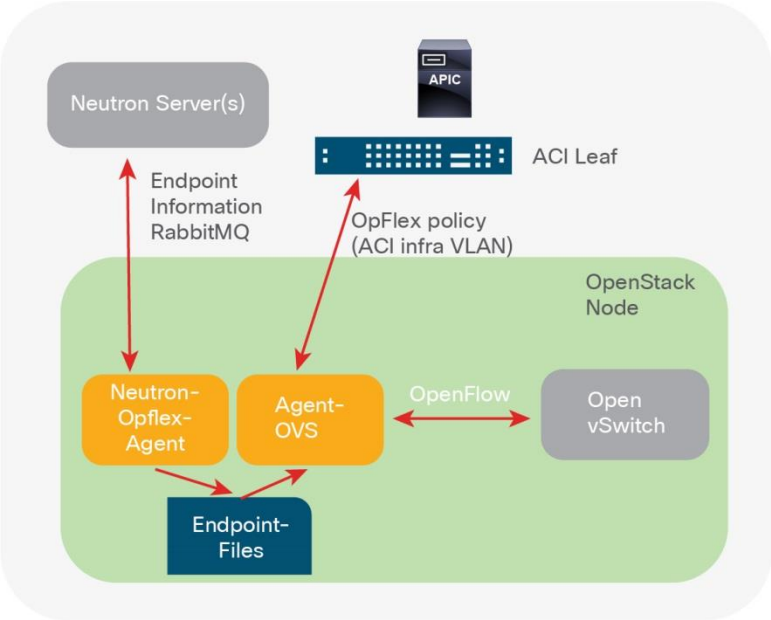


As part of the Unified Plug-in for OpenStack deployment, the following main components are added to the OpenStack nodes:

- **Cisco ACI Integration Module (AIM):** The AIM daemon run on the controller nodes and is responsible for configure Cisco ACI through a REST API call based on the OpenStack policy model defined.
- **Neutron OpFlex agent:** The Neutron OpFlex agent runs on both the compute and the controller. It is responsible to communicate with the neutron server. Please note this component is optional as you may decide not to run OpFlex integration with the Unified Plug-in (when running SR-IOV nodes).

Agent Open Virtual Switch (OVS): The agent OVS runs on the compute and controller nodes. It is responsible for communicating with the OVS and the leaf node to register to the Cisco ACI fabric. As shown in Figure 2, in a typical workflow, the Neutron OpFlex agent receives updates from Neutron about new endpoints and updates endpoint and service files. The agent OVS runs the OpFlex protocol with a Cisco ACI leaf proxy and programs OVS via OpenFlow.

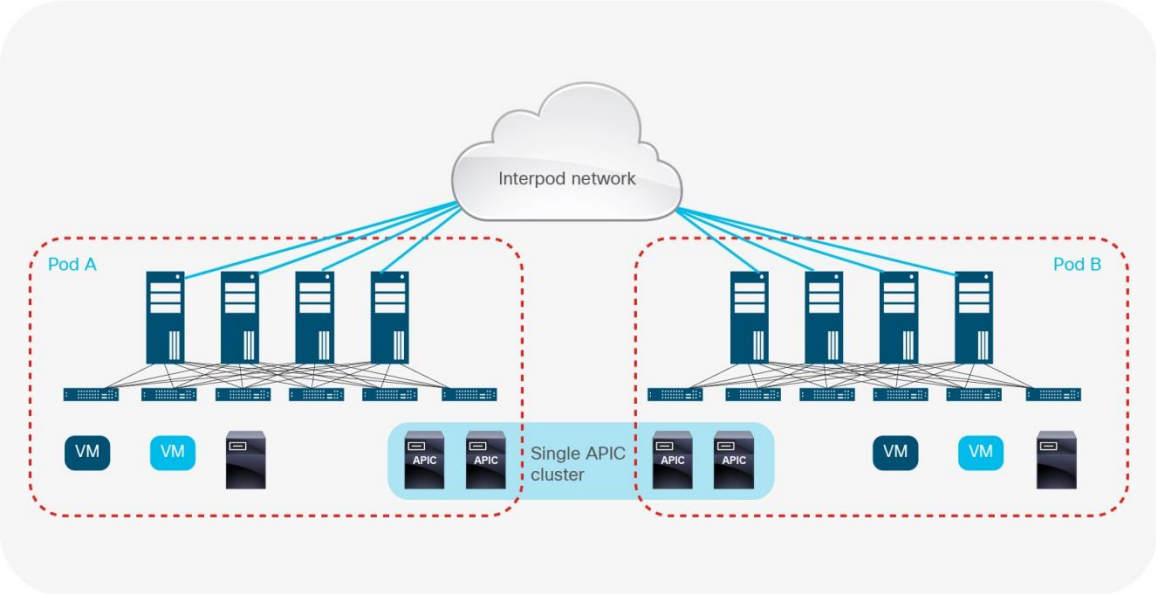
Figure 2. OpFlex architecture



Cisco ACI Multi-Pod

Cisco ACI Multi-Pod allows you to interconnect and centrally manage separate ACI fabrics, as shown in Figure 3. This topology is part of the “single APIC cluster/single domain” family of solutions, as a single APIC cluster is deployed to manage all of the interconnected ACI fabrics. Those separate fabrics are termed “pods” and each looks like a regular two-tier spine-leaf fabrics. A single APIC cluster can manage several pods, and to increase the resiliency of the solution, the various controller nodes that make up the cluster can be deployed across different pods. (More information on Cisco ACI Multi-Pod can be found in the ACI Multi-Pod white paper.)

Figure 3. Cisco ACI Multi-Pod topology



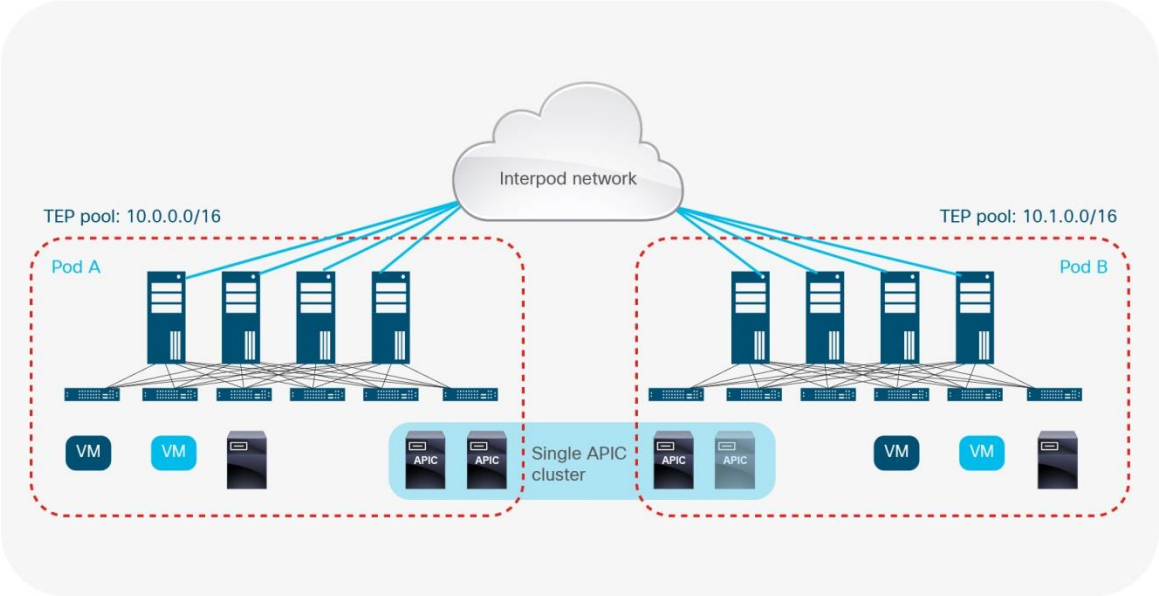
For the purpose of this document, it is fundamental to understand that separate ACI pods use separate tunnel endpoint address pools (TEP pools).

TEP pools are necessary to create an underlay network topology through which APIC controllers can manage and control the entire fabric infrastructure. Switches will also use this network to run protocols such as Intermediate System to Intermediate System (ISIS) and Multiprotocol Border Gateway Protocol (MP-BGP). TEP pool IP addressing is constrained to the ACI underlay only, therefore, it is not reachable from the **overlay** networks or **external** world.

TEP pool information is specified during the **fabric bring-up** process.

In a Multi-Pod ACI topology, each pod will be assigned with its own TEP pool address space, as shown in Figure 4.

Figure 4. ACI Multi-Pod TEP pools

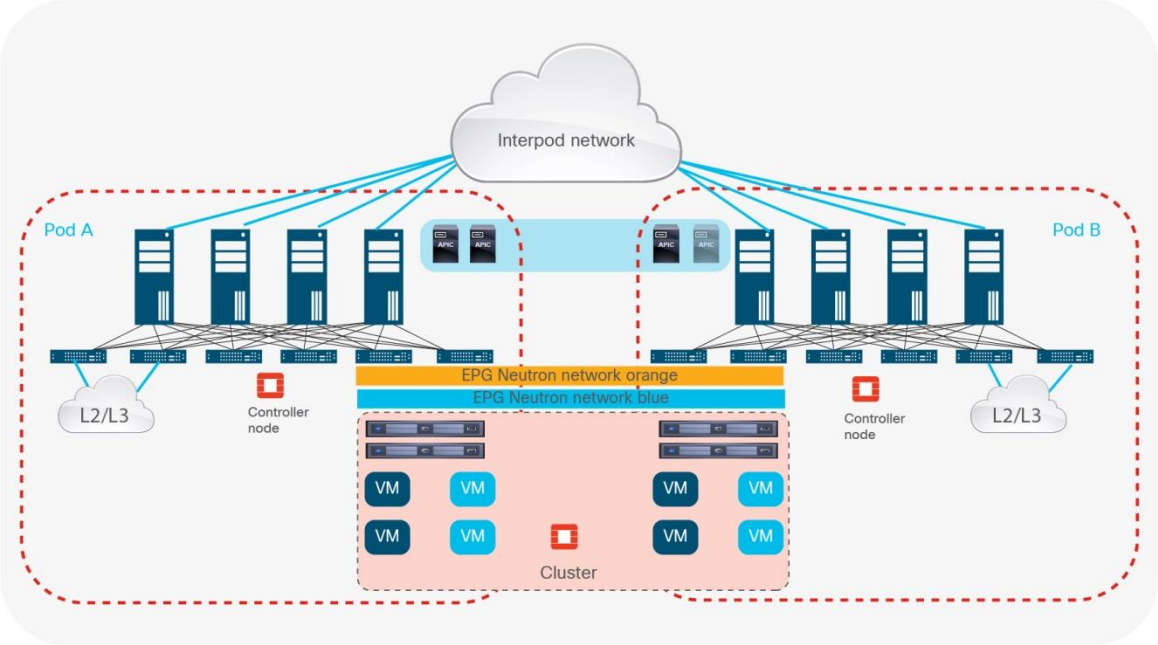


Cisco ACI Multi-Pod and OpenStack

The benefit of combining Cisco ACI Multi-Pod and the OpenStack ACI Unified Plug-in is to seamlessly extend the OpenStack cluster across different data centers, both increasing redundancy and allowing disaster recovery scenarios.

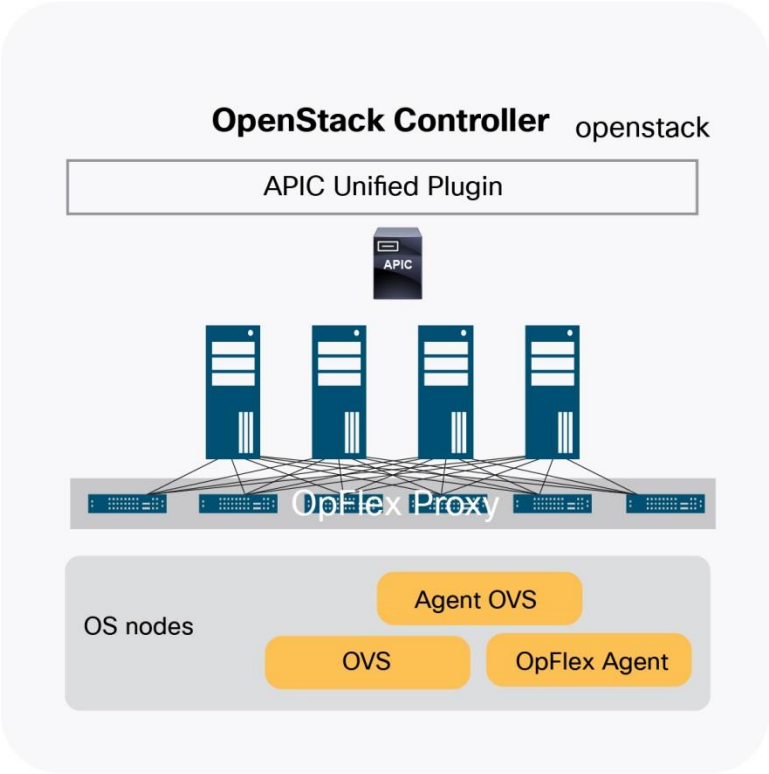
Figure 5 shows an example in which OpenStack controller and compute nodes span different ACI pods connected through an interpod Layer 3 network.

Figure 5. Cisco ACI Multi-Pod with OpenStack nodes distributed in different pods



When integrating a Cisco ACI solution with OpenStack as the VMM domain, an OpFlex agent will be installed in the OpenStack nodes (Figure 6). Through the integration, the OpenStack nodes running the OpFlex agent may later become, effectively, ACI Virtual Tunnel Endpoints (VTEPs), similar to ACI leaf switches. The OpFlex integration enables APIC controllers to push policies down to the compute nodes that will be implemented in the OVS by another agent, called agent-ovs. Effectively, the OVS becomes an virtual switch that is managed by the APIC. As a result, the compute nodes receive from the APIC controller IP addresses that are carved out from the pod VTEP pool they are connected to.

Figure 6. ACI OpFlex integration with OpenStack



Once the OVS agents running in the OpenStack nodes have IP connectivity with the Cisco ACI fabric, they will register to the directly connected ACI leaf switches. To do this, the agents will use the ACI infra anycast gateway IP address. If the default ACI VTEP pool addressing is used (10.0.0.0/16) the infra anycast gateway IP address is 10.X.0.30; X is the pod ID where the OpenStack node belongs.

Additionally, the Cisco ACI fabric also helps with offloading from the node replication for broadcast, unknown unicast, and multicast (BUM) traffic, a key differentiator versus other implementations. To leverage this offloading, the OpenStack node will encapsulate traffic to a unicast IP address that corresponds to the ACI infra subnet gateway and belongs to the directly connected ACI leaf switch. Continuing the example from above, if the default ACI VTEP pool addressing is used (10.0.0.0/16) this IP address is 10.X.0.32; X is the pod ID the OpenStack node is connected.

The agent OVS will have those ACI infra subnet and Anycast gateway IP addresses configured in the file /etc/opflex-agent-ovs/conf.d/opflex-agent-ovs.conf.

Red Hat OSP Director

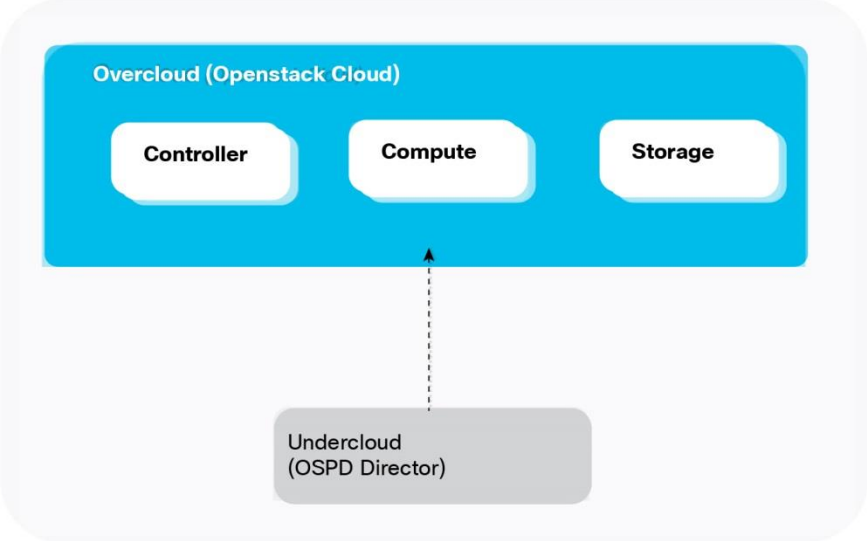
Red Hat OSP Director is a lifecycle management platform for OpenStack. It provides capability for planning and design, OpenStack cloud deployments, and the ongoing operation, administration, and management of the environment.

This document focuses on the predeployment planning stage of OSP Director only.

OSP Director provides configuration files to define the target architecture, including networking and storage topologies, OpenStack service parameters, integrations to third-party plug-ins, etc.—all of the items needed to suit an organization's requirements.

As shown in Figure 7, OSP Director is in charge of deploying all the OpenStack nodes of the "overcloud."

Figure 7. OSP Director overcloud deployment



The Cisco ACI Unified Plug-in fully integrates with OSP Director as of OSP9. (For more information on the general OSP Director planning with Cisco ACI Unified Plug-in, please refer to the Cisco ACI Installation Guide for Red Hat OpenStack Using OSP Director.)

As explained previously, in the context of a ACI Multi-Pod deployment, an extra configuration must be customized based on the pod VTEP pool. In other words, a per-node configuration is required to customize the /etc/opflex-agent-ovs/conf.d/opflex-agent-ovs.conf file, depending on which POD the node is connected to.

To have OSP Director automatically deploy an overcloud with a per-node customized configuration based on the pod ID, you need to follow the procedure described in the next section.

Procedure

OSP Director supports an automated deployment of Red Hat OpenStack overcloud automatically integrated with ACI. Please refer to the Cisco ACI Installation Guide for Red Hat OpenStack Using OSP Director for general information on this subject.

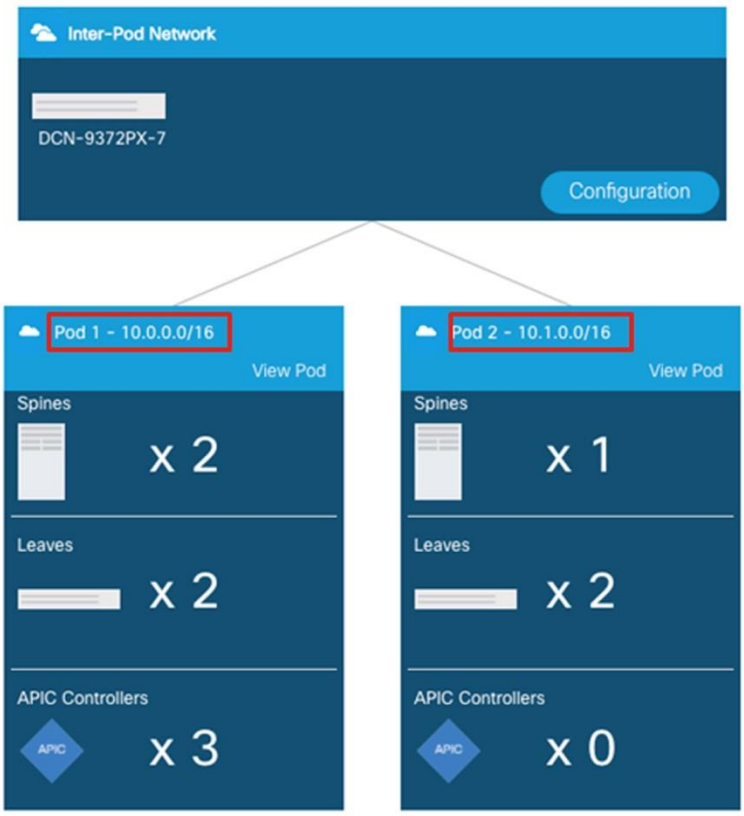
As explained in that installation guide mentioned above, the default information for a Cisco ACI infra subnet and anycast gateway should be configured as a value in the .yaml file for these two keys:

- ACIApicInfraSubnetGateway
- ACIApicInfraAnycastAddr

These values are defined for the **default** pod ID-specific infra anycast and subnet gateways IP addresses. By "default pod ID" we mean the ACI pod where a node would be directly connected, unless further ad hoc configuration is added, as is explained in the following paragraphs.

If you are not sure what TEP pool you defined in the pods of your Cisco ACI infrastructure, you can verify that in the ACI interface: Log in and go to Fabric → Inventory → Topology. From there, as shown in Figure 8, you can see what the TEP pools are defined by pod.

Figure 8. Pod TEP pools verification



When OpenStack nodes span multiple ACI pods, ACIApicInfraSubnetGateway and ACIApicInfraAnycastAddr must be overridden for the nodes not belonging to the Pod 1.

To override, you must use the OS::TripleO::NodeExtraConfig interface and follow the steps described below.

Step 1: Acquire system UUID of nodes belonging to nondefault pod

First step is to acquire the system's Universal Unique Identifier (UUID) saved as part of the introspection data for nodes belonging to a pods other than Pod 1.

For each OpenStack node attached to a pod other than Pod 1, do the following:

1. Retrieve the UUID of the ironic node.
2. Execute the following command against the retrieved UUID of the node:

   ```
   $ openstack baremetal introspection data save <ironic-node-UUID> | jq
   .extra.system.product.uuid
   ```

3. Note the output of this command, for example:

   ```
   F5055C6C-477F-47FB-AFE5-95C6928C407F
   ```

4. Repeat steps 1–3 for all the other nodes not belonging to Pod 1.

Step 2: Create a per-node .yaml configuration file

The UUID identified in the previous step will be used to create a new .yaml file, such as /home/stack/templates/site_environment.yaml.

This new file should look like the following:

```
NodeDataLookup: '{"8EBB0479-330D-4A8D-B90A-5A52C67F6F4F":
         {"ciscoaci::opflex::aci_apic_infra_subnet_gateway" : "10.1.0.30",
"ciscoaci::opflex::aci_apic_infra_anycast_address" : "10.1.0.32"},
                  "AEAE5734-868A-46ED-ACC6-60A2E07ED3B7":
         {"ciscoaci::opflex::aci_apic_infra_subnet_gateway" : "10.1.0.30",
"ciscoaci::opflex::aci_apic_infra_anycast_address" : "10.1.0.32"}
     }'
```

Where:

- 8EBB0479-330D-4A8D-B90A-5A52C67F6F4F and AEAE5734-868A-46ED-ACC6-60A2E07ED3B7 would be UUDIs of two OpenStack nodes belonging to ACI Pod 2.
- 10.1.0.30 is the ACI infra anycast gateway of the pod ID the node belongs to.
- 10.1.0.32 is the ACI infra subnet gateway of the pod ID the node belongs to.

More nodes can be attached by adding more elements to the json file structure.

Step 3: Identify the NodeExtraConfig interface file

In OpenStack director only one OS::TripleO::NodeExtraConfig interface can be used.

Identify the .yaml file where the interface is called, for example:

```
[stack@undercloudmp ~]$ more /home/stack/templates/extra_config.yaml
resource_registry:
  OS::TripleO::NodeExtraConfig: /home/stack/templates/site_nodeextraconfig.yaml
```

Step 4: Add the node-specific configuration parameters, resources, and output

Once you have identified the file of the OS::TripleO::NodeExtraConfig interface (/home/stack/templates/site_nodeextraconfig.yaml in our example), you need to add parameters, resources, and output for the NodeExtra configuration.

Under the existing **parameters** ([snip…] in the example below), add the following:

```
parameters:
[snip…]
server:
    description: ID of the controller node to apply this config to
    type: string

  # Config specific parameters, to be provided via parameter_defaults
  # This would be a lookup of the node UUID as provided by dmidecode
  # to the json required for the node-specific hieradata
  # Note this needs to be a json blob e.g:
  #      parameter_defaults:
  #        NodeDataLookup: |
```

```
#           {"AB4114B1-9C9D-409A-BEFB-D88C151BF2C3": {"foo": "bar"},
#            "8CF1A7EA-7B4B-4433-AC83-17675514B1B8": {"foo2": "bar2"}}
  NodeDataLookup:
    type: string
    default: ''
    description: json string containing per-node configuration map
```

Under the existing **resources** ([snip…] in the example below), add the following:

```
resources:
[snip…]
  NodeSpecificConfig:
    type: OS::Heat::SoftwareConfig
    properties:
      group: script
      inputs:
      - name: node_lookup
      config: |
        #!/bin/sh
        node_id=$(dmidecode --s system-uuid)
        # Create a /etc/puppet/hieradata/UUID.json file to provide
        # the data of the NodeDataLookup parameter that matches the
        # system UUID
        echo $node_lookup | python -c "
        import json
        import sys
        input = sys.stdin.readline() or '{}'
        cnt = json.loads(input)
        print json.dumps(cnt.get('${node_id}', {}))
        " > /etc/puppet/hieradata/${node_id}.json
  NodeSpecificDeployment:
    type: OS::Heat::SoftwareDeployment
    properties:
      name: NodeSpecificDeployment
      config: {get_resource: NodeSpecificConfig}
      server: {get_param: server}
      input_values:
        node_lookup: {get_param: NodeDataLookup}
```

Under the existing **output** ([snip…] in the example below), add the following:

```
outputs:
  deploy_stdout:
    [snip…]
    value: {get_attr: [NodeSpecificDeployment, deploy_stdout]}
```

Step 5: Deploy the overcloud

After preparing the yaml files in the preceding, you can deploy the overcloud environment.

Verification

After a successful deployment, the overcloud nodes should be deployed so that have a specific configuration pointing to the pod local anycast and subnet gateway.

To verify the deployment, you can log in to a node and check the /etc/opflex-agent-ovs/conf.d/opflex-agent-ovs.conf configuration, which should look something like this:

```
[stack@overcloudmp ~]$ ssh heat-admin@10.10.250.75
[heat-admin@overcloud-compute-0 ~]$ more /etc/opflex-agent-ovs/conf.d/opflex-agent-ovs.conf | grep 'hostname\|remote-ip'
          {"hostname": "10.1.0.30", "port": "8009"}
                    "remote-ip": "10.1.0.32",
[heat-admin@overcloud-compute-0 ~]$
```

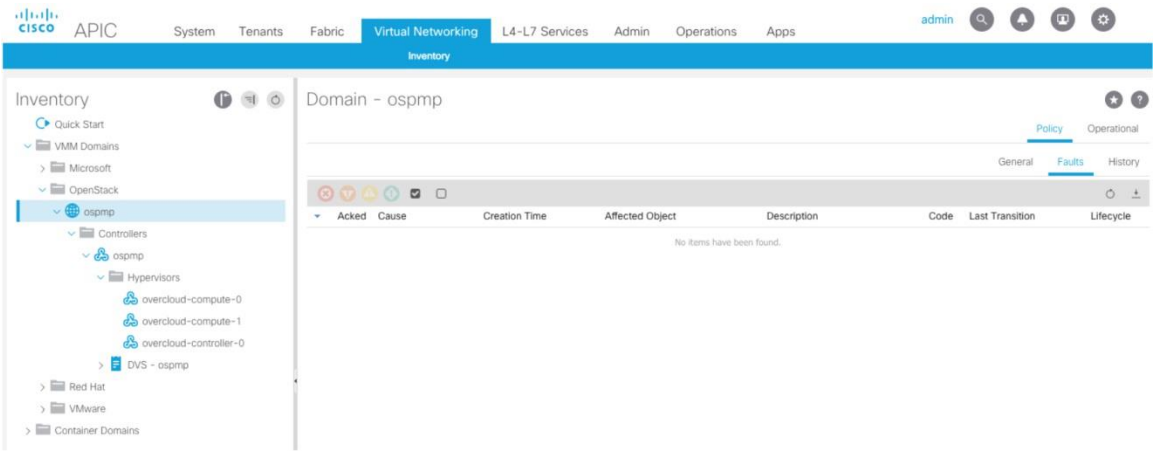
The IP addresses 10.1.0.30 and 10.1.0.32 should be coherent with the pod-specific VTEP addressing specified in the configuration .yaml file. (This statement assumes that the default ACI VTEP pool addressing is used. Note that this VTEP pool can be customized at fabric bring-up, as explained earlier in this document.)

You can also verify that the Agent OVS is not showing errors or being disconnected:

```
[heat-admin@overcloud-compute-0 ~]$ journalctl -f -u agent-ovs | grep -i 'disconnect\|error'
^C
[heat-admin@overcloud-compute-0 ~]$
```

Finally, in the Cisco ACI interface, go to Virtual Networking □ VMM Domains □ OpenStack Panel and verify that all the nodes are correctly shown and registered with ACI, while no faults are shown for the domain (see Figure 9).

Figure 9. Cisco ACI VMM domain verification



Conclusion

Cisco ACI is the right platform for integrating with OpenStack. The Cisco Unified Plug-in provides additional value for customers, adding benefits like scalable virtual networking, hardware-accelerated performance, integrated overlay and underlay, and visibility and telemetry through the APIC controller. Also, distributing Neutron functions through the Cisco ACI integration adds the benefit of a distributed and highly available OpenStack cloud platform.

Red Hat OSP Director 9 and later versions now fully supports Cisco ACI Unified Plug-in integration throughout the full lifecycle of planning, deploying, and operating the OpenStack environment. . This solution decreases time to market and it is considered one of the best performing and solid ways to provide OpenStack services.

When combined with Cisco ACI Multi-Pod technology—assuming sufficient OpenStack storage is provided—this solution seamlessly spans the OpenStack cluster across up to 12 data centers hundreds of kilometers apart (up to 50 milliseconds round trip time). Altogether, this solution makes possible highly available cloud deployments that support disaster recovery scenarios and increase the reliability and scalability of your infrastructure.

For more information

ACI Multi-Pod White Paper:

https://www.cisco.com/c/en/us/solutions/collateral/data-center-virtualization/application-centric-infrastructure/white-paper-c11-737855.html

Cisco ACI with OpenStack OpFlex Architectural Overview:

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/openstack/b_ACI_with_OpenStack_OpFlex_Architectural_Overview/b_ACI_with_OpenStack_OpFlex_Architectural_Overview_chapter_010.html

Cisco ACI Installation Guide for Red Hat OpenStack Using OSP Director 2.3(x) or later:

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/2-x/openstack/osp_director/b_ACI_Installation_Guide_for_Red_Hat_OpenStack_Using_OSP_Director_2_3_x/b_ACI_Installation_Guide_for_Red_Hat_OpenStack_Using_OSP_Director_2_3_x_chapter_010.html

Cisco Application Centric Infrastructure Fundamentals:

https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/aci-fundamentals/b_ACI-Fundamentals/b_ACI-Fundamentals_chapter_01011.html

Introduction to Red Hat OpenStack Platform Director:

https://redhatstackblog.redhat.com/2016/07/25/introduction-to-red-hat-openstack-platform-director/

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA

Asia Pacific Headquarters
Cisco Systems (USA) Pte. Ltd.
Singapore

Europe Headquarters
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Printed in USA C11-740407-00 04/18