Cisco Application Centric Infrastructure and IBM Cloud Orchestrator Integration ESX Environment Solution Overview
Contents

About This Document................................................................................................................. 4
Audience........................................................................................................................................ 4
Introduction................................................................................................................................. 4

Solution: Cisco Application Centric Infrastructure and IBM Cloud Orchestrator........................ 4
  Overview..................................................................................................................................... 5
  Hardware and Software Revisions......................................................................................... 6
  Cisco ACI Fabric.................................................................................................................... 6
  Cisco Application Policy Infrastructure Controller ............................................................ 7
  IBM Cloud Orchestrator ....................................................................................................... 7
  IBM Cloud Manager ............................................................................................................. 8
  Cisco Unified Computing System......................................................................................... 8
  VMware vSphere .................................................................................................................. 9

Cisco ACI: Basics and VMware Integration............................................................................ 9
  Cisco ACI Integration with VMware vCenter ......................................................................... 11

Cisco ACI and IBM Cloud Orchestrator Integration................................................................. 12
  Operation Workflow ............................................................................................................. 13

Preparing the Cisco ACI and VMware vCenter Environment for Integration........................ 14
  Creating a VMware vCenter Domain Profile Using the GUI.................................................. 14
  Creating VMware vSphere Distributed Switch Uplink Port Groups ..................................... 16

Preparing IBM Cloud Orchestrator for Cisco ACI and VMware vCenter integration ............ 16
  Supported Deployment Topologies ...................................................................................... 17
  Deployment Topology Components ..................................................................................... 18
  Deployment Topology Requirements and Overview ........................................................... 19
  Hardware Prerequisites for IBM Cloud Orchestrator ......................................................... 21
  Software Prerequisites for IBM Cloud Orchestrator ........................................................... 22
  Hardware Prerequisites for IBM Cloud Manager with OpenStack ...................................... 22
  Configuration Changes to Support VMware Hypervisors ..................................................... 24

Configuring IBM Cloud Manager for VMware Integration ..................................................... 24
  Connecting to Multiple Clusters ......................................................................................... 25
  Configuring VMware Discovery ........................................................................................... 27
  Configuring VMware DVS Mechanism Driver (Enabling VMware VDS on IBM Cloud Manager with OpenStack) .............................................................................................................................. 28
  Verifying Discovery of the VMware Environment and Networks ....................................... 29
  Creating a Virtual Machine Template as a Snapshot Image for VMware Instances .................. 30
  Assigning Availability Zones to Domains and Projects ........................................................ 31

Deploying Applications in Cisco ACI with IBM Cloud Manager Integrated for VMware ........... 33
  Creating Tenants, Private Networks, and Bridge Domains .................................................. 33
  Creating Application Profiles, EPGs, and Contracts ........................................................... 34
    Create a Filter Using the GUI ............................................................................................. 35
    Create a Contract Using the GUI ....................................................................................... 37
    Create an Application Profile Using the GUI ................................................................. 37
    Create EPGs Using the GUI .............................................................................................. 38
    Consume and Provide Contracts Using the GUI .............................................................. 38
  Creating Subnets in IBM Cloud Manager for Discovered Networks ..................................... 39
  Deploying Virtual Machines with IBM Cloud Orchestrator ................................................ 40
  Verifying That Cisco ACI Discovers the Virtual Machines and That Communication Follows Application Policy ................................................................. 44
About This Document

This document describes the Cisco Application Centric Infrastructure (Cisco ACI™) and IBM Cloud Orchestrator solution for managing VMware vSphere hypervisors in an OpenStack cloud environment.

Cisco ACI simplifies and accelerates the entire application deployment lifecycle for next-generation data center and cloud deployments. Cisco ACI is a holistic architecture that introduces hardware and software innovations built on the new Cisco Nexus® 9000 Series Switches. Cisco ACI provides a centralized policy-based application deployment architecture that is managed through the Cisco Application Policy Controller (APIC). Cisco ACI delivers software flexibility with the scalability and performance of hardware.

IBM Cloud Orchestrator provides cloud management for your IT services, allowing you to accelerate the delivery of software and infrastructure. Based on open standards, it reduces the number of steps needed to manage public, private, and hybrid clouds by providing an easy-to-use interface.

IBM Cloud Orchestrator gives you access to ready-to-use patterns and content packs, helping accelerate configuration, provisioning, and deployment. It integrates management tools such as metering, use information, accounting, monitoring, and capacity management into your cloud services. Go live as quickly as you develop and test applications.

Audience

The audience for this document includes sales engineers, field consultants, professional series, IT managers, partner engineers, and customers who want to take advantage of an infrastructure that is built to deliver IT efficiency and enable IT innovation.

Introduction

IT consumption models are changing. More applications are hosted in the cloud. Cloud, mobility, and big data applications are causing a shift in the data center model. New applications are placing demands on the infrastructure in new ways.

Successful IT businesses require “application velocity”: the capability to deploy new applications much faster than before. The application environment in the data center is becoming more diverse. Distributed applications (for example, big data and Hadoop), database applications (such as those from Oracle and SAP) that run on bare metal, virtualized applications running in multihypervisor environments, and cloud-based applications that are available on demand all impose different demands on infrastructure.

Solution: Cisco Application Centric Infrastructure and IBM Cloud Orchestrator

Cisco ACI is a highly flexible, open, programmable environment that can be transparently integrated into OpenStack and VMware virtual environments. Cisco ACI integration into OpenStack and VMware environments focuses on delivering simplicity without compromising infrastructure scalability, responsiveness, security, or end-to-end visibility.

The Cisco Application Policy Infrastructure Controller, or APIC, is the main architectural component of the Cisco ACI and IBM Cloud Orchestrator cloud environment. It is the unified point of automation and management for the Cisco ACI fabric, policy enforcement, and health monitoring for both physical and virtual environments.
Many enterprise customers operate large VMware-based virtualized infrastructures to support critical existing traditional applications and enterprise databases. These organizations are also turning to OpenStack for its open APIs, flexible architecture, and large ecosystem of vendors to compete with an agile software development and deployment model. They seek to gain the agility offered by the cloud while maintaining the stability of a traditional environment.

Cisco ACI and IBM Cloud Orchestrator integrate with VMware vCenter to transparently extend the Cisco ACI policy framework to VMware vSphere workloads while providing the agility and flexibility needed to operate a multihypervisor environment.

Overview

The Cisco ACI and IBM Cloud Orchestrator solution described here uses a reference implementation consisting of the following components (Figure 1):

- Cisco Nexus 9000 Series leaf and spine switches for Cisco ACI to form the flexible and resilient data center network
- Cisco APIC to orchestrate the application-centric network deployment
- IBM Cloud Manager with OpenStack to deploy the OpenStack cloud and manage VMware vSphere–based traditional hypervisors
- IBM Cloud Orchestrator to deliver cloud services
- Cisco Unified Computing System™ (Cisco UCS®) servers with VMware ESXi
- VMware vCenter and vSphere 5.5

Figure 1. Cisco ACI and IBM Cloud Orchestrator Reference Architecture

This document guides you through the steps required to deploy the base solution and the configuration steps for operation use cases.
Hardware and Software Revisions

Table 1 describes the hardware and software versions used for this solution validation.

Table 1.  Validated Software Versions

<table>
<thead>
<tr>
<th>Layer</th>
<th>Device</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network</td>
<td>Cisco APIC</td>
<td>Release 1.1(2h)</td>
</tr>
<tr>
<td></td>
<td>Cisco Nexus 9000 Series Switches</td>
<td>Release 7.0(3)I2(2)</td>
</tr>
<tr>
<td>VMware</td>
<td>VMware vSphere ESXi</td>
<td>Release 5.5u2</td>
</tr>
<tr>
<td></td>
<td>VMware vCenter</td>
<td>Release 5.5u2</td>
</tr>
<tr>
<td>IBM</td>
<td>IBM Cloud Manager</td>
<td>Release 4.3 FP2</td>
</tr>
<tr>
<td></td>
<td>IBM Cloud Orchestrator</td>
<td>Release 2.5</td>
</tr>
</tbody>
</table>

Cisco ACI Fabric

The Cisco ACI fabric consists of three major components:

- APIC
- Spine switches
- Leaf switches

The Cisco ACI switching architecture is laid out in a leaf-and-spine topology in which every leaf connects to every spine using 40 Gigabit Ethernet interfaces. Figure 2 shows the Cisco ACI fabric architecture.

Figure 2.  Cisco ACI Fabric

The software controller—the APIC—is delivered as an appliance, and three or more such appliances form a cluster for high availability and enhanced performance. The APIC is responsible for all tasks that enable traffic transport, including:

- Fabric activation
- Switch firmware management
- Network policy configuration and instantiation
Though the APIC acts as the centralized point of configuration for policy and network connectivity, it is never in line with the data path or the forwarding topology. The fabric can still forward traffic even when communication with the APIC is lost.

The APIC provides both a command-line interface (CLI) and a GUI to configure and control the Cisco ACI fabric. The APIC also exposes a northbound API through XML and JavaScript Object Notation (JSON) and an open-source southbound API.

**Cisco Application Policy Infrastructure Controller**

The APIC is the unifying point of automation and management for the Cisco ACI fabric. The APIC provides centralized access to all fabric information, optimizes the application lifecycle for scalability and performance, and supports flexible application provisioning across physical and virtual resources. The main benefits of the APIC include the following:

- Centralized application-level policy engine for physical, virtual, and cloud infrastructure
- Detailed visibility, telemetry, and health scores by application and by tenant
- Design based on open standards and open APIs
- Robust implementation of multitenant security, quality of service (QoS), and high availability
- Integration with management systems such as VMware, Microsoft, and OpenStack solutions

The APIC exposes northbound APIs through XML and JSON and provides both a CLI and a GUI that use the APIs to manage the fabric holistically. For redundancy and load distribution, three controllers are recommended for to manage Cisco ACI fabric.


The APIC integrates with VMware vCenter to transparently extend the Cisco ACI policy framework to VMware vSphere workloads. The APIC uses application network profiles to represent the Cisco ACI policy. These profiles model the logical representation of all components of the application and its interdependencies on the Cisco ACI fabric. This policy framework also includes a Layer 4 through Layer 7 service insertion mechanism, providing full-service lifecycle management from workload instantiation through decommissioning.

**IBM Cloud Orchestrator**

IBM Cloud Orchestrator provides cloud management for your IT services, allowing you to accelerate the delivery of software and infrastructure. Based on open standards, it reduces the number of steps needed to manage public, private, and hybrid clouds by providing an easy-to-use interface.

The orchestrator gives you access to ready-to-use patterns and content packs, helping accelerate configuration, provisioning, and deployment. It integrates management tools such as metering, use information, accounting, monitoring, and capacity management into your cloud services. Go live as quickly as you develop and test applications.

The orchestrator helps you:

- Quickly deploy and scale on-premises and off-premises cloud services
- Provision and scale cloud resources
- Reduce administrator workloads and error-prone manual IT administrator tasks
Integration with existing environments using APIs and tool extensions
Deliver services with IBM SoftLayer, existing OpenStack platforms, IBM PowerVM, IBM System z, VMware, and Amazon Elastic Compute Cloud (EC2)


**IBM Cloud Manager**

IBM Cloud Manager, based on OpenStack with IBM enhancements and support, is easy to deploy and use.

This cloud management software provides these benefits:

- Respond rapidly to the changing IT demands of business through self-service provisioning of infrastructure services. End users can redeploy virtual servers through an easy-to-use interface.
- Improve virtualization operation efficiency and overall business effectiveness. Administrators can capture and manage standard virtual machine images and get support for common business processes.
- Support production-class cloud operations and interoperability at scale through an enhanced foundation and full OpenStack API compatibility.
- Optimize infrastructure use, reduce the cost of cloud ownership, and increase workload QoS.
- Deploy an open computing cloud alternative to proprietary vendors, with world-class support from IBM.
- Deliver hybrid capability to IBM SoftLayer.
- IBM Platform Resource Scheduler (PRS) is now included as a standard feature.


**Cisco Unified Computing System**

This solution uses Cisco UCS C220 M3 Rack Servers to install IBM Cloud Manager with OpenStack, IBM Cloud Orchestrator, and VMware ESXi servers as OpenStack computing nodes.

The Cisco UCS C220 M3 is designed for performance and density over a wide range of business workloads, from web serving to distributed databases. This server further extends the capabilities of the Cisco UCS portfolio in a 1-rack-unit (1RU) form factor. With the addition of the Intel® Xeon® processor E5-2600 and E5-2600 v2 product families, it also delivers significant performance and efficiency gains.

The Cisco UCS C220 M3 also offers up to 512 GB of RAM, eight hard disk drives or solid-state disk (SSD) drives, and two 1 Gigabit Ethernet LAN interfaces built into the motherboard, delivering outstanding levels of density and performance in a compact package.

The Cisco UCS C220 M3 offers these main features and capabilities:

- General-purpose rack server, suitable for nearly all 2-socket applications
- Unique Cisco UCS Virtual Interface Card (VIC) 1225, providing two 10 Gigabit Ethernet PCI Express (PCIe) slots that can support up to 256 PCIe virtual interfaces
- Exceptional building block and entry point into Cisco UCS
The specifications for the Cisco UCS C220 M3 include:

- Up to two Intel Xeon processor E5-2600 or E5-2600 v2 CPUs
- Up to 512 GB of RAM with 16 DIMM slots for memory-intensive applications
- Four or eight SAS, SATA, or SSD drives
- Two PCIe Generation 3 slots and two 1 Gigabit Ethernet LAN interfaces on the motherboard
- Trusted Platform Module (TPM) for authentication and tool-free access


**VMware vSphere**

VMware vSphere is a virtualization platform for holistically managing large collections of infrastructure resources—CPUs, storage, and networking resources—as a cohesive, versatile, and dynamic operating environment. Unlike traditional operating systems that manage individual machines, vSphere aggregates the infrastructure of an entire data center to create a single powerhouse with resources that can be allocated quickly and dynamically to any application that needs them.

vSphere delivers a robust application environment. For example, with vSphere, all applications can be protected from downtime with VMware High Availability (HA) without the complexity of conventional clustering. In addition, applications can be scaled dynamically to meet changing loads with capabilities such as hot addition and VMware vSphere Distributed Resource Scheduler (DRS).


**Cisco ACI: Basics and VMware Integration**

Cisco ACI abstracts policy and connectivity from the underlying fundamental network constructs of VLANs, IP addresses, access control lists (ACLs), and QoS policies. It then can be used to describe application network connectivity more abstractly, in terms of endpoint groups, providers and consumers, service-level agreements (SLAs), etc., to make this connectivity relevant for the end user of the fabric.

Cisco ACI provides a secure multitenant solution, allowing the network infrastructure administration and data flows to be segregated. Tenants can be divided into customers, business units, groups, etc. Tenants also can be subdivided into Layer 3 constructs, known as Virtual Routing and Forwarding (VRF) instances. Contexts provide an additional way to separate the organizational and forwarding requirements of a tenant. Within each VRF instance, a bridge domain is created. A bridge domain is a Layer 2 namespace in which the various subnets are defined. All the subnets and default gateways are assigned within the bridge domain. By using separate forwarding instances, IP addressing can be duplicated in separate contexts for multitenancy.

Within a context, a series of objects, called endpoints and endpoint groups (EPGs), define an application (Figure 3). Endpoints are devices (physical or virtual) that connect to the fabric and use it to interface with the network infrastructure. These endpoints can be computing, storage, and network services and security devices that attach to the fabric. At first customer shipment (FCS), Cisco ACI supports endpoints classified as network interface cards (NICs) or virtual NICs (vNIC) and their associated VLANs or Virtual Extensible LANs (VXLANs). In the future, endpoints will be extended to include IP addresses, MAC addresses, Domain Name System (DNS) names, virtual machine attributes, IEEE 802.1x identity, and other common attributes.
An EPG is a collection of endpoints with the same types of attributes and identical network behavior (connectivity, security, QoS requirements, etc.).

Here are some examples of EPGs:

- EPG defined by traditional network VLANs; all endpoints connected to a given VLAN are placed in an EPG
- EPG defined by VXLANs; same as for VLANs except using VXLAN
- EPG defined by security zone
- EPG defined by application tier (web, application, or database)
- EPG mapped to a VMware ESXi port group

The policies used to describe the communication, services insertion, and QoS and SLAs embedded in EPGs are referred to as contracts. A contract is a set of policy requirements that describe how EPGs can communicate with each other and with the outside world. The Cisco ACI contract defines a filter, which includes Layer 4 inbound and outbound ports and an associated action that dictates whether the traffic is permitted, denied, logged, marked, redirected, or copied. The default behavior of Cisco ACI security follows a whitelist model, which denies all traffic. In the default mode, administrators must explicitly allow communication between EPGs. This is a configurable option.

For each tenant, EPGs and policies are summarized in an application network profile (ANP). These ANPs are the logical representation of the application infrastructure requirements. When the application is ready to be provisioned, the APIC can push the ANP and provision the entire stateless Cisco ACI fabric and Layer 4 through Layer 7 service devices instantaneously.
Cisco ACI Integration with VMware vCenter

A main benefit of Cisco ACI is the capability to be hypervisor independent. Traffic forwarding is not limited to or constrained within the encapsulation type or encapsulation overlay network (Figure 4).

Figure 4. Hypervisor-Independent Traffic in Cisco ACI

In Cisco ACI, the virtual machine manager (VMM) is a hypervisor management system that has control over virtual machines. A Cisco ACI fabric can have multiple VMM domains across hypervisors from different vendors. The VMM domain for VMware is VMware vSphere vCenter. Each VMM domain has a local significant association with a VLAN or VXLAN and contains its own mobility domain, which means that mobility is restricted to that VMM domain. Across the various VMM domains, the VLAN or VXLAN namespace can be reused. Using VMM domains, a customer can scale beyond the usual restriction of 4000 VLANs. In a large-scale cloud environment, 4000 VLANs or identifiers are not enough to uniquely identify each tenant, segment, or network. Each VMM domain can scale up to 4000 VLANs. As a result, organizations can span and scale horizontally using VMM domains and achieve up to 64,000 logical network segments. The only restriction, as mentioned earlier, is that virtual machine mobility is limited to the VMM domain.

The APIC integrates with the VMware vCenter instances to transparently extend the Cisco ACI policy framework to VMware vSphere workloads. The APIC uses ANPs to represent the Cisco ACI policy. The APIC creates a VMware vSphere Distributed Switch (VDS) in vCenter to create the virtual network. From that point onward, the APIC manages all application infrastructure components. The network administrator creates EPGs and pushes them to vCenter as port groups on the VDS. Server administrators can then associate the virtual machines and provision them accordingly (Figure 5).
When a virtual endpoint is discovered, the policy is pushed to and configured on the leaf nodes based on the resolution immediacy policy and instrumentation immediacy policy, respectively. Both policies have immediate and on-demand (default) options that are defined when the VMM domain is associated on the APIC. The on-demand option conserves resources and uses the reserved space in the policy content addressable memory (CAM) when necessary.

Cisco ACI and IBM Cloud Orchestrator Integration

The preceding sections described how Cisco ACI transparently integrates with VMware vCenter to extend the Cisco ACI policy framework to VMware vSphere workloads. This section describes how IBM Cloud Orchestrator and Cloud Manager can discover vCenter assets and transparently orchestrate deployment and management of virtual machines through OpenStack. In this scenario, Cloud Manager with OpenStack is used as the control plane to manage a multihypervisor cloud, running both vSphere and alternate hypervisors such as Linux Kernel-based Virtual Machine (KVM) or Citrix Xen. This approach provides common, self-service provisioning and API access. It consolidates cloud management while allowing applications to be hosted on the environments best suited for them (Figure 6).
**Figure 6.** Cisco ACI and IBM Cloud Orchestrator Integration

**Operation Workflow**

The workflow proceeds as follows:

1. The APIC administrator configures the VMware vCenter domain policies in the APIC. The administrator provides the following vCenter connectivity information:
   - vCenter IP address and credentials and VMM domain policies and domain span
   - Policies (VLAN pools, domain type such as VMware VDS or Cisco Nexus 1000V Switch, etc.)
   - Connectivity to physical leaf interfaces (using attach entity profiles)

2. The APIC automatically connects to vCenter.

3. The APIC creates a VDS in a specified data center in vCenter.

4. **Note:** The VDS name is the VMM domain name.

5. The vCenter administrator or the computing management tool adds the VMware ESX host or hypervisor to the APIC VDS and assigns the ESX host hypervisor ports as uplinks on the APIC VDS. These uplinks must connect to the Cisco ACI leaf switches.

6. The APIC learns the location of the connection of the hypervisor host to the leaf switch using the Link Layer Discovery Protocol (LLDP) or Cisco Discovery Protocol information of the hypervisors.

7. The APIC administrator creates and associates application EPG policies.

8. The APIC administrator associates EPG policies with VMM domains.

9. The APIC automatically creates port groups in vCenter under the VDS. This process provisions the network policy in vCenter.
Note: The port group name is a concatenation of the tenant name, the application profile name, and the EPG name. The port group is created under the VDS; it was created earlier by the APIC.

9. IBM Cloud Manager is configured to discover the vCenter environment and enable the vSphere VDS for use by OpenStack Neutron. The vCenter administrator or the computing management tool instantiates and assigns virtual machines to the port groups.

10. The APIC learns about the placement of the virtual machines on the basis of vCenter events. The APIC automatically pushes the application EPG and its associated policy (for example, contracts and filters) to the Cisco ACI fabric.

11. Cisco Nexus 9000 Series Switches in Cisco ACI mode are deployed with the APIC.

12. vCenter is integrated into Cisco ACI for VLAN and VXLAN deployment.

13. A three-tier application profile with web, application, and database EPGs is used.

14. Virtual machines are deployed using IBM Cloud Orchestrator.

Preparing the Cisco ACI and VMware vCenter Environment for Integration

After racking and stacking all your Cisco Nexus 9000 Series Switches and your controllers (APICs), at this point you have everything connected to the network. The basic steps for installing and deploying the APIC are described in the following document: http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/1-x/getting-started/b_APIC_Getting_Started_Guide/b_APIC_Getting_Started_Guide_chapter_01.html.

The configuration steps describe how to:

- Configure clusters
- Configure out-of-band management
- Configure the administrator user
- Establish Layer 2 or Layer 3 connectivity between the API and the VMware vCenter server and establish Layer 2 or Layer 3 connectivity between the APIC and external entities using the in-band management network

Creating a VMware vCenter Domain Profile Using the GUI

The following is an overview of the tasks for creating a vCenter domain:

1. Create or select a switch profile.
2. Create or select an interface profile.
3. Create or select an interface policy group.
4. Create or select a VLAN pool.
5. Create a vCenter domain.
6. Create vCenter credentials.

The steps that follow provide the details.

2. In the navigation pane, click Switch Policies.
3. Right-click Switch Policies and choose Configure Interface, PC, and VPC.
4. In the work pane, in the Configured Switch Interfaces area, expand Switch Profile (Figure 7).

**Figure 7.** Cisco APIC Configuration Settings

5. For Select Switches to Configure Interfaces, verify that the Quick radio button is automatically checked.
6. From the Switches drop-down list, choose the appropriate leaf ID.
7. In the Switch Profile Name field, verify that the switch profile name automatically populates.
8. Click the + icon to configure the switch interfaces.
9. For Interface Type, select the appropriate radio button.
10. In the Interfaces field, enter the desired interface range.
11. In the Interface Selector Name field, enter the name of the selector to which the VMware ESX ports will be connected.
12. From the Link Level Policy drop-down list, choose the desired link-level policy.
13. From the CDP Policy drop-down list, choose the desired Cisco Discovery Protocol policy.
   
   **Note:** Also choose the desired interface policies from the available policy fields.
14. In the Attached Device Type field, choose ESX Hosts.
15. In the Domain Name field, enter the domain name.
16. In the VLAN Range field, enter the appropriate VLAN range.
   
   **Note:** A range of at least 200 VLAN IDs is recommended. Do not define a range that includes the reserved VLAN ID for the infrastructure network, because that VLAN is for internal use.
17. In the vCenter Login Name field, enter the login name.
18. Optional: From the Security Domains drop-down list, choose the appropriate security domain.
19. In the **Password** field, enter a password.
20. In the **Confirm Password** field, reenter the password.
21. Expand **vCenter/vShield**.
22. In the **Create vCenter/vShield Controller** dialog box, enter the appropriate information; then click **Save**.
23. In the **Configure Interface, PC, and VPC** dialog box, in the **vSwitch Policy** field, select the desired check box to enable Cisco Discovery Protocol or LLDP. Click **Save** and then click **Submit**.
24. Verify the new domain and profiles by performing the following actions:
   a. On the menu bar, choose **VM Networking > Inventory**.
   b. In the navigation pane, choose **VMware > domain_name > vCenter_name**.
25. In the work pane, under **Properties**, view the VMM domain name to verify that the controller is online. In the work pane, the vCenter properties are displayed, including the operational status. The displayed information confirms that connection from the APIC to the vCenter server is established and that the inventory is available (Figure 8).

**Figure 8.** Verifying the Connection

---

**Creating VMware vSphere Distributed Switch Uplink Port Groups**

Each VMM domain appears in vCenter as a VDS. The virtualization administrator associates hosts with the VDS created by the APIC and selects the virtual machine network interface cards (vmnics) to use for the specific VDS. The configuration of the VDS uplinks is performed from the APIC by changing the virtual switch (vSwitch) configuration from the attach entity profile, or AEP, that is associated with the VMM domain. You can find the AEP in the APIC GUI in the Fabric Access Policies configuration area.

After these steps are completed, Cisco ACI and vCenter are ready to host tenants and applications.

**Preparing IBM Cloud Orchestrator for Cisco ACI and VMware vCenter integration**

The deployment described in this document uses IBM Cloud Orchestrator to manage and orchestrate cloud deployment using IBM Cloud Manager with OpenStack. Cloud Manager with OpenStack acts as the OpenStack controller.
Before you start to deploy Cloud Orchestrator, you must decide which deployment topology to install for the Cloud Orchestrator management stack. Depending on your needs and the available hardware resources, you can configure your environment as a single-server environment or as a highly available environment spread across two servers.

**Supported Deployment Topologies**

IBM Cloud Orchestrator supports the following deployment topologies:

- Single-server topology
- Single-server topology with external database
- Dual-server high-availability topology with external database

The examples in this section show a Cloud Orchestrator and OpenStack topology with one OpenStack controller and one computing node.

- Single-server topology: All IBM Cloud Orchestrator management components are installed on a single Cloud Orchestrator server (Figure 9).

**Figure 9.** Single-Server Topology

- Single-server topology with external database: The IBM Cloud Orchestrator management components are installed on a single Cloud Orchestrator server pointing to an existing external IBM DB2 database server (Figure 10).
Dual-server high-availability topology with external database: The IBM Cloud Orchestrator management components are installed on two Cloud Orchestrator servers in a high-availability configuration pointing to an existing external IBM DB2 database server (Figure 11).

Note: The public cloud gateway component is a single point of failure and is not a high-availability component.

Note: High availability is not provided for Cloud Manager with OpenStack components.

Deployment Topology Components
The main components of an IBM Cloud Orchestrator deployment topology are as follows:

- IBM Cloud Orchestrator server: This server hosts the core Cloud Orchestrator management components. For the high-availability topology, two Cloud Orchestrator servers are used.
- OpenStack controllers (previously known as region servers): Each OpenStack controller communicates with a specific hypervisor management infrastructure:
  - The IBM Hyper-V OpenStack controller requires one or more Hyper-V computing nodes to provide the computing resources.
© 2016 Cisco and/or its affiliates. All rights reserved. This document is Cisco Public Information.

- The KVM OpenStack controller requires one or more KVM computing nodes to provide the computing resources.
- The IBM Power Virtualization Center (PowerVC) OpenStack controller must connect to an existing PowerVC to provide virtual machines.
- The VMware OpenStack controller must connect to an existing VMware vCenter instance to provide virtual machines.
- The VMware z/VM OpenStack controller must connect to an IBM Extreme Cloud Administration Toolkit (xCAT) management node on z/VM to provide virtual machines.

- Computing nodes (KVM or Hyper-V): The computing nodes manage the virtual machines through the interface that is provided by KVM or Hyper-V.

Deployment Topology Requirements and Overview
Verify that your IBM Cloud Manager with OpenStack installation meets the following requirements:

- Cloud Manager with OpenStack Version 4.3 Fix Pack 2 (OpenStack Kilo release): Verify that Cloud Manager with OpenStack v4.3 Fix Pack 2 is installed (OpenStack Kilo release).
- Shared OpenStack Keystone service: The Cloud Manager with OpenStack installation uses a single Keystone service. In a multiregion installation, the Keystone service must run on the OpenStack controller of the first region that is installed. OpenStack's identity federation is not supported by Cloud Orchestrator.
- Fresh installation: The Cloud Manager with OpenStack installation must be a fresh installation. No users, projects, or domains are defined other than those that are created during the basic OpenStack installation.
- Use of OpenStack capabilities through the Cloud Orchestrator interface: Parallel use of the Cloud Manager with OpenStack installation through IBM Cloud Orchestrator and standard OpenStack is not supported. If a Cloud Manager with OpenStack installation is configured for use by Cloud Orchestrator, all user activities must be performed through the Cloud Orchestrator interface, and not through the OpenStack interfaces. The OpenStack self-service interface must be used only as described in the Cloud Orchestrator documentation.
- One OpenStack region per hypervisor type: Each region must have only one hypervisor type. Cloud Orchestrator does not support the use of multiple hypervisor types within a region.
- Customized simple token setup: The Cloud Manager with OpenStack distribution provides the simple token enhancement that is required and used by Cloud Orchestrator. For more information about how to set up the simple token, see “Customizing Passwords and Secrets” and “Data Bags” in the Cloud Manager with OpenStack documentation.
  https://www.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_customize_passwords.html
  https://www.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_example_databags.html
- No self-service user interface extension: Do not install the self-service user interface extension for Cloud Manager with OpenStack, because this deployment will use Cloud Orchestrator for the self-service user interface.
- Supported hypervisor: Cloud Orchestrator supports the following hypervisors in a Cloud Manager with OpenStack environment: Hyper-V, KVM, PowerVC, and z/VM.
Figure 12 shows the typical deployment topology used for a multihypervisor environment.

**Figure 12.** Typical Topology for a Multihypervisor Environment

The topologies use the following terminology:

- **Deployment server:** The deployment server is the system on which you install the Cloud Manager with OpenStack product. The deployment server runs the Chef server and maintains the Cloud Manager with OpenStack repository.
- Cloud Manager with OpenStack repository: The repository is the location at which all the Cloud Manager with OpenStack components are stored. The repository includes Chef resources, Red Hat Package Managers (RPMs), commands, licenses, dependencies, and more. As an administrator, you manage your topology by using this repository.

- Node: A node is a system that is part of your topology. The applicable Cloud Manager with OpenStack components are deployed to the node according to your cloud environment.

- Region: In a multiregion environment, a region provides a way for you to define separate cloud deployment areas (separated by hypervisor, for example), while still sharing some OpenStack infrastructure, such as Keystone.

The repository, along with the associated OpenStack Chef cookbooks and dependencies, provide the basic building blocks that you can use to deploy Cloud Manager with OpenStack.

The topologies support the following operating systems, databases, message queue, and OpenStack components:

**Note:** The support information provided here applies to the minimum controller +n computing, high-availability controller +n computing, and distributed database topologies only. Also, not all support information applies to all of the topologies previously listed.

- OpenStack components: Identity, image, network, computing, orchestration, block storage, telemetry, and dashboard
- OpenStack networking: Neutron
- OpenStack computing scheduler: Computing scheduler filters and IBM PRS
- Operating system for OpenStack controller: Red Hat Enterprise Linux (RHEL) 7.1
- Database: IBM DB2 (default), MySQL, and MariaDB
- Message queue: RabbitMQ (default) and Apache Qpid
- Hypervisor types: KVM, IBM PowerKVM, QEMU, z/VM, and Hyper-V
- Virtualization manager: PowerVC
- Cloud Manager: Self-service; enabled by default

### Hardware Prerequisites for IBM Cloud Orchestrator

Table 2 lists the hardware prerequisites for a typical IBM Cloud Orchestrator installation.

<table>
<thead>
<tr>
<th>Server</th>
<th>Processor (vCPU)</th>
<th>Memory (GB)</th>
<th>Total Free Hard-Disk Space (GB)</th>
<th>Free Hard-Disk Space by Partition (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM Cloud Orchestrator Server</td>
<td>4</td>
<td>8</td>
<td>66</td>
<td>/ 10, /home 10, /opt 36, /tmp 10</td>
</tr>
</tbody>
</table>

For more information, refer to [http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_hw_prerequisites.html](http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_hw_prerequisites.html).
Software Prerequisites for IBM Cloud Orchestrator

Verify that your IBM Cloud Orchestrator server environment meets the software prerequisites for your Cloud Orchestrator installation.

- Cloud Orchestrator must be installed on the Cloud Orchestrator server. The server operating system must be RHEL 7.0 or 7.1. The architecture must be x86 64-bit.
- The Cloud Orchestrator installer must access one of the following RHEL repositories:
  - Registered Red Hat network
  - Customer-provided Yum repository
  - RHEL ISO
- You must configure the Cloud Orchestrator server as follows:
  - The operating system must be installed with at least the standard minimum installation package. When installing the RHEL OS for the Cloud Orchestrator server, the minimum installation package is sufficient, because the Cloud Orchestrator deployment script installs the required packages from the corresponding Yum repository or Red Hat ISO files.
  - Host name resolution must work between the Cloud Orchestrator server and the OpenStack servers. You can configure the Cloud Orchestrator server with the corporate DNS. If no corporate DNS is available, you must update the /etc/hosts file on each of the required servers (for example, Cloud Orchestrator server, OpenStack controllers, and computing nodes) to include all the Cloud Orchestrator and OpenStack server hosts. Each entry in the /etc/hosts file must specify both the fully qualified domain name and the host name, in that order. To verify that you configured the /etc/hosts file correctly, run the commands listed in Table 3.
  - The ports used by Cloud Orchestrator must be open, not blocked. For a list of the ports that must be open, see "Ports Used by IBM Cloud Orchestrator."
  - (VMware only) If you install Cloud Orchestrator on a VMware virtual machine, install VMware Tools to improve the performance of the virtual machine.

For more information, refer to [http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_sw_prereqs.html](http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_sw_prereqs.html).

Table 3. Commands for Verifying the Host File Configuration

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>host &lt;IP_address&gt;</td>
<td>This command must return the fully qualified domain name (FQDN) of the server (for example, ico_server.subdomain.example.com).</td>
</tr>
<tr>
<td>hostname –fqdn</td>
<td>This command must return the same FQDN as the previous command.</td>
</tr>
<tr>
<td>hostname</td>
<td>This command must return the first part of the FQDN that is the host name (for example, ico_server).</td>
</tr>
</tbody>
</table>

Hardware Prerequisites for IBM Cloud Manager with OpenStack

The IBM Cloud Manager with OpenStack server is supported on the following platforms:

- Intel x86 64-bit (Linux)
- IBM POWER6, POWER7, or POWER8 (PowerLinux)
- IBM z Systems (IBM z/Architecture)
Table 4 lists both the minimum hardware requirements and recommended minimum production hardware requirements for the Cloud Manager with OpenStack deployment server, controller, database, and computing node components.

The minimum requirements listed indicate the absolute minimum hardware levels needed when running with only 5 to 10 concurrent users. The recommended minimum production requirements are recommendations to support a small cloud. As with any software solution, the hardware needs to be properly sized for the specific customer scenario.

Table 4. Minimum Hardware Requirements

<table>
<thead>
<tr>
<th>Component</th>
<th>Minimum Hardware Requirements</th>
<th>Recommended Minimum Hardware Production Requirements</th>
</tr>
</thead>
</table>
| IBM Cloud Manager with OpenStack deployment server¹ | • 4 CPUs  
• Free disk space: 9 GB for /opt/ibm/cmwo  
• Free disk space: 21 GB of temporary space while installing  
• Chef server: 5.0 GB of free disk space in /opt; 5.0 GB of free disk space in /var.  
• 4 GB of physical memory | • 8 CPUs  
• 40 GB of free disk space  
• 8 GB of physical memory |
| OpenStack controller                          | • 4 CPUs  
• 16 GB of physical memory  
• 8 GB of free disk space for installed code and empty databases²  
• 1 network interface card (NIC) without using OpenStack Ironic  
• 2 NICs when using Ironic | • 4 CPUs  
• 16 GB of physical memory  
• 8 GB of free disk space for installed code and empty databases (6 GB for DB2 and databases)²  
• 3 NICs |
| Standalone IBM DB2                            | 6 GB for DB2 and empty databases²                                                           | 6 GB for DB2 and empty databases³                                                 |
| Computing nodes                               | The following computing nodes are installed from Cloud Manager with OpenStack. For specific requirements, see information about the applicable virtualization environment.  
• KVM or QEMU prerequisites  
• KVM for z Systems prerequisites  
• PowerKVM prerequisites  
• Hyper-V prerequisites  
• z/VM prerequisites  
Other supported virtualization environments:  
• PowerVC prerequisites  
• VMware prerequisites |                                                                                           |

¹ Requirements are for the Cloud Manager with OpenStack deployment server only (for example, Chef server). If other servers are installed on the same system, the requirements would need to be higher to account for the additional needs of the other components installed and running there.

²³ For installing DB2 and empty databases, 6 GB is mandatory for overall disk and volume space to accommodate the temporary file during installation. However, if you mount a separate disk and volume for a specific directory, then you must consider the following disk and volume size requirements:

- /opt: 1.2 GB available
- /home: 2.5 GB available
- /tmp: 2.0 GB available
Of particular concern is the /tmp directory, because the DB2 package is copied and extracted before installation and is removed afterward. If the disk or volume mounted to it or the overall space is insufficient, the DB2 installation will fail, and all intermediate results will be eliminated. Note that the sizes listed here are minimum requirements, and you must adjust them if you have a larger configuration according to your actual deployment.

For detailed information about installation and prerequisites, refer to http://www-01.ibm.com/support/knowledgecenter/SST5SW_4.3.0/liaca/liaca_kc_welcome.html.

Configuration Changes to Support VMware Hypervisors
To support VMware hypervisors in an IBM Cloud Manager with OpenStack environment, set the following options in the VMware Nova configuration file:

- In the /etc/nova directory on the VMware OpenStack controller (the default name of the file is nova-vcenter-Cluster.conf):
  
  ```
  force_config_drive=always
  customization_enabled = false
  ```

- Insert `force_config_drive`: Set up in the [DEFAULT] section.
- Insert `customization_enabled`: Set up in the [vmware] section.

- Restart the computing service.

**Note:** The file name nova-vcenter-Cluster.conf is the default name when you install Cloud Manager for the first time. Later you will see that each VMware cluster needs a new Nova computing service. Typically, these Nova computing services are configured with a file name that identifies the vCenter cluster that the service is managing. As a best practice, make the cluster name part of the Nova configuration file name.

**Note:** If you want to deploy Microsoft Windows images, use the latest version of the cloudbase-init tool while creating the Windows image to support the OpenStack updates related to the network interface management of Windows images.

**Note:** If you install fix packs for Cloud Manager with OpenStack, or if you perform other maintenance work that updates the topology through Knife, the configuration for Cloud Orchestrator may be overwritten. In this case, rerun the configuration scripts.

**Note:** If you add a new region to Cloud Manager with OpenStack after it was configured for Cloud Orchestrator, the new region will not be correctly set up to work with Cloud Orchestrator. Run the configuration scripts for the new region.

Configuring IBM Cloud Manager for VMware Integration

After IBM Cloud Orchestrator and Cloud Manager with OpenStack are installed, the environment is ready to start configuration to discover the VMware cloud. The following sections present the steps for configuring Cloud Manager with OpenStack to work with the VMware environment. Here is a summary of the steps:

1. Connect to multiple clusters. This task maps OpenStack availability zones to VMware clusters.
2. Configure VMware discovery.
3. Configure the VMware distributed virtual switch (DVS) mechanism driver.
4. Verify that IBM Cloud Manager discovers the vCenter environment and networks.
5. Create a virtual machine template as a snapshot image for VMware instances.
6. Assign availability zones to domains and projects.

Advanced configuration steps include the following:

- Connect to different data stores in the same cluster.
- Connect to resource pools.
- Enable storage vSphere DRS.
- Enable data-store random selection.
- Configure OpenStack to support thin provisioning.
- Configure OpenStack to support linked clones.

These advanced configuration steps are described here: [http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.doc.2.5/c_advanced_config_vmware_region.html?lang=en](http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.doc.2.5/c_advanced_config_vmware_region.html?lang=en).

**Connecting to Multiple Clusters**

When you configure a VMware region, you can connect to multiple clusters defined in the same vCenter. Each cluster is set as a new availability zone in the same region. To discover the VMware environment in Cloud Manager, follow the steps described in this section.

Because an OpenStack Nova computing service can connect to only one cluster in OpenStack, you must create a new Nova computing service to connect to your new cluster. The new cluster is set as a new host aggregate in a new availability zone. The examples in the following procedure use the names listed in Table 5.

<table>
<thead>
<tr>
<th>Table 5. Names Used in Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td>IBMICO-Availability-Zone</td>
</tr>
<tr>
<td>IBMICO-HostAggregate</td>
</tr>
<tr>
<td>nova-ibmico-cluster.conf</td>
</tr>
<tr>
<td>openstack-nova-compute-ibmico-cluster</td>
</tr>
<tr>
<td>IBMICO</td>
</tr>
</tbody>
</table>

1. Create the host aggregate and associate it with a new availability zone by running the following command:
   ```bash
nova aggregate-create IBMICO-HostAggregate IBMICO-Availability-Zone
   ```

   This command also creates a new availability zone named IBMICO-Availability-Zone. Now you must create a new OpenStack Nova computing service.
2. Create a copy of the Nova computing service configuration file related to VMware:
   
   ```
   cp /etc/nova/nova-discovery.conf /etc/nova/nova-ibmico-cluster.conf
   ```

   Change the file ownership by running the following command:
   
   ```
   chown nova:nova /etc/nova/nova-ibmico-cluster.conf
   ```

3. Modify the `/etc/nova/nova-ibmico-cluster.conf` file to this configuration:

   ```
   [DEFAULT]
   default_availability_zone = IBMICO-Availability-Zone
   default_schedule_zone = IBMICO-Availability-Zone
   storage_availability_zone = IBMICO-Availability-Zone
   host = IBMICO <the name of the new cluster in your vCenter>
   # Use a host name different from the VMware OpenStack Controller to
   # avoid conflict with the first cluster configuration.
   
   [vmware]
   host_ip = <your vCenter IP address or host name>
   cluster_name = <the name of the new cluster in your vCenter>
   ```

4. Create a copy of the `/etc/init.d/openstack-nova-compute-vmware` file:

   ```
   cp /etc/init.d/openstack-nova-compute-vmware /etc/init.d/openstack-nova-compute-ibmico-cluster
   ```

   Change the file ownership by running the following command:
   
   ```
   chown nova:nova /etc/init.d/openstack-nova-compute-ibmico-cluster
   ```

5. In the `/etc/init.d/openstack-nova-compute-ibmico-cluster` file, modify the `tail`, `prog`, and `config` parameters:

   ```
   tail= ibmico-cluster
   proj=nova
   suffix=compute
   prog= openstack-nova-compute-ibmico-cluster
   exec="/usr/bin/$proj-$suffix"
   config="/etc/nova/nova-ibmico-cluster.conf"
   pidfile="/var/run/$proj/$proj-$suffix-$tail.pid"
   logfile="/var/log/$proj/$suffix-$tail.log"
   ```

6. Run the following commands to start the services:

   ```
   /etc/init.d/openstack-nova-compute-ibmico-cluster start
   ```

   Check the status of the process:
   
   ```
   /etc/init.d/openstack-nova-compute-ibmico-cluster status
   ```

   For more information, check the `/var/log/nova/compute-ibmico-cluster.log` file.

7. Add the host that you specified in step 3 for the new computing service to the host aggregate by running the following command:

   ```
   nova aggregate-add-host IBMICO-HostAggregate IBMICO
   ```

   In the preceding command, IBMICO is the cluster name.
8. Verify that the new service is up and running. Run the following command to verify that the new availability zone, which is named IBMICO-Availability-Zone, is shown:
   nova availability-zone-list
   For troubleshooting, see the /var/log/nova/compute-ibmico-cluster.log file.

9. For more information, see http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_connect_multiple_clusters.html.

Figure 13 shows the host aggregates and availability zones on the Cloud Manager dashboard.

Figure 13. Host Aggregates and Availability Zones on IBM Cloud Manager Dashboard

Configuring VMware Discovery
Follow the steps shown here to configure VMware discovery.

1. Back up the configuration file /etc/nova/vmware-discovery.conf using the following command:
   cd /etc/nova/
   cp vmware-discovery.conf vmware-discovery.conf_org

2. Modify /etc/nova/vmware-discovery.conf as shown here.
   a. Add vSwitch information in physical_network_mappings in the [network] section. For example if you have vSphere standard switch vSwitch0 and vSwitch1 and VDS VMM, then the configuration looks like this:
      Physical_network_mappings = physnet1:vSwitch0,physnet2:vSwitch1,physnet3:VMM
   b. Add the cluster name in clusters in the [vmware_discovery] section.
      Note: You can add comma-separated names if you have more than one cluster: for example, clusters = cluster1,cluster2.
      Verify sure the permissions for the configuration file by running ls -la in the /etc/nova folder. The vmware-discovery.conf file should show the owner as nova:nova.

3. Restart the Nova discovery service with the following commands:
   service nova-discovery restart

4. Verify that the service is restarted and running:
   service nova-discovery status
   The status should be listed as Active (running).
   Alternatively, use these commands:
ps -ef | grep discovery

The process should be listed with an active process ID.

5. Verify that the discovery.log file is being updated.

6. Verify that the file /var/log/nova/discovery.log is being updated.

7. For more information, refer to http://www-01.ibm.com/support/knowledgcenter/SST55W_4.3.0/liaca/liaca_cfg_vmware_discovery.html.

Configuring VMware DVS Mechanism Driver (Enabling VMware VDS on IBM Cloud Manager with OpenStack Neutron)

This process is needed to configure Cloud Manager with OpenStack to discover all the vCenter networks.

1. Back up the current Neutron configuration file /etc/neutron/plugins/ml2/ml2_conf.ini using the following commands:
   
   cd /etc/neutron/plugins/ml2/
   cp ml2_conf.ini ml2_conf.ini_org

   2. To use the VDS mechanism, verify that your system meets the following requirements:
      
      a. The Neutron server is installed.
      b. The managed-to vCenter has a VDS that is connected with managed clusters or hosts.
      c. The neutron-vmware-driver package is installed.

   3. Configure the following scripts manually. Perform the configuration at the Cloud Manager command line.

     VCENTER_HOST_IP=<$vcenter_host_ip>
     VCENTER_HOST_USERNAME=<$vcenter_host_username>
     VCENTER_HOST_PASSWORD=<$vcenter_host_password> it should be encrypted. You can copy this from file VMwareDiscovery.conf
     NET_MAPS=<$physical_network_mappings> You can copy this from the vmware-discovery.conf
     DVS_MECH_DRIVER="neutron.plugins.ml2.drivers.mech_dvs.driver:VMwareDVSMechanismDriver"
     NEUTRON_ML2_CONF_FILE=/etc/neutron/plugins/ml2/ml2_conf.ini

     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_type_drivers flat,vlan
     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_tenant_network_types flat,vlan
     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_mechanism_drivers dvs
     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_type_flat flat_networks physnet1
     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_type_vlan network_vlan_ranges physnet1:1:4094,physnet2:1:4094,physnet3:1:4094 (This should match the physical_network_mappings number of switches. In this case we have added vSwitch0,vSwitch1,and VMM. So there are 3 vlan ranges we have defined)

     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_vmware_host_ip $VCENTER_HOST_IP
     openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_vmware_host_username
$VCENTER_HOST_USERNAME
openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_vmware host_password
$VCENTER_HOST_PASSWORD
openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_vmware task_poll_interval 5.0
openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_vmware api_retry_count 10
openstack-config --set $NEUTRON_ML2_CONF_FILE ml2_vmware network_maps
$NET_MAPS
openstack-config --set $NEUTRON_EGG neutron.ml2.mechanism_drivers dvs
$DVS_MECH_DRIVER

Note: Replace <$vcenter_host_ip>, <$vcenter_host_username>, <$vcenter_host_password>, and <$physical_network_mappings> with real values. The physical network mappings represent a map of the Neutron physical network name to the DVS in vCenter: for example, physnet1:dvSwitch.

Verifying Discovery of the VMware Environment and Networks
Follow the steps presented here to verify discovery of the VMware environment and networks.

1. Verify the permissions of the following files. Both files should list the owner as neutron:neutron.
   /etc/neutron/plugin.in (This is a soft link to the following file.)
   /etc/neutron/plugins/ml2/ml2_conf.ini
2. Restart the Neutron server with the following command:
   service neutron-server restart
3. Verify that the Neutron service is restarted and running:
   service neutron-server status
   The status should be listed as Active (running).
4. Verify that the server.log file is being updated.
5. Verify that the file /var/log/neutron/server.log is being updated.
6. For more information, refer to link http://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_cfg_dvs.html.

Figure 14 shows the Networks display on the Cloud Manager dashboard to verify that all port groups from vCenter have been discovered.

Figure 14. Networks Display on IBM Cloud Manager Dashboard to Verify Discovery of All Port Groups from VMware vCenter
Creating a Virtual Machine Template as a Snapshot Image for VMware Instances

For details about creating a virtual machine template as a snapshot for VMware instances, refer to [https://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_create_vm_template.html](https://www-01.ibm.com/support/knowledgecenter/SST55W_4.3.0/liaca/liaca_create_vm_template.html).

Figure 15 shows the Cloud Manager dashboard Images list.

**Figure 15.** IBM Cloud Manager Dashboard Images

![Image Details Discovered by IBM Cloud Manager from vCenter](image1.png)

Figure 16 shows the image details discovered by Cloud Manager from vCenter.

**Figure 16.** Image Details Discovered by IBM Cloud Manager from VMware vCenter

![Image Details Discovered by IBM Cloud Manager from VMware vCenter](image2.png)
Assigning Availability Zones to Domains and Projects

Follow the steps here to assign availability zones to domains and projects. First, log in to Cloud Manager: under Hypervisors, you should see the newly added hosts and clusters (Figure 17).

Figure 17. IBM Cloud Manager Dashboard

The cloud administrator creates domains to organize projects, groups, and users. Domain administrators can update and delete resources in a domain. Here is the procedure:

1. Log in to the OpenStack dashboard as the cloud administrator.
2. In the left navigation pane, choose Identity > Domains. The Domains page opens.
3. Select Create Domain. The Create Domain window is displayed.
4. Specify the domain name and, optionally, a domain description.
5. Optional: Clear the Enabled check box to disable the domain. If the domain is disabled, the domain administrator cannot create, update, or delete resources that are related to the domain. New domains are enabled by default.
6. Choose Create Domain (Figure 18).

Figure 18. Creating a Domain

7. After the domain is created, you can assign a zone to the domain. Use the Edit option on the Actions menu for the selected domain to assign an availability zone to that domain (Figure 19).
8. For detailed steps for assigning a zone to a domain, refer to http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/t_assigning_a_zone_to_a_domain.html?lang=en.

9. To assign a zone to a project, refer to http://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/t_assigning_a_zone_to_a_project.html?lang=en.

Assigning a zone to a project enables users within a zone to access a specific project. Here is the procedure:

1. Log in to the OpenStack dashboard as the cloud administrator.
2. Open the Domains page by choosing Identity > Domains in the navigation pane.
3. On the Domains page, find the entry for the domain and select Set Domain Context in the Actions column. The Identity Panel group is now in the context of the selected domain, and the Domains page is also changed. You are now working within the context of the domain that you created.
4. Choose Identity > Projects (Figure 20).
5. In the Actions column in the table for the project, click the arrow icon. Then click the Edit Project option.
6. Click the Availability Zones tab. The available zones and the assigned zones are listed in the following format: Zone_Name - Region_Name.
7. To assign a zone to a domain, in the Available Zones list click the plus button beside the zone name. The selected zone moves to the Assigned Zones list. To change an assigned zone back to an available zone, select the minus button beside the zone name. Use the Filter field to search for specific zones.
8. When you have assigned all zones, click Save.

Figure 20. Assigning Zones to Projects
Verify the domain and project assignments through Cloud Orchestrator (Figure 21).

**Figure 21.** Verifying Domain and Project Assignments in IBM Cloud Orchestrator

Deploying Applications in Cisco ACI with IBM Cloud Manager Integrated for VMware

After Cisco ACI and IBM Cloud Manager are installed and set up to work with VMware vCenter clusters, prepare the environment for deploying virtual machines to host applications. The following sections provide detailed steps. A summary of the process is shown here:

1. Create tenants, private networks, and bridge domains.
2. Create application profiles, EPGs, and contracts.
3. Assign subnets to discovered vCenter networks through Cloud Orchestrator.
4. Deploy virtual machines with Cloud Orchestrator.
5. Verify that Cisco ACI discovers the virtual machines deployed through Cloud Orchestrator and verify that communication between virtual machines occurs according to application policy.
6. Manage virtual machine instances.

**Creating Tenants, Private Networks, and Bridge Domains**

Follow these steps to create tenants, private networks, and bridge domains.

1. On the menu bar, choose Tenants and perform the following actions:
   b. In the Name field, add the tenant name (IBM_ICO); then click Next.

   **Note:** Create a security domain so that it allows only users in that security domain to have access.

2. Click the plus sign next to Security Domains to open the Create Security Domain dialog box. Then perform the following actions:
   a. In the Name field, specify the security domain name (for instance, ExampleCorp).
   b. Click Submit. In the Create Tenant dialog box, select the check box for the security domain that you created; then click Next.

3. In the Network window, perform the following actions:
   a. Click the plus sign to add the network.
b. In the **Create New Network** area, specify the private tenant network name (IBM_ICO_Network1); then click **Next**. You are prompted to specify a bridge domain in the network.

4. In the **Specify Bridge Domain for the Network** area, perform the following actions:
   a. In the **Name** field, enter a name (IBM_ICO_BD1).
   b. Expand **Subnets**, and in the **Gateway Address** field, enter the address. Click **Update** and then click **OK**.

5. Click **Next** and perform the following actions:
   a. Confirm that the private network (IBM_ICO_Network1) is created and is associated with the bridge domain (IBM_ICO_BD1).
   b. In the **Network** area, click **Finish**.

6. To validate that the tenant has a private network and bridge domain, in the menu bar under the **Tenants** tab, click the name of the new tenant that you created. In the navigation pane, expand the tenant name. Choose **Networking > Bridge Domains**, and you should see the new bridge domain displayed. Under **Private Networks**, the new network is displayed (Figure 22).

**Figure 22.** Verifying Tenants, Bridge Domains, and Networks in the Navigation Pane

**Creating Application Profiles, EPGs, and Contracts**
The following application profile is created here to demonstrate Cisco ACI integration with Cloud Manager and vCenter:

- The tenant role is to host the application 3Tier_Application (Figure 23).
- Application profile, EPGs, and contracts are used.
- The tenant consists of one application profile.
- The application profile 3Tier_Application consists of three EPGs: Application_EPG, Webserver_EPG, and Database_EPG.
  - Webserver_EPG is attached to the VMM domain and provides a port group on the VDS to connect the web servers.
  - Application_EPG is attached to the VMM domain and provides a port group on the VDS to connect the application servers.
Database_EPG is attached to the VMM domain and provides a port group on the VDS to connect the database servers.

- Appropriate contracts are defined to allow traffic between the various application tiers.

**Figure 23.** 3Tier_Application Web Application Profile

---

**Create a Filter Using the GUI**

Create three separate filters. In this example, the filters are HTTP, RMI, and SQL. This task shows how to create the HTTP filter. The process for creating the other filters is identical.

Before you begin, verify that the tenant, network, and bridge domain have been created. Then follow the procedure shown here:

1. On the menu bar, choose Tenants. In the navigation pane, expand the tenant and Security Policies, right-click Filters, and click Create Filter.

**Note:** In the navigation pane, you expand the tenant where you want to add filters.

2. In the Create Filter dialog box, perform the following actions:
   a. In the Name field, enter the filter name (AppServer_Filter).
   b. Expand Entries, and in the Name field, enter the name (app_filter).
   c. From the EtherType drop-down list, choose the EtherType (IP).
   d. From the IP Protocol drop-down list, choose the protocol (tcp).
   e. From the Destination Port/Range drop-down lists, choose 1443 in the From and To fields (http).
   f. Click Update and then click Submit. The newly added filter appears in the navigation pane and in the work pane.
3. In the **Name** field, expand **Entries**. Follow the same process to add another entry (app_icmp) with **IP Protocol** set to **icmp**. Then click **Update**. This new filter rule is added.

4. Follow the same process as in the previous steps to create two more filters (WebServer_Filter and DatabaseServer_Filter). Use the parameters provided in Figures 24 and 25.

**Figure 24.** WebServer_Filter Properties

![WebServer_Filter Properties](image)

**Figure 25.** DatabaseServer_Filter Properties

![DatabaseServer_Filter Properties](image)

Figure 26 shows the created filters.

**Figure 26.** Creating Filters

![Creating Filters](image)
Create a Contract Using the GUI
Use the procedure shown here to use the GUI to create a contract.

1. On the menu bar, choose Tenants and the name of the tenant on which you want to operate. In the navigation pane, expand the tenant and Security Policies.

2. Right-click Contracts and choose Create Contract.

3. In the Create Contract dialog box, perform the following tasks:
   a. In the Name field, enter the contract name (App_Contract).
   b. Click the plus sign next to Subjects to add a new subject.
   c. In the Create Contract Subject dialog box, enter a subject name in the Name field (App-Contract_Sub).
   
   **Note:** This step associates the filters that were created earlier with the contract subject.
   d. In the Filter Chain area, click the plus sign next to Filters.
   e. In the dialog box, from the drop-down menu, choose the filter names (app_filter and app_icmp); then click Update.

4. In the Create Contract Subject dialog box, click OK.

5. Create two more contracts—Webserver_Contract and Database_Contract—following the same procedure as in the preceding steps. Use the subject names and filters as shown in Figure 27.

**Figure 27.** Creating Contracts

Create an Application Profile Using the GUI
Use the following procedure to use the GUI to create an application profile.

1. On the menu bar, choose Tenants. In the navigation pane, expand the tenant, right-click Application Profiles, and choose Create Application Profile.

2. In the Create Application Profile dialog box, in the Name field, add the application profile name (3Tier_Application).
Create EPGs Using the GUI
The port that the EPG uses must belong to one of the VMM or physical domains associated with the EPG. Use the following procedure to create the EPGs.

1. Expand EPGs. In the Create Application EPG dialog box, perform the following actions:
   a. In the Name field, add the EPG name (Application_EPG).
   b. In the Bridge Domain field, choose the bridge domain from the drop-down list (IBM_ICO_BD1).
   c. Expand Associated Domain Profiles and from the drop-down list, choose the domain profile name (VMMICO). This action deploys the EPG in the vCenter in the chosen VMM domain.
   d. Click Update and then click OK.

2. In the Create Application Profile dialog box, create two more EPGs. The three EPGs should be Database_EPG, Web_EPG, and Application_EPG in the same bridge domain and data center (Figure 28).

Figure 28. Creating EPGs

Consume and Provide Contracts Using the GUI
You can associate contracts that were created earlier to create policy relationships between the EPGs. Follow the procedure presented here.

Note: The Database_EPG, Application_EPG, and Webserver_EPG EPGs are displayed as icons.

1. Click and drag across the APIC GUI window from the Database_EPG EPG to the Application_EPG EPG. The Add Consumed Contract dialog box is displayed.
2. In the Name field, from the drop-down list, choose Database_Contract contract. Click OK. This step enables Database_EPG to provide the Database_Contract contract, and Application_EPG to consume the Database_Contract contract.
3. Click and drag across the APIC GUI screen from the Application_EPG EPG to the Webserver_EPG EPG. The Add Consumed Contract dialog box is displayed.
4. In the Name field, from the drop-down list, choose App_Contract contract. Click OK. This step enables Application_EPG to provide the App_Contract contract, and Webserver_EPG to consume the App_Contract contract.

5. Click the Webserver_EPG EPG icon and click the plus sign in the Provided Contracts area. The Add Provided Contract dialog box is displayed.

6. In the Name field, from the drop-down list, choose Webserver_Contract contract. Click OK. Click Submit. You have created a three-tier application profile called 3Tier_Application.

7. To verify your work, in the navigation pane, navigate to and click 3Tier_Application under Application Profiles. In the work pane, the three EPGs—Application_EPG, Database_EPG, and Webserver_EPG—are displayed.

8. In the work pane, choose Operational > Contracts. The EPGs and contracts are displayed in the order that they are consumed and provided (Figure 29).

Figure 29. Conserving and Providing Contracts

Creating Subnets in IBM Cloud Manager for Discovered Networks

Follow the steps presented here to create subnets in Cloud Manager for discovered networks.

1. Log in to Cloud Manager. Choose Project > Network > Networks.

   Verify that the application profile and networks created in Cisco ACI are visible in Cloud Manager (Figure 30).
2. Click the discovered network (Figure 31).

Figure 31. IBM Cloud Manager: Network Details
3. In the **Subnets** section, click **Create Subnet** (Figure 32)

**Figure 32.** IBM Cloud Manager: Create Subnet

4. Provide a subnet name and network address (for example, the 10.1.1.0/24 CIDR address); then click **Next** (Figure 33).

**Figure 33.** IBM Cloud Manager: Create Subnet
5. In the **Subnet Details** section, provide a valid DNS name server address.

6. In the **Subnet Details** section, deselect **Enable DHCP**; then click **Create**.

**Deploying Virtual Machines with IBM Cloud Orchestrator**

Use the procedure shown here to deploy virtual machines with Cloud Orchestrator.

1. Log in to Cloud Orchestrator. Choose **Self-Service Catalog > Deploy Cloud Services** (Figure 34).

   **Figure 34.** IBM Cloud Orchestrator: Self-Service Catalog

2. Click **Deploy Single Virtual Server** (Figure 35).

   **Figure 35.** IBM Cloud Orchestrator: Self-Service Catalog

3. Configure the following fields (Figure 36):
Figure 36. IBM Cloud Orchestrator: Deploy Single Virtual Server

![Figure 36](image)

a. Enter the **server name**.

b. Select **Image** from the drop-down menu.

c. Select **Availability Zone** from the drop-down menu. Here you should see the availability zone you created earlier.

d. Select the flavor from the drop-down menu.

e. Select the network. Here you should see the port group created on the VDS by the APIC application profile.

f. Click **Deploy**. The virtual machine will be created on the VMware cluster and ESXi through Cloud Manager.

4. To view the status of the request in Cloud Orchestrator, select the **Requests** window and click the request in the menu at the left. The status of deployment will be displayed (Figure 37).

Figure 37. Viewing the Deployment Status
5. Repeat the preceding steps in this section to deploy virtual machines for each EPG.

6. You can view the deployed instances in Cloud Orchestrator, Cloud Manager, and Cisco ACI.
   a. To view instances in Cloud Orchestrator, select Resources and verify that the deployed virtual machines are shown in the list and are listed as Active (Figure 38).

   Figure 38. Viewing Deployed Virtual Machines in IBM Cloud Orchestrator

   b. To view virtual machine instances in Cloud Manager, select Instances and verify that the deployed virtual machines are listed (Figure 39).

   Figure 39. Viewing Deployed Virtual Machines in IBM Cloud Manager

Verifying That Cisco ACI Discovers the Virtual Machines and That Communication Follows Application Policy

To verify that Cisco ACI discovered the virtual machines deployed by Cloud Orchestrator, follow this procedure.

1. From VM Networking, select Inventory; then navigate to the VMM domain, select Hypervisors, and view the virtual machines (Figure 40).
2. Select each virtual machine deployed from Cloud Orchestrator and verify that each one belongs to the correct EPG port group. Double-click the virtual machine name in the menu bar at the left to view virtual machine details (Figure 41).

3. Test the connectivity between the virtual machines to verify that it follows the contracts created on the APIC. The Cloud Orchestrator dashboard displays the request status and use statistics (Figure 42).
Managing Virtual Machine Instances

After a virtual machine is deployed through IBM Cloud Orchestrator, you can perform lifecycle management as described here.

To manage virtual machines, first perform the following steps:

1. Log in to the self-service user interface as an end user.
2. Click Resources.
3. Click Virtual Machines to display a table of virtual machines.
4. From the region list, select a region. The table displays only the virtual machines in the specified region. From this view, you can perform the actions described here (Figure 43).

To start one or more virtual machines, follow these steps:

1. In the table, select one or more virtual machines that have the Shutoff status.
2. In the Actions menu at the left side of the table, click Start.
   
   **Note:** This action is available only if all the selected virtual machines have the Shutoff status.

To stop one or more virtual machines, follow these steps:

1. In the table, select one or more virtual machines that have the Active status.
2. In the Actions menu at the left side of the table, click Stop.
   
   **Note:** This action is available only if all the selected virtual machines have the Active status.

To delete one or more virtual machines, follow these steps:

1. In the table, select one or more virtual machines that have the Active status.
2. In the Actions menu at the left side of the table, click Delete.
   
   **Note:** This action is available only if all of the selected virtual machines have the Active status.

Figure 43. Managing Virtual Machines

For more information about managing virtual machine instances, refer to [https://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_managing_vms_instances.html](https://www-01.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/c_managing_vms_instances.html).