



Cisco Cloud Network Controller for Azure User Guide, Release 26.0(x)

First Published: 2021-09-20

Last Modified: 2023-02-28

Americas Headquarters

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



Trademarks

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at:

<http://www.cisco.com/go/softwareterms>. Cisco product warranty information is available at <http://www.cisco.com/go/warranty>. US Federal Communications Commission Notices are found here <http://www.cisco.com/c/en/us/products/us-fcc-notice.html>.

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 products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and-if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

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: [www.cisco.com go trademarks](http://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. (1721R)



CONTENTS

PREFACE

Trademarks iii

CHAPTER 1

New and Changed Information 1

New and Changed Information 1

CHAPTER 2

About Cisco Cloud Network Controller 3

Overview 3

Understanding the overlay-2 (Secondary) VRF 4

External Network Connectivity 5

Understanding Supported Routing and Security Policies 6

Routing and Security Policies: Releases Prior to Release 25.0(1) 6

Routing and Security Policies: Release 25.0(1) 7

Routing Policies: Release 25.0(2) 9

Source Interface Selection for Tunnels 11

Guidelines and Limitations 11

About the Cisco Cloud Network Controller GUI 13

CHAPTER 3

Cisco Cloud Network Controller Policy Model 19

About the CCNC Policy Model 19

Policy Model Key Characteristics 19

Logical Constructs 20

The Cisco CCNC Policy Management Information Model 21

Tenants 23

Understanding Tenants, Identities, and Subscriptions 24

Support for Inter-Tenant Shared Services in Hybrid Cloud Environments 26

Cloud Context Profile 29

CCR	29
About the Cisco Catalyst 8000V	29
Changing the Number of CCRs	31
Private IP Address Support for Cisco Cloud Network Controller and CCR	33
VRFs	33
Cloud Application Profiles	36
Cloud Endpoint Groups	37
Cloud Service Endpoint Groups	38
About Service Types	40
About Deployment Types	43
Security Groups	45
Guidelines and Limitations for ASGs and NSGs	46
Security Rules	47
NSG Behavior With Software Upgrades or Downgrades	48
Contracts	49
Comma-separated Filters Support for Contract Rule Consolidation	50
Filters and Subjects Govern Cloud EPG Communications	51
About the Cloud Template	52
Managed Object Relations and Policy Resolution	55
Default Policies	56
Shared Services	57
<hr/>	
CHAPTER 4	Configuring Cisco Cloud Network Controller Components 59
	About Configuring the Cisco Cloud Network Controller 59
	Configuring the Cisco Cloud Network Controller Using the GUI 59
	Creating a Tenant Using the Cisco Cloud Network Controller GUI 59
	Creating an Application Profile Using the Cisco Cloud Network Controller GUI 65
	Creating a VRF Using the Cisco Cloud Network Controller GUI 66
	Creating an External Network Using the Cisco Cloud Network Controller GUI 66
	Configuring the Global Inter-VRF Route Leak Policy 71
	Configuring Leak Routes Using the Cisco Cloud Network Controller GUI 72
	Configuring Inter-VRF Route Leaking Using the Cisco Cloud Network Controller GUI 72
	Configuring Leak Routes for Internal VRFs Using the Cisco Cloud Network Controller GUI 75
	Enabling Connectivity From the Azure Site to External Devices 77

Downloading the External Device Configuration Files	77
Enabling Connectivity From the Azure Site to the External Devices	77
Creating an EPG Using the Cisco Cloud Network Controller GUI	81
Creating an Application EPG Using the Cisco Cloud Network Controller GUI	81
Creating an External EPG Using the Cisco Cloud Network Controller GUI	85
Creating a Service EPG	88
Creating a Filter Using the Cisco Cloud Network Controller GUI	101
Creating a Contract Using the Cisco Cloud Network Controller GUI	103
Creating an Inter-Tenant Contract Using the Cisco Cloud Network Controller GUI	105
Configuring Network Security Groups Using the Cisco Cloud Network Controller GUI	108
Viewing Security Group Details	111
Specifying Consumer and Provider EPGs Using the Cisco Cloud Network Controller	112
Creating a Cloud Context Profile Using the Cisco Cloud Network Controller GUI	113
Configuring Virtual Machines in Azure	117
Creating a Backup Configuration Using the Cisco Cloud Network Controller GUI	118
Creating a Tech Support Policy Using the Cisco Cloud Network Controller GUI	122
Creating a Scheduler Using the Cisco Cloud Network Controller GUI	123
Creating a Remote Location Using the Cisco Cloud Network Controller GUI	125
Creating a Login Domain Using the Cisco Cloud Network Controller GUI	126
Creating a Security Domain Using the Cisco Cloud Network Controller GUI	128
Creating a Role Using the Cisco Cloud Network Controller GUI	128
Creating a Certificate Authority Using the Cisco Cloud Network Controller GUI	133
Creating a Key Ring Using the Cisco Cloud Network Controller GUI	135
Creating a Local User Using the Cisco Cloud Network Controller GUI	136
Managing Regions (Configuring a Cloud Template) Using the Cisco Cloud Network Controller GUI	139
Configuring Smart Licensing	141
Cloud Resources Naming	142
Variables Available for Naming Rules	142
Naming Rules Guidelines and Limitations	145
Viewing Cloud Resource Naming Rules	146
Configuring Cisco Cloud Network Controller Using the REST API	147
Creating a Tenant Using the REST API	147
Creating a Contract Using the REST API	149

Creating a Cloud Context Profile Using the REST API	150
Managing a Cloud Region Using the REST API	151
Creating a Filter Using the REST API	151
Creating an Application Profile Using the REST API	152
Configuring Network Security Groups Using the REST API	153
Creating an EPG Using the REST API	153
Creating a Cloud EPG Using the REST API	153
Creating an External Cloud EPG Using the REST API	154
Creating a Service EPG Using the REST API	155
Creating a Cloud Template Using the REST API	156
Configuring VRF Leak Routes Using the REST API	157
Configuring the Source Interface Selection for Tunnels Using the REST API	159
Defining Global Cloud Resource Naming Rules or Overriding Specific Object's Name	159

CHAPTER 5**Viewing System Details 163**

Monitoring VM Host Metrics	163
Monitoring VM Host Metrics Using the GUI	163
Monitoring VM Host Metrics Using the REST API	166
Viewing Application Management Details	167
Viewing Cloud Resource Details	168
Viewing Operations Details	169
Viewing Infrastructure Details	171
Viewing Administrative Details	172
Viewing Health Details Using the Cisco Cloud Network Controller GUI	174

CHAPTER 6**Deploying Layer 4 to Layer 7 Services 177**

Overview	177
About Service Graphs	177
Using Service Graphs with Cloud Native and Third-Party Services	178
About Application Load Balancers	179
About Network Load Balancer	180
Understanding Configurations for Multiple Frontend IP Addresses on the Azure Network Load Balancer	181
About Third-Party Load Balancers	182

About Allow All Traffic Option	183
Dynamic Server Attachment to Server Pool	184
About Inter-VNet Services	185
About Multinodes	185
About Layer 4 to Layer 7 Service Redirect	186
Passthrough Rules	186
Redirect Programming	186
Redirect Policy	187
Workflow for Configuring Redirect	187
Example Use Cases	188
Example Use Cases for Service Graphs with Cloud Native and Third-Party Services	203
Example Use Cases Without Redirect	203
Example Use Cases With Redirect	210
Guidelines and Limitations for Redirect	222
Adding a New CIDR to a Secondary VRF Using the Cisco Cloud Network Controller GUI	224
Deploying a Service Graph	226
Deploying a Service Graph Using the GUI	227
Creating Service Devices Using The Cisco Cloud Network Controller GUI	227
Creating a Service Graph Template Using the Cisco Cloud Network Controller GUI	234
Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI	236
Deploying a Service Graph Using the REST API	240
Creating an Internal-Facing Load Balancer Using the REST API	240
Configuring an Internet-Facing Load Balancer Using the REST API	241
Creating a Third-Party Firewall Using the REST API	242
Creating a Third Party Load Balancer Using the REST API	243
Creating a Service Graph Using the REST API for an Application Gateway	243
Creating a Service Graph Using the REST API for Azure Load Balancer	244
Creating a Service Graph Using the REST API for a Third Party Load Balancer	245
Creating a Multi-Node Service Graph Using the REST API	245
Creating a Multi-Node Service Graph With Redirect Using the REST API	248
Attaching a Service Graph Using the REST API	253
Configuring an HTTP Service Policy Using the REST API	253
Configuring a Key Ring Using the REST API	255
Creating an HTTPS Service Policy Using the REST API	256

CHAPTER 7	Cisco Cloud Network Controller Security	259
	Access, Authentication, and Accounting	259
	Configuration	259
	Configuring TACACS+, RADIUS, LDAP and SAML Access	260
	Overview	260
	Configuring Cisco Cloud Network Controller for TACACS+ Access	260
	Configuring Cisco Cloud Network Controller for RADIUS Access	261
	Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the Cisco Cloud Network Controller	263
	Configuring LDAP Access	263
	Configuring Windows Server 2008 LDAP for APIC Access with Cisco AVPair	263
	Configuring Cisco Cloud Network Controller for LDAP Access	263
	Configuring Cisco Cloud Network Controller for SAML Access	265
	About SAML	265
	Configuring Cisco Cloud Network Controller for SAML Access	266
	Setting Up a SAML Application in Okta	267
	Setting Up a Relying Party Trust in AD FS	267
	Configuring HTTPS Access	267
	About HTTPS Access	268
	Guidelines for Configuring Custom Certificates	268
	Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI	268

CHAPTER 8	Restricting Access	271
	Restricting Access by Domains	271
	RBAC Roles	271
	RBAC Rules	275
	Guidelines and Limitations for Restricted Domains	276
	Creating an RBAC Rule Using the Cisco Cloud Network Controller GUI	276

CHAPTER 9	Configuration Drifts	279
	Configuration Drift Notifications and Faults	279
	Accessing the Main Configuration Drift Page	280
	Checking for Missing Contracts Configuration	283

Checking for Missing EPGs Configuration	284
Checking for Missing VRFs Configuration	285
Configuration Drift Troubleshooting	287

CHAPTER 10
Configuring Connectivity Between Cisco Cloud Network Controller-Managed Cloud Site and Non-ACI Remote Site 289

Configuring Connectivity Using Express Route Gateway	289
About Deploying Express Route Gateway Using Redirect	289
Deploying Express Route Gateway Using Redirect	291
About Deploying Express Route Gateway Without Redirect	292
Deploying Express Route Gateway Without Redirect	293
Configuring Connectivity Using VPN Gateway (Virtual Network Gateway)	295
Configuring Connectivity Using VPN Gateway	296

APPENDIX A
Cisco Cloud Network Controller Error Codes 303
 Cisco Cloud Network Controller Error Codes 303

APPENDIX B

Service EPG Configuration Examples 311	
Azure Kubernetes Services (AKS) Service EPG Configuration Example	311
Creating a Subnet in the Cloud Context Profile	311
Creating the Cloud Service EPG for AKS	313
Verifying the Outbound Security Rules	315
Creating a Kubernetes Service	315
Verifying the New Kubernetes Service	319
Installing the Azure and AKS CLI	321



CHAPTER 1

New and Changed Information

This chapter contains the following sections:

- [New and Changed Information, on page 1](#)

New and Changed Information

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

Table 1: New Features and Changed Behavior in Cisco Cloud Network Controller for Release 26.0(3)

Feature or Change	Description	Where Documented
Topology view	This release provides an enhanced topology view.	
Common tenant support for hybrid cloud deployments	Beginning with release 26.0(3), support is now available for inter-tenant shared services between the on-premises tenant common and cloud user tenants.	
Application Management Details Enhancement	Beginning with 26.0(3), the VRF column for the cloud context profiles table under application management details will display the VRF name instead of the VRF count.	
Cisco Catalyst 8000V upgrade support to 17.12.02	Cisco Catalyst 8000V is upgraded to version 17.12.02 using existing upgrade procedure when Cisco Cloud Network Controller gets upgraded.	
Support for inter-tenant shared services between the on-premises tenant common and cloud user tenants	Prior to release 26.0(3), endpoints on an on-premises ACI tenant and endpoints in a user tenant using networking resources from the on-premises tenant common cannot communicate with endpoints on the cloud user tenant. Beginning with release 26.0(3), support is now available for inter-tenant shared services between the on-premises tenant common and cloud user tenants.	Support for Inter-Tenant Shared Services in Hybrid Cloud Environments, on page 26

Table 2: New Features and Changed Behavior in Cisco Cloud Network Controller for Release 26.0(2)

Feature or Change	Description	Where Documented
Multi Account support per tenant	Beginning with 26.0(2), Cisco Cloud Network Controller supports association of multiple cloud accounts under a given tenant.	
Topology view of cloud inventory	This release provides support to view real time topology of cloud networks. This helps view all the connections, routers, and transit networks within a cloud.	
Multiple VPCs/VNETs in a single VRF	This release supports deploying multiple VPCs/VNETs belonging to one VRF in the same region	
Route table support per subnet	Beginning with 26.0(2), segmentation will be possible at the subnet level for infra and user tenants. This will help associate different subnets to different VRFs and route tables. This feature is only available for Azure.	

Table 3: New Features and Changed Behavior in Cisco Cloud Network Controller for Release 26.0(1)

Feature or Change	Description	Where Documented
UI new look and feel	The screenshots and diagrams in this document have been updated to reflect the recent UI changes. While the look and feel of the new UI is different, the arrangement of the screens and configuration options remains the same.	



CHAPTER 2

About Cisco Cloud Network Controller

- [Overview, on page 3](#)
- [Understanding the overlay-2 \(Secondary\) VRF, on page 4](#)
- [External Network Connectivity, on page 5](#)
- [Understanding Supported Routing and Security Policies, on page 6](#)
- [Source Interface Selection for Tunnels, on page 11](#)
- [Guidelines and Limitations, on page 11](#)
- [About the Cisco Cloud Network Controller GUI, on page 13](#)

Overview

Cisco Cloud Network Controller is a software deployed on a cloud-based virtual machine (VM). Amazon Web Services (AWS), Azure, and Google Cloud are the cloud providers supported with the Cisco Cloud Network Controller.

When deployed, the Cisco Cloud Network Controller:

- Provides an interface that is similar to Cisco APIC (applicable for existing ACI customers only) to interact with the Azure public cloud
- Automates the deployment and configuration of cloud connectivity
- Configures the cloud router control plane
- Configures the data path between the on-premises Cisco ACI fabric or NX-OS fabrics and the cloud site
- Translates Cisco CNC policies to cloud native constructs
- Discovers endpoints
- Provides a consistent policy, security, and analytics for workloads deployed either on or across on-premises data centers and the public cloud



Note

- Cisco Nexus Dashboard Orchestrator pushes the MP-BGP EVPN configuration to the on-premises spine switches
 - On-premises VPN routers require a manual configuration for IPsec
-

- Provides an automated connection between on-premises data centers and the public cloud with easy provisioning and monitoring
- Policies are pushed by Cisco Nexus Dashboard Orchestrator to the on-premises and cloud sites, and Cisco Cloud Network Controller translates the policies to the cloud native constructs to keep the policies consistent with the on-premises site

For more information about extending Cisco ACI to the public cloud, see the *Cisco Cloud Network Controller Installation Guide* or the [Hybrid Cloud Connectivity Deployment for Cisco NX-OS](#).

When the Cisco Cloud Network Controller is up and running, you can begin adding and configuring Cisco Cloud Network Controller components. This document describes the Cisco Cloud Network Controller policy model and explains how to manage (add, configure, view, and delete) the Cisco Cloud Network Controller components using the GUI and the REST API.

Understanding the overlay-2 (Secondary) VRF

Previously, the overlay-2 VRF, which is a secondary VRF, was created in the infra tenant implicitly during the Cisco Cloud Network Controller bringup, and you would have to create services for Azure only in the overlay-2 (secondary) VRF. That restriction is removed and the overlay-2 VRF is no longer created implicitly in the infra tenant during the Cisco Cloud Network Controller bringup.

There is no special handling of this overlay-2 (secondary) VRF in either Cisco Cloud Network Controller or Nexus Dashboard Orchestrator (NDO). You can create any secondary VRF with any name and associate the `RsSubnetToCtx` in the infra VNet and deploy services in any of these secondary VRFs for Azure. You can always create a secondary VRF, and overlay-2 is just a secondary VRF.

- You can create cloud EPGs with any secondary VRF, including the overlay-2 VRF, in the infra tenant. When you create new CIDRs in the infra VNet, those CIDRs are not mapped to the overlay-2 VRF implicitly, so it is your responsibility to map the new CIDR to the secondary VRF.
- The cloud context profile of the infra VNet can be mapped to multiple VRFs (the overlay-1 and overlay-2 VRFs in the infra VNet).
- In the cloud, the subnet's route table is the most granular entity for achieving network isolation. So all system-created cloud subnets of the overlay-1 VRF and the user-created subnets of the overlay-2 VRF will be mapped to separate route tables in the cloud for achieving the network segmentation.



Note On Azure cloud, you cannot add or delete CIDRs in a VNet when it has active peering with other VNets. Therefore, when you need to add more CIDRs to the infra VNet, you need to first disable VNet peering in it, which removes all the VNet peerings associated with the infra VNet. After adding new CIDRs to the infra VNet, you need to enable VNet peering again in the infra VNet.

You do not have to disable VNet peering if you are adding a new subnet in an existing CIDR in the hub VNet.

Per Subnet Route Table

Prior to 26.0(2), Cisco Cloud Network Controller had some restrictions on the secondary VRFs and hence could only support VRF association at the CIDR level. Beginning with release 26.0(2), these restrictions are

removed and VRF association is supported at the subnet level. Within a given CIDR, different subnets can be associated to different VRFs.



Note Beginning with release 26.0(2), cloud will be programmed with subnet based routes for greenfield deployments. However if you are upgrading from prior versions, cloud will continue to be programmed with CIDR based routes for legacy CIDRs, unless secondary VRF associations are changed.

This feature provides support for the following cases:

- Support for User and Infra tenants.
- Support for leakTo and contracts routing model.

This feature does not provide support for the following cases:

- Secondary VRF on user tenant gateway subnet
- Secondary VRF on infra overlay-1
- Secondary VRF cannot be same as the main VRF
- L4-L7 Service redirection not automated

This is achieved by mapping the subnets to separate VRFs which are associated with route tables. The VRFs are then used for different contracts and leak routes to allow communication between the VRFs. Route leaking could be across VNets or subnets or across sites which belong to different VRFs. When the subnet creation is triggered, a corresponding route table is created and the subnet is associated to the respective route table.

The following steps are needed to configure Per Subnet Route Table using Cisco Cloud Network Controller. If you have Nexus Dashboard Orchestrator to manage CCNC, these steps are not needed.

- Create three VRFs : vrf100, hv1, hv2. To create a VRF, see [Create a VRF using the Cisco Cloud Network Controller GUI](#)
- Create Cloud Context Profile that maps to VRF vrf100. The cloud context profile will be programmed as a VNet with a CIDR. Inside the CIDR, create 2 subnets (net1 and net2) that map to VRF hv1 and hv2 respectively. At the end, net1 and net2 are in the same VNet but map to 2 different Route Tables on Azure Cloud. To create a cloud context profile, see [Create a Cloud Context Profile using the Cisco Cloud Network Controller GUI](#).



Note Although optional, it is highly recommended to enable VNet peering. VNet will establish VPC peering to Infra VNet.

External Network Connectivity

External network connectivity for Cisco Cloud Network Controller with Azure is available by using EVPN connectivity from the CCRs in the infra VNet. Support is also available for IPv4 connectivity from the infra VNet CCRs to any external device with IPSec/BGP. This IPSec/BGP external connectivity allows Cisco Cloud Network Controller to connect to branch offices.

The following sections provide more information on the components that allow for the new external network connectivity provided in release 25.0(1).

External VRF

An **external VRF** is a unique VRF that does not have any presence in the cloud but is associated with one or more external networks. As opposed to an internal VRF, which is a VRF that is used to host the VNETs and is associated with a cloud context profile, an external VRF is not referred to in any cloud context profile used by Cisco Cloud Network Controller.

An external VRF represents an external network that is connected to other cloud sites or to on-premises branch offices. Multiple cloud VRFs can leak routes to an external VRF or can get the routes from an external VRF. When an external network is created on an external VRF, inter-VRF routing is set up so that routes received and advertised on the external network are received or advertised on the external VRF.

Connections to Non-ACI External Devices

Support is also available for connectivity from Azure CCRs to any non-ACI external device. IPv4 sessions are created on an external VRF from the infra VNET CCRs to these non-ACI external devices, and inter-VRF routing is set up between the external VRF and the site local VRFs.

Following are the guidelines and limitations for this type of connectivity:

- You cannot use both EVPN and IPv4 IPSec/BGP to connect from the cloud to the same remote site.

Guidelines and Limitations

Instead of manually selecting all the regions, you have to set `allRegion` to true for the external network connectivity.

Understanding Supported Routing and Security Policies

Routing and security policies are handled differently, depending on the release that is running on your Cisco Cloud Network Controller.

Routing and Security Policies: Releases Prior to Release 25.0(1)

Prior to release 25.0(1), routing and security policies are tightly coupled together. To allow communication between two endpoints that are across EPGs, you must configure contracts. These contracts are used for the following:

- **Routing policies:** Policies used to define routes to establish traffic flow.
- **Security policies:** Rules used for security purposes, such as security group rules or network security group rules.

In other words, contracts inherently serve the dual purpose of configuring both security policies and routing policies. This means that tearing down contracts not only tears down the security policies that govern which traffic to allow and which to deny, it also tears down any policies used to route that traffic. Prior to release 25.0(1), there is no way to configure routing policies without also configuring security policies, and vice versa.

Routing and Security Policies: Release 25.0(1)

Beginning with release 25.0(1), support is now available for configuring routing separately, independent of the security policies.

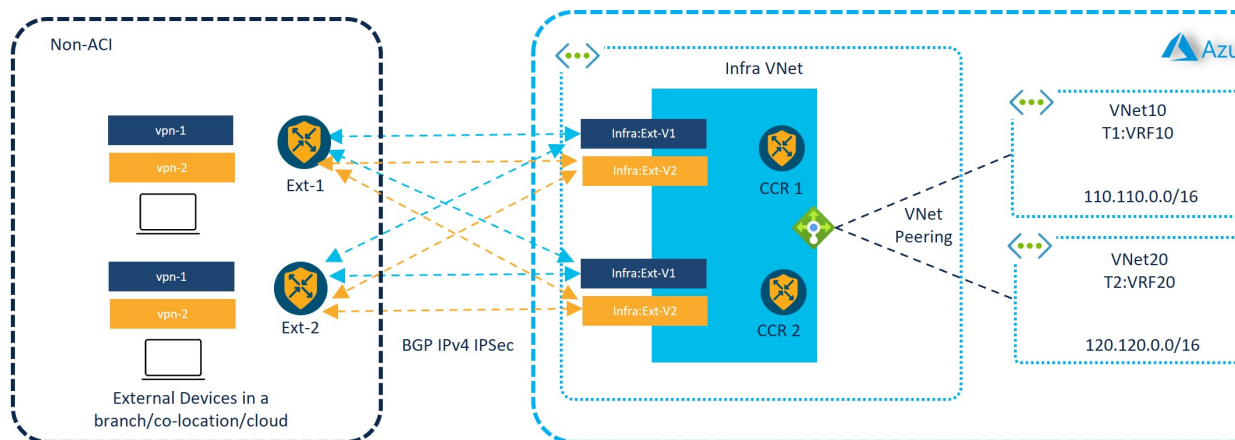


Note The routing and security policies described in this section are specifically for the 25.0(1) release and apply only between internal and external VRFs. For changes in the routing policies in the 25.0(2) release, see [Routing Policies: Release 25.0\(2\), on page 9](#).

The procedures for configuring the routing and security policies are here:

- **Routing policy:** You will use the inter-VRF routing feature introduced in release 25.0(1) to configure the routing policy separately. See [Configuring Inter-VRF Route Leaking Using the Cisco Cloud Network Controller GUI, on page 72](#) for those procedures.
- **Security policy:** After you have configured the routing policy, you will continue to use contracts as you did previously to configure the security policy separately:
 - First create an external EPG. See [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#) for those procedures.
 - Then create a contract between the external EPG and the cloud EPG. See [Creating a Contract Using the Cisco Cloud Network Controller GUI, on page 103](#) for those procedures.

Using inter-VRF routing, you can configure an independent routing policy to specify which routes to leak between a pair of internal and external VRFs when you are setting up routing between a cloud site and a non-ACI site.



The following figure shows an example topology of this sort of configuration. This example topology shows how you can connect to a remote endpoint (vpn-1) behind an external device (Ext-1) which might be located in a non-ACI site. This non-ACI site could be a branch office, co-located or cloud site, or anywhere in the internet that has the capability of BGP IPv4 and IPsec.

In this example, the infra:Ext-V1 is the external VRF on the CCRs in the infra VNet, with BGP IPv4 sessions over IPsec tunnels to the remote devices. The remote endpoint routes are received over these sessions in the infra:Ext-V1 VRF, which are then leaked into the internal VRFs displayed on the right side of the graphic (for example, the T1:VRF10 in VNet10). The reverse leaking routes are also configured.

Route leaking occurs between internal and external VRFs using route maps. Cisco Cloud Network Controller supports using route maps to configure routing policies independent of security policies only from internal VRFs to external VRFs, and from external VRF to internal VRFs. You will continue to use contracts when configuring routing between a pair of internal VRFs, so routing and security policies are tied together in the configuration process when routing between internal VRFs.

The following list provides more information on situations when you can use **route maps** to configure routing policies independent of security policies, and when you have to use **contracts** where the routing and security policies are tied together.

- Routing situations that use contracts-based routing:
 - Intra-site routing (within and across regions)
 - Inter-site routing (cloud-to-ACI on-premises using EVPN)
 - Cloud-to-cloud routing
 - Route leaking between internal VRFs
- Routing situations that use route map-based routing:
 - Cloud-to-non-ACI on-premises site using L3Out external VRF (no EVPN)
 - Leak specific or all routes from an internal VRF to an external VRF
 - Leak specific or all routes from an external VRF to an internal VRF

Guidelines and Restrictions

The following guidelines apply when using inter-VRF routing to leak routes between a pair of VRFs using route maps:

- Routes are always leaked bi-directionally between an internal VRF and the external VRF.
For example, assume there is a user tenant (t1) with an internal VRF (V1) and external VRF (Ext-V1). The route leak must be configured for both of these VRFs bi-directionally.
- You cannot configure "smaller" prefixes to be leaked while a "larger" prefix is already being leaked. For example, configuring the 10.10.10.0/24 prefix will be rejected if you already have the 10.10.0.0/16 prefix configured to be leaked. Similarly, if you configure the 0.0.0.0/0 (leak all) prefix, no other prefix will be allowed to be configured.
- Contracts are not allowed between cloud external EPGs (cloudExtEpgs).
- An external VRF cannot be used for creating cloud EPGs.
- An external VRF always belongs to the infra tenant.
- Leak routing is not supported between external VRFs.

Routing Policies: Release 25.0(2)



Note The routing and security policies described in this section are specifically for the 25.0(2) release. For changes in the routing and security policies in the previous release, see [Routing and Security Policies: Release 25.0\(1\), on page 7](#).

For release 25.0(2), the routing and security policies continue to be split as described in [Routing and Security Policies: Release 25.0\(1\), on page 7](#), but with these additional changes specifically for the routing policies:

- [Route Leaking Between Internal VRFs, on page 9](#)
- [Global Inter-VRF Route Leak Policy, on page 9](#)
- [Guidelines and Limitations, on page 10](#)

Route Leaking Between Internal VRFs

In the previous 25.0(1) release, the inter-VRF route map-based routing feature was introduced, where you can configure an independent routing policy to specify which routes to leak between a pair of internal and external VRFs. This route map-based routing feature applied specifically between internal and external VRFs; when configuring routing between a pair of internal VRFs, you could only use contract-based routing in that situation, as described in [Routing and Security Policies: Release 25.0\(1\), on page 7](#).

Beginning with release 25.0(2), support is now available for route map-based route leaking between a pair of internal VRFs. You will specify how routes are leaked using one of the following options:

- Leak all CIDRS or specific subnet IP addresses associated with the VRF by using:
 - **Leak All** option through the GUI
 - `leakInternalPrefix` field through the REST API
- Leak between a pair of VRFs by using:
 - **Subnet IP** option through the GUI
 - `leakInternalSubnet` field through the REST API

Global Inter-VRF Route Leak Policy

In addition to the support that is now available for route map-based route leaking between a pair of internal VRFs, the internal VRF route leak policy also allows you to choose whether you want to use contract-based routing or route map-based routing between a pair of internal VRFs. This is a global mode configuration available in the First Time Setup to allow a contract-based or route map-based model. Note that when you enable contract-based routing in this global mode, the routes between a pair of internal VRFs can be leaked using contracts only in the absence of route maps.

This policy has the following characteristics:

- This policy is associated with every internal VRF.
- This is a Cisco Cloud Network Controller -created policy.

- Contracts-based routing is disabled by default for greenfield cases (when you are configuring a Cisco Cloud Network Controller for the first time). For upgrades, where you have a Cisco Cloud Network Controller that was already configured prior to release 25.0(2), contract-based routing is enabled.

The internal VRF route leak policy is a global policy that is configured in the First Time Setup screen under **Advanced Settings** followed by **Edit Configuration**, where a Yes or No is used to indicate whether contracts can drive routes in the absence of route maps:

- **No**: Default setting. Routes are not leaked based on contracts, and are leaked based on route maps instead.
- **Yes**: Routes are leaked based on contracts in the absence of route maps. When enabled, contracts drive routing when route maps are not configured. When route maps exist, route maps always drives routing.

You can switch back and forth. Following are the general recommended steps for switching this global VRF route leak policy, with more detailed instructions provided in [Configuring Leak Routes for Internal VRFs Using the Cisco Cloud Network Controller GUI, on page 75](#).

- You should enable contract-based routing in Cisco Cloud Network Controller for multi-cloud and hybrid-cloud deployments with EVPN.
- For multi-cloud and hybrid-cloud deployments without EVPN, routing is driven through route maps only and not through contracts.
- If you want to disable contract-based routing switch and switch from contract-based routing to route map-based routing, this action can be disruptive if route map-based routing is not configured before you've switched this setting to **No**.

You should make the following configuration changes switching to route map-based routing:

1. Enable route map-based route leaking between all pairs of VRFs that have existing contracts.
2. Disable contract-based routing policy in the global policy.

At that point, you can change the routing policy to route map-based routing, and you can then change the routing to reflect any granularity that is required with the new route map-based routing.

- If you want to enable contract-based routing by switching from route map-based routing to contract-based routing, you do not have to make any configuration changes before switching to contract-based routing. That's because this setting is an additive operation. In other words, both contract-based and route map-based routing can be enabled between a pair of VRFs. Route maps take precedence over contracts when enabling routing. With route map-based routing enabled, adding contract-based routing should be non-disruptive.

Guidelines and Limitations

The following guidelines and limitations apply for release 25.0(2):

- Routing between external and internal VRFs continues to use route map-based routing only.
- Layer 4 to Layer 7 service insertion continues to be through contracts, so you must enable contract-based routing at the global level in these situations.
- External connectivity with Azure express route will continue to use contract-based routing.
- The `leakExternalPrefix` should not overlap with the external endpoint selector configured for external EPG to perform SSH, otherwise SSH will be broken. In this case, the prefix will be pointing to the network load balancer instead of Azure's default route to the internet.

- Unless the internet traffic has to be redirected to the remote site, `leakInternalPrefix` (Leak All, or 0.0.0.0/0) should not be used, otherwise SSH will be broken. In this case, the default route to the internet will be overwritten by the new UDR pointing to the network load balancer.

Source Interface Selection for Tunnels

Support is available for having more than one tunnel across different external networks to the same destination. This is done in the GUI by using different source interfaces (2,3, or 4) or through the REST API using `cloudtemplateIpseTunnelSourceInterface`.

The following example shows a situation where only interface 3 is used as the originating interface:

```
<cloudtemplateIpSecTunnel peeraddr="173.36.19.2" preSharedKey="def" poolname="pool1">  
  <cloudtemplateIpSecTunnelSourceInterface sourceInterfaceId="3" />  
</cloudtemplateIpSecTunnel>
```

The following example shows a situation where both interfaces 2 and 3 are used as the originating interfaces:

```
<cloudtemplateIpSecTunnel peeraddr="173.36.19.2" preSharedKey="def" poolname="pool1">  
  <cloudtemplateIpSecTunnelSourceInterface sourceInterfaceId="2" />  
  <cloudtemplateIpSecTunnelSourceInterface sourceInterfaceId="3" />  
</cloudtemplateIpSecTunnel>
```

Guidelines and Limitations

- Increasing the number of interfaces increases the demand of tunnel inner local IP addresses.
- The IPsec tunnel source interfaces feature is supported only with the IKEv2 configuration.

Guidelines and Limitations

This section contains the guidelines and limitations for Cisco Cloud Network Controller.

- You cannot stretch more than one VRF between on-prem and the cloud while using inter-VRF route leaking in the cloud CCRs (cloud routers). For example, in a situation where VRF1 with EPG1 is stretched and VRF2 with EPG2 is also stretched, EPG1 cannot have a contract with EPG2. However, you can have multiple VRFs in the cloud, sharing one or more contracts with one on-premises VRF.
- Set the BD subnet for on-premises sites as advertised externally to advertise to the CSR1kv on the cloud.
- Before configuring an object for a tenant, first check for any stale cloud resource objects. A stale configuration might be present if it was not cleaned properly from the previous Cisco Cloud Network Controller virtual machines that managed the account. Cisco Cloud Network Controller can display stale cloud objects, but it cannot remove them. You must log in to the cloud account and remove them manually.



Note It takes some time for Cisco Cloud Network Controller to detect the stale cloud resources after adding the tenant subscription ID.

Azure allows multiple tenants to share an Azure account owned by one tenant. When the account is shared by multiple tenants, only the owner tenant is able to view the stale objects in the other tenants.

To check for stale cloud resources:

1. From the Cisco Cloud Network Controller GUI, click the **Navigation menu > Application Management > Tenants**. The **Tenants** summary table appears in the work pane with a list of tenants as rows in a summary table.
 2. Double click the tenant you are creating objects for. The Overview, Cloud Resources, Application Management, Statistics, and Event Analytics tabs appear.
 3. Click the **Cloud Resources > Actions > View Stale Cloud Objects**. The **Stale Cloud Objects** dialog box appears.
- Cisco Cloud Network Controller tries to manage the Azure resources that it created. It does not attempt to manage resources created by other applications, other than listing existing resources as inventory. At the same time, it is also expected that Azure IAM users in the Azure infra tenant subscription, and the other tenant subscriptions, do not disturb the resources that Cisco Cloud Network Controller creates. For this purpose, all resources Cisco Cloud Network Controller creates on Azure has at least one of these two tags:
 - AciDnTag
 - AciOwnerTag

Cisco Cloud Network Controller must prevent Azure IAM users who have access to create, delete, or update VM, or any other resources, from accessing or modifying the resources that Cisco Cloud Network Controller created and manages. Such restrictions should apply on both the infra tenant and other user tenant subscriptions. Azure subscription administrators should utilize the above two tags to prevent their unintentional access and modifications. For example, you can have an access policy like the following to prevent access to resources managed by Cisco Cloud Network Controller:

```
{
  "properties": {
    "level": "CanNotDelete",
    "notes": "Optional text notes."
  }
}
```

- When configuring shared L3Out:
 - An on-premises L3Out and cloud EPGs cannot be in tenant common.
 - If an on-premises L3Out and a cloud EPG are in different tenants, define a contract in tenant common. The contract cannot be in the on-premises site or the cloud tenant.
 - Specify the CIDR for the cloud EPG in the on-premises L3Out external EPGs (l3extInstP).

- When an on-premises L3Out has a contract with a cloud EPG in a different VRF, the VRF in which the cloud EPG resides cannot be stretched to the on-premises site and cannot have a contract with any other VRF in the on-premises site.
- When configuring an external subnet in an on-premises external EPG:
 - Specify the external subnet as a non-zero subnet.
 - The external subnet cannot overlap with another external subnet.
 - Mark the external subnet with a shared route-control flag to have a contract with a cloud EPG.
- The external subnet that is marked in the on-premises external EPG should have been learned through the routing protocol in the L3Out or created as a static route.
- For the total supported scale, see the following Scale Supported table:



Note With the scale that is specified in the Scale Supported table, you can have only 4 total managed regions.

Table 4: Scale Supported

Component	Number Supported
Tenants	20
Application Profiles	500
EPGs	500
Cloud Endpoints	1000
VRFs	20
Cloud Context Profiles	40
Contracts	1000
Service Graphs	200
Service Devices	100

About the Cisco Cloud Network Controller GUI

The Cisco Cloud Network Controller GUI is categorized into groups of related windows. Each window enables you to access and manage a particular component. You move between the windows using the **Navigation** menu that is located on the left side of the GUI. When you hover your mouse over any part of the menu, the following list of tab names appear: **Dashboard**, **Application Management**, **Cloud Resources**, **Operations**, **Infrastructure**, and **Administrative**.

Each tab contains a different list of subtabs, and each subtab provides access to a different component-specific window. For example, to view the EPG-specific window, hover your mouse over the **Navigation** menu and click **Application Management > EPGs**. From there, you can use the **Navigation** menu to view the details of another component. For example, you can navigate to the **Active Sessions** window from **EPGs** by clicking **Operations > Active Sessions**.

The **Intent** menu bar icon enables you to create a component from anywhere in the GUI. For example, to create a tenant while viewing the **Routers** window, click the **Intent** icon. A dialog appears with a search box and a drop-down list. When you click the drop-down list and choose **Application Management**, a list of options, including the **Tenant** option, appears. When you click the **Tenant** option, the **Create Tenant** dialog appears displaying a group of fields that are required for creating the tenant.

For more information about configuring Cisco Cloud Network Controller components, see [Configuring Cisco Cloud Network Controller Components, on page 59](#)

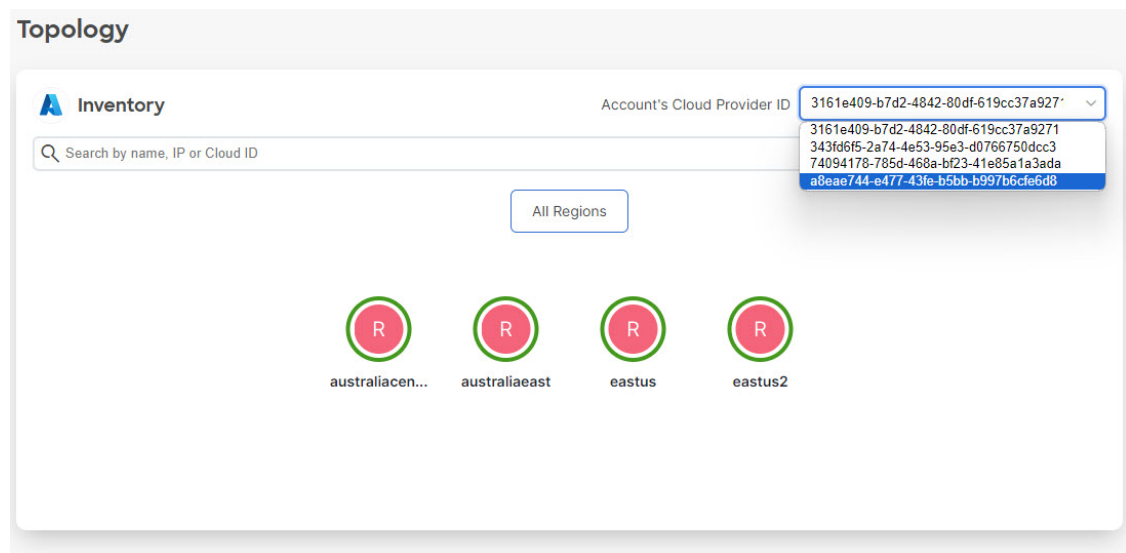
Topology View

Beginning with 26.0(2), Cisco Cloud Network Controller will now provide you with a real time view of their inventory and resources in a topology view. This helps navigate through deployments and establish a clear view of the environment. The deployed resources will be displayed in a hierarchial manner. They can be organized based on accounts, regions or VPCs/VNets.

Topology also offers search so that you can easily navigate resources by name, DN, cloud provider ID, IP address etc.

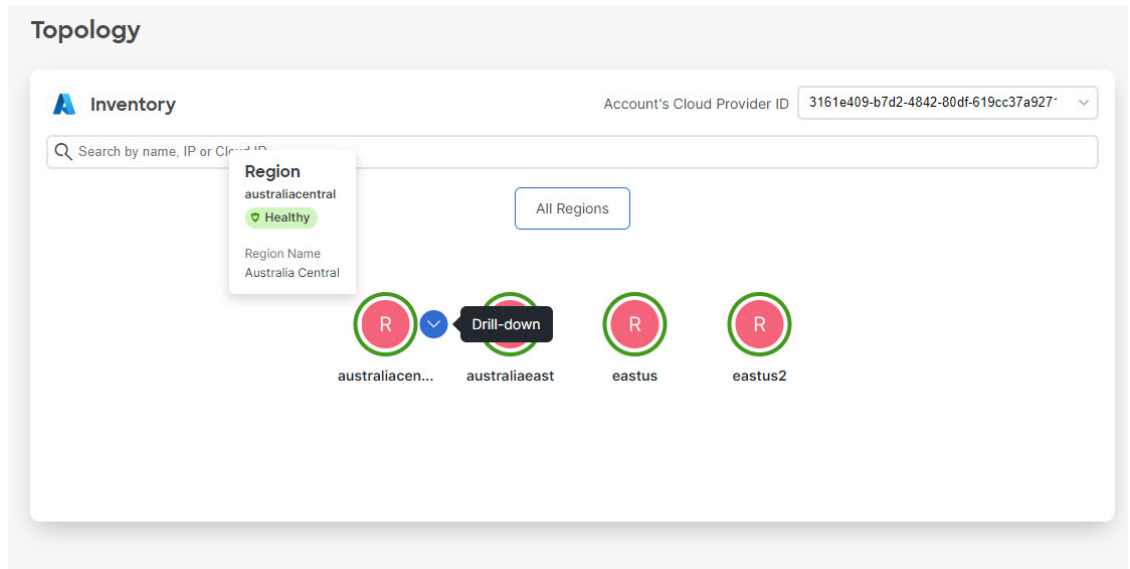
Navigation in Topology

Click Topology from the left navigation pane to view the Topology Main Page.

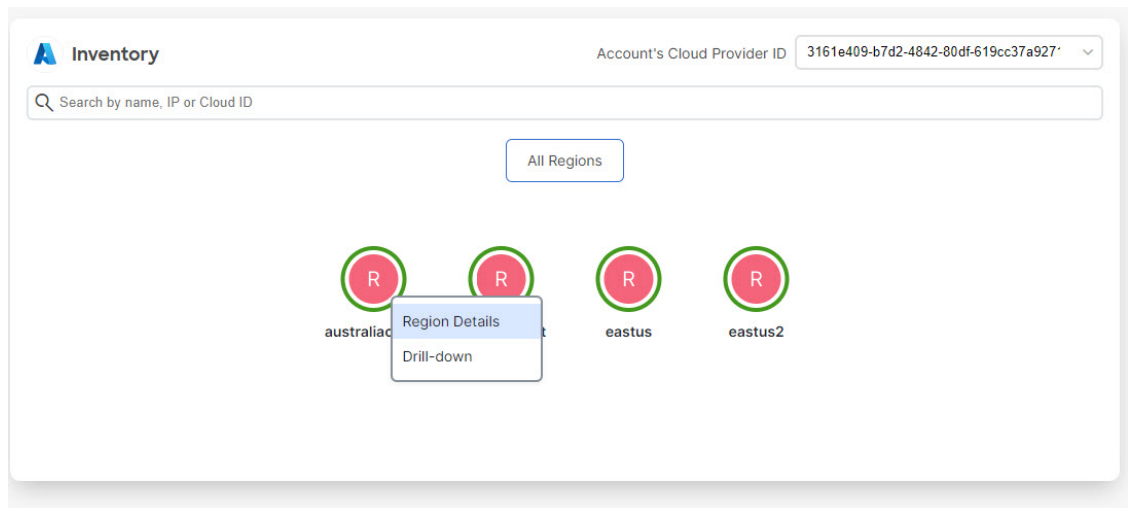


Topology Main Page is where the topology is displayed. There's a drop down box with a list of the accounts deployed or managed by Cisco Cloud Network Controller. One is selected by default. The view displays all the regions in which the account selected is deployed

The Cloud Provider for the topology is mentioned at the top left next to the Inventory header. The 'Cloud Provider ID' provides a drop down with the list of the accounts deployed or managed by Cisco Cloud Network Controller. One is selected by default.



If you hover over any node, you see a box which displays details regarding the node. You can also drill down on the node to view the associated resources (VPCs/VNets). If there are more than eight associated nodes with any node, the 'More' node is created. You can click on that to view the rest of the nodes. The NIC nodes are viewed as a hover up on the VM nodes.



If you right click on the node and select Region Details, a slider on the right side opens with more details about the selected resource.

The screenshot displays the Cisco Cloud Network Controller GUI. On the left, the 'Topology' view shows an 'Inventory' section with a search bar and a button for 'All Regions'. Below this, three region icons are shown: 'australiacen...', 'australiaeast', and 'eastus'. On the right, the 'Region' view for 'australiacentral' is shown, featuring a green 'Healthy' status bar. Below this, a 'Cloud Resources' table provides a summary of resources:

Account	Virtual Networks	Virtual Machines
1	0	0
Endpoints	Network Security Group	Application Security Groups
0	0	0

Below the table, the 'Settings' section shows the 'Name' as 'australiacentral'.

You can further drill down on the VPCs/VNets to view more resources in the hierarchy. Some of the resources available are :

- Subnets
- Route Tables
- Security Groups
- Network Interfaces
- Virtual Machines
- Account
- Region
- VPC
- ASG
- Route Table Entry
- Load Balancers



Note The resources available are inclusive but not limited to the ones mentioned above.




The search bar allows you search for any resource in the topology. It is based on some of the attributes of objects. You can select any object and the bottom-up topology for that will be shown.

Inventory Account's Cloud Provider ID



Search by name, IP or Cloud ID


rt-10.11.1.0_australiaeast
RT: a8eae744-e477-43fe-b5bb-b997b6cfe6d8 → infra → rt-10


1 / 1 results

australiacen... eastus eastus2

 **australiaeast** 

 overlay-1

 rt-10.11.1.0...



CHAPTER 3

Cisco Cloud Network Controller Policy Model

- [About the CCNC Policy Model, on page 19](#)
- [Policy Model Key Characteristics, on page 19](#)
- [Logical Constructs, on page 20](#)
- [The Cisco CCNC Policy Management Information Model, on page 21](#)
- [Tenants, on page 23](#)
- [Cloud Context Profile, on page 29](#)
- [VRFs, on page 33](#)
- [Cloud Application Profiles, on page 36](#)
- [Cloud Endpoint Groups, on page 37](#)
- [Security Groups, on page 45](#)
- [Contracts, on page 49](#)
- [About the Cloud Template, on page 52](#)
- [Managed Object Relations and Policy Resolution, on page 55](#)
- [Default Policies, on page 56](#)
- [Shared Services, on page 57](#)

About the CCNC Policy Model

The CCNC policy model enables the specification of application requirements policies. The Cisco Cloud Network Controller automatically renders policies in the cloud infrastructure. When you or a process initiates an administrative change to an object in the cloud infrastructure, the Cisco Cloud Network Controller first applies that change to the policy model. This policy model change then triggers a change to the actual managed item. This approach is called a model-driven framework.

Policy Model Key Characteristics

Key characteristics of the policy model include the following:

- As a model-driven architecture, the software maintains a complete representation of the administrative and operational state of the system (the model). The model applies uniformly to cloud infrastructure, services, system behaviors, and virtual devices attached to the network.
- The logical and concrete domains are separated; the logical configurations are rendered into concrete configurations by applying the policies in relation to the available resources. No configuration is carried

out against concrete entities. Concrete entities are configured implicitly as a side effect of the changes to the Cisco Cloud policy model.

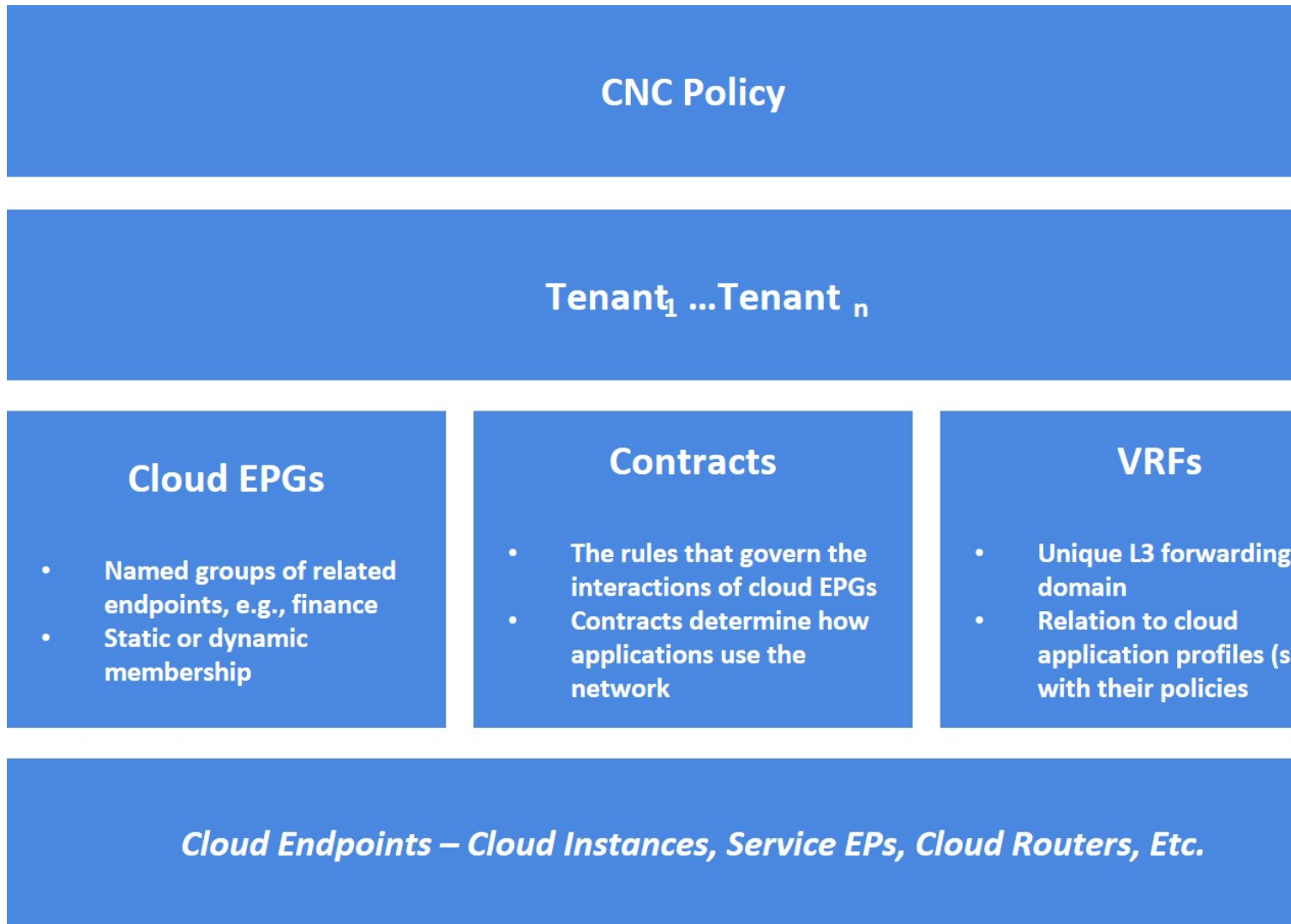
- The system prohibits communications with newly connected endpoints until the policy model is updated to include the new endpoint.
- Network administrators do not configure logical system resources directly. Instead, they define logical (hardware-independent) configurations and the Cisco Cloud Network Controller policies that control different aspects of the system behavior.

Managed object manipulation in the model relieves engineers from the task of administering isolated, individual component configurations. These characteristics enable automation and flexible workload provisioning that can locate any workload anywhere in the infrastructure. Network-attached services can be easily deployed, and the Cisco Cloud Network Controller provides an automation framework to manage the lifecycle of those network-attached services.

Logical Constructs

The policy model manages the entire cloud infrastructure, including the infrastructure, authentication, security, services, applications, cloud infrastructure, and diagnostics. Logical constructs in the policy model define how the cloud infrastructure meets the needs of any of the functions of the cloud infrastructure. The following figure provides an overview of the CCNC policy model logical constructs.

Figure 1: CCNC Policy Model Logical Constructs Overview



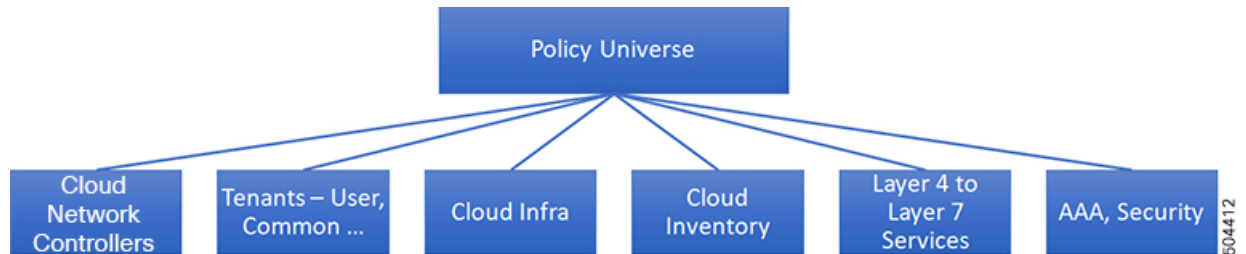
Certain administrators (tenant or cloud infrastructure-wide) create predefined policies that contain application or shared resource requirements. These policies automate the provisioning of applications, network-attached services, security policies, and tenant subnets, which puts administrators in the position of approaching the resource pool in terms of applications rather than infrastructure building blocks. The application needs to drive the networking behavior, not the other way around.

The Cisco CCNC Policy Management Information Model

The cloud infrastructure comprises the logical components as recorded in the Management Information Model (MIM), which can be represented in a hierarchical management information tree (MIT). The Cisco Cloud Network Controller runs processes that store and manage the information model. Similar to the OSI Common Management Information Protocol (CMIP) and other X.500 variants, the Cisco Cloud Network Controller enables the control of managed resources by presenting their manageable characteristics as object properties that can be inherited according to the location of the object within the hierarchical structure of the MIT.

Each node in the tree represents a managed object (MO) or group of objects. MOs are abstractions of cloud infrastructure resources. An MO can represent a concrete object, such as a cloud router, adapter, or a logical object, such as an application profile, cloud endpoint group, or fault. The following figure provides an overview of the MIT.

Figure 2: Cisco CCNC Policy Management Information Model Overview



The hierarchical structure starts with the policy universe at the top (Root) and contains parent and child nodes. Each node in the tree is an MO and each object in the cloud infrastructure has a unique distinguished name (DN) that describes the object and locates its place in the tree.

The following managed objects contain the policies that govern the operation of the system:

- A tenant is a container for policies that enable an administrator to exercise role-based access control. The system provides the following four kinds of tenants:
 - The administrator defines user tenants according to the needs of users. They contain policies that govern the operation of resources such as applications, databases, web servers, network-attached storage, virtual machines, and so on.
 - Although the system provides the common tenant, it can be configured by the cloud infrastructure administrator. It contains policies that govern the operation of resources accessible to all tenants, such as firewalls, load balancers, Layer 4 to Layer 7 services, intrusion detection appliances, and so on.



Note The Cisco Cloud Network Controller only supports load balancers as a Layer 4 to Layer 7 service.

- The infrastructure tenant is provided by the system but can be configured by the cloud infrastructure administrator. It contains policies that govern the operation of infrastructure resources. It also enables a cloud infrastructure provider to selectively deploy resources to one or more user tenants. Infrastructure tenant policies are configurable by the cloud infrastructure administrator.
- The cloud infra policies enable you to manage on-premises and inter-region connectivity when setting up the Cisco Cloud Network Controller. For more information, see the *Cisco Cloud Network Controller Installation Guide*.
- Cloud inventory is a service that enables you to view different aspects of the system using the GUI. For example, you can view the regions that are deployed from the aspect of an application or the applications that are deployed from the aspect of a region. You can use this information for cloud resource planning and troubleshooting.

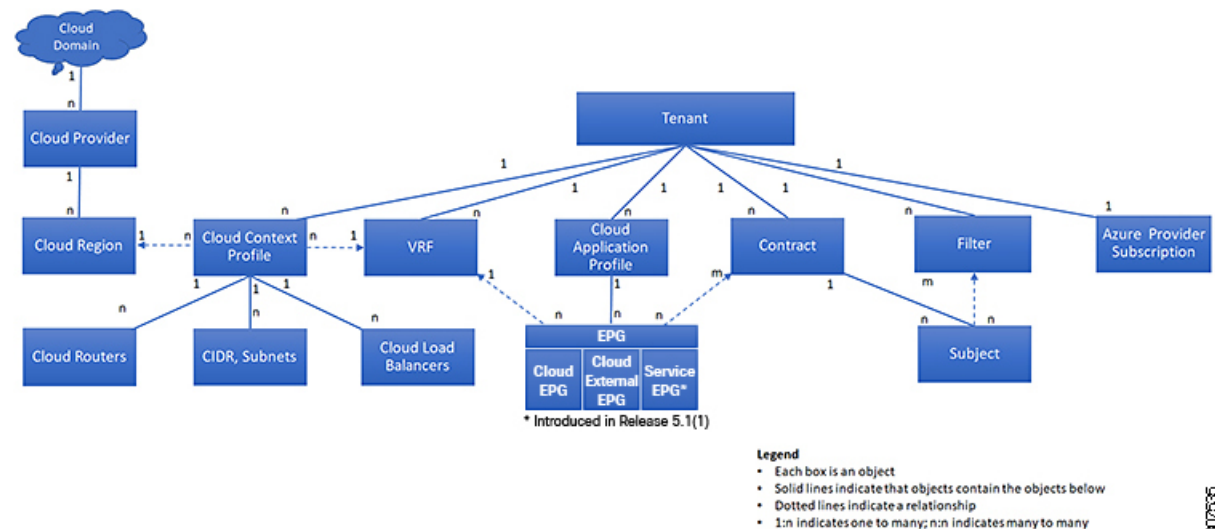
- Layer 4 to Layer 7 service integration lifecycle automation framework enables the system to dynamically respond when a service comes online or goes offline. For more information, see [Deploying Layer 4 to Layer 7 Services, on page 177](#)
- Access, authentication, and accounting (AAA) policies govern user privileges, roles, and security domains of the Cisco Cloud Network Controller cloud infrastructure. For more information, see [Cisco Cloud Network Controller Security, on page 259](#)

The hierarchical policy model fits well with the REST API interface. When invoked, the API reads from or writes to objects in the MIT. URLs map directly into distinguished names that identify objects in the MIT. Any data in the MIT can be described as a self-contained structured tree text document encoded in XML or JSON.

Tenants

A tenant ($fvTenant$) is a logical container for application policies that enable an administrator to exercise domain-based access control. A tenant represents a unit of isolation from a policy perspective, but it does not represent a private network. Tenants can represent a customer in a service provider setting, an organization or domain in an enterprise setting, or just a convenient grouping of policies. The following figure provides an overview of the tenant portion of the management information tree (MIT).

Figure 3: Tenants



Tenants can be isolated from one another or can share resources. The primary elements that the tenant contains are filters, contracts, Virtual Routing and Forwarding (VRF) instances, cloud context profiles, Azure provider configurations, and cloud application profiles that contain cloud endpoint groups (cloud EPGs). Entities in the tenant inherit its policies. VRFs are also known as contexts; each VRF can be associated with multiple cloud context profiles. A cloud context profile, in conjunction with a VRF, tenant and region, represents a resource group in Azure. A VNET is created inside the resource group based on the VRF name.

Tenants are logical containers for application policies. The cloud infrastructure can contain multiple tenants. You must configure a tenant before you can deploy any Layer 4 to Layer 7 services. The CCNC cloud infrastructure supports only IPv4 configurations for tenant networking.

Support for Multiple Cloud Subscriptions Under a Single Tenant

Beginning with 26.0(2), multiple cloud subscriptions can be associated to a given tenant and deploy different cloud resources in multiple Azure cloud subscriptions. Different VNets can also be deployed in different subscriptions under the same VRF for a given tenant.

For example, if you have only cloud deployments where cloud resources have to be deployed in different cloud accounts, you can now create a tenant that has multiple accounts and then have VNets point to the respective cloud subscriptions.



Note Multi-Account tenant is only supported on cloud deployments. This is not supported on configurations deployed in Nexus Dashboard Orchestrator.

Understanding Tenants, Identities, and Subscriptions

Azure has an active directory structure. The top level structure is the organization, and underneath the organization are the directories (also known as Azure tenants). Inside the directories, you can have one or more Azure subscriptions.

The relationship between certain Azure components is as follows:

Tenants > Subscriptions > Resource Groups > Resources

Where:

- One tenant can have multiple subscriptions, but each subscription can belong to only one tenant
- One subscription can have multiple resource groups, but each resource group can belong to only one subscription
- One resource group can have multiple resources, but each resource can belong to only one subscription

The following sections provide more detail about each of these components:

- [Mapping Azure and Cisco Cloud Network Controller Components, on page 24](#)
- [About Azure Subscriptions, on page 24](#)
- [About Tenants and Identities, on page 25](#)

Mapping Azure and Cisco Cloud Network Controller Components

In Cisco Cloud Network Controller, each Azure resource group is mapped to one Cisco Cloud Network Controller tenant, and one Cisco Cloud Network Controller tenant can have multiple Azure resource groups.

The relationship between certain Cisco Cloud Network Controller components is as follows:

Tenants > VRFs > Regions

When you create a VRF in Cisco Cloud Network Controller, a new resource group is also created on Azure.

About Azure Subscriptions

An Azure subscription is used to pay for Azure cloud services. An Azure subscription has a trust relationship with Azure Active Directories (Azure ADs), where the subscription uses the Azure AD to authenticate users,

services, and devices. While multiple subscriptions can trust the same Azure AD, each subscription can trust only one Azure AD.

In Azure, the same Azure subscription ID can be used for multiple CCNC tenants. This means that you could configure the infra tenant using one Azure subscription, and then configure more user tenants in the same subscription. CCNC tenants are tied to Azure subscriptions.

About Tenants and Identities

Following are the different types of tenants and identities available through Azure and Cisco Cloud Network Controller.



Note Both managed identity and service principal is supported as an access type for the infra tenants and the user tenants.

Managed Identity

Managed identities provide an identity for applications to use when connecting to resources that support Azure AD authentication. Applications can use the managed identity to obtain Azure AD tokens. For example, an application could use a managed identity to access resources like [Azure Key Vault](#), where developers can store credentials in a secure manner or to access storage accounts.

Following are several benefits to using managed identities:

- You don't need to manage credentials, since credentials are not even accessible to you.
- You can use managed identities to authenticate to any resource that supports Azure AD authentication, including your own applications.
- Managed identities can be used without any additional cost.

For additional information on managed identities in Azure, see:

<https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/overview>

If you are configuring tenants in the Cisco Cloud Network Controller using **managed identity**, then you will make the following configurations in the Azure portal and in the Cisco Cloud Network Controller:

1. In the Azure portal, you will add a role assignment for a **virtual machine**. You use this option when the Azure subscriptions are in the same Azure directory (of the same organization).



Note If your Azure subscriptions are in different directories and you want to configure tenants using **managed identity**, you can go to the Azure console and click on each of the subscriptions and move the subscriptions under the same Azure directory. You can only do this if the directories (containing the different subscriptions) are a child of the same parent organization.

2. In the Cisco Cloud Network Controller, you will choose the **Managed Identity** option when configuring a tenant in Cisco Cloud Network Controller.

See [Creating a Tenant Using the Cisco Cloud Network Controller GUI, on page 59](#) for more information on making these configurations.

Service Principal

An Azure **service principal** is an identity created for use with applications, hosted services, and automated tools to access Azure resources. You would use the service principal identity when you want to configure tenants in different subscriptions. The subscriptions are either in different Azure directories (Azure tenants) in the same organization, or the subscriptions can be in different organizations.

If you are configuring tenants in the Cisco Cloud Network Controller using **service principal**, then you will make the following configurations in the Azure portal and in the Cisco Cloud Network Controller:

1. In the Azure portal, you will be adding a role assignment for an **app**, where the cloud resources will be managed through a specific application.
2. In the Cisco Cloud Network Controller, you will choose the **Service Principal** option when configuring a tenant in Cisco Cloud Network Controller. The subscriptions that you enter in this page can be in different Azure directories (Azure tenants) in the same organization, or the subscriptions can be in different organizations.

See [Creating a Tenant Using the Cisco Cloud Network Controller GUI, on page 59](#) for more information on making these configurations.

Shared Tenant

You will choose this option when you have already associated Azure subscriptions with either of the two methods above and want to create more tenants in that subscription.

If you are configuring a tenant in the Cisco Cloud Network Controller as **shared tenant**, then you will make the following configurations in the Azure portal and in the Cisco Cloud Network Controller:

1. You do not have to make any configurations in Azure specifically for a shared tenant, because you will have already associated Azure subscriptions with either of the two methods above. With the shared tenant, you will just create more tenants in that existing subscription.
2. In the Cisco Cloud Network Controller, you will choose the **Shared** option when configuring a tenant in Cisco Cloud Network Controller.

See [Creating a Tenant Using the Cisco Cloud Network Controller GUI, on page 59](#) for more information on making these configurations.

Support for Inter-Tenant Shared Services in Hybrid Cloud Environments

In Cisco APIC, a pre-defined tenant (the tenant `common`) is available to provide common services to all tenants, such as shared L3Out, private networks, DNS, DHCP, and Active directory. Prior to release 26.0(3), endpoints on an on-premises ACI tenant and endpoints in a user tenant using networking resources from the on-premises tenant `common` cannot communicate with endpoints on the cloud user tenant. Beginning with release 26.0(3), support is now available for inter-tenant shared services between the on-premises tenant `common` and cloud user tenants.

Cisco Cloud Network Controller, used in conjunction with Nexus Dashboard Orchestrator, supports inter-tenant shared services in a hybrid cloud environment, allowing you to deploy resources in on-premises tenants and cloud tenants, where contracts are deployed in tenant `common`. The tenant `common` still exists on the Cloud Network Controller; however, it is not associated with any cloud account. It is just used for storing filters and contracts that later can be used for a shared service policy. Beginning with release 26.0(3), support is available for having resources in the on-premises Cisco APIC tenant `common` for both Application EPGs and external EPGs, as well as having inter-tenant shared services in a hybrid cloud environment.

For example, assume that you already have an on-premises Cisco APIC tenant `common` deployed with a VRF. You can have bridge domain or EPG in the tenant `common` as you normally would, or you can now create a new user tenant to leverage the VRF and bridge domain in the tenant `common`.

Prior to release 26.0(3), the following variants of standard tenant are supported:

- Regular EPG in a user tenant to a cloud tenant
- External EPG in a user tenant to a cloud tenant

With this update in release 26.0(3), the following variants of the on-premises ACI tenant `common` are also supported:

- Regular EPG in the tenant `common` to a cloud tenant
- External EPG in the tenant `common` to a cloud tenant
- Regular EPG in a user tenant with a bridge domain and VRF in the tenant `common` to a cloud tenant
- External EPG in a user tenant with a VRF in the tenant `common` to a cloud tenant

Use Cases

This section describes several use case examples related to the support for inter-tenant shared services in hybrid cloud environments in release 26.0(3).

On-Premises Cisco APIC Tenant Common Use Case

In this use case, an on-premises Cisco APIC tenant `common` is deployed with either or both of these configurations:

- Application EPGs in the bridge domain or subnet
- External EPG subnet in the L3Out

There is also a contract configured with a user tenant in a cloud site.

The user tenant in the cloud site can be stretched to all the sites, including the on-premises and other cloud sites, and traffic will still flow between the on-premises tenant `common` and the user tenant across all sites.

Site1: On-Premises Site	Site2: Cloud Site
VRF in tenant <code>common</code> in Site1: VRF1	VRF in tenant in Site2: VRF2
EPG in Site1: EPG1	EPG in Site2: EPG2
Tenant in Site2 stretched to Site1	Tenant <code>common</code> in Site1 available in Site2
External EPG available in VRF1 in tenant <code>common</code>	External EPG can be created on Site1

Site User Tenants Use Case

In this use case, a tenant (Tenant1) is deployed only in Site1, which is either an on-premises site or a cloud site, and another tenant (Tenant2) is deployed only in Site2, which is a cloud site, and a contract is shared across tenants.

Site1: On-Premises or Cloud Site	Site2: Cloud Site
VRF in tenant (Tenant1) in Site1: VRF1	VRF in tenant (Tenant2) in Site2: VRF2
EPG in Site1: EPG1	EPG in Site2: EPG2
Tenant2 in Site2 stretched to Site1	Tenant1 in Site1 stretched to Site2
External EPG available in VRF1 in Tenant1	External EPG can be created on Site1

Example Configuration Process

The following general steps provide an example for configuring inter-tenant shared services in hybrid cloud environments. See the [Nexus Dashboard Orchestrator documentation](#) for more details.

1. Define the tenants, if necessary.

In this example scenario, two tenants need to be defined:

- Cloud only tenant that is associated with a cloud account
- On-premises `common` tenant, which is already defined through APIC and exists in both the on-premises ACI and the cloud by default

2. Define the tenant templates in Nexus Dashboard Orchestrator (NDO) that are associated with the two tenants.

In this example scenario, you will define two tenant templates in NDO:

- `cloud-tenant-template`: Tenant template that is associated with the cloud only tenant
- `common-tenant-template`: Tenant template that is associated with the on-premises `common` tenant

3. Create a schema (for example, `common-schema`) with the necessary templates.

You can have multiple templates within a schema. For example, you could create two templates within this schema:

- `common-policy`: In this example scenario, we will make the following configurations in this template:
 - We will associate this template with the `common` tenant in the cloud site. This template is to deploy the contracts and filter to the `common` tenant on the cloud (though the `common` tenant is not associated with any cloud account) and the `common` tenant on the on-premises ACI site.
 - We will also create two contracts in this template:
 - One for the external EPG from the on-premises site to the cloud site
 - One for a regular EPG from the on-premises site to the cloud site
 - We will also configure the necessary policy contract and filters within this template.
- `common-app`: In this example scenario, we will associate this template only with the tenant `common` in the on-premises site, and we will make the necessary configurations with this on-premises site, such as configurations related to an application profile, VRF, bridge domain, L3Out, external EPG, and so on.

4. Create a second schema (for example, `cloud-schema`) with a single template (`cloud-only`), where we will associate this template only with the cloud only tenant, and we will make the necessary configurations with this cloud site, such as configurations related to an application profile, VNet/vPC, and so on.
5. Configure contracts using the contracts that you defined when you created the schemas.
6. Deploy the configurations in NDO.

Cloud Context Profile

The cloud context profile contains information on the following Cisco Cloud Network Controller components:

- CIDRs
- VRFs
- EPGs
- Regions
- Virtual Networks
- Routers
- Endpoints

CCR

The CCR is a virtual router that delivers comprehensive WAN gateway and network services into virtual and cloud environments. The CCR enables enterprises to extend their WANs into provider-hosted clouds. Two CCRs are required for Cisco Cloud Network Controller solution.

The **Cisco Catalyst 8000V** is used with the Cisco Cloud Network Controller. For more information on this type of CCR, see the [Cisco Catalyst 8000V Edge software documentation](#).

About the Cisco Catalyst 8000V

Following are the updates in the Cisco Catalyst 8000V.



Note Cisco Catalyst 8000V needs to be upgraded to version **17.12.02** using existing upgrade procedure when Cisco Cloud Network Controller gets upgraded to version **26.0(3)** . View [Triggering an Upgrade of the C8kVs](#) to upgrade Cisco Catalyst 8000V.

- [Licensing Models, on page 29](#)
- [Throughput Options Based on Licensing Models, on page 30](#)

Licensing Models

The Cisco Catalyst 8000V on Cisco Cloud Network Controller supports two licensing models:

1. **Bring Your Own License (BYOL)**
2. **Pay As You Go (PAYG)**

BYOL Licensing Model

The BYOL licensing model on Cisco Catalyst 8000V which requires you to purchase your Catalyst 8000V Cisco DNA license from Cisco and deploy it in the cloud.

- For instructions on subscribing to one of the tier-based Cisco Catalyst 8000V licenses, see the [Cisco Catalyst 8000V Edge Software](#).
- For more information on different throughputs based on the tiers, see [Throughput Options Based on Licensing Models, on page 30](#).

Cisco Cloud Network Controller makes use of the “Cisco DNA Advantage” subscription. For features supported by the “Cisco DNA Advantage” subscription, see the [Cisco DNA Software SD-WAN and Routing Matrices](#).

PAYG Licensing Model

Cisco Cloud Network Controller supports Pay-As-You-Go (PAYG) Licensing Model on Cisco Catalyst 8000V which allows users to deploy a Catalyst 8000V instance in the cloud based on the VM size and purchase the usage on an hourly basis.

As you completely depend on the VM size to get the throughput, the PAYG licensing model can be enabled only by first un-deploying the current Cisco Catalyst 8000V and then re-deploying it using the First Time Set Up with the new VM size. For more information, see "Configuring Cisco Cloud Network Controller Using the Setup Wizard" in the [Cisco Cloud Network Controller for Azure Installation Guide](#)



Note The procedure for switching between licenses can also be used if you would like to switch between the two licensing types available.



Note There are two PAYG options for consuming licenses in the Azure marketplace: **Catalyst 8000V Cisco DNA Essentials** and **Catalyst 8000V Cisco DNA Advantage**. Cisco Cloud Network Controller will make use of **Catalyst 8000V Cisco DNA Advantage**. For features supported by the “Cisco DNA Advantage” subscription, see [Cisco DNA Software SD-WAN and Routing Matrices](#)

Throughput Options Based on Licensing Models

The Cisco Catalyst 8000V on Cisco Cloud Network Controller supports two licensing models:

1. **Bring Your Own License (BYOL)**
2. **Pay As You Go (PAYG)**

1. Bring Your Own License (BYOL)

For this model, the Cisco Catalyst 8000V supports tier-based (T0/T1/T2/T3) throughput options. The following table lists what Azure VM sizes are needed for different router throughput settings for the Cisco Catalyst 8000V:

CCR Throughput	Azure VM Size
T0 (up to 15M throughput)	DS3_v2
T1 (up to 100M throughput)	DS3_v2
T2 (up to 1G throughput)	DS3_v2
T3 (up to 10G throughput)	F16s_v2

Tier2 (T2) is the default throughput supported by Cisco Cloud Network Controller.

2. Pay-As-You-Go Licensing Model

For this model, Cisco Cloud Network Controller supports a range of Azure compute instances for cloud networking needs powered by Cisco's Catalyst 8000V virtual router.

The table below shows the cloud instance type supported by Cisco Cloud Network Controller on Azure.

VmName on Azure	Memory	vCPUs	NetworkBw
DS3V2	14GiB	4	Up to 3 Gigabit
DS4V2	28GiB	8	Up to 6 Gigabit
F16SV2	32GiB	16	Up to 12.5 Gigabit
F32SV2	64GiB	32	Up to 16 Gigabit

Changing the Number of CCRs

The maximum number of CCRs supported per region increased from 4 to 8. These procedures provide instructions for increasing the number of CCRs above 4, or for reducing the number of CCRs back to 4, if necessary.

Note the following:

- You do not have to use these instructions if you are increasing or decreasing the number of CCRs in a range between 2-4 CCRs. Use these instructions only if you are increasing the number of CCRs above 4, or if you are decreasing the number of CCRs from a range of 5-8 CCRs.
- Changing the number of CCRs can impact traffic for up to 30 minutes.

Step 1

Disable Azure VNet peering at the local level on all infra cloud context profiles.

- Navigate to the **Create Cloud Context Profile** page:

Application Management > Cloud Context Profiles

- Click the link under the **Name** column for the infra cloud context profile.

A panel showing details for this cloud context profile slides in from the right side of the window.

- Click the Details icon (.

Another window appears that provides more detailed information for this cloud context profile.

- d) Click the pencil icon in the upper right corner of the window.


The **Edit Cloud Context Profile** window appears.

- e) Uncheck (disable) the **Hub Network Peering** field.
- f) Click **Save** when finished.

Repeat these steps to disable Azure VNet peering on all infra cloud context profiles.

Step 2

If you are increasing the number of CCRs above 4, add additional subnet pools for the additional CCRs, if necessary. You will see an error message if you attempt to increase the number of CCRs above 4 and the system determines that additional subnet pools are required.

- a) In the Cisco Cloud Network Controller GUI, click the Intent icon () and select **Cloud Network Controller Setup**.
- b) In the **Region Management** area, click **Edit Configuration**.
- c) In the **Regions to Manage** window, click **Next**.


The **General Connectivity** window appears.

- d) Under the **General** area, in the **Subnet Pools for Cloud Routers** field, click **Add Subnet Pool for Cloud Routers** if you want to add additional subnets for CCRs.

Addresses from this subnet pool will be used for inter-region connectivity for any additional regions that are added that need to be managed by the Cisco Cloud Network Controller. This must be a valid IPv4 subnet with mask /24.

Step 3

Increase the number of CCRs above 4, or decrease the number of CCRs from a range of 5-8 CCRs.

- a) In your Cisco Cloud Network Controller GUI, click the Intent icon () and choose **Cloud Network Controller Setup**.
- b) In the **Region Management** area, click **Edit Configuration**.

The **Regions to Manage** window appears.

- c) Click **Next** to leave the previously-selected regions and CCRs as-is.

The **General Connectivity** window appears.

- d) Locate the **CCRs** area in the **General Connectivity** window and, in the **Number of Routers Per Region** field, make the necessary changes to increase or decrease the number of CCRs.
- e) Click **Next**, then enter the necessary information in the following page and click **Save and Continue**.

The process of adding or removing the CCRs might take roughly a half hour.

Step 4

Enable Azure VNet peering again at the local level on all infra cloud context profiles.

- a) Navigate to the **Create Cloud Context Profile** page:

Application Management > Cloud Context Profiles

- b) Click the link under the **Name** column for the infra cloud context profile.

A panel showing details for this cloud context profile slides in from the right side of the window.

- c) Click the Details icon ().

Another window appears that provides more detailed information for this cloud context profile.

- d) Click the pencil icon in the upper right corner of the window.

The **Edit Cloud Context Profile** window appears.

- e) Check (enable) the **Hub Network Peering** field.
- f) Click **Save** when finished.

Repeat these steps to enable Azure VNet peering on all infra cloud context profiles.

Private IP Address Support for Cisco Cloud Network Controller and CCR

By default, CCR interfaces are assigned private IP addresses only and assignment of public IP addresses to CCR interfaces is optional. Private IP addresses are always assigned to all the interfaces of a CCR. The private IP of GigabitEthernet1 of a CCR is used as BGP and OSPF router IDs. Hybrid cloud with on-premise ACI sites over express route is supported when CCRs are assigned private IP addresses.

To enable CCR private IP addresses for inter-site connectivity, where you are disabling public IP addresses for the CCR interfaces, see the [Managing Regions \(Configuring a Cloud Template\) Using the Cisco Cloud Network Controller GUI, on page 139](#) procedure.

By default, a private IP address is assigned to the management interface of the Cisco Cloud Network Controller and assigning a public IP address is optional. To disable public IP to the Cisco Cloud Network Controller so that a private IP address is used for connectivity, see *Deploying the Cisco Cloud Network Controller in Azure* procedure in the *Cisco Cloud Network Controller for Azure Installation Guide*.

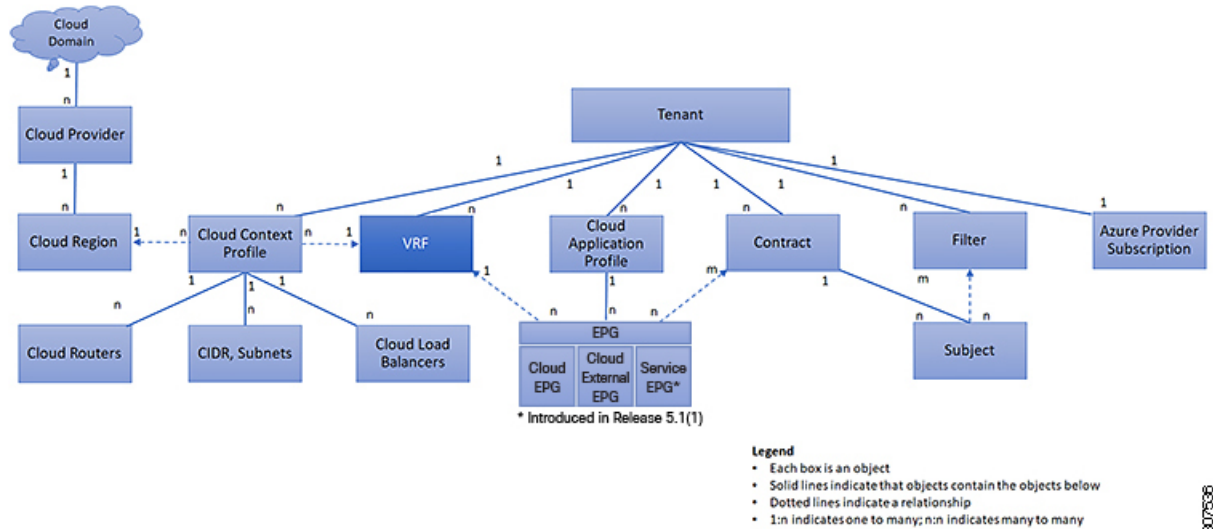
Restrictions for CCR with Private IP Address:

- No support for multicloud deployments as intersite communication needs IPsec.

VRFs

A Virtual Routing and Forwarding (VRF) object (`fvrctx`) or context is a tenant network (called a VRF in the Cisco Cloud Network Controller GUI). A tenant can have multiple VRFs. A VRF is a unique Layer 3 forwarding and application policy domain. The following figure shows the location of VRFs in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 4: VRFs



A VRF defines a Layer 3 address domain. One or more cloud context profiles are associated with a VRF. You can only associate one cloud context profile with a VRF in a given region. All the endpoints within the Layer 3 domain must have unique IP addresses because it is possible to forward packets directly between these devices if the policy allows it. A tenant can contain multiple VRFs. After an administrator creates a logical device, the administrator can create a VRF for the logical device, which provides a selection criteria policy for a device cluster. A logical device can be selected based on a contract name, a graph name, or the function node name inside the graph.

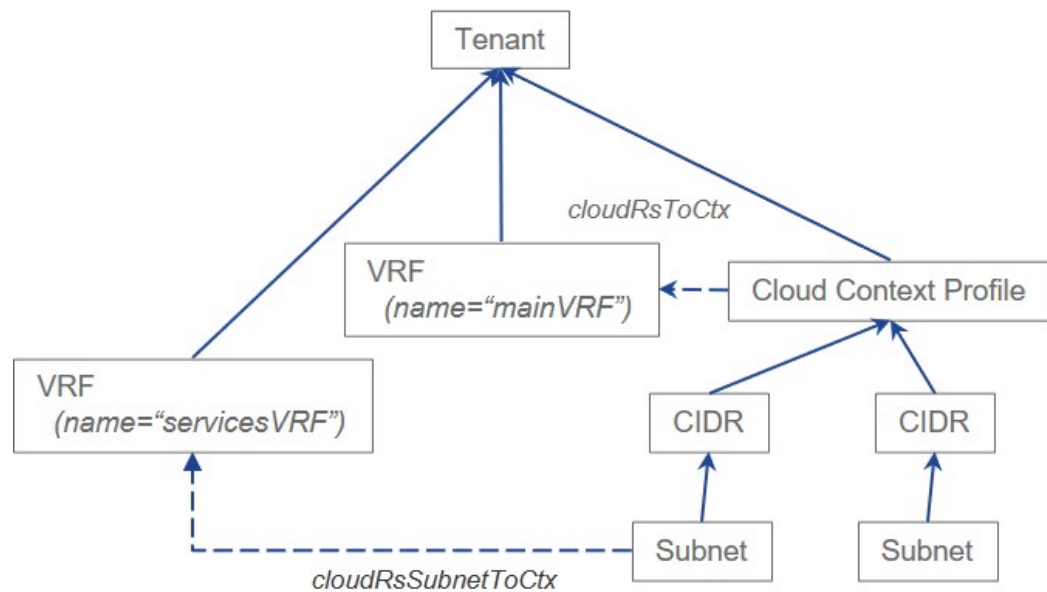
Support for Multiple VRFs Under Single VNet

Support is now available for multiple VRFs under a single VNet.

You can have an infra (hub) VNet (a `cloudCtxProfile` in the infra tenant) that can be carved out into multiple VRFs. All subnets in the respective VRFs will have separate route tables in the cloud for VRF segregation.

You can also carve out multiple VRFs beyond the infra VNet so that you can divide any VNet into multiple VRFs under the same tenant, where multiple VRFs can exist in a single VNet. This is useful for situations such as cloud service access, where you might want to carve out multiple networks (VRFs) within a given VNet, allowing you to have separate routing by having unique route tables for each VRF within the VNet in the cloud.

The following graphic shows an example managed object (MO) relationship tree with multiple VRFs under the same tenant (VNet).



In this example, two VRFs exist under the same tenant (VNet):

- The primary VRF with the name `mainVRF`
- A secondary VRF with the name `servicesVRF`

A second CIDR block and subnet exists in the same cloud context profile, under the same tenant (VNet), but that second CIDR block and subnet is associated with the secondary VRF in that same VNet.

Support for VRF to span across multiple VNets within a Region

Prior to 26.0(2), multiple VNets belonging to the same VRF would need to be deployed in different regions. Beginning with 26.0(2), Cisco Cloud Network Controller will have the ability to configure multiple VNets in a VRF inside one region. This helps enable automatic route propagation across all the VNets grouped under one VRF.

The advantages are as follows :

- This is necessary in situations where a customer may want multiple VNets configured in the same VRF.
- This is also beneficial in cases if the customer wants multiple VNets in the same region. Prior to this feature, the customer would have to deploy the VNets in different VRFs and setup contracts and leak routes to allow communication of the VPCs with each other. This feature helps simplify the configuration.



Note VNets with overlapping subnets cannot be associated with a single VRF in a region. This applies to brownfield VNets as well.

As VNets are created in resource groups, there are two options that are available :

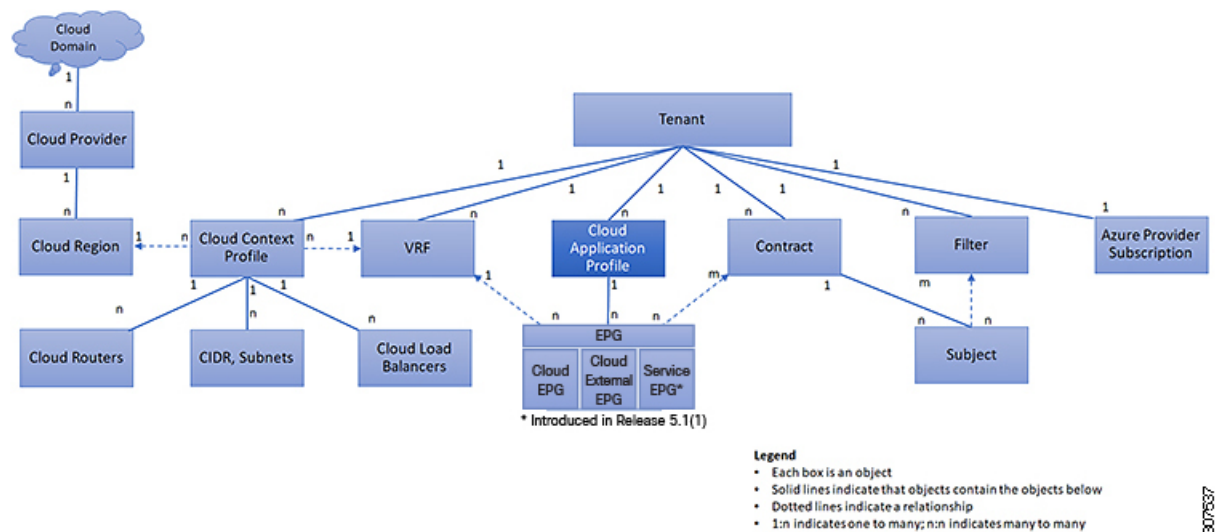
- VNets in the same Resource Group (Greenfield deployments cannot be done using this) - Route tables, ASGs and VNets should have different names to distinguish between cloud context profiles.

- VNets in different Resource Groups (Greenfield deployments can be only be done using this) - Resource groups should have different names in cloud for brownfield resource groups.

Cloud Application Profiles

A cloud application profile (`ccloudAp`) defines the policies, services and relationships between cloud EPGs. The following figure shows the location of cloud application profiles in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 5: Cloud Application Profiles



Cloud application profiles contain one or more cloud EPGs. Modern applications contain multiple components. For example, an e-commerce application could require a web server, a database server, data located in a storage service, and access to outside resources that enable financial transactions. The cloud application profile contains as many (or as few) cloud EPGs as necessary that are logically related to providing the capabilities of an application.

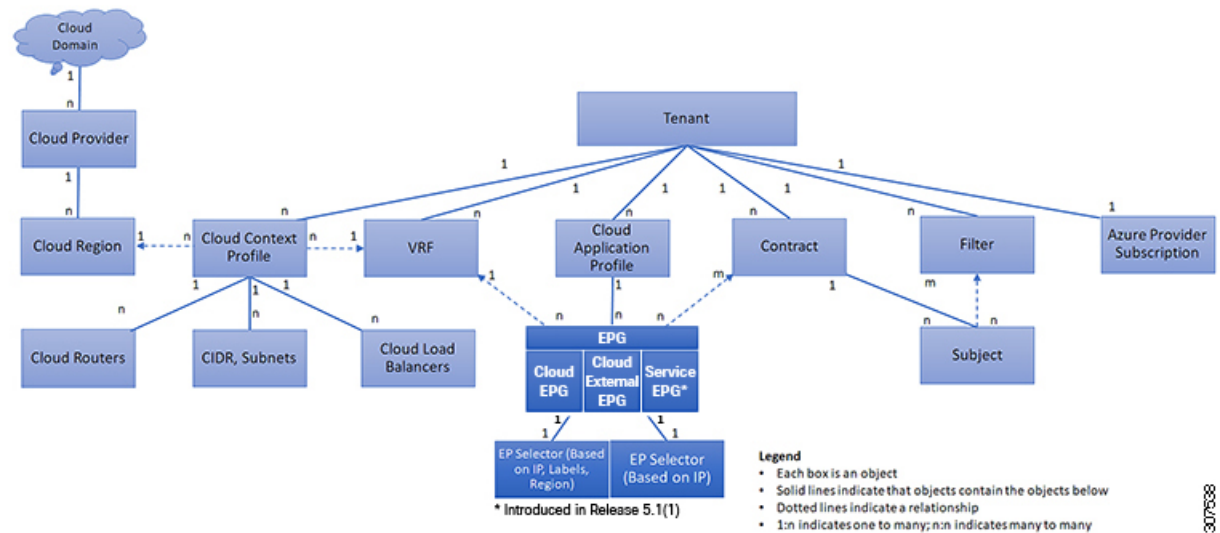
Cloud EPGs can be organized according to one of the following:

- The application they provide, such as a DNS server or SAP application (see *Tenant Policy Example* in *Cisco APIC REST API Configuration Guide*).
- The function they provide (such as infrastructure)
- Where they are in the structure of the data center (such as DMZ)
- Whatever organizing principle that a cloud infrastructure or tenant administrator chooses to use

Cloud Endpoint Groups

The cloud endpoint group (cloud EPG) is the most important object in the policy model. The following figure shows where application cloud EPGs are located in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 6: Cloud Endpoint Groups



A cloud EPG is a managed object that is a named logical entity that contains a collection of endpoints. Endpoints are devices that are connected to the network. They have an address (identity), a location, attributes (such as version or patch level), and are virtual. Knowing the address of an endpoint also enables access to all its other identity details. Cloud EPGs are fully decoupled from the physical and logical topology. Endpoint examples include servers, virtual machines, storage services, or clients on the Internet. Endpoint membership in a cloud EPG can be dynamic or static.

The CCNC cloud infrastructure can contain the following types of cloud EPGs:

- Cloud endpoint group (`cloudEPg`)
- Cloud external endpoint group (`cloudExtEPg`)
- Cloud service endpoint group (`cloudSvcEPg`): Introduced in Release 5.1(2). See [Cloud Service Endpoint Groups](#), on page 38 for more information.

Cloud EPGs contain endpoints that have common policy requirements such as security or Layer 4 to Layer 7 services. Rather than configure and manage endpoints individually, they are placed in a cloud EPG and are managed as a group.

Policies apply to cloud EPGs, never to individual endpoints.

Regardless of how a cloud EPG is configured, cloud EPG policies are applied to the endpoints they contain.

WAN router connectivity to the cloud infrastructure is an example of a configuration that uses a static cloud EPG. To configure WAN router connectivity to the cloud infrastructure, an administrator configures a `cloudExtEPg` cloud EPG that includes any endpoints within an associated WAN subnet. The cloud infrastructure learns of the cloud EPG endpoints through a discovery process as the endpoints progress through their

connectivity life cycle. Upon learning of the endpoint, the cloud infrastructure applies the `cloudExtEPg` cloud EPG policies accordingly. For example, when a WAN connected client initiates a TCP session with a server within an application (`cloudEPg`) cloud EPG, the `cloudExtEPg` cloud EPG applies its policies to that client endpoint before the communication with the (`cloudEPg`) cloud EPG web server begins. When the client server TCP session ends, and communication between the client and server terminates, the WAN endpoint no longer exists in the cloud infrastructure.

The Cisco Cloud Network Controller uses endpoint selectors to assign endpoints to Cloud EPGs. The endpoint selector is essentially a set of rules that are run against the cloud instances that are assigned to the Azure VNET managed by Cisco CCNC. Any endpoint selector rules that match endpoint instances assign that endpoint to the Cloud EPG. The endpoint selector is similar to the attribute-based microsegmentation available in Cisco ACI.

Cloud Service Endpoint Groups

A cloud service EPG, introduced in Release 5.1(2), is a managed object that is a named logical entity that contains a collection of cloud native or third-party service instances or endpoints. In this situation, an endpoint refers to a particular service instance. For example, an SQL server would be considered an endpoint, and a collection of SQL servers would form a service endpoint group. Other examples of service EPGs would be a collection of Storage Accounts, a collection of Key Vaults, and so on.

Service EPGs have several unique attributes:

- **Service Type:** This attribute indicates what type of cloud service is being grouped. Examples of available service types include **Azure SQL**, **Azure Containter Registry**, **Azure ApiManagement Services**, and so on. The service type **Custom** is used when configuring a third-party service EPG.
- **Deployment Type:** This attribute indicates how and where the service is deployed. Following are the available deployment types:
 - **Cloud Native:** In this type of deployment, the service is instantiated in the cloud provider's network and the user or applications consuming it have a handle to the service. For example, an Azure storage account might reside inside Azure's own VNet, and you would have a URL to access the storage contents.
 - **Cloud Native Managed:** In this type of deployment, the service is instantiated in your VNet or subnet (created through the Cisco Cloud Network Controller). For example, an Azure Kubernetes cluster (AKS) could be deployed in a subnet that is managed by the Cisco Cloud Network Controller.
 - **Third-Party:** This is a deployment where a third-party (not Azure) is providing services through the market place. Access to this service is provided through the private links feature.
- **Access Type:** This indicates how the service will be accessed. Following are the available access types:
 - **Public:** The service will be accessed using the public IP address assigned to it. Access to the public IP address range of a particular service is achieved using the Azure "Service Tags" in the NSG rules.
 - **Private:** The service will be accessed using a private IP address assigned to it. This assignment is done through the creation of private endpoints when the deployment is of type **Cloud Native** and **Third-Party**. In the case of a **Cloud Native Managed** deployment, the private IP is assigned by the service from the subnet IP space.

Only certain deployment types, and certain access types within each deployment type, are supported for each service type, described in the previous bullets. The following table provides more information on the deployment types and access types that are supported for each service type.

Service Type	Provider	Deployment Type/Access Type		
		Cloud Native	Cloud Native Managed	Third-Party
Azure Storage Blob	Microsoft.Storage	Private	N/A	N/A
Azure SQL	Microsoft.Sql	<ul style="list-style-type: none"> • Public • Private 	N/A	N/A
Azure Cosmos DB	Microsoft.DocumentDB	<ul style="list-style-type: none"> • Public • Private 	N/A	N/A
Azure Databricks	Microsoft.Databricks	Public	<ul style="list-style-type: none"> • Private • Public and Private 	N/A
Azure Storage	Microsoft.Storage	<ul style="list-style-type: none"> • Public • Private 	N/A	N/A
Azure Storage File	Microsoft.Storage	Private	N/A	N/A
Azure Storage Queue	Microsoft.Storage	Private	N/A	N/A
Azure Storage Table	Microsoft.Storage	Private	N/A	N/A
Azure Kubernetes Services (AKS)	Microsoft.ContainerService	Private	<ul style="list-style-type: none"> • Private • Public and Private 	N/A
Azure Active Directory Domain Services	Microsoft.AAD	Public	<ul style="list-style-type: none"> • Private • Public and Private 	N/A
Azure Container Registry	Microsoft.ContainerRegistry	<ul style="list-style-type: none"> • Public • Private 	N/A	N/A
Azure ApiManagement Services	Microsoft.ApiManagement	Public	<ul style="list-style-type: none"> • Private • Public and Private 	N/A
Azure Key Vault	Microsoft.KeyVault	<ul style="list-style-type: none"> • Public • Private 	N/A	N/A
Redis Cache	Microsoft.Cache	N/A	<ul style="list-style-type: none"> • Private • Public and Private 	N/A

Service Type	Provider	Deployment Type/Access Type		
		Cloud Native	Cloud Native Managed	Third-Party
Custom Service		<ul style="list-style-type: none"> Public Private 	N/A	Private

• **Service Endpoint Selectors:** Service endpoints can be selected using the existing selectors (used in the cloud EPG selection) as well as the new types of selectors listed below:

- **Resource Name:** The service resource's name
- **Resource ID:** The cloud provider's ID for the resource
- **URL:** The alias or FQDN that identifies the service (the private link alias is used in Azure)

The following table provides more information on the endpoint selectors that are supported for each deployment type.



Note Information for the Cloud Native (Public) deployment type is not provided in the following table because that deployment type does not support endpoint selectors.

Deployment Type	Tags	Region	IP	Resource Name	Resource ID	URL
Cloud Native (Private)	Y	Y	N	Y	Y	N
Cloud Native Managed	N	N	Y	N	N	N
Third-Party	N	N	N	N	N	Y (applicable only for private link connection)

Guidelines and Restrictions for Cloud Service EPGs

You must have the **NSG-per-subnet** configuration enabled if you are configuring cloud service EPGs. See [Security Groups, on page 45](#) for more information.

About Service Types

Additional information specific to certain service types are provided below:

- [Azure Storage, on page 41](#)
- [Azure ApiManagement Services, on page 41](#)
- [Azure Databricks Services, on page 42](#)
- [Azure Active Directory Domain Services, on page 42](#)

- [Azure Kubernetes Services, on page 42](#)
- [Azure Redis Cache, on page 42](#)

Azure Storage

The Azure Storage service type is a general service type that can be broken down into four subtypes:

- Blob
- File
- Table
- Queue

If you were to configure a service EPG with the following values, using the general Azure Storage service type:

- **Service type:** Azure Storage
- **Deployment type:** Cloud Native
- **Access type:** Private

Then four private endpoints are automatically configured for this service EPG, one for each of the four subtypes listed above.

However, if you were to configure a service EPG with the following values, using a more specific Azure Storage service type:

- **Service type:** One of these service types:
 - Azure Storage Blob
 - Azure Storage File
 - Azure Storage Table
 - Azure Storage Queue
- **Deployment type:** Cloud Native
- **Access type:** Private

Then only one private endpoint is automatically configured for this particular subtype for this service EPG.

Note that the four specific Azure Storage subtypes (Blob, File, Table, and Queue) are not allowed if you have an access type of Public with the deployment type of Cloud Native. This is because Azure service tags are not storage subtype specific.

Azure ApiManagement Services

For an Azure ApiManagement (APIM) Services instance to be deployed in a VNet, it needs to be able to access a lot of other Azure services. In order to do this, the security group rules that allow this access must be programmed.

Cisco Cloud Network Controller automates this and configures the rules listed here:

<https://docs.microsoft.com/en-us/azure/api-management/api-management-using-with-vnet#-common-network-configuration-issues>

Azure Databricks Services

Azure Databricks requires the following:

- Access to other services
- Two subnets for deployment, where the subnets are delegated to Microsoft

For Azure Databricks, make the following configurations:

- Before configuring the service EPG, you must configure two subnets specifically for the Azure Databricks Services.
- When configuring the service EPG, you must create two service endpoint selectors that will be used to match the two service subnets.

Once the subnet is identified with the Azure Databricks service EPG through the configured endpoint selectors, Cisco Cloud Network Controller delegates subnets to Azure and configures the rules listed here:

<https://docs.microsoft.com/en-us/azure/databricks/administration-guide/cloud-configurations/azure/vnet-inject>

Azure Active Directory Domain Services

Azure Active Directory Domain Services (ADDS) requires the following:

- Access to other services
- No routing table is attached to the subnet when it is being deployed

The action of de-associating the routing table from the subnet should be done through the Azure portal after configuring the service EPG and before deploying ADDS. The routing table can be attached to the subnet after the deployment is completed.

Cisco Cloud Network Controller automates the programming of the rules listed here:

<https://docs.microsoft.com/en-us/azure/active-directory-domain-services/network-considerations>

Azure Kubernetes Services

Azure Kubernetes Services (AKS) requires access to other services.

Cisco Cloud Network Controller automates the programming of the rules listed here:

<https://docs.microsoft.com/en-us/azure/aks/limit-egress-traffic#required-outbound-network-rules-and-fqdns-for-aks-clusters>

See [Service EPG Configuration Examples, on page 311](#) for an example configuration of the AKS service EPG.

Azure Redis Cache

Azure Redis cache requires access to other services.

Cisco Cloud Network Controller automates the programming of the rules listed here:

<https://docs.microsoft.com/en-us/azure/azure-cache-for-redis/cache-how-to-premium-vnet#outbound-port-requirements>

About Deployment Types

Additional information specific to certain deployment types are provided below:

- [Cloud Native](#), on page 43
- [Cloud Native Managed](#), on page 44

Cloud Native

In this type of deployment, the service is instantiated in the cloud provider's network and the user or applications consuming it have a handle to the service. For example, an Azure storage account might reside inside Azure's own VNet, and you would have a URL to access the storage contents.

The following is an example service EPG with a Cloud Native deployment type:

- **Service Type:** Azure SQL
- **Deployment type:** Cloud Native
- **Access type:** Private

In this example scenario, you would make the following configurations in this order:

1. In the Cisco Cloud Network Controller GUI, create a private link label in a cloud context profile to be used by the Azure SQL service EPG.

Follow the procedures in [Creating a Cloud Context Profile Using the Cisco Cloud Network Controller GUI](#), on page 113. Configure a private link label to be used by the Azure SQL service EPG (for example, SQL-PLL).

2. In the Cisco Cloud Network Controller GUI, create a service EPG of the service type Azure SQL.

Follow the procedures in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI](#), on page 92, using the following parameters:

- **Service Type:** Azure SQL
- **Deployment type:** Cloud Native
- **Access type:** Private

When you are configuring the endpoint selector as part of the process of configuring this type of service EPG, configure the endpoint selector to match the appropriate value for the SQL server.

For example, if you wanted to select an SQL server with the name `ProdSqlServer`, you would make the following selections:

- **Key:** Name
- **Operator:** equals
- **Value:** `ProdSqlServer`

As another example, if you wanted to select an SQL server using the cloud provider's resource ID of `/subscriptions/{subscription-id}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/ProdSqlServer`, you would make the following selections:

- **Key:** Resource ID

- **Operator:** equals
- **Value:**
/subscriptions/{subscription-id}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/ProdSqlServer

3. In the Azure portal, configure the Azure SQL resources in the cloud.

Cloud Native Managed

In this type of deployment, the service is instantiated in your VNet or subnet (created through the Cisco Cloud Network Controller). For example, an Azure ApiManagement Services could be deployed in a subnet that is managed by the Cisco Cloud Network Controller.

The following is an example service EPG with a Cloud Native Managed deployment type:

- **Service Type:** Azure ApiManagement Services
- **Deployment type:** Cloud Native Managed
- **Access type:** Private

In this example scenario, you would make the following configurations in this order:

1. In the Cisco Cloud Network Controller GUI, create a subnet in a cloud context profile to be used by the Azure ApiManagement Services service EPG.

Follow the procedures in [Creating a Cloud Context Profile Using the Cisco Cloud Network Controller GUI, on page 113](#). Configure a subnet to be used by the Azure ApiManagement Services service EPG (for example, 10.50.0.0/16).

2. In the Cisco Cloud Network Controller GUI, create a service EPG of the service type Azure ApiManagement Services.

Follow the procedures in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#), using the following parameters:

- **Service Type:** Azure ApiManagement Services
- **Deployment type:** Cloud Native Managed
- **Access type:** Private

When you are configuring the endpoint selector as part of the process of configuring this type of service EPG, configure the endpoint selector to match the IP address that you used when you created a subnet in the cloud context profile in the first step.

For example, using the example provided in the first step, you would configure this endpoint selector for this service EPG:

- **Key:** IP
- **Operator:** equals
- **Value:** 10.50.0.0/16

3. In the Azure portal, configure the Azure ApiManagement Services resources in the cloud.

Security Groups

In Azure, two types of security groups are used to administer and control network traffic within a virtual network (VNet):

- **Network security groups:** Network security groups, or NSGs, are used in Azure to filter network traffic to and from Azure resources. An NSG is used to define incoming and outgoing security policies, and contains security rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources. For each rule, you can specify source and destination, port, and protocol.

In Cisco Cloud Network Controller, an NSG is automatically configured based on a contract.

- **Application security groups:** Application security groups, or ASGs, are used in Azure to group virtual machine (VM) NICs according to the applications that run on them and define network security policies based on those groups. ASGs are used within an NSG to define these security policies and to apply a network security rule to a specific workload or group of virtual machines.

In Cisco Cloud Network Controller, an ASG is a collection of endpoints for each EPG and is referenced as the source or destination in the NSG security policies.

The way that these security groups are configured, and what they are mapped to, differ depending on the release.

- [Releases Prior to Release 5.1\(2\): NSG-Per-EPG Configurations, on page 45](#)
- [Release 5.1\(2\) and Later: NSG-Per-Subnet Configurations, on page 45](#)
- [Release 5.1\(2g\) and Later: IP-Based Rules for Inter-VRF Contracts in the Same VNet, on page 46](#)

Releases Prior to Release 5.1(2): NSG-Per-EPG Configurations

For releases prior to Release 5.1(2), there is a one-to-one mapping between NSGs in Azure and EPGs on Cisco Cloud Network Controller (these configurations are also referred to as **NSG-per-EPG** configurations throughout this document). These NSGs for the Cisco Cloud Network Controller EPGs are populated with security rules based on contracts associated with the EPGs.

For releases prior to Release 5.1(2), the creation of an EPG in Cisco Cloud Network Controller results in the creation of the following Azure components:

- An ASG, which is used to group all endpoints or virtual machine NICs for each EPG based on the endpoint selectors
- An NSG, which gets associated with all of the NICs in that ASG and provides the security policy definition for that EPG

Release 5.1(2) and Later: NSG-Per-Subnet Configurations

Beginning with Release 5.1(2), in addition to the existing NSG-per-EPG configurations available previously, NSGs in Azure can also have a one-to-one mapping with subnets rather than EPGs on Cisco Cloud Network Controller (these configurations are also referred to as **NSG-per-subnet** configurations throughout this document). By default, NSGs are no longer created for EPGs beginning with Release 5.1(2), and NSGs are no longer associated with the endpoints and VM NICs in the ASG for that EPG. Instead, the NSG for each subnet will contain all of the rules based on the contracts for the ASGs, which have their endpoints discovered in the subnet.

For NSG-per-subnet configurations, the creation of an EPG in Cisco Cloud Network Controller results in the creation of the following Azure components:

- An ASG, which is used to group all endpoints or virtual machine NICs for each EPG based on the endpoint selectors [essentially no change in behavior for ASGs from releases prior to Release 5.1(2)]
- An NSG, which continues to provide the security policy definition for that EPG, but now gets associated with a subnet in a Cisco Cloud Network Controller -managed VNet

Looked at from another perspective:

- Every EPG in a Cisco Cloud Network Controller -managed VNet will have an ASG associated with it, which will group all the endpoints based on the endpoint selectors configured for the EPG.
- Every subnet in a Cisco Cloud Network Controller -managed VNet will have an NSG associated with it.

The default setting for a Greenfield or a fresh Cisco Cloud Network Controller deployment is **NSG-per-subnet**. When manually setting this configuration, as described previously, you can choose either a newer **NSG-per-subnet** configuration or the older **NSG-per-EPG** configuration beginning with Release 5.1(2). However, we recommend choosing the newer **NSG-per-subnet** configuration for several reasons:

- Using the **NSG-per-subnet** configuration reduces the number of NSGs in the VNet, and also reduces the number of rules for deployments with a large number of subnets accessing common shared services. This provides for easier management, since all of the rules can be checked in one NSG for a subnet, rather than for each NSG mapped to individual EPGs or ASGs.
- You must use the **NSG-per-subnet** configuration if you are configuring service EPGs. See [Cloud Service Endpoint Groups, on page 38](#) for more information.

See [Configuring Network Security Groups Using the Cisco Cloud Network Controller GUI, on page 108](#) for instructions on enabling or disabling the NSG-per-EPG or NSG-per-subnet configurations.

Release 5.1(2g) and Later: IP-Based Rules for Inter-VRF Contracts in the Same VNet

Prior to release 5.1(2g), if two EPGs had a contract and were in the same VNet but belonged to different VRFs, ASG-based rules were used to enable communication between those hosted VRFs in that VNet. Azure has a limit of 100 ASGs in rules for every NSG, and this limit could be reached quickly in some situations (for example, if you have one VNet for all of your shared services).

Beginning with release 5.1(2g), if two EPGs have a contract and are in the same VNet but belong to different VRFs, IP-based rules are now used to enable communication between those hosted VRFs in that VNet, which is preferable because an NSG can support 4000 IP addresses in the rules. These IP-based rules are based on endpoints discovered or on subnet selectors used in the EPG.

Guidelines and Limitations for ASGs and NSGs

Following are the guidelines and limitations for ASGs and NSGs.

- [Guidelines and Limitations for Releases Prior to 5.1\(2\), on page 47](#)
- [Guidelines and Limitations for Release 5.1\(2\) or Later, on page 47](#)

Guidelines and Limitations for Releases Prior to 5.1(2)

For releases prior to Release 5.1(2), support is only available for NSG-to-EPG mapping for Cisco Cloud Network Controller.

Guidelines and Limitations for Release 5.1(2) or Later

- Beginning with Release 5.1(2), support is also available for NSG-to-subnet mapping for Cisco Cloud Network Controller. However, you can have *either* the newer NSG-per-subnet configuration or the NSG-per-EPG configuration, but not both in the same Cisco Cloud Network Controller system.
- You can configure one NSG per subnet in a Cisco Cloud Network Controller -managed VNET. Having one NSG per a group of subnets is not supported for Cisco Cloud Network Controller at this time.
- Passthrough devices, such as transparent firewall, will not have NSGs attached to their NICs. If there are multiple passthrough devices sharing a subnet, the passthrough rules for each device will apply to all endpoints in the subnet.

Security Rules

The security rules for NSG differ, depending on whether they are rules for NSG-per-EPG configurations or for NSG-per-subnet configurations. A major distinction on the processing of the security rules between the two types of configurations is the trigger for installing and deleting the rules.

- [NSG-Per-EPG Security Rules, on page 47](#)
- [NSG-Per-Subnet Security Rules, on page 47](#)

NSG-Per-EPG Security Rules

- Once the EPGs and the contract are defined on the Cisco Cloud Network Controller, the NSG security rules that use ASGs as the source and destination are always programmed, regardless of whether an endpoint for the ASG that is referenced in the NSG security rule is discovered or not.
- For inter-VRF contracts:
 - If either the consumer or the provider EPG uses an endpoint selector based on subnet, then the NSG security rules that have the source or destination as the subnet from the EPG selector are always programmed, regardless of the discovery of an endpoint.
 - If the consumer or provider EPG does not use an endpoint selector based on subnet, then the NSG security rules using the endpoint's IP address as the source and destination are programmed, depending on the discovery of an endpoint.
- The rules created for an inter-site contract, where a cloud external EPG (`cloudExtEPG`) is involved, also get pre-programmed without the endpoint getting discovered.

NSG-Per-Subnet Security Rules

The NSG security rules for an EPG are not programmed in a subnet-based NSG until the EPG has at least one endpoint discovered in that subnet.

NSG Behavior With Software Upgrades or Downgrades

Because only NSG-per-EPG mapping is supported for releases prior to Release 5.1(2), and support for NSG-per-subnet mapping became available beginning with Release 5.1(2), certain system configuration changes might have to take place when you are upgrading or downgrading your software in certain situations. The following sections describe these situations and what must occur during these upgrade or downgrade operations.

- [NSG Behavior With Software Upgrades, on page 48](#)
- [NSG Behavior With Software Downgrades, on page 48](#)

NSG Behavior With Software Upgrades

When you perform a standard upgrade from a release prior to Release 5.1(2) to Release 5.1(2) or later, NSGs that were configured using the NSG-per-EPG mapping that was supported for the release prior to Release 5.1(2) will remain as-is after the upgrade. This is because either NSG-per-EPG or NSG-per-subnet configurations are supported for Release 5.1(2) or later, so the older NSG-per-EPG configurations will be retained automatically when performing a standard upgrade to Release 5.1(2) or later.

However, there are benefits to the NSG-per-subnet configuration, so we recommend that you convert the NSG-per-EPG configurations to NSG-per-subnet to take advantage of those benefits. See [Security Groups, on page 45](#) for more information on the different NSG configurations, and [Configuring Network Security Groups Using the Cisco Cloud Network Controller GUI, on page 108](#) for instructions on enabling or disabling the NSG-per-EPG or NSG-per-subnet configurations.

Keep in mind that, after the upgrade, you can have either older NSG-per-EPG or the newer NSG-per-subnet configuration, but you cannot have both in the same Cisco Cloud Network Controller system. See [Guidelines and Limitations for ASGs and NSGs, on page 46](#) for more information.

However, if you backed up your existing Cisco Cloud Network Controller configuration using the procedures in [Creating a Backup Configuration Using the Cisco Cloud Network Controller GUI, on page 118](#), then performed an upgrade and imported the backed-up configuration after the upgrade, the NSG-per-subnet configuration is turned on automatically, and any older NSG-per-EPG configurations are automatically converted to the newer NSG-per-subnet configuration.

NSG Behavior With Software Downgrades

When you downgrade from Release 5.1(2) or later to a release prior to Release 5.1(2), you must manually move any NSG-per-subnet configurations back to the NSG-per-EPG configuration that was supported for releases prior to Release 5.1(2).

Following is the general process that you will follow to transition from NSG-per-subnet configurations to NSG-per-EPG configurations before downgrading the software:

1. Before downgrading the software from Release 5.1(2) or later to a release prior to Release 5.1(2), disable the NSG-per-subnet configuration using the procedures provided in [Configuring Network Security Groups Using the Cisco Cloud Network Controller GUI, on page 108](#). The Cisco Cloud Network Controller software begins the transition from NSG-per-subnet mapping to NSG-per-EPG mapping.
2. Wait until the transition is complete, where the Cisco Cloud Network Controller software has deleted all of the NSGs that were configured as part of the NSG-per-subnet mapping process and has created new NSGs for the NSG-per-EPG mapping configuration. If you attempt to proceed with the downgrade before the transition is complete, you will see an error message and the Cisco Cloud Network Controller software

will not allow you to proceed with the downgrade until this transition from NSG-per-subnet mapping to NSG-per-EPG mapping has completed.



Note You will get an error message if you attempt a software downgrade before the transition is complete when downgrading through the GUI; however, you will not get an error message if you attempt a software downgrade too early when downgrading through the REST API. For that reason, we recommend that you do not downgrade your software through the REST API if you are in this situation.

If you decide to downgrade your software through the REST API, monitor the following MO:

```
hcloudReconcileDone
```

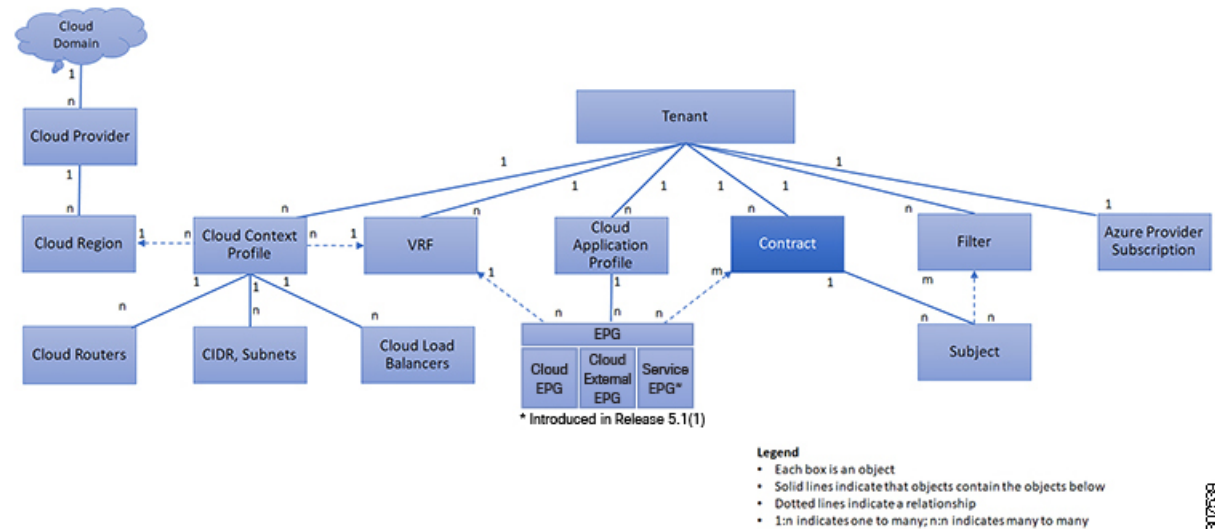
Verify that the property `sgForSubnetModeConverged` is set to `yes` before proceeding with the downgrade through the REST API.

- When you have confirmation that the system has successfully completed the transition back to the NSG-per-EPG mapping, you can downgrade the Cisco Cloud Network Controller software using the instructions provided in the *Cisco Cloud Network Controller for Azure Installation Guide*.

Contracts

In addition to cloud EPGs, contracts (`vzBRCP`) are key objects in the policy model. Cloud EPGs can only communicate with other cloud EPGs according to contract rules. The following figure shows the location of contracts in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 7: Contracts



An administrator uses a contract to select one or more types of traffic that can pass between cloud EPGs, including the protocols and ports allowed. If there is no contract, inter-EPG communication is disabled by default. There is no contract required for intra-EPG communication; intra-EPG communication is always implicitly allowed.

Contracts govern the following types of cloud EPG communications:

- Between cloud EPGs (`cloudEPG`), both intra-tenant and inter-tenant



Note In the case of a shared service mode, a contract is required for inter-tenant communication. A contract is used to specify static routes across VRFs, although the tenant VRF does not enforce a policy.

- Between cloud EPGs and cloud external EPGs (`cloudExtEPG`)

Contracts govern the communication between cloud EPGs that are labeled providers, consumers, or both. The relationship between a cloud EPG and a contract can be either a provider or consumer. When a cloud EPG provides a contract, communication with the cloud endpoints in that cloud EPG can be initiated from cloud endpoints in other cloud EPGs as long as the communication complies with the provided contract. When a cloud EPG consumes a contract, the cloud endpoints in the consuming cloud EPG may initiate communication with any cloud endpoint in a cloud EPG that is providing that contract.



Note A cloud EPG can both provide and consume the same contract. A cloud EPG can also provide and consume multiple contracts simultaneously.

Comma-separated Filters Support for Contract Rule Consolidation

After a contract is created, some of the rules defined in the contract are consolidated and displayed in Azure based on certain criteria. You can combine multiple ports and multiple IP addresses and ranges into a single, easy-to-understand rule. The criteria for consolidation of rules are:

- Rules are consolidated only within a contract. Two rules resulting from two different contracts are not consolidated in Azure.
- The source/ destination address prefixes and destination port(s) are consolidated.
- The conditions for multiple rules to get consolidated together in an NSG are:
 - Same contract
 - Same protocol (UDP, TCP, ICMP)
 - Same direction (inbound, outbound)
 - Same type (SG, IP)
- Overlapping port ranges for same protocol (TCP/UDP) in the same contract are consolidated to one range.

For example, TCP ports 100-200, 150-250 are consolidated to 100-250.

- If 1.2.3.4/32 (any address prefixes) is allowed, and an ext EPG with 0.0.0.0/0 is added, then the allowed Source/Destination IP would be *Any*, not [1.2.3.4/32, 0.0.0.0/0].

Example below shows the EPG1 outbound rules and the consolidated EPG1 outbound rules, based on contracts C1 and C2.

```
Contract C1:
Consumer: EPG1, Provider: EPG2
Filter: TCP (ports 53)
Filter: UDP (port 53, 5000)
```

```
Contract C2:
Consumer: EPG1, Provider: EPG2
Filter: TCP (ports 80, 8080)
```

```
EPG1 outbound rules:
EPG1 -> EPG2    TCP    80
EPG1 -> EPG2    TCP    8080
EPG1 -> EPG2    TCP      53
EPG1 -> EPG2    UDP    53
EPG1 -> EPG2    UDP    5000
EPG1 -> 1.1.1.1/32 TCP    80
EPG1 -> 1.1.1.1/32 TCP    8080
EPG1 -> 1.1.1.1/32 TCP    53
EPG1 -> 1.1.1.1/32 UDP    53
EPG1 -> 1.1.1.1/32 UDP    5000
EPG1 -> 2.2.2.2/32 TCP    80
EPG1 -> 2.2.2.2/32 TCP    8080
EPG1 -> 2.2.2.2/32 TCP      53
EPG1 -> 2.2.2.2/32 UDP    53
EPG1 -> 2.2.2.2/32 UDP    5000
```

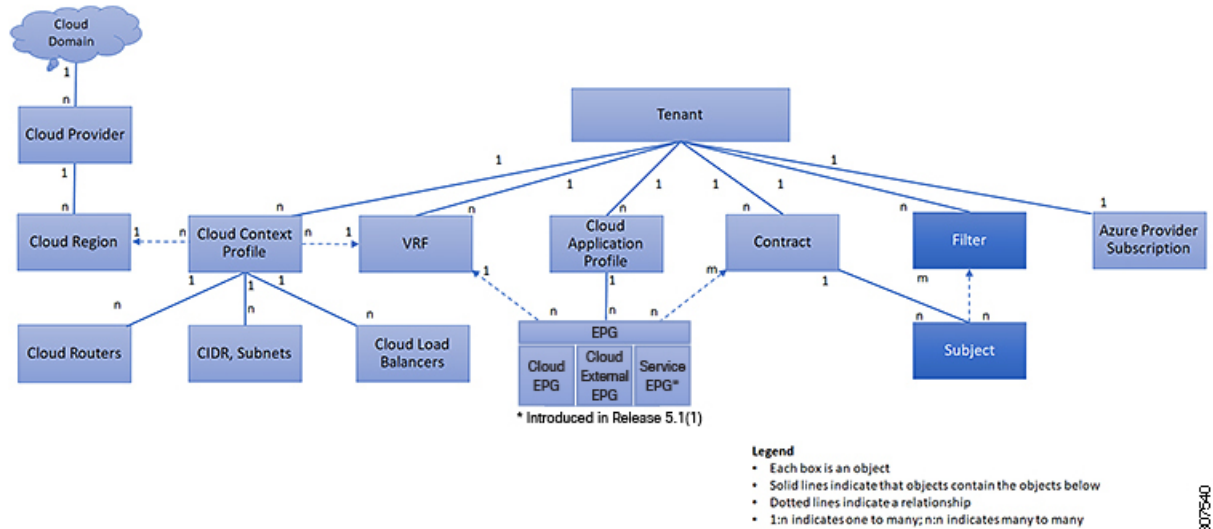
Rules are consolidated by comma-separated filters (consolidated based on C1 and C2):

```
EPG1 -> EPG2    TCP    80,8080
EPG1 -> EPG2    UDP    53,5000
EPG1 -> EPG2    TCP    53
EPG1 -> 1.1.1.1/32, 2.2.2.2/32 TCP    80,8080
EPG1 -> 1.1.1.1/32, 2.2.2.2/32 UDP    53,5000
EPG1 -> 1.1.1.1/32, 2.2.2.2/32 TCP    53
```

Filters and Subjects Govern Cloud EPG Communications

Subject and filter managed-objects enable mixing and matching among cloud EPGs and contracts so as to satisfy various applications or service delivery requirements. The following figure shows the location of application subjects and filters in the management information tree (MIT) and their relation to other objects in the tenant.

Figure 8: Subjects and Filters



Contracts can contain multiple communication rules and multiple cloud EPGs can both consume and provide multiple contracts. A policy designer can compactly represent complex communication policies and re-use these policies across multiple instances of an application.



Note Subjects are hidden in Cisco Cloud Network Controller and not configurable. For rules installed in Azure, source port provided in the filter entry is not taken into account.

Subjects and filters define cloud EPG communications according to the following options:

- Filters are Layer 3 to Layer 4 fields, TCP/IP header fields such as Layer 3 protocol type, Layer 4 ports, and so forth. According to its related contract, a cloud EPG provider dictates the protocols and ports in both the in and out directions. Contract subjects contain associations to the filters (and their directions) that are applied between cloud EPGs that produce and consume the contract.
- Subjects are contained in contracts. A subject within a contract uses filters to specify the type of traffic that can be communicated and how it occurs. For example, for HTTPS messages, the subject specifies the direction and the filters that specify the IP address type (for example, IPv4), the HTTP protocol, and the ports allowed. Subjects determine if filters are unidirectional or bidirectional. A unidirectional filter is used in one direction. Unidirectional filters define in or out communications but not the same for both. Bidirectional filters are the same for both; they define both in and out communications.
- CCNC contracts rendered in Azure constructs are always stateful, allowing return traffic.

About the Cloud Template

The cloud template provides a template that configures and manages the Cisco Cloud Network Controller infra network. The template requires only the most essential elements for the configuration. From these elements, the cloud template generates a detailed configuration necessary for setting up the Cisco Cloud Network Controller infra network. However, it is not a one-time configuration generation—it is possible to

add, modify, or remove elements of the template input. The cloud template updates the resulting configuration accordingly.

One of the central things in the Azure network configuration is the Virtual Private Cloud (VNET). Azure supports many regions worldwide and one VNET is specific to one region.

The cloud template accepts one or more region names and generates the entire configuration for the infra VNETs in those regions. They are the infra VNETs. The Cisco Cloud Network Controller -managed object (MO) corresponding to the Azure VNET is `cloudCtxProfile`. For every region specified in the cloud template, it generates the `cloudCtxProfile` configuration. A `cloudCtxProfile` is the topmost MO for all the configuration corresponding to a region. Underneath, it has many of other MOs organized as a tree to capture a specific configuration. The `cloudCtxProfile` MO for the infra VNet is generated by the cloud template. It carries `ctxProfileOwner == SYSTEM`, which means that this MO is generated by the system. For the non-infra network, it is possible to configure `cloudCtxProfile` directly; in this case, `cloudCtxProfile` carries `ctxProfileOwner == USER`.

A primary property of an Azure VNet is the CIDR. In Cisco Cloud Network Controller, you can choose and deploy CIDRs in the user VNETs. The CIDRs for the infra VNet are provided by users to the cloud template during the initial setup of the cloud site, and are deployed to the Azure cloud by the cloud template.

Beginning with release 5.0(2), a new property called `createdBy` is added for the CIDR. The default value for this `createdBy` property is `USER`.

- For all user-created CIDRs, the value for the `createdBy` property is set to `USER`.
- For cloud template-created CIDRs, the value for the `createdBy` property is set to `SYSTEM`.

Multiple CIDR and subnet blocks can be configured on the infra VNet. You can create CIDRs and associate subnets in the infra VNet. The cloud template subnets will be mapped to the overlay-1 VRF, but for user-created subnets, you have to manually configure the subnet-to-VRF mapping to the secondary VRF in the same infra VNet. All subnets in the respective VRFs will have separate route tables in the cloud for VRF segregation.

You can create cloud EPGs and cloud external EPGs in the infra tenant, where all the cloud EPGs and cloud external EPGs will be associated with the secondary VRF in the infra tenant. A cloud EPG in the secondary VRF can communicate with other cloud EPGs and cloud external EPGs in the secondary VRF, and can also communicate with cloud EPGs in other user tenant VRFs. We recommend that you do not use existing "cloud-infra" application profiles, and instead create a new application profile in the infra tenant and associate that new application profile to the cloud EPGs and cloud external EPGs in the secondary VRF.

For more information, see [Creating an Application EPG Using the Cisco Cloud Network Controller GUI, on page 81](#).

The cloud template generates and manages a huge number of MOs in the `cloudCtxProfile` subtree including, but not limited to, the following:

- Subnets
- Cloud routers
- IP address allocation for the cloud router interfaces
- IP address allocation and configuration for tunnels
- IP address allocation and configuration for loopbacks

Without the cloud template, you would be responsible for configuring and managing these.

The *Cisco Cloud Template MO* table contains a brief summary of the inputs (MOs) to the cloud template.

Table 5: Cloud Template MOs

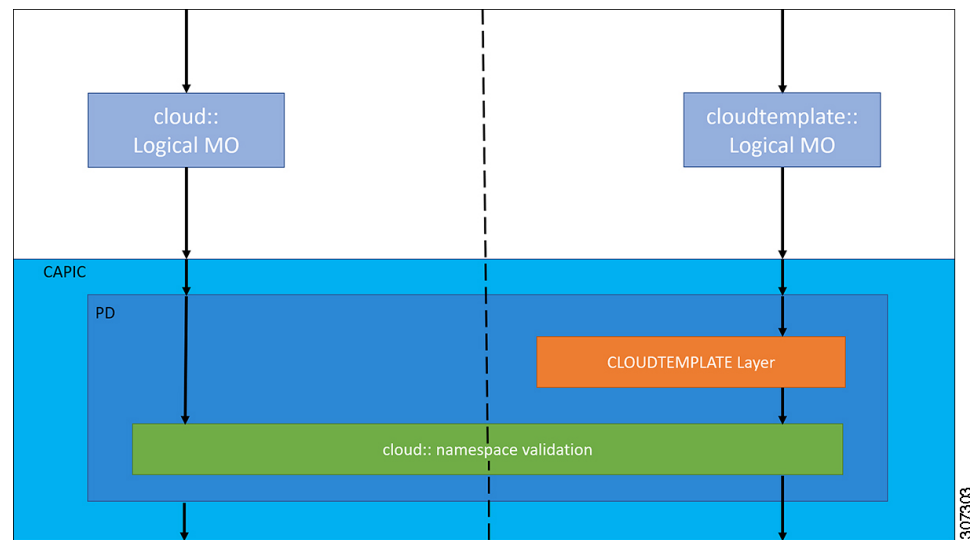
MO	Purpose
cloudtemplateInfraNetwork	The root of the cloud template configuration. Attributes include: numRoutersPerRegion—The number of cloud routers for each cloudRegionName specified under cloudtemplateIntNetwork.
cloudtemplateProfile	Configuration profile for all the cloud routers. Attributes include: <ul style="list-style-type: none"> routerUsername <p>Note</p> <ul style="list-style-type: none"> The username cannot be "admin." Any username restrictions from Azure applies. <ul style="list-style-type: none"> routerPassword routerThroughput routerLicenseToken routeDataInterfacePublicIP routerMgmtInterfacePublicIP
cloudtemplateIntNetwork	Contains a list of regions, which specify where you deploy the cloud routers. Each region is captured through a cloudRegionName child MO
cloudtemplateExtNetwork	Contains infra network configuration input that is external of the cloud. Contains a list of regions where cloud routers are configured for external networking. Each region is captured through a cloudRegionName child MO
cloudtemplateVpnNetwork	Contains information for setting up a VPN with an ACI on-premises site or another Cisco Cloud Network Controller site.
cloudtemplateIpSecTunnel	Captures the IP address of the IPsec peer in the ACI on-premises site.
cloudtemplateOspf	Captures the OSPF area to be used for the VPN connections.
cloudtemplateBgpEvpn	Captures the peer IP address, ASN, and so forth, for setting up the BGP session with the on-premises site.

In Cisco Cloud Network Controller, the layering of MOs is slightly different from a regular Cisco APIC due to the cloud template. In a regular Cisco APIC, you post logical MOs that go through two layers of translation:

1. Logical MO to resolved MO
2. Resolved MO to concrete MO

In Cisco Cloud Network Controller, there is an additional layer of translation for the infra network. This additional layer is where the cloud template translates logical MOs in the `cloudtemplate::` namespace to logical MOs in the cloud namespace. For configurations outside of the infra network, you post logical MOs in the cloud namespace. In this case, the MOs go through the usual two-layer translation as in the regular Cisco APIC.

Figure 9: Cloud and Cloud Template MO Conversion



Note For information about configuring the cloud template, see [Configuring Cisco Cloud Network Controller Components, on page 59](#)

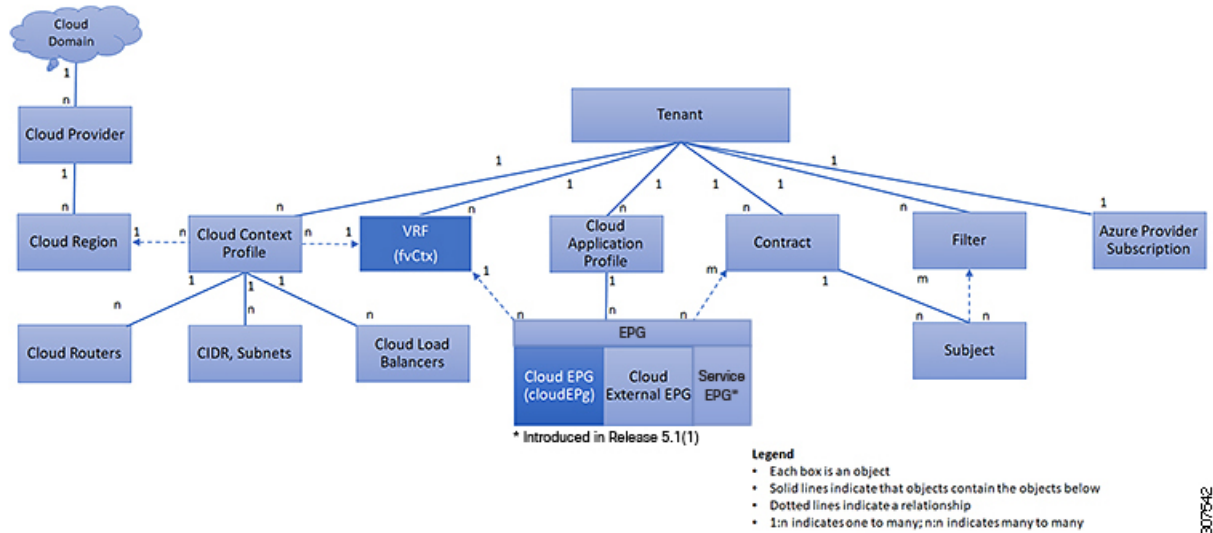
Managed Object Relations and Policy Resolution

Relationship-managed objects express the relation between managed object instances that do not share containment (parent-child) relations. MO relations are established between the source MO and a target MO in one of the following two ways:

- An explicit relation, such as with `cloudRsCloudEPgCtx`, defines a relationship that is based on the target MO distinguished name (DN).
- A named relation defines a relationship that is based on the target MO name.

The dotted lines in the following figure show several common MO relations.

Figure 10: MO Relations



For example, the dotted line between the cloud EPG and the VRF defines the relation between those two MOs. In this figure, the cloud EPG (`cloudEPg`) contains a relationship MO (`cloudRsCloudEPgCtx`) that is named with the name of the target VRF MO (`fvCtx`). For example, if production is the VRF name (`fvCtx.name=production`), then the relation name is production (`cloudRsCloudEPgCtx.tnFvCtxName=production`).

In the case of policy resolution based on named relations, if a target MO with a matching name is not found in the current tenant, the CCNC cloud infrastructure tries to resolve in the common tenant. For example, if the user tenant cloud EPG contained a relationship MO targeted to a VRF that did not exist in the tenant, the system tries to resolve the relationship in the common tenant. If a named relation cannot be resolved in either the current tenant or the common tenant, the CCNC cloud infrastructure attempts to resolve to a default policy. If a default policy exists in the current tenant, it is used. If it does not exist, the CCNC cloud infrastructure looks for a default policy in the common tenant. Cloud context profile, VRF, and contract (security policy) named relations do not resolve to a default.

Default Policies



Warning Default policies can be modified or deleted. Deleting a default policy can result in a policy resolution process to complete abnormally.

The CCNC cloud infrastructure includes default policies for many of its core functions. Examples of default policies include the following:

- Cloud Azure provider (for the infra tenant)
- Monitoring and statistics



Note To avoid confusion when implementing configurations that use default policies, document changes made to default policies. Be sure that there are no current or future configurations that rely on a default policy before deleting a default policy. For example, deleting a default firmware update policy could result in a problematic future firmware update.

A default policy serves multiple purposes:

- Allows a cloud infrastructure administrator to override the default values in the model.
- If an administrator does not provide an explicit policy, the Cisco Cloud Network Controller applies the default policy. An administrator can create a default policy and the Cisco Cloud Network Controller uses that unless the administrator provides any explicit policy.

The following scenarios describe common policy resolution behavior:

- A configuration explicitly refers to the default policy: if a default policy exists in the current tenant, it is used. Otherwise, the default policy in tenant **common** is used.
- A configuration refers to a named policy (not default) that does not exist in the current tenant or in tenant **common**: if the current tenant has a default policy, it is used. Otherwise, the default policy in tenant **common** is used.



Note The scenario above does not apply to a VRF in a tenant.

- A configuration does not refer to any policy name: if a default policy exists in the current tenant, it is used. Otherwise, the default policy in tenant **common** is used.

The policy model specifies that an object is using another policy by having a relation-managed object (MO) under that object and that relation MO refers to the target policy by name. If this relation does not explicitly refer to a policy by name, then the system tries to resolve a policy that is called default. Cloud context profiles and VRFs are exceptions to this rule.

Shared Services

Cloud EPGs in one tenant can communicate with cloud EPGs in another tenant through a contract interface that is contained in a shared tenant. Within the same tenant, a cloud EPG in one VRF can communicate with another cloud EPG in another VRF through a contract defined in the tenant. The contract interface is an MO that can be used as a contract consumption interface by the cloud EPGs that are contained in different tenants. By associating to an interface, a cloud EPG consumes the subjects that are represented by the interface to a contract contained in the shared tenant. Tenants can participate in a single contract, which is defined at some third place. More strict security requirements can be satisfied by defining the tenants, contract, subjects, and filter directions so that tenants remain isolated from one another.

Follow these guidelines when configuring shared services contracts:

- A shared service is supported only with non-overlapping and non-duplicate CIDR subnets. When configuring CIDR subnets for shared services, follow these guidelines:
 - CIDR subnets leaked from one VRF to another must be disjointed and must not overlap.

- CIDR subnets advertised from multiple consumer networks into a VRF or vice versa must be disjointed and must not overlap.
- Inter-tenant contracts require a global scope.



CHAPTER 4

Configuring Cisco Cloud Network Controller Components

- [About Configuring the Cisco Cloud Network Controller](#), on page 59
- [Configuring the Cisco Cloud Network Controller Using the GUI](#), on page 59
- [Configuring Cisco Cloud Network Controller Using the REST API](#), on page 147

About Configuring the Cisco Cloud Network Controller

You create the Cisco Cloud Network Controller components using either the Cisco Cloud Network Controller GUI or the REST API. This section explains how to create configuration, application management, operations, and administrative components.



Note

- For information about configuring a load balancer and service graph, see [Deploying Layer 4 to Layer 7 Services](#), on page 177.
 - For information about the GUI, such as navigation and a list of configurable components, see [About the Cisco Cloud Network Controller GUI](#), on page 13.
-

Configuring the Cisco Cloud Network Controller Using the GUI

Creating a Tenant Using the Cisco Cloud Network Controller GUI

This section explains how to create a tenant using the Cisco Cloud Network Controller GUI.

Before you begin

- You can create a tenant that is managed by the Cisco Cloud Network Controller or a tenant that is unmanaged. To establish a managed tenant, you must first obtain the Azure subscription ID from the Azure portal. You enter the subscription ID in the appropriate field of the Cisco Cloud Network Controller when creating the tenant. Before you can use the managed tenant, you must explicitly grant the Cisco Cloud Network Controller permission to manage the subscription. The steps for doing so are displayed

in the Cisco Cloud Network Controller GUI during tenant creation. The steps for the infra tenant, however, are displayed in the infra tenant details view:

1. Click the **Navigation** menu > **Application Management** subtab.
2. Double-click the infra tenant.
3. Click **View Azure Role Assignment Command**. The steps for granting the Cisco Cloud Network Controller permission to manage the subscription are displayed.



Note For information about obtaining the Azure subscription ID, see the Microsoft Azure documentation.

- Creating an unmanaged tenant requires obtaining a directory (Azure Tenant) ID, an Azure enterprise application ID, and a client secret from the enterprise application. For more information, see the Microsoft Azure documentation.



Note Cisco Cloud Network Controller does not disturb Azure resources created by other applications or users. It only manages the Azure resources created by itself.

- The required steps to explicitly grant the Cisco Cloud Network Controller permission to manage a given subscription are located in the Cisco Cloud Network Controller GUI. When creating a tenant, the steps are displayed after entering the client secret.
- Cisco Cloud Network Controller enforces ownership checks to prevent deployment of policies in the same tenant-region combination done either intentionally or by mistake. For example, assume that Cisco Cloud Network Controller is deployed in Azure subscription IA1 in region R1. Now you want to deploy a tenant TA1 in region R2. This tenant deployment i.e. account-region combination TA1-R2 is now owned by IA1-R1. If another Cisco Cloud Network Controller attempts to manage the same tenant-region combination later (say CNC2 in Azure subscription IA2 deployed in region R3), this will not be allowed because the current owner for the deployment TA1-R2 is IA1-R1. In other words, only one account in one region can be managed by one Cisco Cloud Network Controller. Example below shows some valid and wrong deployment combinations.

```
CNC1 :
IA1-R1: TA1-R1- ok
        TA1-R2- ok

CNC2 :
IA1-R2: TA1-R1- not allowed
        TA1-R3- ok

CNC3 :
IA2-R1: TA1-R1- not allowed
        TA1-R4- ok
        TA2-R4- ok
```

- Ownership enforcement is done using Azure Resource Groups. When a new tenant in subscription TA1 in region R2 is managed by Cisco Cloud Network Controller, a Resource Group CNC_TA1_R2 (e.g. CNC_123456789012__eastus2) is created in the subscription. This Resource Group has a resource tag AciOwnerTag with value IA1_R1_TA1_R2, assuming it was managed by Cisco Cloud Network Controller

in subscription IA1 and deployed in region R1. If the AciOwnerTag mismatch happens, tenant-region management is aborted.

Here is a summary of AciOwnerTag mismatch cases:

- Initially Cisco Cloud Network Controller is installed in a subscription, and then taken down and Cisco Cloud Network Controller is installed in a different subscription. All existing tenant-region deployment will fail.
- Another Cisco Cloud Network Controller is managing the same tenant-region.

In ownership mismatch cases, **retry** (to setup tenant-region again) is not currently supported. As a workaround, if you are certain that no other Cisco Cloud Network Controller is managing the same tenant-region combination, logon to the tenant's Azure subscription and manually remove the affected Resource Group (for example: CNC_123456789012__eastus2). Next, reload Cisco Cloud Network Controller or delete and add the tenant again.

- For both the infra tenants and the user tenants, support is available for both managed identity and unmanaged identity/service principal when dealing with authorization or credentials.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create Tenant**. The **Create Tenant** dialog box appears.

Step 4 Choose the appropriate options and enter the appropriate values in each field as listed in the following *Create Tenant Dialog Box Fields* table then continue.

Table 6: Create Tenant Dialog Box Fields

Properties	Description
Name	Enter the name of the tenant.
Description	Enter a description of the tenant.
Settings	
Add Security Domain	To add a security domain for the tenant: <ol style="list-style-type: none"> Click Add Security Domain. The Select Security Domains dialog appears with a list of security domains in the left pane. Click to choose a security domain. Click Select to add the security domain to the tenant.
Azure Subscription	

Properties	Description
Azure Active Directory ID	<p>Note This field is only valid for the Service Principal or Unmanaged Identity access type.</p> <p>Enter the active directory ID.</p> <p>Note For information about obtaining the active directory ID, see the Azure documentation or support.</p>
Default Subscription	Select a default Azure subscription.

Properties	Description
Azure Subscription	

Properties	Description
	<p>Beginning with 26.0(2), you can now add multiple Azure subscriptions under a single tenant. Click Add Account to enter the Azure subscription ID. You can also check the box to set this account as a default account. Enter the name for the subscription.</p> <p>Note You can only select one account as default.</p> <ul style="list-style-type: none"> • Choose an access type: <ul style="list-style-type: none"> • Service Principal or Unmanaged Identity—Choose this option if the tenant subscription is not managed by the Cisco Cloud Network Controller. • Managed Identity—Choose this option if the tenant subscription is managed by the Cisco Cloud Network Controller. <p>Note You can assign either Service Principal or Managed Identity to the infra tenant.</p> <p>For more information, see Understanding Tenants, Identities, and Subscriptions, on page 24.</p> <ul style="list-style-type: none"> • Enter the Application ID. <p>Note This field is only valid for the Service Principal or Unmanaged Identity access type.</p> <p>Note For information about obtaining the application ID, see the Azure documentation or support.</p> • Enter the Client Secret. <p>Note This field is only valid for the Service Principal or Unmanaged Identity access type.</p> <p>Note <ul style="list-style-type: none"> • For information about creating a client secret, see the Azure documentation or support. • You must explicitly grant Cisco Cloud Network Controller permission to manage a given subscription. Go to the Azure portal and follow these steps: <ol style="list-style-type: none"> a. Open the Cloud Shell </p>

Properties	Description
	<ul style="list-style-type: none"> b. Choose 'Bash' c. Copy and paste the command displayed in the Cisco Cloud Network Controller GUI. <ul style="list-style-type: none"> • To add a security domain for the account: <ul style="list-style-type: none"> a. Click Add Security Domain. The Select Security Domains dialog appears with a list of security domains in the left pane. b. Click to choose a security domain. c. Click Select to add the security domain to the tenant.

Step 5 Click **Save** when finished.

Creating an Application Profile Using the Cisco Cloud Network Controller GUI

This section explains how to create an application profile using the Cisco Cloud Network Controller GUI.

Before you begin

Create a tenant.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create Application Profile**. The **Create Application Profile** dialog box appears.

Step 4 Enter a name in the **Name** field.

Step 5 Choose a tenant:

a) Click **Select Tenant**.

The **Select Tenant** dialog box appears.

b) From the **Select Tenant** dialog, click to choose a tenant in the left column then click **Select**.

You return to the **Create Application Profile** dialog box.

Step 6 Enter a description in the **Description** field.

Step 7 Click **Save** when finished.

Creating a VRF Using the Cisco Cloud Network Controller GUI

This section explains how to create a VRF using the Cisco Cloud Network Controller GUI.

Before you begin

Create a tenant.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create VRF**. The **Create VRF** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create VRF Dialog Box Fields* table then continue.

Table 7: Create VRF Dialog Box Fields

Properties	Description
General	
Name	Enter a name for the VRF in the Name field. All VRFs are assigned a <i>vrfEncoded</i> value. If the Tenant and VRF name combination has more than 32 characters, then, a VRF name (which also contains the tenant name) is identified in the cloud router using the <i>vrfEncoded</i> value. To see the <i>vrfEncoded</i> value, navigate to Application Management > VRFs subtab. Click a VRF on the right hand pane and look for <i>Encoded VRF Name in Cloud Router</i> .
Tenant	To choose a tenant: a. Click Select Tenant . The Select Tenant dialog box appears. b. From the Select Tenant dialog, click to choose a tenant in the left column then click Select . You return to the Create VRF dialog box.
Description	Enter a description of the VRF.

Step 5 When finished, click **Save**.

Creating an External Network Using the Cisco Cloud Network Controller GUI

This procedure describes how to create an external network. You can have a single external network that can connect to multiple routers on the on-premises site, or you can have multiple external networks with multiple VRFs that you can use to connect to CCRs.

Before you begin

You must have a hub network created before you can create an external network.

- Step 1** In the left navigation bar, navigate to **Application Management > External Networks**.
The configured external networks are displayed.
- Step 2** Click **Actions**, then choose **Create External Network**.
The **Create External Network** window appears.
- Step 3** Enter the appropriate values in each field as listed in the following *Create External Network Dialog Box Fields* table then continue.

Table 8: Create External Network Dialog Box Fields

Properties	Description
General	
Name	Enter the name for the external network.
VRF	<p>This external VRF will be used for external connectivity with external non-ACI devices. You can create multiple external VRFs for this purpose.</p> <p>This VRF will be identified as an external VRF if the VRF has all three of the following characteristics:</p> <ul style="list-style-type: none"> • Configured under the infra tenant • Associated with an external network • Not associated with a cloud context profile <p>Any VRF that is associated with an external network becomes an external VRF. The external VRF is not allowed to be associated with a cloud context profile or subnet.</p> <p>To choose an external VRF:</p> <ol style="list-style-type: none"> a. Click Select VRF. The Select VRF dialog box appears. b. From the Select VRF dialog, click to choose a VRF in the left column. You can also create a VRF using the + Create VRF option. c. Click Select. You return to the Create External Network dialog box.
Host Router Name	This field is not editable. The default host router is automatically selected.
Settings	

Properties	Description
Regions	<p>To choose a region:</p> <ol style="list-style-type: none"><li data-bbox="373 338 1487 422">a. Click Add Regions. The Select Regions dialog box appears. The regions that you selected as part of the First Time Setup are displayed here.<li data-bbox="373 485 1487 562">b. From the Select Regions dialog, click to choose a region in the left column then click Select. You return to the Create External Network dialog box.

Properties	Description
VPN Networks	

Properties	Description
	<p>The VPN networks entries are used for external connectivity. All configured VPN networks will be applied to all the selected regions.</p> <p>To add a VPN network:</p> <ol style="list-style-type: none"> a. Click Add VPN Network. The Add VPN Network dialog box appears. b. In the Name field, enter a name for the VPN network. c. Click + Add IPSec Peer. A tunnel is created for each IPSec peer entry. d. Enter values for the following fields for the IPSec tunnel that you want to add: <ul style="list-style-type: none"> • Public IP of IPSec Tunnel Peer • Pre-Shared Key • IKE Version: Select ikev1 or ikev2 for IPSec tunnel connectivity • BGP Peer ASN • Subnet Pool Name: Click Select Subnet Pool Name. The Select Subnet Pool Name dialog box appears. Select one of the available subnet pools that are listed, then click Select. <p>Note Additional IPsec tunnel subnet pools can be added in the External Networks page, or through the Cloud Network Controller First Time Set Up, if necessary. For more information on adding additional subnet pools through the Cloud Network Controller First Time Set Up, see the chapter "Configuring Cisco Cloud Network Controller Using the Setup Wizard" in the <i>Cisco Cloud Network Controller for Azure Installation Guide</i>, Release 25.1(x). The subnet pool size should be large enough to accommodate the number of IPsec tunnels that will get created.</p> <ul style="list-style-type: none"> • IPsec Tunnel Source Interfaces: Using the entries in this field, the Cisco Cloud Network Controller creates one IPsec tunnel from each selected source interface to the destination IP address. <p>Note ikev2 is the default option in this field. The IPsec tunnel source interfaces feature is supported only with the IKEv2 configuration.</p> <p>gig3 is selected by default. Choose one or more from the following interfaces:</p> <ul style="list-style-type: none"> • gig2: The GigabitEthernet2 interface • gig3: The GigabitEthernet3 interface • gig4: The GigabitEthernet4 interface <p>Note After you have configured the IPsec tunnel source interfaces in this external network, you can then configure IPsec tunnel source interfaces in additional networks where tunnels to the same destination can be formed, as described in Routing Policies: Release 25.0(2), on page 9.</p>

Properties	Description
	<p>e. Click the checkmark to add this IPsec tunnel.</p> <p>Click + Add IPsec Tunnel if you want to add another IPsec tunnel.</p> <p>f. Click Add in the Add VPN Network dialog box.</p> <p>You return to the Create External Network dialog box.</p>

- Step 4** When you have finished creating the external network, click **Save**.
After you click **Save** in the **Create External Network** window, cloud routers are then configured in AWS.

Configuring the Global Inter-VRF Route Leak Policy

The global inter-VRF route leak policy feature is introduced in release 25.0(2).

Before you begin

Review the information provided in [Route Leaking Between Internal VRFs, on page 9](#) before making any changes in the **Contract Based Routing** area in the **Cisco Cloud Network Controller Setup** window.

- Step 1** Click the **Intent** icon. The **Intent** menu appears.

- Step 2** A list of options appear in the **Intent** menu.

Under **Workflows**, click **Cisco Cloud Network Controller Setup**.

The **Setup - Overview** dialog box appears.

- Step 3** Under **Advanced Settings**, click **Edit Configuration**. In the **Contract Based Routing** area, note the current setting for the **Contract Based Routing** field.

The **Contract Based Routing** setting reflects the current internal VRF route leak policy, which is a global policy under the infra tenant where a **Yes** or **No** is used to indicate whether contracts can drive routes in the absence of route maps:

- **No:** Default setting. Indicates that routes are not leaked based on contracts, and are leaked based on route maps instead.
- **Yes:** Indicates that routes are leaked based on contracts in the absence of route maps. When enabled, contracts drive routing when route maps are not configured. When route maps exist, route maps always drives routing.

- Step 4** Determine if you want to change the current setting for the **Contract Based Routing** field.

Follow these procedures if you would like to change from one setting to another:

- **Yes setting to No(disabling contract-based routing):** In this situation, the assumption is that you have contract-based routing configured currently and you want to shift over to route map-based routing. This can be disruptive if the route map-based routing is not configured before you shift from contract-based routing to map-based routing.

Before moving from the **Yes** setting to the **No** setting in this situation, make the following changes:

- Between all pairs of VRFs that have existing contracts, enable route maps-based route leaking.

Follow the procedures provided in [Configuring Inter-VRF Route Leaking Using the Cisco Cloud Network Controller GUI, on page 72](#).

- b. Disable the contract-based route policy in the global policy.

Switch in the **Contract Based Routing** field from the **Yes** setting to the **No** setting to move from contract-based routing to route map-based routing.

- c. Change the routing to reflect any granularity that is required based on the new route map-based routing that you enabled.

- **Changing from No setting to Yes (enabling contract-based routing):** In this situation, the assumption is that you have map-based routing configured currently and you want to shift over to contract-based routing. This is not a disruptive operation, but rather is an additive operation, since both contracts and route maps can be enabled between a pair of VRFs. In that situation, route maps take precedence over contracts when enabling routing. With map-based routing enabled, adding contract-based routing should be non-disruptive.

For that reason, you do not have to make any changes before shifting from the **No** setting to the **Yes** setting in this situation. However, if you do not want to have both contracts and route maps enabled between a pair of VRFs, and you want to move completely to contract-based routing, you should completely set up contracts between the VRFs and delete the route maps between the VRFs before shifting to the **Yes** setting in the **Contract Based Routing** field.

Step 5 If you want to change the current setting for the **Contract Based Routing** area, change the setting based on the type of routing that you want.

Step 6 Click **Save and Continue** when you have finished the **Cisco Cloud Network Controller Setup** configurations.

Configuring Leak Routes Using the Cisco Cloud Network Controller GUI

The procedures for configuring leak routes using the Cisco Cloud Network Controller GUI will vary slightly, depending on the release:

- For releases prior to 25.0(2), you can configure an independent routing policy to specify which routes to leak between internal and external VRFs when you are setting up routing between an ACI cloud site and an external destination using the external connectivity feature. See [Configuring Inter-VRF Route Leaking Using the Cisco Cloud Network Controller GUI, on page 72](#) for those procedures.
- For releases 25.0(2) and later, support is available for route maps-based route leaking between a pair of internal VRFs. See [Configuring Leak Routes for Internal VRFs Using the Cisco Cloud Network Controller GUI, on page 75](#) for those procedures.

Configuring Inter-VRF Route Leaking Using the Cisco Cloud Network Controller GUI

Configuring leak routes is part of the release 25.0(1) update where routing and security policies are configured separately. Using inter-VRF routing, you can configure an independent routing policy to specify which routes to leak between internal and external VRFs when you are setting up routing between an ACI cloud site and an external destination using the external connectivity feature. See [Understanding Supported Routing and Security Policies, on page 6](#) for more information.

The external destination must be configured manually using the [Enabling Connectivity From the Azure Site to External Devices, on page 77](#) procedures. The external destination could be another cloud site, an ACI on-premises site or a branch office.

**Note**

- Use these procedures to configure routing policies independent of security policies only between internal and external VRFs.
- Do not use these procedures to configure routing between a pair of internal VRFs; use contracts as you normally would prior to release 25.0(1) in that case.

- Step 1** In the left navigation bar, navigate to **Application Management > VRFs**.
The configured VRFs are displayed.
- Step 2** Click the **Leak Routes** tab.
Any already-configured leak routes are displayed.
- Step 3** Click **Actions**, then choose **Create Leak Route**.
The **Create Leak Route** window appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create Leak Routes Dialog Box Fields* table then continue.

Table 9: Create Leak Routes Dialog Box Fields

Properties	Description
Source VRF	<p>To choose a source VRF:</p> <ol style="list-style-type: none"> Click Select a Source VRF. The Select a VRF dialog box appears. From the Select a VRF dialog, click to choose a VRF in the left column to use for the source VRF. Note that the source VRF can be an internal or an external VRF. Click Select to select this source VRF. You return to the Create Leak Route dialog box.
Destination VRF	<p>To choose a destination VRF:</p> <ol style="list-style-type: none"> Click Select a Destination VRF. The Select a VRF dialog box appears. From the Select a VRF dialog, click to choose a VRF in the left column to use for the destination VRF. Note that the destination VRF cannot be an internal VRF if the source VRF is also internal VRF. Click Select to select this destination VRF. You return to the Create Leak Route dialog box.

Properties	Description
Type	<p>Choose the type of leaked route that you want to configure:</p> <ul style="list-style-type: none"> • Leak All: Select to configure all routes to leak from the source VRF to the destination VRF. The entry <code>0.0.0.0/0</code> is entered automatically in the subnet IP area by default in this case. • Subnet IP: Select to configure a specific subnet IP address as the route to leak from the source VRF to the destination VRF. The Subnet IP box appears. In the Subnet IP box, enter a subnet IP address as the route to leak between VRFs.

Step 5 When finished, click **Save**.
The **Success** window appears.

Step 6 Determine if you want to configure additional inter-VRF route leaking.

- If you want to add another route to leak between a pair of VRFs, click the **Add Another Leak Route** option in the **Success** window.

You are returned to the **Add Leak Route** window. Repeat [Step 4, on page 73](#) through [Step 5, on page 74](#) to configure another route to leak between a pair of VRFs.

- If you want to add a reverse route, where:

- The destination VRF from the previous configuration now becomes the source VRF, and
- The source VRF from the previous configuration now becomes the destination VRF

Then click the **Add Reverse Leak Route** option in the **Success** window.

You are returned to the **Add Leak Route** window. Repeat [Step 4, on page 73](#) through [Step 5, on page 74](#) to configure another route, but this time:

- In the **Source VRF** field, select the VRF that you had selected as a destination VRF in the previous configuration.
- In the **Destination VRF** field, select the VRF that you had selected as a source VRF in the previous configuration.

Step 7 When you have finished configuring leak routes, click **Done**.

The **Leak Routes** tab in the main **VRFs** page is displayed again, with the newly configured leak route displayed.

Step 8 To get more information on a source or destination VRF, or to make changes to a configured leak route, double-click the VRF in the **Leak Routes** tab in the main **VRFs** page.
The **Overview** page for that VRF is displayed.

Step 9 Click the **Application Management** tab at the top of the VRF page, then click the **Leak Routes** tab in the left nav bar. The leak routes associated with this particular VRF are displayed.

Step 10 Configure additional leak routes associated with this VRF, if necessary.

- To add a leak route from this VRF, click **Actions**, then choose **Add Leak Route from <VRF_name>**.

The **Add Leak Route** window appears. Enter the necessary information as you did previously using the information in [Step 4, on page 73](#). Note that the entry in the **Source VRF** is pre-selected and cannot be changed in this situation.

- To add a leak route to this VRF, click **Actions**, then choose **Add Leak Route to <VRF_name>**.

The **Add Leak Route** window appears. Enter the necessary information as you did previously using the information in [Step 4, on page 73](#). Note that the entry in the **Destination VRF** is pre-selected and cannot be changed in this situation.

What to do next

You have now configured the routing policy. Since the routing and security policies are separate, you now need to configure the security policy separately:

- [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#): Use these procedures to create an external EPG.
- [Creating a Contract Using the Cisco Cloud Network Controller GUI, on page 103](#): Use these procedures to create a contract between the external EPG and the cloud EPG.

Configuring Leak Routes for Internal VRFs Using the Cisco Cloud Network Controller GUI

Beginning with release 25.0(2), support is available for route maps-based route leaking between a pair of internal VRFs, as described in [Route Leaking Between Internal VRFs, on page 9](#). This feature is an extension of the routing and security split update provided in release 25.0(1), where routing and security policies are configured separately.

- Step 1** In the left navigation bar, navigate to **Application Management > VRFs**.
The configured VRFs are displayed.
- Step 2** Click the **Leak Routes** tab.
Any already-configured leak routes are displayed.
- Step 3** Click **Actions**, then choose **Create Leak Route**.
The **Create Leak Route** window appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create Leak Routes Dialog Box Fields* table then continue.

Table 10: Create Leak Routes Dialog Box Fields

Properties	Description
Source VRF	<p>To choose a source VRF:</p> <ol style="list-style-type: none"> Click Select a Source VRF. The Select a VRF dialog box appears. From the Select a VRF dialog, click to choose a VRF in the left column to use for the source VRF. Because this procedure is for route maps-based route leaking between a pair of internal VRFs, choose an internal VRF for the source VRF. Click Select to select this source VRF. You return to the Create Leak Route dialog box.

Properties	Description
Destination VRF	<p>To choose a destination VRF:</p> <ol style="list-style-type: none"> Click Select a Destination VRF. The Select a VRF dialog box appears. From the Select a VRF dialog, click to choose a VRF in the left column to use for the destination VRF. Because this procedure is for route maps-based route leaking between a pair of internal VRFs, choose an internal VRF for the destination VRF. Click Select to select this destination VRF. You return to the Create Leak Route dialog box.
Type	<p>Choose the type of leaked route that you want to configure:</p> <ul style="list-style-type: none"> Leak All: Select to configure all routes to leak from the source VRF to the destination VRF. The entry <code>0.0.0.0/0</code> is entered automatically in the subnet IP area by default in this case. Subnet IP: Select to configure a specific subnet IP address as the route to leak from the source VRF to the destination VRF. The Subnet IP box appears. In the Subnet IP box, enter a subnet IP address as the route to leak between VRFs.

Step 5 When finished, click **Save**.

The **Success** window appears.

Step 6 Determine if you want to configure additional inter-VRF route leaking.

- If you want to add another route to leak between a pair of VRFs, click the **Add Another Leak Route** option in the **Success** window.

You are returned to the **Add Leak Route** window. Repeat [Step 4, on page 75](#) through [Step 5, on page 76](#) to configure another route to leak between a pair of VRFs.

- If you want to add a reverse route, where:
 - The destination VRF from the previous configuration now becomes the source VRF, and
 - The source VRF from the previous configuration now becomes the destination VRF

Then click the **Add Reverse Leak Route** option in the **Success** window.

You are returned to the **Add Leak Route** window. Repeat [Step 4, on page 75](#) through [Step 5, on page 76](#) to configure another route, but this time:

- In the **Source VRF** field, select the VRF that you had selected as a destination VRF in the previous configuration.
- In the **Destination VRF** field, select the VRF that you had selected as a source VRF in the previous configuration.

Step 7 When you have finished configuring leak routes, click **Done**.

The **Leak Routes** tab in the main **VRFs** page is displayed again, with the newly configured leak route displayed.

- Step 8** To get more information on a source or destination VRF, or to make changes to a configured leak route, double-click the VRF in the **Leak Routes** tab in the main **VRFs** page. The **Overview** page for that VRF is displayed.
- Step 9** Click the **Application Management** tab at the top of the VRF page, then click the **Leak Routes** tab in the left nav bar. The leak routes associated with this particular VRF are displayed.
- Step 10** Configure additional leak routes associated with this VRF, if necessary.
- To add a leak route from this VRF, click **Actions**, then choose **Add Leak Route from <VRF_name>**.
The **Add Leak Route** window appears. Enter the necessary information as you did previously using the information in [Step 4, on page 75](#). Note that the entry in the **Source VRF** is pre-selected and cannot be changed in this situation.
 - To add a leak route to this VRF, click **Actions**, then choose **Add Leak Route to <VRF_name>**.
The **Add Leak Route** window appears. Enter the necessary information as you did previously using the information in [Step 4, on page 75](#). Note that the entry in the **Destination VRF** is pre-selected and cannot be changed in this situation.

Enabling Connectivity From the Azure Site to External Devices

Follow these procedures to manually enable IPv4 connectivity from the infra VNet CCRs to any external device with IPsec/BGP.

Downloading the External Device Configuration Files

- Step 1** In the Cisco Cloud Network Controller GUI, click on **Dashboard**.
The **Dashboard** view for the Cisco Cloud Network Controller appears.
- Step 2** Navigate to **Infrastructure > External Connectivity**.
The **External Connectivity** window appears.
- Step 3** Click **Actions > Download External Device Configuration Files**.
The **Download External Device Configuration Files** pop-up appears.
- Step 4** Select the external device configuration files to download and click **Download**.
This action downloads a zip file that contains configuration information that you will use to manually configure the external device for IPv4 connectivity to the CCRs.

Enabling Connectivity From the Azure Site to the External Devices

- Step 1** Gather the necessary information that you will need to manually enable IPv4 connectivity from the infra VNet CCRs to any external device without EVPN.
- Step 2** Log into the external device.
- Step 3** Enter the configuration information to connect an external networking device.

If you downloaded the external device configuration files using the instructions in [Downloading the External Device Configuration Files, on page 77](#), locate the configuration information for the first tunnel and enter that configuration information.

Following is an example of what the external device configuration file might look like for the first tunnel:

```
! The following file contains configuration recommendation to connect an external networking device
with the cloud ACI Fabric
! The configurations here are provided for an IOS-XE based device. The user is expected to understand
the configs and make any necessary amends before using them
! on the external device. Cisco does not assume any responsibility for the correctness of the config.

! Tunnel to 128.107.72.122 1.100 [ikev2] for
hctunnIf.acct-[infra]/region-[westus]/context-[overlay-1]-addr-[10.115.9.128/25]/csr-[ct_routerp_westus_0:0]/tunn-34
! USER-DEFINED: please define gig-gateway: GIG-GATEWAY
! USER-DEFINED: please define GigabitEthernet2 if required
! USER-DEFINED: please define tunnel-id: 100 if required
! USER-DEFINED: please define vrf-name: infra:externalvrf1 if required
! USER-DEFINED: please define gig3-public-ip: 13.88.168.176 if 0.0.0.0 ip still not provided by AWS.
! Device:          128.107.72.122
! Tunnel ID:       100
! Tunnel counter:  1
! Tunnel address:  5.16.1.9
! Tunnel Dn:
acct-[infra]/region-[westus]/context-[overlay-1]-addr-[10.115.9.128/25]/csr-[ct_routerp_westus_0:0]/tunn-34
! VRF name:        infra:externalvrf1
! ikev:            ikev2
! Bgp Peer addr:   5.16.1.10
! Bgp Peer asn:    65015
! Gig3 Public ip:  13.88.168.176
! PreShared key:   devicelazure
! ikev profile name: ikev2-100

vrf definition infra:externalvrf1
  rd 1:1

  address-family ipv4
    route-target export 64550:1
    route-target import 64550:1
  exit-address-family
exit

crypto ikev2 proposal ikev2-infra:externalvrf1
  encryption aes-cbc-256 aes-cbc-192 aes-cbc-128
  integrity sha512 sha384 sha256 sha1
  group 24 21 20 19 16 15 14 2
exit

crypto ikev2 policy ikev2-infra:externalvrf1
  proposal ikev2-infra:externalvrf1
exit

crypto ikev2 keyring keyring-ikev2-100
  peer peer-ikev2-keyring
  address 13.88.168.176
  pre-shared-key devicelazure
  exit
exit

crypto ikev2 profile ikev2-100
  match address local interface GigabitEthernet2
  match identity remote address 13.88.168.176 255.255.255.255
  identity local address 128.107.72.122
```

```
    authentication remote pre-share
    authentication local pre-share
    keyring local keyring-ikev2-100
    lifetime 3600
    dpd 10 5 on-demand
exit

crypto ipsec transform-set ikev2-100 esp-gcm 256
    mode tunnel
exit

crypto ipsec profile ikev2-100
    set transform-set ikev2-100
    set pfs group14
    set ikev2-profile ikev2-100
exit

interface Tunnel100
    vrf forwarding infra:externalvrf1
    ip address 5.16.1.10 255.255.255.252
    ip mtu 1400
    ip tcp adjust-mss 1400
    tunnel source GigabitEthernet2
    tunnel mode ipsec ipv4
    tunnel destination 13.88.168.176
    tunnel protection ipsec profile ikev2-100
exit

ip route 13.88.168.176 255.255.255.255 GigabitEthernet2 GIG-GATEWAY

router bgp 65015

address-family ipv4 vrf infra:externalvrf1
    redistribute connected
    maximum-paths eibgp 32

    neighbor 5.16.1.9 remote-as 65008
    neighbor 5.16.1.9 ebgp-multihop 255
    neighbor 5.16.1.9 activate
    neighbor 5.16.1.9 send-community both

    distance bgp 20 200 20
exit-address-family
```

The following figures provide more information on what each set of fields is used for in the external device configuration file:

- The fields shown in the following figure are used to configure these areas:
 - VRF definition
 - IPSec global configurations

```

vrf definition Ext-V1
rd 1:10
!
address-family ipv4
  route-target export 64550:10
  route-target import 64550:10
!
crypto isakmp policy 10
encryption aes
authentication pre-share
group 2
lifetime 28800
!
crypto isakmp keepalive 10 10 periodic
crypto isakmp aggressive-mode disable
!

```

VRF Definition

IPSec Global Configurations

- The fields shown in the following figure are used to configure these areas:

- IPSec and ikev1 per tunnel configurations
- BGP configurations for the VRF neighbor

```

!
crypto keyring Ext-V1-1000-ike
  pre-shared-key address <50.18.55.126>[cAPIC CSR Gig3 Public IP] key <abodef12345>
!
crypto isakmp profile Ext-V1-1000-ike
  keyring Ext-V1-1000-ike
  match identity address <50.18.55.126>[cAPIC CSR1 gig3 Public IP] 255.255.255.255
!
crypto ipsec transform-set Ext-V1-1000-ike esp-aes esp-sha-hmac
mode tunnel
!
crypto ipsec profile Ext-V1-1000-ike
set security-association lifetime kilobytes disable
set security-association replay window-size 512
set transform-set Ext-V1-1000-ike
set pfs group14
!
interface Tunnel1000
vrf forwarding Ext-V1
ip address 50.50.0.2[cAPIC CSR BGP Peer Addr] 255.255.255.252
ip mtu 1400
ip tcp adjust-mss 1400
tunnel source GigabitEthernet2
tunnel mode ipsec ipv4
tunnel destination <50.18.55.126>[cAPIC CSR1 gig3 Public IP]
tunnel protection ipsec profile Ext-V1-1000-ike
!
router bgp 64550
!
address-family ipv4 vrf Ext-V1
 redistribute connected
 neighbor <50.50.0.1>[cAPIC CSR1 Tunnel Inner IP Addr] remote-as 1234
 neighbor 50.50.0.1 ebgp-multihop 255
 neighbor 50.50.0.1 activate
 neighbor 50.50.0.1 send-community both
 neighbor <50.50.0.5>[cAPIC CSR1 Tunnel Inner IP Addr] remote-as 1234
 neighbor 50.50.0.5 ebgp-multihop 255
 neighbor 50.50.0.5 activate
 neighbor 50.50.0.5 send-community both
 distance bgp 20 200 20
!
ip route 50.18.55.126[cAPIC CSR1 gig3 Public IP] 255.255.255.255 GigabitEthernet2 10.10.0.103
!

```

IPSec and Ikev1 Per Tunnel Configurations

BGP Configurations for VRF Neighbor

- The fields shown in the following figure are used to configure these areas:

- Ikev2 global configurations
- IPSec and ikev2 per tunnel configurations

```

crypto ikev2 proposal ikev2-1
  encryption aes-cbc-256 aes-cbc-192 aes-cbc-128
  integrity sha512 sha384 sha256 sha1
  group 24 21 20 19 16 15 14 2
  !
crypto ikev2 policy ikev2-1
  proposal ikev2-1
  !
crypto ikev2 keyring ikev2-2000
  peer peer-ikev2-keyring
  address 35.81.94.248 [cAPIC CSR1 gig3 Public IP]
  pre-shared-key abcdefg12345
  !
crypto ikev2 profile ikev2-2000
  match address local interface GigabitEthernet3
  match identity remote address 35.81.94.248[cAPIC CSR1 gig3 Public IP] 255.255.255.255
  identity local address 52.53.49.193 [Local Device tunnel source interface Public IP (Gig3 public IP)]
  authentication remote pre-share
  authentication local pre-share
  keyring local keyring-ikev2-2000
  lifetime 3600
  dpd 10 5 on-demand
  !
crypto ipsec transform-set ikev2-2000 esp-gcm 256
  mode tunnel
  !
crypto ipsec profile ikev2-2000
  set transform-set ikev2-2000
  set pfs group14
  set ikev2-profile ikev2-2000
  !
interface Tunnel2000
  vrf forwarding Ext-V1
  ip address 50.50.0.14 [cAPIC CSR1 BGP Peer Addr] 255.255.255.252
  ip mtu 1400
  ip tcp adjust-mss 1400
  tunnel source GigabitEthernet3
  tunnel mode ipsec ipv4
  tunnel destination 35.81.94.248[cAPIC CSR1 gig3 Public IP]
  tunnel protection ipsec profile ikev2-2000

```

Ikev2 Global Configurations

IPSec and Ikev2
Per Tunnel Configurations

Step 4 Repeat the previous step to configure additional tunnels.

Creating an EPG Using the Cisco Cloud Network Controller GUI

Use the procedures in this section to create an application EPG, an external EPG, or a service EPG. The available configuration options vary, depending on which type of EPG you are creating.

Creating an Application EPG Using the Cisco Cloud Network Controller GUI

This section explains how to create an application EPG using the Cisco Cloud Network Controller GUI. Each service needs at least one consumer EPG and one provider EPG.



Note You can create cloud EPGs and cloud external EPGs in the infra tenant, where all the cloud EPGs and cloud external EPGs will be associated with the secondary VRF in the infra tenant. A cloud EPG in the secondary VRF can communicate with other cloud EPGs and cloud external EPGs in the secondary VRF, and can also communicate with cloud EPGs in other user tenant VRFs. We recommend that you do not use existing "cloud-infra" application profiles, and instead create a new application profile in the infra tenant and associate that new application profile to the cloud EPGs and cloud external EPGs in the secondary VRF.

Before you begin

Create an application profile and a VRF.

Step 1 Click the **Intent** icon.

The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create EPG**.

The **Create EPG** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create EPG Dialog Box Fields* table then continue.

Table 11: Create EPG Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the EPG.
Tenant	<p>To choose a tenant:</p> <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column. You can select the infra tenant and can create cloud EPGs and cloud external EPGs in the infra tenant, as described earlier in this section. Click Select. You return to the Create EPG dialog box.
Application Profile	<p>To choose an application profile:</p> <ol style="list-style-type: none"> Click Select Application Profile. The Select Application Profile dialog box appears. From the Select Application Profile dialog, click to choose an application profile in the left column. Note If you are creating an EPG in the infra tenant, we recommend that you do not choose the <code>cloud-infra</code> application profile because that application profile is used by EPGs in the overlay-1 VRF. Select a different application profile or click Create Application Profile to create a new one. Click Select. You return to the Create EPG dialog box.
Description	Enter a description of the EPG.
Settings	
Type	Because this will be an application EPG, choose Application as the EPG type.
VRF	<p>To choose a VRF:</p> <ol style="list-style-type: none"> Click Select VRF. The Select VRF dialog box appears. From the Select VRF dialog, click to choose a VRF in the left column. If you are creating an EPG in the infra tenant, select the secondary VRF in this step. A cloud EPG in the secondary VRF can communicate with other cloud EPGs and cloud external EPGs in the secondary VRF, and can also communicate with cloud EPGs in other user tenant VRFs. Click Select. You return to the Create EPG dialog box.

Properties	Description
Endpoint Selectors	

Properties	Description
	<p>Note See Configuring Virtual Machines in Azure, on page 117 for instructions on configuring virtual machines in Azure as part of the endpoint selector configuration process.</p> <p>To add an endpoint selector:</p> <ol style="list-style-type: none"> a. Click Add Endpoint Selector to open the Add Endpoint Selector dialog. b. In the Add Endpoint Selector dialog, enter a name in the Name field. c. Click Selector Expression. The Key, Operator, and Value fields are enabled. d. Click the Key drop-down list to choose a key. The options are: <ul style="list-style-type: none"> • Choose IP if you want to use an IP address or subnet for the endpoint selector. <p>Note IPv6 is not supported for Cisco Cloud Network Controller in Azure. You must use a valid IPv4 address for this field.</p> • Choose Region if you want to use the Azure region for the endpoint selector. • Choose Custom if you want to create a custom key for the endpoint selector. <p>Note When choosing the Custom option, the drop-down list becomes a text box. You need to enter a name for the key in the spaces after custom: (for example, custom: Location).</p> e. Click the Operator drop-down list to choose an operator. The options are: <ul style="list-style-type: none"> • equals: Used when you have a single value in the Value field. • not equals: Used when you have a single value in the Value field. • in: Used when you have multiple comma-separated values in the Value field. • not in: Used when you have multiple comma-separated values in the Value field. • has key: Used if the expression contains only a key. • does not have key: Used if the expression contains only a key. f. Enter a value in the Value field then click the check mark to validate the entries. The value you enter depends on the choices you made for the Key and Operator fields. For example, if the Key field is set to IP and the Operator field is set to equals, the Value field must be an IP address or subnet. However, if the Operator field is set to has key, the Value field is disabled. g. When finished, click the check mark to validate the selector expression. h. Determine if you want to create additional endpoint selector expressions to the endpoint selector. If you create more than one expression under a single endpoint selector, a logical AND exists between those expressions. <p>For example, assume you created two sets of expressions under a single endpoint selector:</p> <ul style="list-style-type: none"> • Endpoint selector 1, expression 1: <ul style="list-style-type: none"> • Key: Region

Properties	Description
	<ul style="list-style-type: none"> • Operator: equals • Value: westus <ul style="list-style-type: none"> • Endpoint selector 1, expression 2: <ul style="list-style-type: none"> • Key: IP • Operator: equals • Value: 192.0.2.1/24 <p>In this case, if <i>both</i> of these expressions are true (if the region is westus AND if the IP address belongs to subnet 192.0.2.1/24), then that endpoint is assigned to the Cloud EPG.</p> <p>i. Click the check mark after every additional expression that you want to create under this endpoint selector then click Add when finished.</p> <p>If you create more than one endpoint selector under an EPG, a logical OR exists between those endpoint selectors. For example, assume you had created endpoint selector 1 as described in the previous step, and then you created a second endpoint selector as described below:</p> <ul style="list-style-type: none"> • Endpoint selector 2, expression 1: <ul style="list-style-type: none"> • Key: Region • Operator: in • Value: eastus, centralus <p>In this case:</p> <ul style="list-style-type: none"> • If the region is westus AND the IP address belongs to the 192.0.2.1/24 subnet (endpoint selector 1 expressions) <p>OR</p> <ul style="list-style-type: none"> • If the region is either eastus or centralus (endpoint selector 2 expression) <p>Then that end point is assigned to the Cloud EPG.</p>

Step 5 Click **Save** when finished.

Creating an External EPG Using the Cisco Cloud Network Controller GUI

This section explains how to create an external EPG using the Cisco Cloud Network Controller GUI. Each service needs at least one consumer EPG and one provider EPG.



Note You can create cloud EPGs and cloud external EPGs in the infra tenant, where all the cloud EPGs and cloud external EPGs will be associated with the secondary VRF in the infra tenant. A cloud EPG in the secondary VRF can communicate with other cloud EPGs and cloud external EPGs in the secondary VRF, and can also communicate with cloud EPGs in other user tenant VRFs. We recommend that you do not use existing "cloud-infra" application profiles, and instead create a new application profile in the infra tenant and associate that new application profile to the cloud EPGs and cloud external EPGs in the secondary VRF.

Before you begin

Create an application profile and a VRF.

-
- Step 1** Click the **Intent** icon.
The **Intent** menu appears.
- Step 2** Click the drop-down arrow below the **Intent** search box and choose **Application Management**.
A list of **Application Management** options appear in the **Intent** menu.
- Step 3** From the **Application Management** list in the **Intent** menu, click **Create EPG**.
The **Create EPG** dialog box appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create EPG Dialog Box Fields* table then continue.

Table 12: Create EPG Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the EPG.
Tenant	<p>To choose a tenant:</p> <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column. You can select the infra tenant and can create cloud EPGs and cloud external EPGs in the infra tenant, as described earlier in this section. Click Select. You return to the Create EPG dialog box.

Properties	Description
Application Profile	<p>To choose an application profile:</p> <ol style="list-style-type: none"> Click Select Application Profile. The Select Application Profile dialog box appears. From the Select Application Profile dialog, click to choose an application profile in the left column. <ul style="list-style-type: none"> Note If you are creating an EPG in the infra tenant, we recommend that you do not choose the <code>cloud-infra</code> application profile because that application profile is used by EPGs in the overlay-1 VRF. Select a different application profile or click Create Application Profile to create a new one. Click Select. You return to the Create EPG dialog box.
Description	Enter a description of the EPG.
Settings	
Type	Because this will be an external EPG, choose External as the EPG type.
VRF	<p>To choose a VRF:</p> <ol style="list-style-type: none"> Click Select VRF. The Select VRF dialog box appears. From the Select VRF dialog, click to choose a VRF in the left column. <p>If you are creating an EPG in the infra tenant, select the secondary VRF in this step. A cloud EPG in the secondary VRF can communicate with other cloud EPGs and cloud external EPGs in the secondary VRF, and can also communicate with cloud EPGs in other user tenant VRFs.</p> Click Select. You return to the Create EPG dialog box.
Route Reachability	<p>Choose the type of route reachability for the external EPG. The options are:</p> <ul style="list-style-type: none"> • Internet • External-Site

Properties	Description
Endpoint Selectors	<p>Note See Configuring Virtual Machines in Azure, on page 117 for instructions on configuring virtual machines in Azure as part of the endpoint selector configuration process.</p> <p>To add an endpoint selector:</p> <ol style="list-style-type: none"> Click Add Endpoint Selector to add an endpoint selector. Enter a name in the Name field. Enter a subnet in the Subnet. <p>Note IPv6 is not supported for Cisco Cloud Network Controller in Azure. You must use a valid IPv4 address for this field.</p> <ol style="list-style-type: none"> When finished, click the check mark to validate the endpoint selector. Determine if you want to create additional endpoint selectors. <p>If you create more than one endpoint selector under an EPG, a logical OR exists between those endpoint selectors. For example, assume you created two endpoint selectors:</p> <ul style="list-style-type: none"> • Endpoint selector 1: <ul style="list-style-type: none"> • Name: EP_Sel_1 • Subnet: 192.1.1.1/24 • Endpoint selector 2: <ul style="list-style-type: none"> • Name: EP_Sel_2 • Subnet: 192.2.2.2/24 <p>In this case:</p> <ul style="list-style-type: none"> • If the IP address belongs to the 192.1.1.1/24 subnet (endpoint selector 1) <p style="text-align: center;">OR</p> <ul style="list-style-type: none"> • If the IP address belongs to the 192.2.2.2/24 subnet (endpoint selector 2) <p>Then that end point is assigned to the Cloud EPG.</p>

Step 5 Click **Save** when finished.

Creating a Service EPG

Use the procedures in the following sections to create a service EPG.

Tasks To Perform Prior to Configuring Service EPGs

Before you can configure a service EPG, there are certain tasks that you might have to perform beforehand. If you are using subnets or private link labels with your service EPG, you must first configure the subnets and/or private link label outside of the service EPG.

Step 1

Create a VRF, if necessary.

- a) Click the **Intent** icon. The **Intent** menu appears.
- b) Click the drop-down arrow below the **Intent** search box and choose **Application Management**.
A list of **Application Management** options appear in the **Intent** menu.
- c) From the **Application Management** list in the **Intent** menu, click **Create VRF**. The **Create VRF** dialog box appears.
- d) Make the following selections:
 - **Name**: Enter the name for the VRF.
 - **Tenant**: Select a tenant.
- e) Click **Save**.

Step 2

Configure a cloud context profile.

- a) Click the **Intent** icon. The **Intent** menu appears.
- b) Click the drop-down arrow below the **Intent** search box and choose **Application Management**.
A list of **Application Management** options appear in the **Intent** menu.
- c) From the **Application Management** list in the **Intent** menu, click **Create Cloud Context Profile**. The **Create Cloud Context Profile** dialog box appears.

Step 3

Enter the appropriate values in each field as listed in the following *Cloud Context Profile Dialog Box Fields* table then continue.

Table 13: Create Cloud Context Profile Dialog Box Fields

Properties	Description
Name	Enter the name of the cloud context profile.
Tenant	To choose a tenant: <ol style="list-style-type: none"> a. Click Select Tenant. The Select Tenant dialog box appears. b. From the Select Tenant dialog, click to choose a tenant in the left column then click Select. You return to the Create Cloud Context Profile dialog box.
Description	Enter a description of the cloud context profile.
Settings	
Region	To choose a region: <ol style="list-style-type: none"> a. Click Select Region. The Select Region dialog box appears. b. From the Select Region dialog, click to choose a region in the left column then click Select. You return to the Create Cloud Context Profile dialog box.

Properties	Description
VRF	To choose a VRF: <ol style="list-style-type: none"><li data-bbox="381 338 1015 369">a. Click Select VRF. The Select VRF dialog box appears.<li data-bbox="381 390 1458 453">b. From the Select VRF dialog box, click to choose a VRF in the left column then click Select. You return to the Create Cloud Context Profile dialog box.

Properties	Description
Add CIDR	<p>Note You cannot add, delete, or edit a CIDR when VNet peering is enabled. You must disable VNet peering before adding, deleting or editing a CIDR. To disable VNet peering:</p> <ul style="list-style-type: none"> • For the infra tenant, disable the Hub Network Peering option in the cloud context profile • For a user (non-infra) tenant, disable the VNet Peering option in the cloud context profile <p>Enable VNet peering again after you have made the changes to the CIDR configuration.</p> <p>The following features are supported, depending on the release:</p> <ul style="list-style-type: none"> • You can add additional secondary CIDRs and subnets for infra VNets (cloudCtxProfiles created by the cloud template). You cannot add primary CIDRs or modify the existing CIDRs created by the cloud template. After subnets are created under the user-created CIDRs, the subnets will be implicitly mapped to the secondary VRF. • You can add also additional secondary CIDRs and subnets for VNets other than the infra VNet. <p>See Support for Multiple VRFs Under Single VNet, on page 34 for more information.</p> <p>To add a CIDR:</p> <ol style="list-style-type: none"> a. Click Add CIDR. The Add CIDR dialog box appears. b. Enter the address in the CIDR Block Range field. c. Click to check (enabled) or uncheck (disabled) the Primary check box. If you are adding additional secondary CIDRs and subnets for VNets, leave the Primary box unchecked. d. Click Add Subnet and enter the following information: <ul style="list-style-type: none"> • In the Address field, enter the subnet address. • In the Name field, enter the name for this subnet. • In the Private Link Label field, choose Create New and enter a unique name for the private link label to associate with this subnet. e. In the VRF field, make a selection, if necessary. <ul style="list-style-type: none"> • If you checked the box next to the Primary field, this CIDR is automatically associated with the primary VRF. • If you did not check the box next to the Primary field, you can associate this CIDR with a secondary VRF. Click the X next to the VRF, then click on Select VRF to select the secondary VRF to associate with this CIDR. f. When finished, click Add.
VNet Gateway Router	Click to check (enable) or uncheck (disable) in the VNet Gateway Router check box.

Properties	Description
VNet Peering	Click to check (enable) or uncheck (disable) the Azure VNet peering feature. For more information on the VNet peering feature, see the <i>Configuring VNet Peering for Cisco Cloud Network Controller for Azure</i> document in the Cisco Cloud Network Controller documentation page .

Step 4 Click **Save**.

Creating a Service EPG Using the Cisco Cloud Network Controller GUI

This section explains how to create a service EPG using the Cisco Cloud Network Controller GUI. Each service needs at least one consumer EPG and one provider EPG.

Before you begin

- Review the information in [Cloud Service Endpoint Groups, on page 38](#).
- Verify that you have the **NSG-per-subnet** configuration enabled.
You must have the **NSG-per-subnet** configuration enabled if you are configuring cloud service EPGs. See [Security Groups, on page 45](#) for more information.
- Create an application profile and a VRF.

Step 1 Click the **Intent** icon.

The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create EPG**.

The **Create EPG** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create EPG Dialog Box Fields* table then continue.

Table 14: Create EPG Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the EPG.
Tenant	To choose a tenant: <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column. Click Select. You return to the Create EPG dialog box.

Properties	Description
Application Profile	<p>To choose an application profile:</p> <ol style="list-style-type: none"> Click Select Application Profile. The Select Application Profile dialog box appears. From the Select Application Profile dialog, click to choose an application profile in the left column. <ul style="list-style-type: none"> Note If you are creating a service EPG in the infra tenant, we recommend that you do not choose the <code>cloud-infra</code> application profile because that application profile is used by EPGs in the overlay-1 VRF. Select a different application profile or click Create Application Profile to create a new one. Click Select. You return to the Create EPG dialog box.
Description	Enter a description of the EPG.
Settings	
Type	Because this will be a service EPG, choose Service as the EPG type.
VRF	<p>To choose a VRF:</p> <ol style="list-style-type: none"> Click Select VRF. The Select VRF dialog box appears. From the Select VRF dialog, click to choose a VRF in the left column. Click Select. You return to the Create EPG dialog box.
Deployment Type	<p>Choose the EPG deployment type.</p> <p>Services are differentiated based on their deployment mode:</p> <ul style="list-style-type: none"> • Cloud Native: A Cloud Native service deployed in the provider network • Cloud Native Managed: A Cloud Native service deployed in your network • Third-Party: A third-party service from the market place

Properties	Description
Access Type	<p>Choose the EPG deployment access type. The access type indicates how the other services or VMs will connect to the service.</p> <p>The choices vary, depending on the selection you made in the Deployment Type field:</p> <ul style="list-style-type: none"> • Cloud Native deployment type: <ul style="list-style-type: none"> • Public: Access the public IP of the service. • Private: Use private links and private endpoints to access the service. • Cloud Native Managed deployment type: <ul style="list-style-type: none"> • Private: Choose this type if the service deployed in the managed subnet has only private IP addresses. • Public and Private: Use public and private endpoints to access the service. This is used for services that also expose public IP addresses when deployed in Cisco Cloud Network Controller -managed subnets. • Third-Party deployment type: Private is the only option available to you as an access type. This means that you will use only private endpoints to the service, if the service offers it.

Properties	Description
Service Type	<p>Choose the Azure service type.</p> <p>Certain service types are only supported with certain specific deployment types. See Cloud Service Endpoint Groups, on page 38 for more information on the service types that are supported with specific deployment types.</p> <p>The options are:</p> <ul style="list-style-type: none"> • Azure Storage Blob (see Azure Storage, on page 41) • Azure SQL • Azure Cosmos DB • Azure Databricks (see Azure Databricks Services, on page 42) • Azure Storage (see Azure Storage, on page 41) • Azure Storage File (see Azure Storage, on page 41) • Azure Storage Queue (see Azure Storage, on page 41) • Azure Storage Table (see Azure Storage, on page 41) • Azure Kubernetes Services (AKS) (see Azure Kubernetes Services, on page 42) • Azure Active Directory Domain Services (see Azure Active Directory Domain Services, on page 42) • Azure Container Registry • Azure ApiManagement Services (see Azure ApiManagement Services, on page 41) • Azure Key Vault • Redis Cache (see Azure Redis Cache, on page 42) • Custom Service (used if you choose Third-Party as the Deployment Type)

Step 5 Enter the necessary information in the **Endpoint Selector** area, depending on the selection you made in the **Deployment Type** field:

- If you chose **Cloud Native** as the deployment type, go to [Configuring Cloud Native as the Deployment Type, on page 95](#).
- If you chose **Cloud Native Managed** as the deployment type, go to [Configuring Cloud Native Managed as the Deployment Type, on page 98](#).
- If you chose **Third-Party** as the deployment type, go to [Configuring Third-Party as the Deployment Type, on page 100](#).

Configuring Cloud Native as the Deployment Type

Use the procedures in this section to configure **Cloud Native** as the deployment type for the service EPG.

Before you begin

Review the information provided in [Cloud Native, on page 43](#) to understand tasks that you might have to perform prior to using these instructions.

-
- Step 1** Verify that you have completed the steps in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) before beginning these procedures.
- These procedures are a continuation of the procedures provided in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#), where you would have set a service type, such as `Azure SQL`, before configuring the deployment type in these procedures.
- Step 2** If you selected **Private** as the access type, the **Select Private Link Label** option becomes available.
- A private link label is used to associate the subnets to the service EPGs.
- Step 3** Click **Select Private Link Label**.
- The **Select Private Link Label** window appears.
- Step 4** Search for the appropriate private link label.
- Search for the private link label that you created using the procedures provided in [Tasks To Perform Prior to Configuring Service EPGs, on page 89](#).
- Step 5** In the **Select Private Link Label** window, select the appropriate private link label.
- You are returned to the **Create EPG** window.
- Next, add an endpoint selector in the **Endpoint Selectors** field.
- Step 6** Click **Add Endpoint Selector**.
- The **Add Endpoint Selector** window appears.
- Step 7** In the **Add Endpoint Selector** window, enter a name in the **Name** field.
- Step 8** Click the **Key** drop-down list to choose a key.
- The options are:
- Choose **Custom** if you want to create a custom endpoint selector.
 - Choose **Region** if you want to use the Azure region for the endpoint selector.
 - Choose **Name** if you want to use the service resource's name for the endpoint selector.
- For example, to select an SQL server with the name `ProdSqlServer`, you would choose **Name** in the **Key** field, and you would enter `ProdSqlServer` in the **Value** field later in these procedures.
- Choose **Resource ID** if you want to use the cloud provider's ID for the endpoint selector.
- For example, to select an SQL server using the cloud provider's resource ID, you would choose **Resource ID** in the **Key** field, and you would enter the value of the selector, such as `/subscriptions/{subscription-id}/resourceGroups/{resourceGroupName}/providers/Microsoft.Sql/servers/ProdSqlServer`, in the **Value** field later in these procedures.
- Step 9** Click the **Operator** drop-down list to choose an operator.
- The options are:

- **equals**: Used when you have a single value in the Value field.
- **not equals**: Used when you have a single value in the Value field.
- **in**: Used when you have multiple comma-separated values in the Value field.
- **not in**: Used when you have multiple comma-separated values in the Value field.
- **has key**: Used if the expression contains only a key.
- **does not have key**: Used if the expression contains only a key.

Step 10 Enter a value in the **Value** field then click the check mark to validate the entries.

The value you enter depends on the choices you made for the **Key** and **Operator** fields.

For example, if the **Key** field is set to **IP** and the **Operator** field is set to **equals**, the **Value** field must be an IP address or subnet. However, if the **Operator** field is set to **has key**, the **Value** field is disabled.

Step 11 When finished, click the check mark to validate the selector expression.

Step 12 Determine if you want to create additional endpoint selector expressions for the endpoint selector.

If you create more than one expression under a single endpoint selector, a logical AND exists between those expressions.

For example, assume you created two sets of expressions under a single endpoint selector:

- Endpoint selector 1, expression 1:
 - **Key**: Region
 - **Operator**: equals
 - **Value**: westus
- Endpoint selector 1, expression 2:
 - **Key**: Name
 - **Operator**: equals
 - **Value**: ProdSqlServer

In this case, if *both* of these expressions are true (if the region is `westus` AND if the name attached to the resource is `ProdSqlServer`), then that endpoint is assigned to the service EPG.

Step 13 Click the check mark after every additional expression that you want to create under this endpoint selector, then click **Add** when finished.

You are returned to the **Create EPG** screen, with the new endpoint selector and the configured expressions shown.

Step 14 If you want to create additional endpoint selectors, click **Add Endpoint Selector** again and repeat these steps to create additional endpoint selectors.

If you create more than one endpoint selector under an EPG, a logical OR exists between those endpoint selectors. For example, assume you had created endpoint selector 1 as described in the previous step, and then you created a second endpoint selector as described below:

- Endpoint selector 2, expression 1:
 - **Key**: Region

- **Operator:** in
- **Value:** eastus, centralus

In this case:

- If the region is `westus` AND the name attached to the resource is `ProdSqlServer` (endpoint selector 1 expressions)
OR
- If the region is either `eastus` or `centralus` (endpoint selector 2 expression)

Then that end point is assigned to the service EPG.

Step 15 Click **Save** when finished.

Configuring Cloud Native Managed as the Deployment Type

Use the procedures in this section to configure **Cloud Native Managed** as the deployment type for the service EPG.

Before you begin

Review the information provided in [Cloud Native Managed, on page 44](#) to understand tasks that you might have to perform prior to using these instructions.

Step 1 Verify that you have completed the steps in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) before beginning these procedures.

These procedures are a continuation of the procedures provided in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#), where you would have set a service type, such as `Azure ApiManagement Services`, before configuring the deployment type in these procedures.

Step 2 Click **Add Endpoint Selector**.

The **Add Endpoint Selector** window appears.

Step 3 In the **Add Endpoint Selector** window, enter a name in the **Name** field.

Step 4 Click the **Key** drop-down list to choose a key.

At this time, **IP** is the only option available as a key for this access type.

Note IPv6 is not supported for Cisco Cloud Network Controller in Azure. You must use a valid IPv4 address for this field.

Step 5 Click the **Operator** drop-down list to choose an operator.

The options are:

- **equals:** Used when you have a single value in the Value field.
- **not equals:** Used when you have a single value in the Value field.
- **in:** Used when you have multiple comma-separated values in the Value field.

- **not in**: Used when you have multiple comma-separated values in the Value field.
- **has key**: Used if the expression contains only a key.
- **does not have key**: Used if the expression contains only a key.

Step 6 Enter the appropriate IP address or a subnet in the **Value** field then click the check mark to validate the entries.

Enter the IP address or subnet that you created using the procedures provided in [Tasks To Perform Prior to Configuring Service EPGs](#), on page 89.

Step 7 When finished, click the check mark to validate the selector expression.

Step 8 Determine if you want to create additional endpoint selector expressions to the endpoint selector.

If you create more than one expression under a single endpoint selector, a logical AND exists between those expressions.

For example, assume you created two sets of expressions under a single endpoint selector:

- Endpoint selector 1, expression 1:
 - **Key**: IP
 - **Operator**: equals
 - **Value**: 192.1.1.1/24
- Endpoint selector 1, expression 2:
 - **Key**: IP
 - **Operator**: not equals
 - **Value**: 192.1.1.2

In this case, if *both* of these expressions are true (if the IP address belongs to subnet 192.1.1.1/24 AND the IP address is not 192.1.1.2), then that endpoint is assigned to the service EPG.

Step 9 Click the check mark after every additional expression that you want to create under this endpoint selector, then click **Add** when finished.

You are returned to the **Create EPG** screen, with the new endpoint selector and the configured expressions shown.

Step 10 If you want to create additional endpoint selectors, click **Add Endpoint Selector** again and repeat these steps to create additional endpoint selectors.

If you create more than one endpoint selector under an EPG, a logical OR exists between those endpoint selectors. For example, assume you had created endpoint selector 1 as described in the previous step, and then you created a second endpoint selector as described below:

- Endpoint selector 2, expression 1:
 - **Key**: IP
 - **Operator**: equals
 - **Value**: 192.2.2.2/24

In this case:

- If the IP address belongs to subnet 192.1.1.1/24 AND the IP address is not 192.1.1.2 (endpoint selector 1 expressions)
OR
- If the IP address belongs to subnet 192.2.2.2/24

Then that end point is assigned to the service EPG.

Step 11 Click **Save** when finished.

Configuring Third-Party as the Deployment Type

Use the procedures in this section to configure **Third-Party** as the deployment type for the service EPG.



Note You must choose **Custom** as the **Service Type** if you choose **Third-Party** as the **Deployment Type**.

Step 1 Verify that you have completed the steps in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) before beginning these procedures.

These procedures are a continuation of the procedures provided in [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#), where you would have set the service type as `Custom Service` before configuring the deployment type in these procedures.

Step 2 Make the necessary selections for the access type for the **Third-Party** deployment type.

Private is the only option available to you as an access type. This means that you will use only private endpoints to the service, if the service offers it.

The **Select Private Link Label** option becomes available with this access type. A private link label is used to associate the subnets to the service EPGs.

Step 3 Search for the appropriate private link label.

Search for the private link label that you created using the procedures provided in [Tasks To Perform Prior to Configuring Service EPGs, on page 89](#).

Step 4 In the **Select Private Link Label** window, select the appropriate private link label.

You are returned to the **Create EPG** window.

Next, add an endpoint selector in the **Endpoint Selectors** field.

Step 5 Click **Add Endpoint Selector**.

The **Add Endpoint Selector** window appears.

Step 6 In the **Add Endpoint Selector** window, enter a name in the **Name** field.

Step 7 Click the **Key** drop-down list to choose a key.

At this time, **URL** is the only option available as a key for this access type, where you will use the alias or fully qualified domain name (FQDN) that identifies the service for the endpoint selector.

Step 8 Click the **Operator** drop-down list to choose an operator.

The options are:

- **equals**: Used when you have a single value in the Value field.
- **not equals**: Used when you have a single value in the Value field.
- **in**: Used when you have multiple comma-separated values in the Value field.
- **not in**: Used when you have multiple comma-separated values in the Value field.
- **has key**: Used if the expression contains only a key.
- **does not have key**: Used if the expression contains only a key.

Step 9 Enter a valid URL in the **Value** field then click the check mark to validate the entries.

Step 10 When finished, click the check mark to validate the selector expression, then click **Add**.

You are returned to the **Create EPG** screen, with the new endpoint selector and the configured expression shown.

Step 11 If you want to create additional endpoint selectors, click **Add Endpoint Selector** again and repeat these steps to create additional endpoint selectors.

If you create more than one endpoint selector under an EPG, a logical OR exists between those endpoint selectors.

For example, assume you created two endpoint selectors as described below:

- Endpoint selector 1:
 - **Key**: URL
 - **Operator**: equals
 - **Value**: `www.acme1.com`
- Endpoint selector 2:
 - **Key**: URL
 - **Operator**: equals
 - **Value**: `www.acme2.com`

In this case:

- If the URL is `www.acme1.com`
- OR
- If the URL is `www.acme2.com`

Then that end point is assigned to the service EPG.

Step 12 Click **Save** when finished.

Creating a Filter Using the Cisco Cloud Network Controller GUI

This section explains how to create a filter using the Cisco Cloud Network Controller GUI.

- Step 1** Click the **Intent** icon. The **Intent** menu appears.
- Step 2** Click the drop-down arrow below the **Intent** search box and choose **Application Management**.
A list of **Application Management** options appear in the **Intent** menu.
- Step 3** From the **Application Management** list in the **Intent** menu, click **Create Filter**. The **Create Filter** dialog box appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create Filter Dialog Box Fields* table then continue.

Table 15: Create Filter Dialog Box Fields

Properties	Description
Name	Enter a name for the filter in the Name field.
Tenant	To choose a tenant: <ol style="list-style-type: none"> a. Click Select Tenant. The Select Tenant dialog box appears. b. From the Select Tenant dialog, click to choose a tenant in the left column then click Select. You return to the Create Filter dialog box.
Description	Enter a description of the filter.

Properties	Description
Add Filter	<p>To add a filter:</p> <ol style="list-style-type: none"> a. Click Add Filter Entry. The Add Filter Entry dialog box appears. b. Enter a name for the filter entry in the Name field. c. Click the Ethernet Type drop-down list to choose an ethernet type. The options are: <ul style="list-style-type: none"> • IP • Unspecified <p>Note When Unspecified is chosen, any traffic type is allowed, including IP, and the remaining fields are disabled.</p> d. Click the IP Protocol drop-down menu to choose a protocol. The options are: <ul style="list-style-type: none"> • tcp • udp • Unspecified <p>Note The remaining fields are enabled only when tcp or udp is chosen.</p> e. Enter the appropriate port range information in the Destination Port fields. f. When finished entering filter entry information, click Add. You return to the Create Filter dialog box where you can repeat the steps to add another filter entry.

Step 5 When finished, click **Save**.

Creating a Contract Using the Cisco Cloud Network Controller GUI

This section explains how to create a contract using the Cisco Cloud Network Controller GUI.

Before you begin

Create filters.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

- Step 3** From the **Application Management** list in the **Intent** menu, click **Create Contract**. The **Create Contract** dialog box appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create Contract Dialog Box Fields* table then continue.

Table 16: Create Contract Dialog Box Fields

Properties	Description
Name	Enter the name of the contract.
Tenant	<p>To choose a tenant:</p> <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column. <p>Note Beginning in Release 5.0(2), you can create contracts in the infra tenant. You can also export contracts from and import contracts to the infra tenant for shared services use cases.</p> Click Select. You return to the Create Contract dialog box.
Description	Enter a description of the contract.
Settings	
Scope	<p>The scope limits the contract to any endpoint groups within the same application profile, within the same VRF instance, throughout the fabric (globally), or within the same tenant.</p> <p>Note Shared services enables communication between EPGs in different tenants and between EPGs in different VRFs.</p> <p>To enable EPGs in one tenant to communicate with EPGs in another tenant, choose Global scope.</p> <p>To enable an EPG in one VRF to communicate with another EPG in a different VRF, choose Global or Tenant scope.</p> <p>For more information about shared services, see Shared Services, on page 57.</p> <p>Click the drop-down arrow to choose from the following scope options:</p> <ul style="list-style-type: none"> • Application Profile • VRF • Global • Tenant

Properties	Description
Add Filter	To choose a filter: <ol style="list-style-type: none"> Click Add Filter. The filter row appears with a Select Filter option. Click Select Filter. The Select Filter dialog box appears. From the Select Filter dialog, click to choose a filter in the left column then click Select. You return to the Create Contract dialog box.

Step 5 Click **Save** when finished.

Creating an Inter-Tenant Contract Using the Cisco Cloud Network Controller GUI

This section explains how to create an inter-tenant contract using the Cisco Cloud Network Controller GUI. See [Shared Services, on page 57](#) for more information on situations where you might want to create an inter-tenant contract.

Before you begin

Create filters.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create Contract**. The **Create Contract** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Contract Dialog Box Fields* table then continue.

Table 17: Create Contract Dialog Box Fields

Properties	Description
Name	Enter the name of the contract.
Tenant	To choose a tenant: <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column. <p>Note Beginning in Release 5.0(2), you can create contracts in the infra tenant. You can also export contracts from and import contracts to the infra tenant for shared services use cases.</p> Click Select. You return to the Create Contract dialog box.

Properties	Description
Description	Enter a description of the contract.
Settings	
Scope	<p>The scope limits the contract to any endpoint groups within the same application profile, within the same VRF instance, throughout the fabric (globally), or within the same tenant.</p> <p>For inter-tenant communication, you will first create a contract with the Global scope in one of the tenants (for example, tenant1). This tenant's EPG will always be the provider of this contract.</p> <p>This contract will then be exported to the other tenant (for example, tenant2). For the other tenant that imports this contract, its EPG will be the consumer of the imported contract. If you want tenant2's EPG to be the provider and tenant1's EPG to be the consumer, then create a contract in tenant2 and then export it to tenant1.</p>
Add Filter	<p>To choose a filter:</p> <ol style="list-style-type: none"> Click Add Filter. The filter row appears with a Select Filter option. Click Select Filter. The Select Filter dialog box appears. From the Select Filter dialog, click to choose a filter in the left column then click Select. You return to the Create Contract dialog box.

Step 5 Click **Save** when finished.

Step 6 Export the contract that you just created to another tenant.

For example, assume the following:

- The contract that you created in the procedure above is named **contract1** in tenant **tenant1**.
 - The contract that you want to export is named **exported_contract1** and you are exporting it to tenant **tenant2**.
- Navigate to the Contracts page (**Application Management > Contracts**).
The configured contracts are listed.
 - Select the contract that you just created.
For example, scroll through the list until you see the contract **contract1** and click the box next to it to select it.
 - Go to **Actions > Export Contract**.
The **Export Contract** window appears.
 - Click **Select Tenant**.
The **Select Tenant** window appears.
 - Select the tenant that you want to export the contract to, then click **Save**.
For example, **tenant2**. You are returned to the **Export Contract** window.
 - In the **Name** field, enter a name for the exported contract.
For example, **exported_contract1**.

- g) In the **Description** field, enter a description for the exported contract, if necessary.
- h) Click **Save**.

The list of contracts appears again.

Step 7

Configure the first tenant's EPG as the provider EPG, with the original contract, as the first part of the EPG communication configuration.

- a) Click the **Intent** button, then choose **EPG Communication**.

The **EPG Communication** window appears.

- b) Click **Let's Get Started**.
- c) In the **Contract** area, click **Select Contract**.

The **Select Contract** window appears.

- d) Locate and select the contract that you created at the beginning of these procedures.

In this example, you would locate and select **contract1**.

- e) Click **Select**.

The **EPG Communication** window appears.

- f) In the **Provider EPGs** area, click **Add Provider EPGs**.

The **Select Provider EPGs** window appears.

- g) Leave the **Keep selected items** box checked, then select the first tenant's (**tenant1**) EPG.
- h) Click **Select**.

The **EPG Communication** window appears.

- i) Click **Save**.

Step 8

Configure the second tenant's EPG as the consumer EPG, with the exported contract, as the second part of the EPG communication configuration.

- a) Click the **Intent** button, then choose **EPG Communication**.

The **EPG Communication** window appears.

- b) Click **Let's Get Started**.
- c) In the **Contract** area, click **Select Contract**.

The **Select Contract** window appears.

- d) Locate and select the contract that you created at the beginning of these procedures.

In this example, you would locate and select **exported_contract1**.

- e) Click **Select**.

The **EPG Communication** window appears.

- f) In the **Consumer EPGs** area, click **Add Consumer EPGs**.

The **Select Consumer EPGs** window appears.

- g) Leave the **Keep selected items** box checked, then select the second tenant's (**tenant2**) EPG.
- h) Click **Select**.

The **EPG Communication** window appears.

- i) Click **Save**.

Configuring Network Security Groups Using the Cisco Cloud Network Controller GUI

As described in [Security Groups, on page 45](#), the way the network security groups are configured differ, depending on the release:

- For releases prior to Release 5.1(2), there is a one-to-one mapping between NSGs in Azure and EPGs on Cisco Cloud Network Controller (these configurations are also referred to as **NSG-per-EPG** configurations throughout this document).
- Beginning with Release 5.1(2), in addition to the existing NSG-per-EPG configurations available previously, NSGs in Azure can also have a one-to-one mapping with subnets rather than EPGs on Cisco Cloud Network Controller (these configurations are also referred to as **NSG-per-subnet** configurations throughout this document).



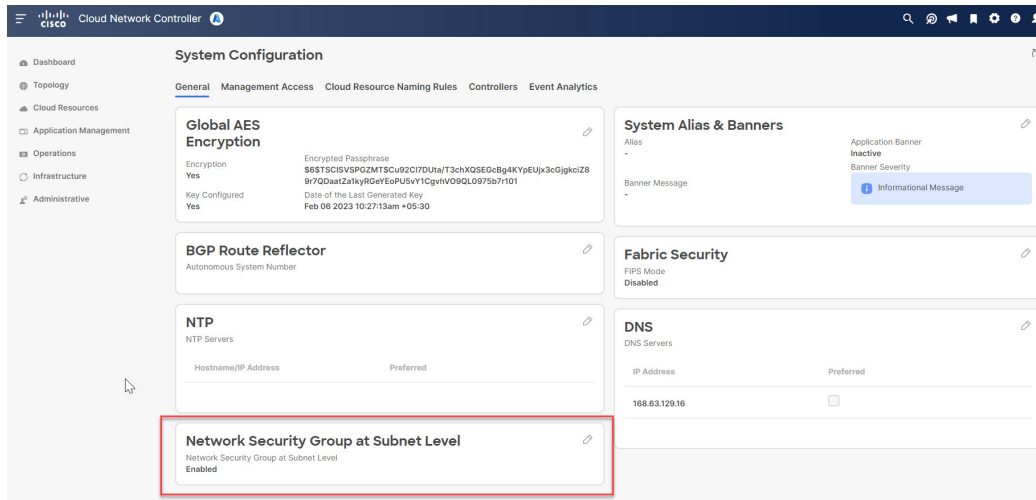
Note You can have either the newer **NSG-per-subnet** configuration *or* the older **NSG-per-EPG** configuration in your Cisco Cloud Network Controller. You cannot have both configurations in the same Cisco Cloud Network Controller system.

These procedures describe how to select either the newer **NSG-per-subnet** configuration or the older **NSG-per-EPG** configuration for your Cisco Cloud Network Controller for Release 5.1(2) or later.

Before you begin

Review the information provided in [Security Groups, on page 45](#) to better understand how security groups are configured, depending on the release, and to understand the guidelines and limitations for security groups.

-
- Step 1** Log in to the Cisco Cloud Network Controller, if you are not logged in already.
 - Step 2** In the left navigation bar, navigate to **Infrastructure > System Configuration**.
The **General** tab is displayed by default.
 - Step 3** In the **General** area in the **System Configuration** window, locate the **Network Security Group at Subnet Level** field.



Step 4 Determine the current setting for the **Network Security Group at Subnet Level** field.

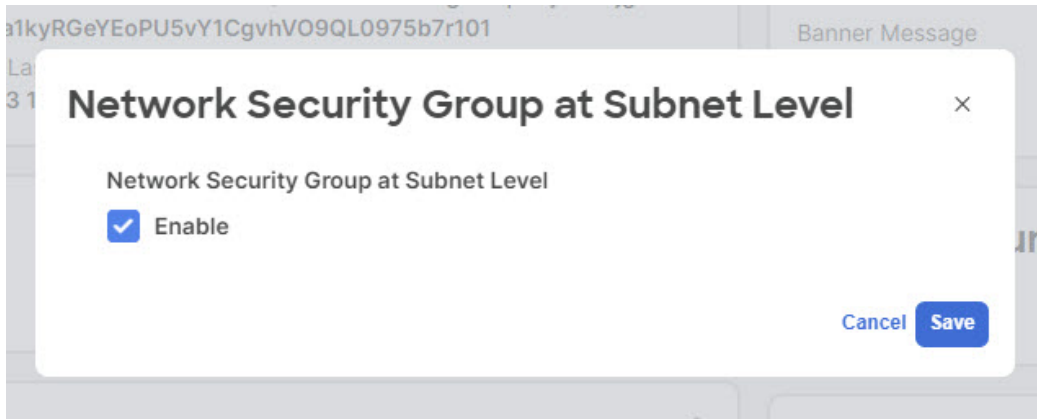
- If you see **Enabled** as the value in this field, that means that you have the newer **NSG-per-subnet** configuration for your Cisco Cloud Network Controller.
- If you see **Disabled** as the value in this field, that means that you have the older **NSG-per-EPG** configuration for your Cisco Cloud Network Controller.

Step 5 Determine if you want to change the setting for the **Network Security Group at Subnet Level** field or leave it as-is.

Desired Configuration	Existing Configuration	Action
If you want to have the newer NSG-per-subnet configuration for your Cisco Cloud Network Controller, and:	You see Enabled as the value in the Network Security Group at Subnet Level field, then:	Your Cisco Cloud Network Controller is already set up with the NSG-per-subnet configuration that you want. You do not have to make any changes.
	You see Disabled as the value in the Network Security Group at Subnet Level field, then:	You will have to change the setting in the Network Security Group at Subnet Level field. Go to Step 6, on page 109 .
If you want to have the older NSG-per-EPG configuration for your Cisco Cloud Network Controller, and:	You see Enabled as the value in the Network Security Group at Subnet Level field, then:	You will have to change the setting in the Network Security Group at Subnet Level field. Go to Step 6, on page 109 .
	You see Disabled as the value in the Network Security Group at Subnet Level field, then:	Your Cisco Cloud Network Controller is already set up with the NSG-per-EPG configuration that you want. You do not have to make any changes.

Step 6 If you have to change the setting in the **Network Security Group at Subnet Level** field, click the pencil icon in the upper right corner of the field.

The **Settings** window for **Network Security Group at Subnet Level** appears.



Step 7 Make the necessary changes in this window.

Note Changing the network security group setting will result in traffic loss. If you have to change the network security group setting, we recommend that you make the change during a maintenance window.

- If you want to have the newer **NSG-per-subnet** configuration for your Cisco Cloud Network Controller and you do not see a check in the box next to the **Enabled** field in this window, then click the box to add the check mark. This allows you to enable the newer **NSG-per-subnet** configuration for your Cisco Cloud Network Controller.
- If you want to have the older **NSG-per-EPG** configuration for your Cisco Cloud Network Controller and you see a check in the box next to the **Enabled** field in this window, then click the box to remove the check mark. This allows you to disable the newer **NSG-per-subnet** configuration, and to enable the older **NSG-per-EPG** configuration, for your Cisco Cloud Network Controller.

Note the following:

- Changing the setting from the newer **NSG-per-subnet** to the older **NSG-per-EPG** configuration is not recommended. Disabling the **NSG-per-subnet** setting means losing support for service EPG configurations and will result in traffic loss.
- If you have a service EPG or a private link label configured, you will not be able to disable the **NSG-per-subnet** configuration. You must disable the configured service EPG and/or a private link label before you can disable the **NSG-per-subnet** configuration.
 - To disable a configured service EPG:
 - a. Navigate to **Application Management > EPGs**.
 - b. Locate the EPGs with **Service** shown in the **Type** column.
 - c. Select the service EPG that you want to delete, then click **Actions > Delete EPG**.
 - To disable a configured private link label:
 - a. Navigate to **Application Management > Cloud Context Profiles**.
 - b. Locate the necessary cloud context profile and click on that profile.
A panel showing details for this cloud context profile slides in from the right side of the window.
 - c. Click the Details icon (🔍).

Another window appears that provides more detailed information for this cloud context profile. In the **CIDRs** area, you should see the text **Private Link Labels** in the **Subnets** column.

- d. Click the pencil icon in the upper right corner of the window.

The **Edit Cloud Context Profile** window appears.

- e. In the **Settings** area, locate the **CIDRs** area again and click the pencil icon in that row.

The **Edit CIDR** window appears.

- f. In the **Subnets** area, locate the row with an entry in the **Private Link Label** column and click on the pencil icon for that subnet row.

The entries on this subnet row become editable.

- g. Click the **X** next to the entry in the **Private Link Label** column for that subnet row.

This removes the private link label.

- Step 8** Click **Save** after you have made the necessary changes in the **Network Security Group at Subnet Level** window.

The **General** area in the **System Configuration** window appears again, and the setting in the **Network Security Group at Subnet Level** field reflects the change that you made in the previous step.

Viewing Security Group Details

- Step 1** Log into your Cisco Cloud Network Controller GUI, if you aren't logged in already.

- Step 2** Navigate to **Cloud Resources > Security Groups**.

The **Security Groups** window appears.

- Step 3** Click on the **Network Security Groups (NSG)** tab or the **Application Security Groups ASG** tab, depending on which type of security group that you want to get details on.

The following information is provided in each tab:

- **Network Security Groups** tab:

- **Name:** The name of the network security group.

- **Cloud Provider ID:** The cloud provider ID that is associated with the network security group.

Note that the value provided in the **Name** and the **Cloud Provider ID** fields will show whether the NSGs are configured with the newer NSG-per-subnet configuration (shown as **subnet-** in the **Cloud Provider ID** column) or with the older NSG-per-EPG configuration (shown as **epg-** in the **Cloud Provider ID** column). See [Security Groups, on page 45](#) for more information on the different types of NSG configurations available, depending on the software release.


- **EPGs:** The EPG that is associated with the network security group, if you have the older NSG-per-EPG configuration.

- **Virtual Machines:** The virtual machine that is associated with the network security group.

- **Endpoints:** The endpoints that are associated with the network security group.
- **Subnets:** The subnets that are associated with the network security group, if you have the newer NSG-per-subnet configuration.
- **Application Security Groups** tab:
 - **Health:** The health status for the application security group.
 - **Name:** The name of the application security group.
 - **Cloud Provider ID:** The cloud provider ID that is associated with the application security group.
 - **EPGs:** The EPG that is associated with the application security group.
 - **Virtual Machines:** The virtual machine that is associated with the application security group.
 - **Endpoints:** The endpoints that are associated with the application security group.

Step 4 Click on the value in any of the columns to get more detailed information.

For example, clicking on a value in the **Name** column in the **Network Security Groups** tab will bring up more detailed information about that particular network security group.

In this window, clicking on the Details icon () brings up another window that provides more detailed information for this security group, such as cloud resources information, including ingress and egress rules.

Specifying Consumer and Provider EPGs Using the Cisco Cloud Network Controller

This section explains how to specify an EPG as a consumer or a provider.

Before you begin

- You have configured a contract.
- You have configured an EPG.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 A list of options appears in the **Intent** menu. Under **Workflows**, click **EPG Communication**. The **EPG Communication** dialog box appears with the **Consumer EPGs**, **Contract**, and **Provider EPGs** information.

Step 3 To choose a contract:

- a) Click **Select Contract**. The **Select Contract** dialog appears.
- b) In the pane on the left side of the **Select Contract** dialog, click to choose a contract then click **Select**. The **Select Contract** dialog box closes.

Step 4 To add a consumer EPG:

- a) Click **Add Consumer EPGs**. The **Select Consumer EPGs** dialog appears.

Note EPGs within the tenant (where the contract is created) are displayed.

- b) In the pane on the left side of the **Select Consumer EPGs** dialog, click to place a check in a check box to choose an EPG.

Step 5 To add a provider EPG:

- a) Click **Add Provider EPGs**. The **Select Provider EPGs** dialog appears.

Note EPGs within the tenant (where the contract is created) are displayed.

- b) In the pane on the left side of the **Select Provider EPGs** dialog, click to place a check in a check box to choose a provider EPG.

Note If the chosen contract is an Imported Contract, the provider EPG selection is disabled.

- c) When finished, click **Select**. The **Select Provider EPGs** dialog box closes, and you return to the **EPG Communication Configuration** window.
d) Click **Save**.

Creating a Cloud Context Profile Using the Cisco Cloud Network Controller GUI

This section explains how to create a cloud context profile using the Cisco Cloud Network Controller GUI.

Before you begin

Create a VRF.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Create Cloud Context Profile**. The **Create Cloud Context Profile** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Cloud Context Profile Dialog Box Fields* table then continue.

Table 18: Create Cloud Context Profile Dialog Box Fields

Properties	Description
Name	Enter the name of the cloud context profile.
Tenant	To choose a tenant: <ol style="list-style-type: none"> a. Click Select Tenant. The Select Tenant dialog box appears. b. From the Select Tenant dialog, click to choose a tenant in the left column then click Select. You return to the Create Cloud Context Profile dialog box.

Properties	Description
Description	Enter a description of the cloud context profile.
Settings	
Region	<p>To choose a region:</p> <ol style="list-style-type: none"> a. Click Select Region. The Select Region dialog box appears. b. From the Select Region dialog, click to choose a region in the left column then click Select. You return to the Create Cloud Context Profile dialog box. <p>Note Beginning with 26.0(2), multiple cloud context profiles can be under same VRF and same region.</p>
VRF	<p>To choose a VRF:</p> <ol style="list-style-type: none"> a. Click Select VRF. The Select VRF dialog box appears. b. From the Select VRF dialog box, click to choose a VRF in the left column then click Select. You return to the Create Cloud Context Profile dialog box. <p>Note Beginning with 26.0(2), multiple cloud context profiles can be under same VRF and same region.</p>

Properties	Description
Add CIDR	

Properties	Description
	<p>Note The following subnet is reserved and should not be used in this Add CIDR field: 192.168.100.0/24 (reserved by the CCR for the bridge domain interface)</p> <p>Note You cannot add, delete, or edit a CIDR when VNet peering is enabled. You must disable VNet peering before adding, deleting or editing a CIDR. To disable VNet peering:</p> <ul style="list-style-type: none"> • For the infra tenant, disable the Hub Network Peering option in the cloud context profile • For a user (non-infra) tenant, disable the VNet Peering option in the cloud context profile <p>Enable VNet peering again after you have made the changes to the CIDR configuration.</p> <p>The following features are supported, depending on the release:</p> <ul style="list-style-type: none"> • You can add additional secondary CIDRs and subnets for infra VNets (cloudCtxProfiles created by the cloud template). You cannot add primary CIDRs or modify the existing CIDRs created by the cloud template. After subnets are created under the user-created CIDRs, the subnets will be implicitly mapped to the secondary VRF. • You can add also additional secondary CIDRs and subnets for VNets other than the infra VNet. <p>See Support for Multiple VRFs Under Single VNet, on page 34 for more information.</p> <p>To add a CIDR:</p> <ol style="list-style-type: none"> a. Click Add CIDR. The Add CIDR dialog box appears. b. Enter the address in the CIDR Block Range field. c. Click to check (enabled) or uncheck (disabled) the Primary check box. If you are adding additional secondary CIDRs and subnets for VNets, leave the Primary box unchecked. d. Click Add Subnet and enter the following information: <ul style="list-style-type: none"> • In the Address field, enter the subnet address. • In the Name field, enter the name for this subnet. • In the Private Link Label field, choose one of the following: <ul style="list-style-type: none"> • Select Existing: Click Select Private Link Label, then choose an existing private link label to associate with this subnet. • Create New: Enter a unique name for the private link label to associate with this subnet. • In the VRF field, make a selection.

Properties	Description
	<p>Note Beginning with 26.0(2), you can map subnets to different VRFs.</p> <p>e. When finished, click Add.</p>
VNet Gateway Router	Click to check (enable) or uncheck (disable) in the VNet Gateway Router check box.
VNet Peering	Click to check (enable) or uncheck (disable) the Azure VNet peering feature. For more information on the VNet peering feature, see the <i>Configuring VNet Peering for Cisco Cloud Network Controller for Azure</i> document in the Cisco Cloud Network Controller documentation page .

Step 5 Click **Save** when finished.

Configuring Virtual Machines in Azure

When you configure endpoint selectors for Cisco Cloud Network Controller, you will also need to configure the virtual machines that you will need in Azure that will correspond with the endpoint selectors that you configure for Cisco Cloud Network Controller.

This topic provides the requirements for configuring the virtual machines in Azure. You can use these requirements to configure the virtual machines in Azure either before you configure the endpoint selectors for Cisco Cloud Network Controller or afterward. For example, you might go to your account in Azure and create a custom tag or label in Azure first, then create an endpoint selector using a custom tag or label in Cisco Cloud Network Controller afterward. Or you might create an endpoint selector using a custom tag or label in Cisco Cloud Network Controller first, then go to your account in Azure and create a custom tag or label in Azure afterward.

Before you begin

You must configure a cloud context profile as part of the Azure virtual machine configuration process. When you configure a cloud context profile, the configurations, such as the VRF and region settings, are pushed out to Azure afterward.

Step 1 Review your cloud context profile configuration to get the following information:

- VRF name
- Subnet information
- Subscription Id
- The resource group that corresponds to where the cloud context profile is deployed.

Note In addition to the information above, if you are using tag-based EPGs, you also need to know the tag names. The tag names are not available in the cloud context profile configuration.

To obtain the cloud context profile configuration information:

- a) From the **Navigation** menu, choose the **Application Management** tab.
When the **Application Management** tab expands, a list of subtab options appear.
- b) Choose the **Cloud Context Profiles** subtab option.
A list of the cloud context profiles that you have created for your Cisco Cloud Network Controller are displayed.
- c) Select the cloud context profile that you will use as part of this Azure virtual machine configuration process.
Various configuration parameters are displayed for this cloud context profile, such as the region, VRF, IP address and subnets. Use the information displayed in this window when you configure the Azure virtual machine.

Step 2 Log in to the Azure portal account for the Cisco Cloud Network Controller user tenant and begin creating an Azure VM using the information you gathered from the cloud context profile configuration.

Note For information about how to create the VM in the Azure portal, see the Microsoft Azure documentation.

Creating a Backup Configuration Using the Cisco Cloud Network Controller GUI

This section explains how to create a backup configuration.

Before you begin

Create a remote location and a scheduler, if needed.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Operations**.

A list of **Operations** options appear in the **Intent** menu.

Step 3 From the **Operations** list in the **Intent** menu, click **Create Backup Configuration**. The **Create Backup Configuration** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Backup Configuration Dialog Box Fields* table then continue.

Table 19: Create Backup Configuration Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the backup configuration.
Description	Enter a description of the backup configuration.
Settings	

Properties	Description
Backup Destination	Choose a backup destination. <ul style="list-style-type: none">• Local• Remote

Properties	Description
Backup Object	

Properties	Description
	<p>Choose the root hierarchical content to consider for the backup</p> <ul style="list-style-type: none"> • Policy Universe • Selector Object—When chosen, this option adds the Object Type drop-down list and Object DN field. <ul style="list-style-type: none"> a. From the Object Type drop-down list, choose from the following options: <ul style="list-style-type: none"> • Tenant—When chosen the Select Tenant option appears. • Application Profile—When chosen the Select Application Profile option appears. • EPG—When chosen the Select EPG option appears. • Contract—When chosen the Select Contract option appears. • Filter—When chosen the Select Filter option appears. • VRF—When chosen the Select VRF option appears. • Device—When chosen the Select fvcloudLBCTX option appears. • Service Graph—When chosen the Select Service Graph option appears. • Cloud Context Profile—When chosen the Select Cloud Context Profile option appears. b. Click the Select <object_name>. The Select <object_name> dialog appears. c. From the Select <object_name> dialog, click to choose from the options in the left column then click Select. You return to the Create Backup Configuration dialog box. <p>Note The Object DN field is automatically populated with the DN of the object it will use as root of the object tree to backup</p> • Enter DN—When chosen, this option displays the Object DN field. <ul style="list-style-type: none"> a. From the Object DN field, enter the DN of a

Properties	Description
	specific object to use as the root of the object tree to backup.
Scheduler	<ol style="list-style-type: none"> Click Select Scheduler to open the Select Scheduler dialog and choose a scheduler from the left-side column. Click the Select button at the bottom-right corner when finished.
Trigger Backup After Creation	Choose one of the following: <ul style="list-style-type: none"> Yes—(Default) Trigger a backup after creating the backup configuration. No—Do not trigger a backup after creating the backup configuration.

Step 5 Click **Save** when finished.

Creating a Tech Support Policy Using the Cisco Cloud Network Controller GUI

This section explains how to create a tech support policy.

Before you begin

When creating a tech support policy for a remote location, you must first create the remote location.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Operations**.

A list of **Operations** options appear in the **Intent** menu.

Step 3 From the **Operations** list in the **Intent** menu, click **Create Tech Support**. The **Create Tech Support** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Tech Support Dialog Box Fields* table then continue.

Table 20: Create Tech Support Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the tech support policy.
Description	Enter a description of the tech support.
Settings	

Properties	Description
Export Destination	Choose an export destination. <ul style="list-style-type: none"> • Controller • Remote Location—When chosen the Select Remote Location option appears. <ol style="list-style-type: none"> Click Select Remote Location. The Select Remote Location dialog box appears. From the Select Remote Location dialog, click to choose a remote location in the left column then click Select. You return to the Create Tech Support dialog box.
Include Pre-Upgrade Logs	Click to place a check in the Enabled check box if you want to include pre-upgrade logs in the tech support policy.
Trigger After Creation	Click to place a check in the Enabled (the default) check box if you want to create the tech support policy after the policy creation. To disable, click the check box to uncheck.

Step 5 Click **Save** when finished.

Creating a Scheduler Using the Cisco Cloud Network Controller GUI

This section explains how to create a scheduler, which would be in User Laptop Browser local time and will be converted to the Cisco Cloud Network Controller default UTC time.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Operations**.

A list of **Operations** options appear in the **Intent** menu.

Step 3 From the **Operations** list in the **Intent** menu, click **Create Scheduler**. The **Create Scheduler** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Scheduler Dialog Box Fields* table then continue.

Table 21: Create Scheduler Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the trigger scheduler policy.
Description	Enter a description of the trigger scheduler.
Settings	

Properties	Description
Recurring Windows	<p>Click Add Recurring Window. The Add Recurring Window dialog appears.</p> <ol style="list-style-type: none"> a. From the Schedule drop-down list, choose from the following. <ul style="list-style-type: none"> • every-day • Monday • Tuesday • Wednesday • Thursday • Friday • Saturday • Sunday • odd-day • even-day b. From the Start Time field, enter a time. c. From the Maximum Concurrent Tasks field, enter a number or leave the field empty to specify unlimited. d. From the Maximum Running Time, click to choose Unlimited or Custom. e. Click Add when finished.
Add One Time Window	<p>Click Add One Time Window. The Add One Time Window dialog appears.</p> <ol style="list-style-type: none"> a. From the Start Time field, enter a date and time. b. From the Maximum Concurrent Tasks field, enter a number or leave the field blank to specify unlimited. c. From the Maximum Running Time, click to choose Unlimited or Custom. d. Click Add when finished.

Step 5 Click **Save** when finished.

Creating a Remote Location Using the Cisco Cloud Network Controller GUI

This section explains how to create a remote location using the Cisco Cloud Network Controller.

- Step 1** Click the **Intent** icon. The **Intent** menu appears.
- Step 2** Click the drop-down arrow below the **Intent** search box and choose **Operations**.
A list of **Operations** options appear in the **Intent** menu.
- Step 3** From the **Operations** list in the **Intent** menu, click **Create Remote Location**. The **Create Remote Location** dialog box appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create Remote Location Dialog Box Fields* table then continue.

Table 22: Create Remote Location Dialog Box Fields

Properties	Description
General	
Name	Enter the name of the remote location policy.
Description	Enter a description of the remote location policy.
Settings	
Hostname/IP Address	Enter the hostname or IP address of the remote location
Protocol	Choose a protocol: <ul style="list-style-type: none"> • FTP • SFTP • SCP
Path	Enter the path for the remote location.
Port	Enter the port for the remote location.
Username	Enter a username for the remote location.
Authentication Type	When using SFTP or SCP, choose the authentication type: <ul style="list-style-type: none"> • Password • SSH Key
SSH Key Content	Enter the SSH key content.
SSH Key Passphrase	SSH key passphrase.
Password	Enter a password for accessing the remote location.
Confirm Password	Reenter the password for accessing the remote location.

Step 5 Click **Save** when finished.

Creating a Login Domain Using the Cisco Cloud Network Controller GUI

This section explains how to create a login domain using the Cisco Cloud Network Controller GUI.

Before you begin

Create a provider before creating a non-local domain.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appear in the **Intent** menu.

Step 3 From the **Administrative** list in the **Intent** menu, click **Create Login Domain**. The **Create Login Domain** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Login Domain Dialog Box Fields* table then continue.

Table 23: Create Login Domain Dialog Box Fields

Properties	Description
Name	Enter the name of the login domain.
Description	Enter a description of the login domain.
Realm	Choose a realm: <ul style="list-style-type: none"> • Local • LDAP—Requires adding providers and choosing an authentication type. • RADIUS—Requires adding providers. • TACACS+—Requires adding providers. • SAML—Requires adding providers.
Providers	To add a provider: <ol style="list-style-type: none"> a. Click Add Providers. The Select Providers dialog appears with a list of providers in the left pane. b. Click to choose a provider. c. Click Select to add the provider.
Advanced Settings	Displays the Authentication Type and LDAP Group Map Rules fields.

Properties	Description
Authentication Type	<p>When LDAP is chosen for realm option, choose one of the following authentication types:</p> <ul style="list-style-type: none"> • Cisco AV Pairs—(Default) • LDAP Group Map Rules—Requires adding LDAP group map rules.
LDAP Group Map Rules	<p>To add an LDAP group map rule:</p> <ol style="list-style-type: none"> a. Click Add LDAP Group Map Rule. The Add LDAP Group Map Rule dialog appears with a list of providers in the left pane. b. Enter a name for the rule in the Name field. c. Enter a description for the rule in the Description field. d. Enter a group DN for the rule in the Group DN field. e. Add security domains: <ol style="list-style-type: none"> 1. Click Add Security Domain. The Add Security Domain dialog box appears. 2. Click Select Security Domain. The Select Security Domain dialog box appears with a list of security domains in the left pane. 3. Click to choose a security domain. 4. Click Select to add the security domain. You return to the Add Security Domain dialog box. 5. Add a user role: <ol style="list-style-type: none"> a. From the Add Security Domain dialog box, click Select Role. The Select Role dialog box appears with a list of roles in the left pane. b. Click to choose a role. c. Click Select to add the role. You return to the Add Security Domain dialog box. d. From the Add Security Domain dialog box, click the Privilege Type drop-down list and choose Read Privilege or Write Privilege. e. Click the check mark on the right side of the Privilege Type drop-down list to confirm. f. Click Add when finished. You return to the Add LDAP Group Map Rule dialog box where you can add another security domain.

Step 5 Click **Save** when finished.

Creating a Security Domain Using the Cisco Cloud Network Controller GUI

A security domain restricts the tenant to the security domains that you add. If you do not add a security domain, all security domains will have access to this tenant. This section explains how to create a security domain using the GUI.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appear in the **Intent** menu.

Step 3 From the **Administrative** list in the **Intent** menu, click **Security > Security Domains > Create Security Domain**. The **Create Security Domain** dialog box appears.

Step 4 In the **Name** field, enter the name of the security domain.

Step 5 In the **Description** field, enter a description of the security domain.

Step 6 Set the **Restricted Domain** control to **Yes** or **No**.

If the security domain is configured as a restricted domain (**Yes**), users who are assigned to this domain will not be able to see policies, profiles, or users configured in other security domains.

Step 7 Click **Save** when finished.

Creating a Role Using the Cisco Cloud Network Controller GUI

This section explains how to create a role using the Cisco Cloud Network Controller GUI.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appear in the **Intent** menu.

Step 3 From the **Administrative** list in the **Intent** menu, click **Create Role**. The **Create Role** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Role Dialog Box Fields* table then continue.

Table 24: Create Role Dialog Box Fields

Properties	Description
General	
Name	Enter a name for the role in the Name field.
Description	Enter a description of the role.
Settings	

Properties	Description
Privilege	

Properties	Description
	<p>Click to place a check mark in the check boxes of the privileges you want to assign the user. The privileges are:</p> <ul style="list-style-type: none"> • aaa—Used for configuring authentication, authorization, accounting and import/export policies. • access-connectivity-11—Used for Layer 1 configuration under infra. Example: selectors and port Layer 1 policy configurations. • access-connectivity-12—Used for Layer 2 configuration under infra. Example: Encap configurations on selectors, and attachable entity. • access-connectivity-13—Used for Layer 3 configuration under infra and static route configurations under a tenant's L3Out. • access-connectivity-mgmt—Used for management infra policies. • access-connectivity-util—Used for tenant ERSPAN policies. • access-equipment—Used for access port configuration. • access-protocol-11—Used for Layer 1 protocol configurations under infra. • access-protocol-12—Used for Layer 2 protocol configurations under infra. • access-protocol-13—Used for Layer 3 protocol configurations under infra. • access-protocol-mgmt—Used for fabric-wide policies for NTP, SNMP, DNS, and image management. • access-protocol-ops—Used for operations-related access policies such as cluster policy and firmware policies. • access-protocol-util—Used for tenant ERSPAN policies. • access-qos—Used for changing CoPP and QoS-related policies. • admin—Complete access to everything (combine ALL roles) • fabric-connectivity-11—Used for Layer 1 configuration under the fabric. Example: selectors and port Layer 1 policy and VNET protection.

Properties	Description
	<ul style="list-style-type: none"> • fabric-connectivity-l2—Used in firmware and deployment policies for raising warnings for estimating policy deployment impact. • fabric-connectivity-l3—Used for Layer 3 configuration under the fabric. Example: Fabric IPv4 and MAC protection groups. • fabric-connectivity-mgmt—Used for atomic counter and diagnostic policies on leaf switches and spine switches. • fabric-connectivity-util—Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches. • fabric-equipment—Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches. • fabric-protocol-l1—Used for Layer 1 protocol configurations under the fabric. • fabric-protocol-l2—Used for Layer 2 protocol configurations under the fabric. • fabric-protocol-l3—Used for Layer 3 protocol configurations under the fabric. • fabric-protocol-mgmt—Used for fabric-wide policies for NTP, SNMP, DNS, and image management. • fabric-protocol-ops—Used for ERSPAN and health score policies. • fabric-protocol-util—Used for firmware management traceroute and endpoint tracking policies. • none—No privilege. • nw-svc-device—Used for managing Layer 4 to Layer 7 service devices. • nw-svc-devshare—Used for managing shared Layer 4 to Layer 7 service devices. • nw-svc-params—Used for managing Layer 4 to Layer 7 service policies. • nw-svc-policy—Used for managing Layer 4 to Layer 7 network service orchestration.

Properties	Description
	<ul style="list-style-type: none"> • ops—Used for operational policies including monitoring and troubleshooting policies such as atomic counter, SPAN, TSW, tech support, traceroute, analytics, and core policies. • tenant-connectivity-11—Used for Layer 1 connectivity changes, including bridge domains and subnets. • tenant-connectivity-12—Used for Layer 2 connectivity changes, including bridge domains and subnets. • tenant-connectivity-13—Used for Layer 3 connectivity changes, including VRFs. • tenant-connectivity-mgmt—Used for tenant in-band and out-of-band management connectivity configurations and for debugging/monitoring policies such as atomic counters and health score. • tenant-connectivity-util—Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches. • tenant-epg—Used for managing tenant configurations such as deleting/creating endpoint groups, VRFs, and bridge domains. • tenant-ext-connectivity-12—Used for managing tenant L2Out configurations. • tenant-ext-connectivity-13—Used for managing tenant L3Out configurations. • tenant-ext-connectivity-mgmt—Used as write access for firmware policies. • tenant-ext-connectivity-util—Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk. • tenant-ext-protocol-11—Used for managing tenant external Layer 1 protocols. Generally only used for write access for firmware policies. • tenant-ext-protocol-12—Used for managing tenant external Layer 2 protocols. Generally only used for write access for firmware policies. • tenant-ext-protocol-13—Used for managing tenant external Layer 3 protocols such as BGP, OSPF, PIM, and IGMP. • tenant-ext-protocol-mgmt—Used as write access for firmware policies.

Properties	Description
	<ul style="list-style-type: none"> • tenant-ext-protocol-util—Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk. • tenant-network-profile—Used for managing tenant configurations, such as deleting and creating network profiles, and deleting and creating endpoint groups. • tenant-protocol-l1—Used for managing configurations for Layer 1 protocols under a tenant. • tenant-protocol-l2—Used for managing configurations for Layer 2 protocols under a tenant. • tenant-protocol-l3—Used for managing configurations for Layer 3 protocols under a tenant. • tenant-protocol-mgmt—Only used as write access for firmware policies. • tenant-protocol-ops—Used for tenant traceroute policies. • tenant-protocol-util—Used for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk. • tenant-qos—Only used as Write access for firmware policies. • tenant-security—Used for Contract related configurations for a tenant. • vmm-connectivity—Used to read all the objects in APIC's VMM inventory required for VM connectivity. • vmm-ep—Used to read VM and Hypervisor endpoints in the APIC's VMM inventory. • vmm-policy—Used for managing policies for VM networking. • vmm-protocol-ops—Not used by VMM policies. • vmm-security—Used for Contract related configurations for a tenant.

Step 5 Click **Save** when finished.

Creating a Certificate Authority Using the Cisco Cloud Network Controller GUI

This section explains how to create a certificate authority using the GUI.

Before you begin

- Have the certificate chain.
- If the certificate authority is for a tenant, create the tenant.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appears in the **Intent** menu.

Step 3 From the **Administrative** list in the **Intent** menu, click **Create Certificate Authority**. The **Create Certificate Authority** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Certificate Authority Dialog Box Fields* table then continue.

Table 25: Create Certificate Authority Dialog Box Fields

Properties	Description
Name	Enter the name of the certificate authority.
Description	Enter a description of the certificate authority.
Used for	Choose from the following options: <ul style="list-style-type: none"> • Tenant—Choose if the certificate authority is for a specific tenant. When chosen, the Select Tenant option appears in the GUI. • System—Choose if the certificate authority is for the system.
Select Tenant	To choose a tenant: <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column then click Select. You return to the Create Certificate Authority dialog box.
Certificate Chain	Enter the certificate chain in the Certificate Chain text box. <p>Note Add the certificates for a chain in the following order:</p> <ol style="list-style-type: none"> CA Sub-CA Subsub-CA Server

Step 5 Click **Save** when finished.

Creating a Key Ring Using the Cisco Cloud Network Controller GUI

This section explains how to create a key ring using the Cisco Cloud Network Controller GUI.

Before you begin

- Create a certificate authority.
- Have a certificate.
- If the key ring is for a specific tenant, create the tenant.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appear in the **Intent** menu.

Step 3 From the **Administrative** list in the **Intent** menu, click **Create Key Ring**. The **Create Key Ring** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Key Ring Dialog Box Fields* table then continue.

Table 26: Create Key Ring Dialog Box Fields

Properties	Description
Name	Enter the name of the key ring.
Description	Enter a description of the key ring.
Used for	<ul style="list-style-type: none"> • System—The key ring is for the system. • Tenant—The key ring is for a specific tenant. Displays a Tenant field for specifying the tenant.
Select Tenant	To choose a tenant: <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog box appears. From the Select Tenant dialog, click to choose a tenant in the left column then click Select. You return to the Create Key Ring dialog box.
Settings	

Properties	Description
Certificate Authority	To choose a certificate authority: <ol style="list-style-type: none"> Click Select Certificate Authority. The Select Certificate Authority dialog appears. Click to choose a certificate authority in the column on the left. Click Select. You return to the Create Key Ring dialog box.
Private Key	Choose one of the following: <ul style="list-style-type: none"> • Generate New Key—Generates a new key. • Import Existing Key—Displays the Private Key text box and enables you to use an existing key.
Private Key	Enter an existing key in the Private Key text box (for the Import Existing Key option).
Modulus	Click the Modulus drop-down list to choose from the following: <ul style="list-style-type: none"> • MOD 512 • MOD 1024 • MOD 1536 • MOD 2048—(Default)
Certificate	Enter the certificate information in the Certificate text box.

Step 5 Click **Save** when finished.

Creating a Local User Using the Cisco Cloud Network Controller GUI

This section explains how to create a local user using the Cisco Cloud Network Controller GUI.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appear in the **Intent** menu.

Step 3 From the **Administrative** list in the **Intent** menu, click **Create Local User**. The **Create Local User** dialog box appears.

Step 4 Enter the appropriate values in each field as listed in the following *Create Local User Dialog Box Fields* table then continue.

Table 27: Create Local User Dialog Box Fields

Properties	Description
Name	Enter the username of the local user.
Password	Enter the password for the local user.
Confirm Password	Reenter the password for the local user.
Description	Enter a description of the local user.
Settings	
Account Status	To choose the account status: <ul style="list-style-type: none">• Active—Activates the local user account.• Inactive—Deactivates the local user account.
First Name	Enter the first name of the local user.
Last Name	Enter the last name of the local user.
Email Address	Enter the email address of the local user.
Phone Number	Enter the phone number of the local user.

Properties	Description
Security Domains	<p>To add a security domain:</p> <ol style="list-style-type: none"> a. Click Add Security Domain. The Add Security Domain dialog box appears. b. Click Select Security Domain. The Select Security Domain dialog box appears with a list of security domains in the left pane. c. Click to choose a security domain. d. Click Select to add the security domain. You return to the Add Security Domain dialog box. e. Add a user role: <ol style="list-style-type: none"> 1. From the Add Security Domain dialog box, click Select Role. The Select Role dialog box appears with a list of roles in the left pane. 2. Click to choose a role. 3. Click Select to add the the role. You return to the Add Security Domain dialog box. 4. From the Add Security Domain dialog box, click the Privilege Type drop-down list and choose Read Privilege or Write Privilege. 5. Click the check mark on the right side of the Privilege Type drop-down list to confirm. 6. Click Add when finished. You return to the Create Local User dialog box where you can add another security domain.

Step 5 Click **Advanced Settings** and enter the appropriate values in each field as listed in the following *Create Local User Dialog Box Fields: Advanced Settings* table then continue.

Table 28: Create Local User Dialog Box Fields: Advanced Settings

Property	Description
Account Expires	If you choose Yes , the account is set to expire at the time that you choose.
Password Update Required	If you choose Yes , the user must change the password upon the next login.
OTP	Put a check in the box to enable the one-time password feature for the user.

Property	Description
User Certificates	<p>To add a user certificate:</p> <ol style="list-style-type: none"> Click Add X509 Certificate. The Add X509 Certificate dialog box appears. Enter a name in the Name field. Enter the X509 certificate in the User X509 Certificate text box. Click Add. The X509 certificate in the User X509 Certificate dialog box closes. You return to the Local User dialog box.
SSH Keys	<p>To add a an SSH key:</p> <ol style="list-style-type: none"> Click Add SSH Key. The Add SSH Key dialog box appears. Enter a name in the Name field. Enter the SSH key in the Key text box. Click Add. The Add SSH Key dialog box closes. You return to the Local User dialog box.

Step 6 Click **Save** when finished.

Managing Regions (Configuring a Cloud Template) Using the Cisco Cloud Network Controller GUI

Regions are configured during the first-time setup. When configured, you specify the regions that are managed by Cisco Cloud Network Controller and the region's inter-site and inter-region connectivity. This section explains how to manage regions with the cloud template using the Cisco Cloud Network Controller GUI after the initial installation.

For more information about cloud templates, see [About the Cloud Template, on page 52](#).

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 A list of options appear in the **Intent** menu. Under **Workflows**, click **Cisco Cloud Network Controller Setup**. The **Set up- Overview** dialog box appears with options for **DNS and NTP Servers**, **Region Management**, **Advanced Settings**, and **Smart Licensing**.

Step 3 For **Region Management**, click **Edit Configuration**.

The **Setup- Region Management** dialog box appears. and the first step in the **Setup- Region Management** series of steps appears, **Regions to Manage**, with a list of managed regions.

Step 4 If you want inter-site connectivity, click to place a check mark in the **Enabled** box in the **Inter-Site Connectivity** area. The **Inter-Site Connectivity** step is added in the **Setup- Region Management** steps at the top of the page.

Step 5 To choose a region that you want to be managed by the Cisco Cloud Network Controller, click to place a check mark in check box of that region.

Step 6 To deploy cloud routers locally to this region, click to place a check mark in the **Cloud Routers** check box for that region.

Step 7 To configure the fabric infra connectivity for the cloud site, click **Next**.
The next step in the **Setup- Region Management** series of steps appears, **General Connectivity**.

Step 8 To add a subnet pool for the CCRs, click **Add Subnet Pool for Cloud Router** and enter the subnet in the text box.

Note The /24 subnet provided during the Cisco Cloud Network Controller deployment would be sufficient for up to two cloud sites. If you need to manage more than two cloud sites, you need to add more subnets.

Step 9 Enter a value in the **BGP Autonomous System Number for CCRs** field.

The BGP ASN can be in the range of 1- 65534.

Step 10 In the **Assign Public IP to CCR Interface** field, determine if you want to assign public IP addresses to the Catalyst 8000V interfaces.

Private IP addresses are assigned to the Catalyst 8000V interfaces by default. The **Assign Public IP to CCR Interface** option determines whether public IP addresses will also be assigned to the Catalyst 8000V interfaces or not.

Note that CCRs require a public IP address for intersite communication.

By default, the **Enabled** check box is checked. This means that public IP addresses can be assigned to the Catalyst 8000Vs.

- If you want *public* IP addresses assigned to the Catalyst 8000Vs in addition to the private IP addresses, leave the check in the box next to **Enabled**.
- If you want only *private* IP addresses assigned to the Catalyst 8000Vs, remove the check in the box next to **Enabled** to disable this option.

Note Changing a CCR address from a public IP address to a private IP address (or vice-versa) is a disruptive operation and can result in traffic loss.

Both the public and private IP addresses assigned to a CCR are displayed with the other details of the router in the **Cloud Resources** area. If public IP addresses are not assigned to a CCR, only the private IP addresses are displayed.

Step 11 To chose the number of routers per region, click the **Number of Routers Per Region** drop-down list and click **2, 3, 4, 6, or 8**.

Step 12 Enter a username in the **Username** text box.

Note Do not use admin as a username for the CCR when connecting to an Azure cloud site.

Step 13 Enter a password in the **Password** and **Confirm Password** text boxes.

Step 14 To choose the throughput value, click the **Throughput of the routers** drop-down list.

- Note**
- Cloud routers should be undeployed from all regions before changing the throughput or login credentials.
 - For information on the throughput values for the Cisco Catalyst 8000V, see [About the Cisco Catalyst 8000V, on page 29](#).

Step 15 Enter the necessary information in the **TCP MSS** field, if applicable.

The **TCP MSS** option is available to configure the TCP maximum segment size (MSS). This value will be applied to all cloud router tunnel interfaces, including VPN tunnels towards the cloud and external tunnels towards the on-premises site or other cloud sites. For VPN tunnels towards the cloud, if the cloud provider's MSS value is less than the value that you enter in this field, then the lower value is used; otherwise, the value that you enter in this field is used.

The MSS value affects only TCP traffic, and has no impact on other types of traffic, such as ping traffic.

Step 16 (Optional) To specify the license token, enter the product instance registration token in the **License Token** text box.

- Note**
- For licensing information for the Cisco Catalyst 8000V, see [About the Cisco Catalyst 8000V, on page 29](#).
 - If no token is entered, the CCR will be in EVAL mode.
 - If you assigned private IP addresses to the CCRs in [Step 10, on page 140](#), the only supported option is **Direct connect to Cisco Smart Software Manager (CSSM)** when registering smart licensing for CCRs with private IP addresses (available by navigating to **Administrative > Smart Licensing**). You must provide reachability to the CSSM through express route in this case.

Step 17 Click **Next**.

- If you placed a check mark in the **Enabled** box in the **Inter-Site Connectivity** area earlier in these procedures, **Inter-Site Connectivity** appears as the next step in the **Setup- Region Management** series of steps. Go to [Step 18, on page 141](#).
- If you did not place a check mark in the **Enabled** box in the **Inter-Site Connectivity** area earlier in these procedures, **Cloud Resource Naming Rules** appears as the next step in the **Setup- Region Management** series of steps. Go to [Step 22, on page 141](#).

Step 18 To enter a peer public IP address of the IPsec Tunnel peer on-premises in the text box, click **Add Public IP of IPsec Tunnel Peer**.

Step 19 Enter the OSPF area ID in the **OSPF Area Id** text box.

Step 20 To add an external subnet pool, click **Add External Subnet** and enter a subnet pool in the text box.

Step 21 When you have configured all the connectivity options, click **Next** at the bottom of the page.

The **Cloud Resource Naming Rules** page appears.

Step 22 In the **Cloud Resource Naming Rules** page, configure the cloud resource naming rules, if necessary.

The cloud resource naming rules are described in detail in the [Cloud Resources Naming, on page 142](#) section. If you don't need to make any changes to the naming rules, you can skip this page.

Step 23 Click **Save and Continue** when finished.

Configuring Smart Licensing

This task demonstrates how to set up smart licensing in the Cisco Cloud Network Controller.

Before you begin

You need the product instance registration token.

-
- Step 1** Click the **Intent** icon. The **Intent** menu appears.
- Step 2** A list of options appear in the **Intent** menu. Under **Workflows**, click **Cloud Network Controller Setup**. The **Set up-Overview** dialog box appears with options for **DNS Servers**, **Region Management**, **Advanced Settings**, and **Smart Licensing**.
- Step 3** To register the Cisco Cloud Network Controller to Cisco's unified license management system: From **Smart Licensing**, click **Register**. The **Smart Licensing** dialog appears.
- Step 4** Choose a transport setting:
- **Direct to connect to Cisco Smart Software Manager (CSSM)**
 - **Transport Gateway/Smart Software Manager Satellite**
 - **HTTP/HTTPS Proxy**
- Note** An IP address is also required when choosing **HTTP/HTTPS Proxy**.
- Step 5** Enter the product instance registration token in the provided text box.
- Step 6** Click **Register** when finished.
-

Cloud Resources Naming

You can create a global naming policy on the Cisco Cloud Network Controller, which allows you to define a custom cloud resources naming convention for all objects deployed from the Cisco Cloud Network Controller into the Azure cloud. You can define custom naming rules for all cloud resources during the first time setup wizard of the Cisco Cloud Network Controller, with the exception of the **Resource group** name used for the Cisco Cloud Network Controller ARM template deployment. The resource group name for the template is defined when you first deploy it and cannot be changed after. In addition to the global policy, you can also explicitly define the names of the cloud resources created from each Cisco Cloud Network Controller object using the REST API.

Starting with Cisco Cloud Network Controller Release 5.1(2), for Layer 4 to Layer 7 service deployments, you can provide custom names to cloud resources, such as, Network Load Balancers, Application Load Balancers and Device Application Security Groups.



Note Keep in mind that even with custom naming policy, once a cloud resource is created, you will not be able to modify the name. If you want to change the name of an existing cloud resource, you would need to delete all configured cloud resources and recreate them. Cloud resources to be deleted include secondary CIDR and subnets, CCRs deployed by Cisco Cloud Network Controller and therefore IPsec tunnels from the CCRs to every remote site.

Variables Available for Naming Rules

When creating your cloud resources naming policy, you can use the following variables to dynamically define the name of the cloud resource based on the Cisco Cloud Network Controller objects:

- `${tenant}` – the resource will include the name of the Tenant

- `${ctx}` – the resource will include the name of the VRF
- `${ctxprofile}` – the resources will include the cloud context profile, which is a VRF deployed in a given cloud region
- `${subnet}` – the resource will include the string `subnet` followed by the subnet IP address
- `${app}` – the resource will include the name of the application profile.
- `${epg}` – the resource will include the name of the EPG.
- `${contract}` – the resource will include the name of the contract
- `${region}` – the resource will include the name of the cloud region
- `${priority}` – the resource will include the name of the network security group (NSG) rule priority. This number is allocated automatically to ensure that each NSG rule name is unique
- `${serviceType}` – the resource will include an abbreviation of the service Type (only valid for private endpoint resources)
- `${resourceName}` – the resource will include the name of the target resource (only valid for private endpoint resources)
- `${device}` – the resource will include the name of the Layer 4 to Layer 7 device.
- `${interface}` – the resource will include the name of the Layer 4 to Layer 7 device interface.
- `${deviceInterfaceDn}` – the resource will include the DN of the Layer to Layer 7 device interface.

For private endpoints, the combination of the

`${app}-${svcepg}-${subnet}-${serviceType}-${resourceName}` makes the private endpoint name unique. Removing any of these variables might form a name of a private endpoint that already exists. This would result in a fault raised by the Cisco Cloud Network Controller. Also, the max length requirements vary from Azure service to service.

When you define a global naming policy using one or more of the above variables, Cisco Cloud Network Controller validates the string to ensure that all mandatory variables are present and no invalid string is specified.

There is a maximum name length limit in Azure. If the length of the name exceeds the length supported by the cloud provider, it rejects the config and Cisco Cloud Network Controller raises a fault that the resource creation failed. You can then check the fault for details and correct the naming rules. The maximum length limits at the time of Cisco Cloud Network Controller, Release 5.0(2) are listed below, for the latest up-to-date information and any changes to the length limit, consult the Azure documentation.

The following table provides a summary of which cloud resources support each of the naming variables above. Cells denoted with an asterisk (*) indicate variables that are mandatory for that type of cloud resource. Cells denoted with a plus sign (+) indicate that at least one of these variables is mandatory for that type of cloud resource; for example, for VNET resources you can provide `${ctx}`, or `${ctxprofile}`, or both.



Note For configuring multiple VNets in different resource groups but in the same VRF, the default naming for Resource Group has been changed to CAPIC_\${tenant}_\${ctx}_\${region}_\${ctxprofile}. This is done to distinguish between the different VNets in the same VRF.

The keyword \${ctxprofile} is a new allowed keyword for resource group naming, and it's selected by default. It cannot be removed. The old naming can be kept only in the case of an upgrade from a prior release to 26.0(2).

Table 29: Supported Variables for Cloud Resources

Azure Resource	\${tenant}	\${ctx}	\${ctxprofile}	\${subnet}	\${app}	\${epg}	\${contract}	\${region}	\${priority}
Resource Group Max Length: 90	Yes*	Yes*	Yes*					Yes*	
Virtual Network (VNET) Max Length: 64	Yes	Yes+	Yes+					Yes	
Subnet Max Length: 80	Yes	Yes	Yes	Yes*				Yes	
Application Security Group (ASG) Max Length: 80	Yes				Yes*	Yes*		Yes	
Network Security Group (NSG) Max Length: 80	Yes				Yes*	Yes*		Yes	

Azure Resource	\${tenant}	\${ctx}	\${ctxprofile}	\${subnet}	\${app}	\${epg}	\${contract}	\${region}	\${priority}
Network Security Group Rule Max Length: 80	Yes						Yes		Yes* (auto)

Table 30: Supported Variables for Cloud Resources (Layer 4 to Layer 7 device services)

Azure Resource	\${tenant}	\${region}	\${ctxprofile}	\${device}	\${interface}	\$(deviceInterfaceDN)
Internal Network Load Balancer Max Length: 80	Yes	Yes	Yes	Yes*		
Internet-facing Network Load Balancer Max Length: 80	Yes	Yes	Yes	Yes*		
Internal Application Load Balancer Max Length: 80	Yes	Yes	Yes	Yes*		
Internet-facing Application Load Balancer Max Length: 80	Yes	Yes	Yes	Yes*		
Device ASG Max Length: 80	Yes	Yes		Yes*	Yes*	Yes*

Naming Rules Guidelines and Limitations

When configuring custom rules for naming cloud resources, the following restrictions apply:

- You define global naming policy during the Cisco Cloud Network Controller 's first time setup using two sets of naming rules:
 - Hub Resource Naming Rules** define names for the Hub Resource Group, Hub VNET, overlay-1 CIDR, secondary CIDR subnet in the Infra Tenant, as well as the subnet prefixes for subnets that are created automatically by the system in the Infra tenant.
 - Cloud Resource Naming Rules** define the names of the Network Security Group (NSG), Application Security Group (ASG), Network Load Balancer, Application Load Balancer, Device Application

Security Group, and subnets you create in the Infra Tenant, as well as the names of all resources (Resource Groups, Virtual Networks, Subnets, NSG, ASG, Network Load Balancer, Application Load Balancer) in user Tenants.

After you define the naming rules, you will be required to review and confirm them. Keep in mind that you must confirm the naming rules before any cloud resources are deployed.

- Once a cloud resource is created, its name cannot be changed and the naming policy cannot be updated in the GUI. If you upgrade your Cisco Cloud Network Controller to Release 5.0(2) with some resources already deployed in Azure, you will also not be able to change the global custom naming rules.

If you want to change the names of the existing cloud resources or the policy, you would need to delete the deployed resources before being able to update the global naming policy in the GUI.

In these cases you can use the REST API to explicitly assign custom names to any new resources you create.

- When updating cloud resources naming via REST API, we recommend you do not import configuration at the same time.

We recommend you define any naming rules first. Then any tenant configuration.

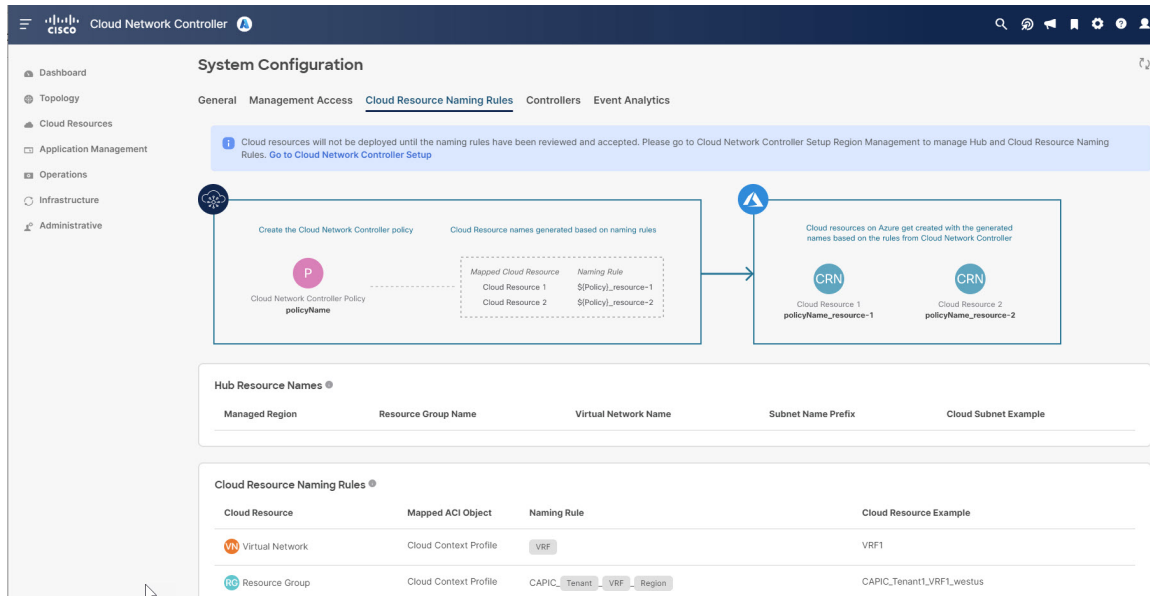
We recommend that you do not change the naming policy after the tenant configuration is deployed.

Viewing Cloud Resource Naming Rules

You initially define the cloud resource naming rules in the Region Management part of the first time setup wizard when you deploy your Cisco Cloud Network Controller, which is described in the *Cisco Cloud Network Controller Installation Guide*. After the initial setup, you can view the rules you configured in the **System Configuration** screen of your Cisco Cloud Network Controller GUI as described in this section.

Note that the information in this screen is presented in read-only view and if you want to change the rules any time after the original deployment, you will need to re-run the first time setup wizard.

-
- Step 1** Log in to your Cisco Cloud Network Controller GUI.
- Step 2** Navigate to the **Cloud Resource Naming Rules** screen.



- In the **Navigation** sidebar, expand the **Infrastructure** category.
- From the **Infrastructure** category, select **System Configuration**.
- In the **System Configuration** screen, select the **Cloud Resource Naming Rules** tab.

In the **Cloud Resource Naming Rules** tab, you can see a summary of the currently configured rules for the names of resources that you deploy in the cloud site from your Cisco Cloud Network Controller.

If you did not configure custom naming rules before, the default rules are listed here, which use the Cisco Cloud Network Controller object names for cloud resources.

If you have not accepted the naming rules you have defined during the first time setup, a warning banner will be displayed across the top of the screen.

Note Keep in mind that you must confirm the naming rules before any cloud resources are deployed.

Configuring Cisco Cloud Network Controller Using the REST API

Creating a Tenant Using the REST API

There are two types of subscriptions: own and shared. Each subscription type has a primary tenant. You choose the own subscription when creating a new managed or unmanaged tenant. You choose the shared subscription when creating a tenant that inherits the managed or unmanaged settings of an existing primary tenant. This section demonstrates how to create a managed and unmanaged tenant with the own type of subscription and how to create a shared subscription.



Note Beginning with 26.0(2), support is now available for having multiple cloud accounts under a single tenant. For more information, see [“Tenants”](#).

This section demonstrates how to create a tenant using the REST API using sample POST requests from the body of Postman.

Step 1 Create an own subscription.

a) To create an unmanaged tenant using a client secret:

```
POST https://<cloud-net-controller-ip-address>/api/mo/uni.xml

<fvTenant name="{{primary-tenant-name}}">
  <cloudAccount id="{{user-tenant-subscription-id}}" vendor="azure" accessType="credentials"
status="">
    <cloudRsCredentials tDn="uni/tn-{{primary-tenant-name }}/credentials-{{ primary-tenant-name
}}"/>
  </cloudAccount>
  <cloudCredentials name="{{ primary-tenant-name }}" keyId="{{application_key_id}}"
key="{{client_secret_key}}">
    <cloudRsAD tDn="uni/tn-{{ primary-tenant-name }}/ad-{{active_directory_id}}"/>
  </cloudCredentials>
  <cloudAD name="{{active_directory_name}}" id="{{active_directory_id}}"/>
  <fvRsCloudAccount tDn="uni/tn-{{ primary-tenant-name }}/act-{{ user-tenant-subscription-id
}}-vendor-azure" status=""/>
</fvTenant>
```

b) To create a managed tenant:

```
POST https://<cloud-net-controller-ip-address>/api/mo/uni.xml

<fvTenant name="{{ primary-tenant-name }}">
  <cloudAccount id="{{ user-tenant-subscription-id }}" vendor="azure" accessType="managed"
status="" />
  <fvRsCloudAccount tDn="uni/tn-{{ primary-tenant-name }}/act-{{ user-tenant-subscription-id
}}-vendor-azure" status=""/>
</fvTenant>
```

Step 2 Create a shared subscription:

```
POST https://<cloud-net-controller-ip-address>/api/mo/uni.xml

<fvTenant name="{{ primary-tenant-name }}">
  <fvRsCloudAccount tDn="uni/tn-{{ primary-tenant-name }}/act-{{ user-tenant-subscription-id
}}-vendor-azure" status=""/>
</fvTenant>
```

Step 3 Create multiple cloud subscriptions for a single tenant

```
<fvTenant name="AzTenant" status="">
  <cloudAccount name="acct1" id="111111111111" vendor="azure" accessType="credentials" status="">
    <cloudRsCredentials tDn="uni/tn-infra/credentials-cApicInfraApp" status=""/>
  </cloudAccount>
  <cloudCredentials name="cApicInfraApp" keyId="xxxx" key="xxxx" httpProxy="">
```



```

    <cloudRsAD tDn="uni/tn-infra/123456789"/>
  </cloudCredentials>

  <fvRsCloudAccount tDn="uni/tn-infra/act-[111111111111]-vendor-azure" />
<cloudAccount name="acct2" id="222222222222" vendor="azure" accessType="credentials" status="">

  <cloudRsCredentials tDn="uni/tn-AzTenant/credentials-cApicApp1" status="" />
</cloudAccount>
<cloudCredentials name="cApicApp1" keyId="xxxx" key="xxxxx" httpProxy="">
  <cloudRsAD tDn="uni/tn-AzTenant/123456789"/>
</cloudCredentials>

  <cloudAD name="CiscoINSBUAd" id="123456789" />

  <fvCtx name="vrf20"/>
  <cloudCtxProfile name="VNET1-central-us" status="" azVirtualNetwork="vnet5" status="">
    <cloudRsCtxProfileToRegion status="" tDn="uni/clouddomp/provp-azure/region-centralus"
status ="" />
    <cloudRsToCtx tnFvCtxName="vrf20" />
    <cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status="" />
    <cloudCidr name="cidr1" addr="42.10.0.0/16" primary="yes" status="">
      <cloudSubnet ip="42.10.10.0/24" status="">
        <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-centralus/zone-default"/>
      </cloudSubnet>
    </cloudCidr>
  </cloudCtxProfile>
</fvTenant>

```

Step 4 Add cloud context profile to non default account

```

<cloudCtxProfile name="VNET4-east-us" status="" azVirtualNetwork="vnet2" status="">
  <cloudRsCtxProfileToRegion status="" tDn="uni/clouddomp/provp-azure/region-eastus" status
="" />
  <cloudRsCtxProfileToAccount tDn="uni/tn-AzTenant/act-[222222222222]-vendor-azure"
status="" />
  <cloudRsToCtx tnFvCtxName="vrf20" />
  <cloudRsCtxProfileToAccount tDn="uni/tn-tenant1/act-[222222222222]-vendor-azure" status="" />

  <cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status="" />
  <cloudCidr name="cidr1" addr="52.12.0.0/16" primary="yes" status="">
    <cloudSubnet ip="52.12.12.0/24" status="">
      <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-eastus/zone-default"/>
    </cloudSubnet>
  </cloudCidr>
</cloudCtxProfile>

```

Creating a Contract Using the REST API

This example demonstrates how to create a contract for the Cisco Cloud Network Controller using the REST API.

Before you begin

Create filters.

To create a contract:

Example:

```
<polUni>
  <fvTenant name="t2" status="">
    <vzFilter descr="" name="http-family-destination" ownerKey="" ownerTag="">
      <vzEntry name="http" prot="tcp" etherT="ip" dFromPort="http" dToPort="http"/>
      <vzEntry name="https" prot="tcp" etherT="ip" dFromPort="https" dToPort="https"/>
    </vzFilter>
    <vzBrCP name="httpFamily">
      <vzSubj name="default" revFltPorts="yes" targetDscp="unspecified">
        <vzRsSubjFiltAtt action="permit" directives="" tnVzFilterName="http-family-destination"/>
      </vzSubj>
    </vzBrCP>
  </fvTenant>
</polUni>
```

Creating a Cloud Context Profile Using the REST API

This section demonstrates how to create a cloud context profile.



Note Beginning with 26.0(2), multiple cloud context profiles can be under same VRF and same region.

Before you begin

Create a VRF.

Step 1 To create a basic cloud context profile:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <cloudCtxProfile name="cProfilewestus151">
      <cloudRsCtxProfileToRegion tDn="uni/clouddomp/provp-azure/region-westus"/>
      <cloudRsToCtx tnFvCtxName="ctx151"/>
      <cloudCidr addr="15.151.0.0/16" primary="true" status="">
        <cloudSubnet ip="15.151.1.0/24" name="GatewaySubnet" usage="gateway">
          <cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
        </cloudSubnet>
        <cloudSubnet ip="15.151.2.0/24" name="albsubnet" >
          <cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
        </cloudSubnet>
        <cloudSubnet ip="15.151.3.0/24" name="subnet" usage="">
          <cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
        </cloudSubnet>
      </cloudCidr>
    </cloudCtxProfile>
  </fvTenant>
</polUni>
```

Step 2 To create a cloud context profile where you are adding a secondary VRF, CIDR, and subnet for a VNet:

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tenant1" status="">
    <fvCtx name="VRF1" />
    <fvCtx name="VRF2" />
    <cloudCtxProfile name="vpcl" status="">
      <cloudRsCtxProfileToRegion tDn="uni/cloudcomp/provp-azure/region-centralus" status=""/>
      <cloudRsToCtx tnFvCtxName="VRF1" />
      <cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status=""/>
      <cloudCidr name="cidr1" addr="192.0.2.0/16" primary="yes" status="">
        <cloudSubnet ip="192.0.3.0/24" usage="gateway" status="">
          <cloudRsZoneAttach status=""
            tDn="uni/cloudcomp/provp-azure/region-centralus/zone-default"/>
        </cloudSubnet>
      </cloudCidr>
      <cloudCidr name="cidr1" addr="193.0.2.0/16" primary="no" status="">
        <cloudSubnet ip="193.0.3.0/24" usage="" status="">
          <cloudRsSubnetToCtx tnFvCtxName="VRF2"/>
          <cloudRsZoneAttach status=""
            tDn="uni/cloudcomp/provp-azure/region-centralus/zone-default"/>
        </cloudSubnet>
      </cloudCidr>
    </cloudCtxProfile>
  </fvTenant>
</polUni>
```

Managing a Cloud Region Using the REST API

This section demonstrates how to manage a cloud region using the REST API.

To create a cloud region:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <cloudDomP name="default">
    <cloudProvP vendor="azure">
      <cloudRegion adminSt="managed" name="eastus"><cloudZone name="default"/></cloudRegion>
      <cloudRegion adminSt="managed" name="eastus2"><cloudZone name="default"/></cloudRegion>
      <cloudRegion adminSt="managed" name="westus"><cloudZone name="default"/></cloudRegion>
    </cloudProvP>
  </cloudDomP>
</polUni>
```

Creating a Filter Using the REST API

This section demonstrates how to create a filter using the REST API.

To create a filter:

```

https://<IP_Address>/api/node/mo/.xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="t15">
    <vzFilter name="rule1">
      <vzEntry etherT="ip" dToPort="22" prot="tcp" dFromPort="22" name="ssh"/>
      <vzEntry etherT="ip" prot="unspecified" name="any"/>
    </vzFilter>
    <vzFilter name="rule2">
      <vzEntry etherT="ip" dToPort="http" prot="tcp" dFromPort="http" name="http"/>
    </vzFilter>
    <vzFilter name="rule3">
      <vzEntry etherT="ip" dToPort="22" prot="tcp" dFromPort="22" name="ssh"/>
    </vzFilter>
    <vzFilter name='all_rule'>
      <vzEntry etherT="ip" prot="unspecified" name="any"/>
    </vzFilter>

    <vzBrCP name="c1">
      <vzSubj name="c1">
        <vzRsSubjFiltAtt tnVzFilterName="rule2"/>
        <vzRsSubjGraphAtt tnVnsAbsGraphName="c13_g1"/>
        <vzRsSubjFiltAtt tnVzFilterName="rule3"/>
        <vzRsSubjFiltAtt tnVzFilterName="all_rule"/>
      </vzSubj>
    </vzBrCP>

  </fvTenant>
</polUni>

```

Creating an Application Profile Using the REST API

This section demonstrates how to create an application profile using the REST API.

Before you begin

Create a tenant.

To create an application profile:

```

https://<IP_Address>/api/node/mo/.xml
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[<subscription id>]-vendor-azure" />

    <fvCtx name="ctx151"/>

    <cloudVpnGwPol name="VgwPol1"/>

```

```

    <cloudApp name="a1">

  </cloudApp>

</fvTenant>
</polUni>

```

Configuring Network Security Groups Using the REST API

This example demonstrates how to set the newer **NSG-per-subnet** configuration for your Cisco Cloud Network Controller using the REST API.

Before you begin

Review the information provided in [Security Groups, on page 45](#).

To set the NSG-per-subnet configuration for your Cisco Cloud Network Controller :

Example:

```

<polUni>
  <cloudDomP status="">
    <cloudProvP vendor="azure">
      <cloudProvResPolCont><cloudProvSGForSubnetP enableSGForSubnet="true"
status=""/></cloudProvResPolCont>
    </cloudProvP>
  </cloudDomP>
</polUni>

```

Creating an EPG Using the REST API

Use the procedures in this section to create an application EPG, an external EPG, or a service EPG using the REST API.

Creating a Cloud EPG Using the REST API

This example demonstrates how to create a cloud EPG using the REST API.

Before you begin

Create an application profile and a VRF.

To create a cloud EPG:

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->

```

```

<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[<subscription id>]-vendor-azure" />

    <fvCtx name="ctx151"/>

    <cloudVpnGwPol name="VgwPol1"/>
    <cloudApp name="a1">

      <cloudEPg name="epg1">
        <cloudRsCloudEPgCtx tnFvCtxName="ctx151"/>
        <cloudEPSelector matchExpression="custom:tag1=='value1'" name="selector-1"/>
      </cloudEPg>

    </cloudApp>
  </fvTenant>
</polUni>

```

Creating an External Cloud EPG Using the REST API

This example demonstrates how to create an external cloud EPG using the REST API.

Before you begin

Create an application profile and a VRF.

Step 1 To create an external cloud EPG:

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[<subscription id>]-vendor-azure" />
    <fvCtx name="ctx151"/>
    <cloudVpnGwPol name="VgwPol1"/>
    <cloudApp name="a1">
      <cloudExtEPg routeReachability="internet" name="extEpg-1">
        <fvRsCons tnVzBrCPName="extEpg-1"/>
        <cloudRsCloudEPgCtx tnFvCtxName="ctx151"/>
        <cloudExtEPSelector name="extSelector1" subnet="0.0.0.0/0"/>
      </cloudExtEPg>
    </cloudApp>
  </fvTenant>
</polUni>

```

Step 2 To create an external cloud EPG with type **site-external**:

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="infra">
    <cloudApp name="a1">

```

```

    <cloudExtEPg routeReachability="site-ext" name="extEpg-1">
      <fvRsCons tnVzBrCPName="extEpg-1"/>
      <cloudRsCloudEPgCtx tnFvCtxName="overlay-2"/>
      <cloudExtEPSelector name="extSelector1" subnet="10.100.0.0/16"/>
    </cloudExtEPg>
  </cloudApp>
</fvTenant>
</polUni>

```

Creating a Service EPG Using the REST API

This example demonstrates how to create a service EPG using the REST API.

Before you begin

- Review the information in [Cloud Service Endpoint Groups, on page 38](#).
- Create an application profile and a VRF.

Step 1 To create a service EPG with a deployment type of Cloud Native:

Example:

```

<cloudSvcEPg name="Storage" type="Azure-Storage" accessType="Private" deploymentType="CloudNative">
  <cloudPrivateLinkLabel name="ProductionSubnets"/>
  <cloudRsCloudEPgCtx tnFvCtxName="HUB-SERVICES-VRF"/>
  <cloudSvcEPSelector matchExpression="ResourceName=='StorageAcct1'" name="selector-1"/>
  <cloudSvcEPSelector matchExpression="custom:Tag=='ProdStorage'" name="selector-2"/>
</cloudSvcEPg>

```

Step 2 To create a service EPG with a deployment type of Cloud Native Managed:

Example:

```

<cloudSvcEPg name="APIM" type="Azure-ApiManagement" accessType="Private"
deploymentType="CloudNativeManaged" status="">
  <cloudRsCloudEPgCtx tnFvCtxName="infra-SvcCtx" />
  <fvRsCons tnVzBrCPName="infra-APIM-Mock"/>
  <fvRsProv tnVzBrCPName="infra-managedAPIM" status=""/>
  <cloudSvcEPSelector matchExpression="IP=='10.21.52.0/28'" name="sel1" status=""/>
</cloudSvcEPg>

```

Step 3 To create a service EPG with a deployment type of Third-Party:

Example:

```

<cloudSvcEPg name="SaaS-Hub" type="Custom" accessType="Private" deploymentType="Third-party" status="">
  <cloudRsCloudEPgCtx tnFvCtxName="infra-SvcCtx" status=""/>
  <cloudSvcEPSelector
matchExpression="URL=='saassvcepg.286b0377-a9b7-40d7-a94f-67abe03ce5f4.centralus.azure.privatelinkservice'"
name="s1" status=""/>
  <cloudPrivateLinkLabel name="saas-hub" status=""/>

```

```
<fvRsProv tnVzBrCPName="SaaS-Hub" status="" />
</cloudSvcEPg>
```

Creating a Cloud Template Using the REST API

This section demonstrates how to create a cloud template using the REST API. For more information about cloud templates, see [About the Cloud Template, on page 52](#).

The REST API will change depending on the type of Licensing model selected. The license type of the Cisco Catalyst 8000V is captured by the property `routerThroughput` in the `cloudtemplateProfile` managed object.

If the `routerThroughput` value belongs to **T0/T1/T2/T3** then **BYOL** Cisco Catalyst 8000V is deployed on Cisco Cloud Network Controller. If `routerThroughput` value is **PAYG** then **PAYG** Cisco Catalyst 8000V is deployed on Cisco Cloud Network Controller.

Before you begin

Step 1 To create a cloud template post to deploy a **BYOL** Cisco Catalyst 8000V:

```
<polUni>
  <fvTenant name="infra">
    <cloudtemplateInfraNetwork name="default" numRemoteSiteSubnetPool="2" numRoutersPerRegion="2"
status="" vrfName="overlay-1">
      <cloudtemplateProfile name="default" routerPassword="cisco123" routerUsername="cisco"
routerThroughput="250M" routerLicenseToken="thisismyscrtoken" />
        </cloudtemplateProfile>
      <cloudtemplateExtSubnetPool subnetpool="10.20.0.0/16"/>

      <cloudtemplateIntNetwork name="default">
        <cloudRegionName provider="azure" region="westus"/>
        <cloudRegionName provider="azure" region="westus2"/>
      </cloudtemplateIntNetwork>

      <cloudtemplateExtNetwork name="default">
        <cloudRegionName provider="azure" region="westus2"/>

      <cloudtemplateVpnNetwork name="default">

        <cloudtemplateIpSecTunnel peeraddr="23.2.1.1/32" />
        <cloudtemplateIpSecTunnel peeraddr="23.0.1.1/32" />
        <cloudtemplateIpSecTunnel peeraddr="23.1.1.1/32" />

        <cloudtemplateOspf area="0.0.0.1"/>

      </cloudtemplateVpnNetwork>

    </cloudtemplateExtNetwork>
  </cloudtemplateInfraNetwork>
</fvTenant>
</polUni>
```

Step 2 To create a cloud template post to deploy a **PAYG** Cisco Catalyst 8000V:

```
<polUni>
  <fvTenant name="infra">
    <cloudtemplateInfraNetwork name="default" numRemoteSiteSubnetPool="2" numRoutersPerRegion="2"
status="" vrfName="overlay-1">
```



```

    <cloudtemplateProfile name="default" routerPassword="cisco123" routerUsername="cisco"
routerThroughput="PAYG" vmType="DS2V2" />
  </cloudtemplateProfile>
cloudtemplateProfile name="default" routerPassword="cisco123" routerUsername="cisco"
routerThroughput="250M" routerLicenseToken="thisismyscrtoken" />
  </cloudtemplateProfile>
  <cloudtemplateExtSubnetPool subnetpool="10.20.0.0/16"/>

  <cloudtemplateIntNetwork name="default">
    <cloudRegionName provider="azure" region="westus"/>
    <cloudRegionName provider="azure" region="westus2"/>
  </cloudtemplateIntNetwork>

  <cloudtemplateExtNetwork name="default">
    <cloudRegionName provider="azure" region="westus2"/>

  <cloudtemplateVpnNetwork name="default">

    <cloudtemplateIpSecTunnel peeraddr="23.2.1.1/32" />
    <cloudtemplateIpSecTunnel peeraddr="23.0.1.1/32" />
    <cloudtemplateIpSecTunnel peeraddr="23.1.1.1/32" />

    <cloudtemplateOspf area="0.0.0.1"/>

  </cloudtemplateVpnNetwork>

  </cloudtemplateExtNetwork>
</cloudtemplateInfraNetwork>
</fvTenant>
</polUni>

```

On selecting PAYG throughput, the user must also select the **vmName** from a list of vmNames which is created by Cisco Cloud Network Controller and represented by the managed object `vmType`.

The following table lists the vmNamesTypes that are indicated by the property `vmType` in the `cloudtemplateProfile`.

VmName on Azure	Memory	vCPUs	NetworkBw
DS3V2	14GiB	4	Up to 3 Gigabit
DS4V2	28GiB	8	Up to 6 Gigabit
F16SV2	32GiB	16	Up to 12.5 Gigabit
F32SV2	64GiB	32	Up to 16 Gigabit

Configuring VRF Leak Routes Using the REST API

Before you begin

Review the information provided in [Route Leaking Between Internal VRFs, on page 9](#) and [Global Inter-VRF Route Leak Policy, on page 9](#) before proceeding with the instructions in this section.

Step 1 Enter a post similar to the following to enable or disable contract-based routing.

```
<fvTenant name="infra">
  <cloudVrfRouteLeakPol name="default" allowContractBasedRouting="true"/>
</fvTenant>
```

Where the `allowContractBasedRouting` field has either of the following settings:

- **true**: Indicates that routes are leaked based on contracts in the absence of route maps. When enabled, contracts drive routing when route maps are not configured. When route maps exist, route maps always drives routing.
- **false**: Default setting. Indicates that routes are not leaked based on contracts, and are leaked based on route maps instead.

Step 2 Enter a post similar to the following to use the `leakInternalPrefix` field to configure route leaking for all cloud CIDRs associated with the VRFs.

```
<fvTenant name="t1">
  <fvCtx name="v1">
    <leakRoutes>
      <leakInternalPrefix ip="0.0.0.0/0" le="32">
        <leakTo tenantName="t2" ctxName="v2" scope="public"/>
      </leakInternalPrefix>
    </leakRoutes>
  </fvCtx>
</fvTenant>

<fvTenant name="t2">
  <fvCtx name="v2">
    <leakRoutes>
      <leakInternalPrefix ip="0.0.0.0/0" le="32">
        <leakTo tenantName="t1" ctxName="v1" scope="public"/>
      </leakInternalPrefix>
    </leakRoutes>
  </fvCtx>
</fvTenant>
```

Step 3 Enter a post similar to the following to use the `leakInternalSubnet` field to leak specific routes between a pair of VRFs.

```
<fvTenant name="anyTenant" status="">
  <fvCtx name="VRF1" >
    <leakRoutes status="">
      <leakInternalSubnet ip="110.110.1.0/24" >
        <leakTo ctxName="VRF2" scope="public" tenantName=" anyTenant " />
      </leakInternalSubnet>
    </leakRoutes>
  </fvCtx>
  <fvCtx name="VRF2" status="" >
    <leakRoutes status="">
      <leakInternalSubnet ip="110.110.2.0/24" >
        <leakTo ctxName="VRF1" scope="public" tenantName=" anyTenant " />
      </leakInternalSubnet>
    </leakRoutes>
  </fvCtx>
</fvTenant>
```

Configuring the Source Interface Selection for Tunnels Using the REST API

Before you begin

Review the information provided in [Source Interface Selection for Tunnels](#), on page 11 before proceeding with these instructions.

Enter a post similar to the following to configure the source interface selection for tunnels.

```
<cloudtemplateInfraNetwork name="default" vrfName="overlay-1">
  <cloudtemplateProfile name="defaultxyz" routerUsername="james" routerPassword="bond@7" />

  <cloudtemplateIpSecTunnelSubnetPool subnetpool="10.20.0.0/16" poolname="pool1" />

  <cloudtemplateIntNetwork name="default">
    <cloudRegionName provider="aws" region="us-west-1"/>
    <cloudRegionName provider="aws" region="us-west-2"/>
  </cloudtemplateIntNetwork>

  <cloudtemplateExtNetwork name="something" vrfName="xyz" >
    <cloudRegionName provider="aws" region="us-west-2"/>
    <cloudtemplateVpnNetwork name="default">
      <cloudtemplateIpSecTunnel peeraddr="23.2.1.1/32" poolname="" presharedkey="abcd"
ikeVersion="v1|v2">
        <cloudtemplateIpSecTunnelSourceInterface sourceInterfaceId="2" />
      </cloudtemplateIpSecTunnel>
    </cloudtemplateVpnNetwork>
  </cloudtemplateExtNetwork>
</cloudtemplateInfraNetwork>
```

Defining Global Cloud Resource Naming Rules or Overriding Specific Object's Name

This section provides an example REST API POST you can use to configure a global policy for naming your cloud resources or override a specific cloud resource's name.



Note

To ensure that any custom naming conventions can be supported, cloud resource names can be defined on a per-object basis. These explicit name overrides are not available in the Cisco Cloud Network Controller GUI and can be done using REST API only. We recommend using the global cloud resource naming policy to define the names. Explicit name overrides should be used only when naming requirements cannot be met using the global naming policy.

Step 1 To create Hub Resource Naming Rules:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="infra">
```

```

<cloudtemplateInfraNetwork name="default" numRemoteSiteSubnetPool="2"
numRoutersPerRegion="2" status="" vrfName="overlay-1">
  <cloudtemplateIntNetwork name="default">
    <cloudRegionName provider="azure" region="west's" status="">
      <cloudtemplateRegionNameCustomization ctxProfileName="infra-vnet"
resourceGroupName="infra-rh" subnetNamePrefix="snet-" />
    </cloudRegionName>
  </cloudtemplateIntNetwork>
</cloudtemplateInfraNetwork>
</fvTenant>
</polUni>

```

Step 2 To create Cloud Resource Naming Rules:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <cloudDomP name="default">
    <cloudNaming
      azResourceGroup="${tenant}-network-${ctx}-${region}-rg"
      azVirtualNetwork="${tenant}-${ctxprofile}-vnet"
      azSubnet="${tenant}-${ctxprofile}-snet-${subnet}"
      azNetworkSecurityGroup="${app}-${epg}-nsg"
      azApplicationSecurityGroup="${app}-${epg}-asg"
      azNetworkSecurityGroupRule="${contract}--${priority}"
      internetApplicationBalancer="agw-e-${device}"
      internalApplicationBalancer="agw-i-${device}"
      internetNetworkBalancer="lbe-${device}"
      internalNetworkBalancer="lbi-${device}"
      l4L7DeviceApplicationSecurityGroup="${deviceInterfaceDn}"
      reviewed="yes" />
    </cloudDomP>
  </polUni>

```

Step 3 To override an Azure cloud resource name corresponding to a specific Cisco Cloud Network Controller object:

You can use the same variables (for example, `${tenant}`) when specifying the custom name using the API.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<fvTenant name="ExampleCorp" status="">
  <fvRsCloudAccount status="" tDn="uni/tn-infra/act-[<i>infra-subscription</i>]-vendor-azure"/>
  <fvCtx name="VRF1"/>
  <cloudApp name="App1">
    <cloudEPg name="Db" azNetworkSecurityGroup="db-nsg" azApplicationSecurityGroup="db-asg-${region}">
      <cloudRsCloudEPgCtx tnFvCtxName="VRF1"/>
      <cloudEPSelector matchExpression="custom:EPG=='db'" name="100"/>
    </cloudEPg>
  </cloudApp>
  <cloudCtxProfile name="c02" azResourceGroup="custom-tc-rg1" azVirtualNetwork="vnet1">
    <cloudRsCtxProfileToRegion tDn="uni/clouddomp/provp-azure/region-westus"/>
    <cloudRsToCtx tnFvCtxName="VRF1"/>
    <cloudCidr addr="10.20.20.0/24" name="cidr1" primary="yes" status="">
      <cloudSubnet ip="10.20.20.0/24" name="subnet1" azSubnet="s1" status="">
        <cloudRsZoneAttach status="" tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
      </cloudSubnet>
    </cloudCidr>
  </cloudCtxProfile>
</fvTenant>

```

Step 4 To override a Layer 4 to Layer 7 Azure cloud resource name corresponding to a specific Cisco Cloud Network Controller object:

You can use the same variables (for example, `${tenant}`) when specifying the custom name using the API.

Override policy for load balancer:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<fvTenant>
  <cloudLB name="ALB" type="application" scheme="internet" size="small" instanceCount="2" status=""
nativeLBName="ALB" >
    <cloudRsLDevToCloudSubnet
      tDn="uni/tn-{{tenantName}}/ctxprofile-cl/cidr-[31.10.0.0/16]/subnet-[31.10.80.0/24]" status="" />
    </cloudLB>
  </fvTenant>
```

Override policy for device ASG:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<fvTenant>
  <cloudLDev name="{{FWName}}" status="" 14L7DeviceApplicationSecurityGroup="Group1" >
    <cloudRsLDevToCtx tDn="uni/tn-{{tenantName}}/ctx-VRFl" status="" />
    </cloudLIf>
  </cloudLDev>
</fvTenant>
```



CHAPTER 5

Viewing System Details

- [Monitoring VM Host Metrics, on page 163](#)
- [Viewing Application Management Details, on page 167](#)
- [Viewing Cloud Resource Details, on page 168](#)
- [Viewing Operations Details, on page 169](#)
- [Viewing Infrastructure Details, on page 171](#)
- [Viewing Administrative Details, on page 172](#)
- [Viewing Health Details Using the Cisco Cloud Network Controller GUI, on page 174](#)

Monitoring VM Host Metrics

Monitoring metrics for the VM host is supported where the Cisco Cloud Network Controller is deployed using the Prometheus Node Exporter. The Prometheus Node Exporter provides visibility to a wide variety of hardware and kernel-related metrics, where it collects technical information from Linux nodes, such as CPU, disk, and memory statistics. For overview information on the Prometheus Node Exporter, see:

<https://prometheus.io/docs/introduction/overview/>

If your Cisco Cloud Network Controller is running on release 25.0(1) or later, the Prometheus Node Exporter is automatically available by default.

Guidelines and Limitations

HTTP is not supported for monitoring metrics using the Prometheus Node Exporter. Only HTTPS is supported for monitoring metrics using the Prometheus Node Exporter.

Monitoring VM Host Metrics Using the GUI

These procedures describe how to enable the Prometheus Node Exporter to monitor VM host metrics using the GUI.

-
- Step 1** In the Cisco Cloud Network Controller GUI, navigate to **Infrastructure > System Configuration**, then click on the **Management Access** tab.
- Step 2** In the **HTTPS** area to the right of the window, note the entry in the **Node Exporter** field.

The screenshot shows the Cisco Cloud Network Controller GUI. The main content area is titled "System Configuration" and has tabs for "General", "Management Access", "Cloud Resource Naming Rules", "Controllers", and "Event Analytics". The "Management Access" tab is selected. On the left, there is a navigation menu with options: Dashboard, Topology, Cloud Resources, Application Management, Operations, Infrastructure, and Administrative. The main configuration area is divided into three sections: SSH, SSH Access Via Web, and Telnet. The SSH section is "Enabled" and shows various settings like Password Authentication State (Disabled), Ciphers, KEX Algorithms, and MACs. The SSH Access Via Web section is "Disabled" and shows a port of 4200. The Telnet section is "Disabled" and shows a port of 23. The HTTPS section is "Disabled" and shows various settings like Admin State (Enabled), Port (443), Allow Credentials (Disabled), DH Parameters (none), Client Certificate Authentication State (Disabled), and SSL Ciphers. A red box highlights the "Node Exporter Disabled" field in the HTTPS section, with the text "Allow Origins" below it.

- **Enabled:** The Prometheus Node Exporter has already been enabled. You do not have to continue with these instructions in that case.
- **Disabled:** The Prometheus Node Exporter is not enabled yet. Proceed with these instructions to enable the Prometheus Node Exporter.

Step 3 Click the pencil icon in the **HTTPS** area to edit the HTTPS settings.

The **HTTPS Settings** window appears.

Step 4 Locate the **Node Exporter** field and click **Enable**.

HTTPS Settings

Admin State
 Enable

Node Exporter
 Enable

Port
443

Allow Origins

Allow Credentials
 Enable

SSL Protocols
 TLSv1 TLSv1.1 TLSv1.2

DH Parameter
None

Admin Key Ring
default

Client Certificate Authentication State
 Enable

Client Certificate Trust Point
[Select Client Certificate Trust Point](#)

SSL Ciphers

ID *	State	
CHACHA20	Enabled	

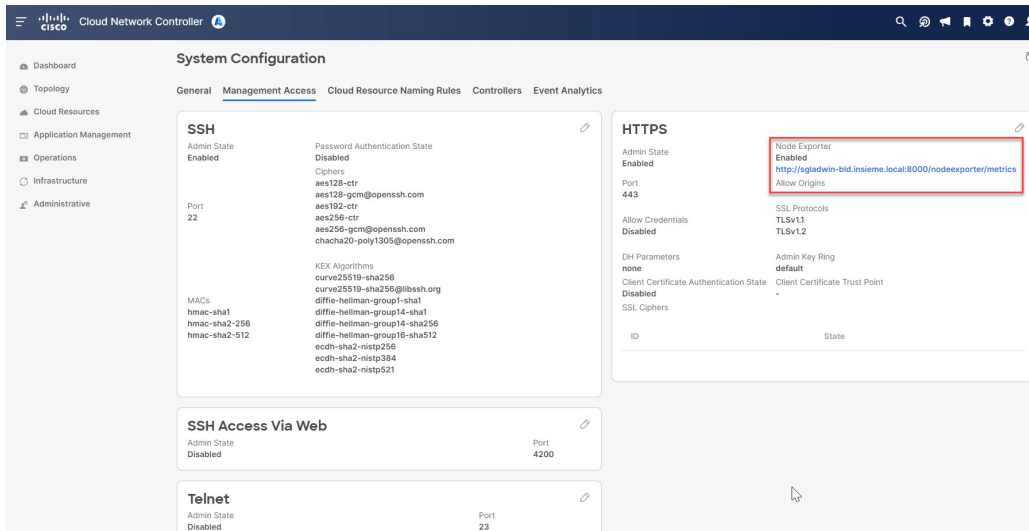
Cancel Save

A warning message appears, telling you that saving these settings will restart the web service, and that it will take a moment for it to resume responding to requests. Click **OK** to confirm these changes.

Step 5 At the bottom of the window, click **Save**.

You are returned to the **System Configuration/Management Access** window. The web service reboots and comes back online in a few seconds.

Step 6 In the **HTTPS** area to the right of the window, verify that the entry in the **Node Exporter** field is set to **Enabled**. This verifies that the Prometheus Node Exporter is enabled.



Step 7 Click the link under the **Enabled** text in the **Node Exporter** area.

Another tab in your browser appears, showing the metrics for the VM host where the Cisco Cloud Network Controller is deployed.

Monitoring VM Host Metrics Using the REST API

These procedures describe how to enable the Prometheus Node Exporter to monitor VM host metrics using the REST API.

Step 1 To determine if the Prometheus Node Exporter is enabled or not, send the following GET call:

```
GET https://<cloud-network-controller-ip-address>/api/mo/uni/fabric/comm-default/https.xml
```

Locate the `nodeExporter` field to determine if it is set to `enabled` or `disabled`.

Step 2 To monitor VM host metrics, send the following post to enable the Prometheus Node Exporter:

```
POST https://<cloud-network-controller-ip-address>/api/mo/uni/fabric/comm-default/https.xml
<commHttps nodeExporter="enabled" />
```

The metrics are displayed for the VM host where the Cisco Cloud Network Controller is deployed.

Step 3 To view the metrics using REST API, send the following GET call:

```
GET https://<cloud-network-controller-ip-address>/nodeexporter/metrics
```

Step 4 To disable the Prometheus Node Exporter, send the following post:

```
POST https://<cloud-network-controller-ip-address>/api/mo/uni/fabric/comm-default/https.xml
```

```
<commHttps nodeExporter="disabled" />
```

Viewing Application Management Details

This section explains how to view application management details using the Cisco Cloud Network Controller GUI. The application management details include the information of a specific tenant, application profile, EPG, contract, filter, VRF, service, or cloud context profile.

Step 1 From the **Navigation** menu, choose the **Application Management** tab.

When the **Application Management** tab expands, a list of subtab options appear. See the *Application Management Options* table for more information.

Table 31: Application Management Subtabs

Subtab Name	Description
Tenants	Displays tenants as rows in a summary table.
Application Profiles	Displays application profiles as rows in a summary table.
EPGs	Displays an EPGs as rows in a summary table.
Contracts	Displays a contracts as rows in a summary table.
Filters	Displays filters as rows in a summary table.
VRFs	Displays VRFs in a summary table.
Services	Contains the following two subtabs and information: <ul style="list-style-type: none"> • Devices—Displays the devices as rows in a summary table. • Service Graphs—Displays service graphs as rows in a summary table.
Cloud Context Profiles	Displays cloud context profiles as rows in a summary table. Beginning with 26.0(3), the VRF column will display the VRF name instead of the VRF count.

Step 2 Click the tab that represents the component with the details you want to view.

A summary table appears with items as rows in the table. For example, if you chose the **Tenants** subtab, a list of tenants appear as rows in a summary table

You can filter the rows by clicking the *Filter by Attributes* bar. Choose the attribute, operator and filter-value. For example, for filtering based on a tenant, choose Name == T1 (where T1 is the name of a tenant).

Step 3 To view a summary pane, click the row that represents the specific component you want to view.

Step 4 For more information, double-click the summary table row that represents the specific component you want to view.

A new dialog box appears over the **work** pane with any of the following tabs:

Note The tabs that appear differ between components and configurations.

- **Overview**—Provides a general overview of cloud resources, configuration relationships, and settings of the component.
- **Topology**—Provides visual relationship between an object and other related objects. The chosen object is displayed at the center.
- **Cloud Resources**—Contains a list of subtabs that display the cloud resource information related to the component.
- **Application Management**—Contains a list of subtabs that display the ACI relation information related to the component.
- **Statistics**—Enables you to view statistics based on a chosen sampling interval and statistics type. The **Statistics** tab may contain subtabs, depending on the component you are viewing.
- **Event Analytics**—Contains a list of subtabs that display faults, events, and audit logs.

Note The dialog box that appears over the **work** pane contains an **edit** button in the top-right corner between the **refresh** button and the **Actions** button. When clicked, the **edit** button enables you to edit the chosen component.

Viewing Cloud Resource Details

This section explains how to view cloud resource details using the Cisco Cloud Network Controller GUI. The cloud resource details include the information about a specific region, VNET, router, security group (application security group/network security group), endpoint, VM, and cloud service.

Search based on *Cloud Tag* attribute is supported for the **Endpoints** subtab.

Step 1 From the **Navigation** menu, choose the **Cloud Resources** tab.

When the **Cloud Resources** tab expands, a list of subtab options appear. See the *Cloud Resource Options* table for more information.

Table 32: Cloud Resource Subtabs

Subtab Name	Description
Regions	Displays regions as rows in a summary table.
Virtual Networks	Displays VNETs as rows in a summary table.
Routers	Displays routers as rows in a summary table.
Security Groups	Displays security groups as rows in a summary table.

Subtab Name	Description
Endpoints	Displays endpoints as rows in a summary table.
Virtual Machines	Displays the VMs as rows in a summary table.
Cloud Services	Contains the following subtabs: <ul style="list-style-type: none"> • Cloud Service Tab—Displays cloud services as rows in a summary table. • Target Groups Tab—Displays target groups as rows in a summary table.

Step 2 Click the tab that represents the component with the details you want to view.

A summary table appears with items as rows in the table. For example, if you chose the **Endpoints** subtab, a list of endpoints appear as rows in a summary table.

You can filter the rows by selecting an attribute from the drop-down menu when you click the *Filter by attributes* bar. The attributes displayed in the drop-down menu depend on the selected subtab.

For the **Endpoints** subtab, you can narrow down the search based on a cloud tag, by entering a **key** or **value** term. If you want to search based on both terms, click the (+) displayed as a superscript to the **key** or **value** term (depending on which was entered first). Cloud tag filters cannot be edited. To modify a search, first delete the filters, and then enter the desired **key** or **value** term again. Search based on multiple cloud tag filters is supported.

Step 3 To view a summary pane, click the row that represents the specific component you want to view.

Step 4 For more information, double-click the summary table row that represents the specific component you want to view.

A new dialog box appears over the **work** pane with any of the following tabs:

Note The tabs that appear differ between components and configurations.

- **Overview**—Provides a general overview of cloud resources, configuration relationships, and settings of the component. The cloud tags associated with endpoints are also displayed.
- **Cloud Resources**—Contains a list of subtabs that display the cloud resource information related to the component.
- **Application Management**—Contains a list of subtabs that display the ACI relation information related to the component.
- **Statistics**—Enables you to view statistics based on a chosen sampling interval and statistics type. The **Statistics** tab may contain subtabs, depending on the component you are viewing.
- **Event Analytics**—Contains a list of subtabs that display faults, events, and audit logs.

Viewing Operations Details

This section explains how to view operations details using the Cisco Cloud Network Controller GUI. The operations details include the information of a specific fault, event, audit log, active sessions, backup and restore policies, tech support policies, firmware management, scheduler policies, and remote locations.

Step 1 From the **Navigation** menu, choose the **Operations** tab.

When the **Operations** tab expands, a list of subtab options appear. See the *Operations Options* table for more information.

Table 33: Operations Subtabs

Subtab Name	Description
Event Analytics	Contains the following subtabs: <ul style="list-style-type: none"> • Faults Tab—Displays faults as rows in a summary table. • Fault Records Tab—Displays fault records as rows in a summary table. • Events Tab—Displays events as rows in a summary table. • Audit Logs Tab—Displays audit logs as rows in a summary table.
Active Sessions	Displays a list of active users who are logged into Cisco Cloud Network Controller.
Backup & Restore	Contains the following subtabs: <ul style="list-style-type: none"> • Backups Tab—Displays backup as rows in a summary table. • Backup Policies Tab—Displays backup policies as rows in a summary table. • Job Status Tab—Displays the job status as rows in a summary table. • Event Analytics Tab—Contains the following subtabs: <ul style="list-style-type: none"> • Faults Tab—Displays faults as rows in a summary table. • Events Tab—Displays events as rows in a summary table. • Audit Logs Tab—Displays audit logs as rows in a summary table.
Tech Support	Contains the following subtabs: <ul style="list-style-type: none"> • Tech Support Tab—Displays tech support policies as rows in a summary table. • Core Logs Tab—Displays core logs as rows in a summary table.

Subtab Name	Description
Firmware Management	<p>Contains the following subtabs:</p> <ul style="list-style-type: none"> • General Tab—Displays general firmware management information, such as Current Firmware Version, Upgrade Status. • Images Tab—Displays a list of images. • Event Analytics Tab—Contains the following subtabs: <ul style="list-style-type: none"> • Faults Tab—Displays faults as rows in a summary table. • Events Tab—Displays events as rows in a summary table. • Audit Logs Tab—Displays audit logs as rows in a summary table.
Schedulers	Displays scheduler policies as rows in a summary table.
Remote Locations	Displays remote locations as rows in a summary table.

Step 2 Click the tab that represents the component you want to view.

A summary table appears with items as rows in the table. For example, if you chose the **Active Sessions** subtab, a list of active sessions appear as rows in a summary table.

You can filter the rows by clicking the *Filter by Attributes* bar. Choose the attribute, operator and filter-value. For example, for filtering based on a username, choose username == user1 (where user1 is a user logged into Cisco Cloud Network Controller).

Step 3 To view a summary pane, click the row that represents the specific component you want to view.

Step 4 For more information, double-click the summary table row that represents the specific item you want to view.

A new dialog box appears over the **work** pane that displays additional information about the item you chose from the summary table.

Viewing Infrastructure Details

This section explains how to view infrastructure details using the Cisco Cloud Network Controller GUI. The infrastructure details include information about system configuration, inter-region connectivity, and external connectivity.

Step 1 From the **Navigation** menu, choose the **Infrastructure** tab.

When the **Infrastructure** tab expands, a list of subtab options appear. See the *Infrastructure Options* table for more information.

Table 34: Infrastructure Subtabs

Subtab Name	Description
System Configuration	Displays General system configuration information, Management Access information, Controllers , Cloud Resource Naming Rules , and Event Analytics .
Inter-Region Connectivity	Displays one pane with a map that contains the inter-region connectivity view and additional panes for each region.
Inter-Site Connectivity	Displays one pane with a map that contains the inter-site connectivity view and additional panes for each site.

Step 2 Click the tab that represents the component with the details you want to view.

Viewing Administrative Details

This section explains how to view administrative details using the Cisco Cloud Network Controller GUI. The administrative details include the information about authentication, security, users, and smart licensing..

Step 1 From the **Navigation** menu, choose the **Administrative** tab.

When the **Administrative** tab expands, a list of subtab options appear. See the *Administrative Options* table for more information.

Table 35: Administrative Subtabs

Subtab Name	Description
Authentication	<p>Displays the Authentication Default Settings, Login Domains, Providers and Event Analytics subtabs, which contain the information described below:</p> <ul style="list-style-type: none"> • Authentication Default Settings Tab—Displays settings information. • Login Domains Tab—Displays the login domains as rows in a summary table. • Providers Tab—Displays the providers as rows in a summary table. • Event Analytics Tab—Displays the Faults, Events, and Audit Logs subtabs, each with the corresponding information displayed as rows in a summary table.

Subtab Name	Description
Security	Contains the following list of subtabs: <ul style="list-style-type: none"> • Security Default Settings Tab—Enables you to view the default security settings information. • Security Domains Tab—Enables you to view security domain information in a summary table. • Roles Tab—Enables you to view the role information in a summary table. • RBAC Rules Tab—Enables you to view RBAC rule information in a summary table. • Certificate Authorities Tab—Enables you to view the certificate authority information in a summary table. • Key Rings Tab—Enables you to view key ring information in a summary table. • User Activity Tab—Enables you to view user activity.
Users	Contains the following subtabs: <ul style="list-style-type: none"> • Local Tab—Displays local users as rows in a summary table. • Remote Tab—Displays remote users as rows in a summary table.
Smart Licensing	Contains the following subtabs: <ul style="list-style-type: none"> • General Tab—Displays the licenses as rows in a summary table. • CCRs Tab—Displays CCRs as rows in a summary table. • Faults Tab—Displays faults as rows in a summary table.

Step 2 Click the tab that represents the component you want to view.

For some options, a summary table appears with items as rows in the table (For example, if you choose the **Users** tab, a list of users appear as rows in a summary table). To view a summary pane, click the row that represents the specific component you want to view. To view more information, double-click the summary table row that represents the specific item you want to view. A new dialog box appears over the **work** pane that displays additional information about the item you chose from the summary table.

You can filter the rows by clicking the *Filter by Attributes* bar. Choose the attribute, operator and filter-value. For example, for filtering based on a user, choose User ID == admin (where admin is a user ID.).

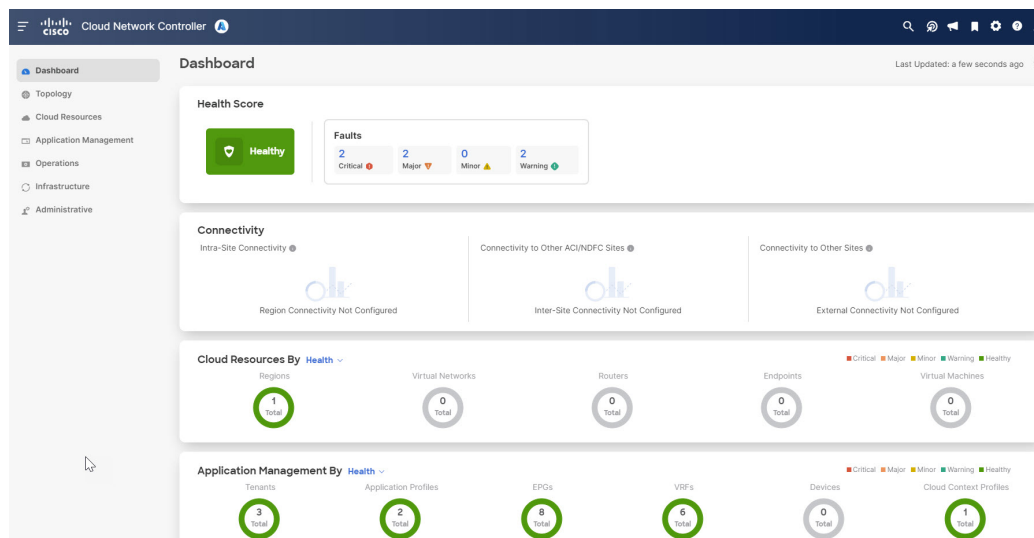
Viewing Health Details Using the Cisco Cloud Network Controller GUI

This section explains how to view health details using the Cisco Cloud Network Controller GUI. You can view health details for any object that you can see in the Cloud Resources area in the Cisco Cloud Network Controller GUI, such as the following:

- Regions
- Availability Zones (for AWS cloud sites)
- VPCs (for AWS cloud sites)
- VNETs (for Azure cloud sites)
- Routers
- Security Groups
- Endpoints
- Instances
- Cloud Services

Step 1 From the **Navigation** menu, choose the **Dashboard** tab.

The **Dashboard** window for the Cisco Cloud Network Controller system appears. From this window, you can view the overall health status of your system.



Step 2 Click within the Fault Summary area in the **Dashboard** window.

The **Event Analytics** window appears, showing more detailed information for the specific fault level that you clicked. The following screen shows an example **Event Analytics** window for the faults listed with critical severity.

The screenshot shows the Cisco Cloud Network Controller GUI with the 'Event Analytics' window open. The 'Severity' filter is set to 'Critical'. The table displays two critical events:

Acked	Severity	Code	Affected object	Description	Lifecycle	Creation Time
<input type="checkbox"/>	Critical	F0104	topology/pod-1/node-1/sys/cagr-(po1)	Bond Interface po1 on node 1 of fabric mininet with hostname capc1 is now down	raised	Feb 06 2023 10:29:28am +05:30
<input type="checkbox"/>	Critical	F0104	topology/pod-1/node-1/sys/cagr-(po1)	Bond Interface po1 on node 1 of fabric mininet with hostname capc1 is now down	raised	Feb 06 2023 10:29:28am +05:30

Step 3 Click the **X** next to the Severity level to display Event Analytics information for all faults.

The information provided in the **Event Analytics** window changes to show the events with critical, major, and warning levels of severity.

The screenshot shows the Cisco Cloud Network Controller GUI with the 'Event Analytics' window open. The 'Severity' filter is set to 'All'. The table displays six events of various severity levels:

Acked	Severity	Code	Affected object	Description	Lifecycle	Creation Time
<input type="checkbox"/>	Critical	F0104	topology/pod-1/node-1/sys/cagr-(po1)	Bond Interface po1 on node 1 of fabric mininet with hostname capc1 is now down	raised	Feb 06 2023 10:29:28am +05:30
<input type="checkbox"/>	Critical	F0104	topology/pod-1/node-1/sys/cagr-(po1)	Bond Interface po1 on node 1 of fabric mininet with hostname capc1 is now down	raised	Feb 06 2023 10:29:28am +05:30
<input type="checkbox"/>	Major	F0101	topology/pod-1/node-1/sys/chp-[dev/vdb]-f-[dev/vdb]	Storage unit /dev/vdb on node 1 with hostname capc1 has failed.	raised	Feb 06 2023 10:29:28am +05:30
<input type="checkbox"/>	Major	F0101	topology/pod-1/node-1/sys/chp-[dev/vda]-f-[dev/vda]	Storage unit /dev/vda on node 1 with hostname capc1 has failed.	raised	Feb 06 2023 10:29:28am +05:30
<input type="checkbox"/>	Warning	F4007	topology/pod-1/node-1/local/svc-policylist-id-1/un/cloudomp/cloudnaming	Resources will not be deployed to cloud until the naming rules have been reviewed and accepted.	raised	Feb 06 2023 10:25:47am +05:30
<input type="checkbox"/>	Warning	F3057	licensecont/manager	APIC smart licensing network settings are not configured. Navigate to System -> Smart Licensing window to configure network settings.	raised	Feb 06 2023 10:29:30am +05:30

Step 4 From the **Navigation** menu, choose the **Cloud Resources** tab.

When the **Cloud Resources** tab expands, a list of subtab options appear. See the *Administrative Options* table for more information.

Step 5 Choose any item under the **Cloud Resources** tab to display health information for that component.

For example, the following figure shows health information that might be displayed when you click on **Cloud Resources** > **Regions**, then you select a specific region.

The screenshot shows the Cisco Cloud Network Controller GUI. The main content area is titled 'Regions' and contains a table of regions. The table has columns for Health, Name, Admin State, Tenants, EPOs, and Virtual Networks. A 'Healthy' status indicator is visible in the top right corner. The table lists 14 regions, all of which are 'Healthy'.

Health	Name	Admin State	Tenants	EPOs	Virtual Networks
Healthy	Central US	managed	N/A	N/A	N/A
Healthy	South India	unmanaged	N/A	N/A	N/A
Healthy	UAE North	unmanaged	N/A	N/A	N/A
Healthy	UK West	unmanaged	N/A	N/A	N/A
Healthy	West Central US	unmanaged	N/A	N/A	N/A
Healthy	South Central US	unmanaged	N/A	N/A	N/A
Healthy	South Africa North	unmanaged	N/A	N/A	N/A
Healthy	North Europe	unmanaged	N/A	N/A	N/A
Healthy	North Central US	unmanaged	N/A	N/A	N/A
Healthy	Jio India West	unmanaged	N/A	N/A	N/A
Healthy	Australia Southeast	unmanaged	N/A	N/A	N/A
Healthy	Australia Central 2	unmanaged	N/A	N/A	N/A
Healthy	Australia Central	unmanaged	N/A	N/A	N/A
Healthy	Norway East	unmanaged	N/A	N/A	N/A

The 'Settings' panel on the right shows 'Admin State' as 'Managed'.



CHAPTER 6

Deploying Layer 4 to Layer 7 Services

- [Overview, on page 177](#)
- [Example Use Cases, on page 188](#)
- [Example Use Cases for Service Graphs with Cloud Native and Third-Party Services, on page 203](#)
- [Guidelines and Limitations for Redirect, on page 222](#)
- [Adding a New CIDR to a Secondary VRF Using the Cisco Cloud Network Controller GUI, on page 224](#)
- [Deploying a Service Graph, on page 226](#)

Overview

The Cisco Cloud Network Controller enables you to deploy Layer 4 to Layer 7 service devices to the public cloud.

Four types of Layer 4 to Layer 7 services are supported for deployments in Azure:

- ALB refers to Azure Application gateway or Application Load balancer
- NLB refers to Azure Load balancer or Network Load balancer
- Third Party Firewall
- Third Party Load Balancer

About Service Graphs

A service graph is used to represent a set of Layer 4 to Layer 7 services devices inserted between two or more pair of EPGs. EPGs can represent your applications running within a cloud (for example, Cloud EPG) or internet (cloudExtEPG) or from other sites (for example, on-premises or remote cloud sites). Layer 4 to Layer 7 services devices can be NLB, ALB, a cluster of third party firewalls or a third party load balancer.

A service graph in conjunction with contracts (and filters) is used to specify communication between two EPGs. A Cisco Cloud Network Controller automatically derives security rules (network security group/NSG and ASG) and forwarding routes (UDRs) based on the policy specified in Contract and Service Graph

Multiple service graphs can be specified to represent different traffic flows or topologies.

Following combinations are possible with service graphs:

- Same device can be used in multiple service graphs.

- Same service graph can be used between multiple consumer and provider EPGs.

By using a service graph, the user can specify the policy once and deploy the service chain within regions or inter-regions. Each time the graph is deployed, Cisco ACI takes care of changing the network configuration to enable the forwarding in the new logical topology.

For Third party firewalls, the configuration inside the device is not managed by Cisco Cloud Network Controller.

A service graph represents the network using the following elements:

- **Service Graph Nodes**—A node represents a function that is applied to the traffic, such as a load balancer. A function within the service graph might require one or more parameters and have one or more connectors.
- **Connector**—A connector enables input and output from a node.

After the graph is configured, the Cisco APIC automatically configures the services according to the service function requirements that are specified in the service graph. The Cisco APIC also automatically configures the network according to the needs of the service function that is specified in the service graph, which does not require any change in the service device.

Using Service Graphs with Cloud Native and Third-Party Services

You can use service graphs with cloud native and third-party services. You can use service graphs in these situations either with or without redirect. See [Example Use Cases for Service Graphs with Cloud Native and Third-Party Services, on page 203](#) for example use cases, with or without redirect.

You will use the cloud service endpoint group (service EPG), also introduced in Release 5.1(2), with this type of service graph. See [Cloud Service Endpoint Groups, on page 38](#) for more information about the service EPG, and the deployment types and access types that are available for service EPGs.

The following deployment types and access types are supported with service graphs used with service EPGs for this purpose.

Table 36: Provider Service EPG Types

Deployment Types	Access Types
Cloud Native	Private
Cloud Native Managed	Public and Private
Third-Party	Private

Table 37: Consumer Service EPG Types

Deployment Types	Access Types
Cloud Native Managed	Public and Private

Guidelines and Limitations

- You must have the newer NSG-per-subnet configuration enabled in order to use service graphs with cloud native and third-party services, using the service EPGs. See [Security Groups, on page 45](#) for more information on the NSG-per-subnet configuration.

- Any restrictions that apply for cloud EPG and service graph combinations also apply to service EPG and service graph combinations. For example, the cloud EPG/service graph restriction that a consumer and provider that is tag-based cannot be in the same VRF in the same region would also apply for service EPGs and service graphs.
- For two node graphs that don't perform redirect, SNAT and DNAT are enabled. It is assumed that the DNATed address is a device that is equivalent to a load balancer, which can take care of spraying traffic across different targets that may be in different subnets.

Note that if those targets are in different subnets, the service graph doesn't provide route reachability rules for those targets. It is assumed that the service EPG will take care of the reachability in this case.

- For cases involving AKS and service graphs, the service graph will only establish route reachability to the load balancer's subnet of the AKS cluster.

About Application Load Balancers

Application Load Balancer (also called Azure Application Gateway or ALB) is a Layer 7 load balancer, which balances the web traffic based on attributes like HTTP request, URL filtering etc. For more details please refer to [Microsoft Documentation](#).

In Cisco ACI, there are two ways to deploy an Application Load Balancer:

- Internet-facing: inserts the Application Load Balancer as a service between the consumer external EPG and the provider cloud EPG.
- Internal-facing: inserts the Application Load Balancer as a service between the consumer cloud EPG and the provider cloud EPG.

You can consume an Application Load Balancer using a service graph. A typical configuration involves:

- Creation of Layer 4 to Layer 7 services device as Application Load Balancer
- Consume the ALB as a node in the service graph
- Creation of one or more listeners in EPG communication when a service graph is associated with a contract.

Listeners enable you to specify the ports and protocols (HTTP or HTTPS) that the Application Load Balancer accepts traffic on. When specifying HTTPS, you also choose a security policy and an SSL certificate.



Note A listener can have multiple certificates.

All listeners require you to configure at least one rule (a default rule, which does not have a condition). Rules enable you to specify the action that the load balancer takes when a condition is met. For example, you can create a rule that redirects traffic to a specified URL when a request is made to a specified hostname or path.

An Application load balancer (ALB) should be in a separate subnet which should not be used to deploy other applications. Cisco Cloud Network Controller creates and attaches ALB's NSG to the subnet associated with the ALB. Cisco Cloud Network Controller supports Standard and Standard_v2 SKUs of Azure Application Gateway.

About Network Load Balancer

A Network Load Balancer (Azure Load Balancer or NLB) is a Layer 4 device that distributes the in-bound flow packets based on Layer 4 ports. For more details, please refer to [Microsoft Documentation](#).

Similar to ALB, NLB can be deployed using a service graph. You can specify these actions by configuring one or more listeners.

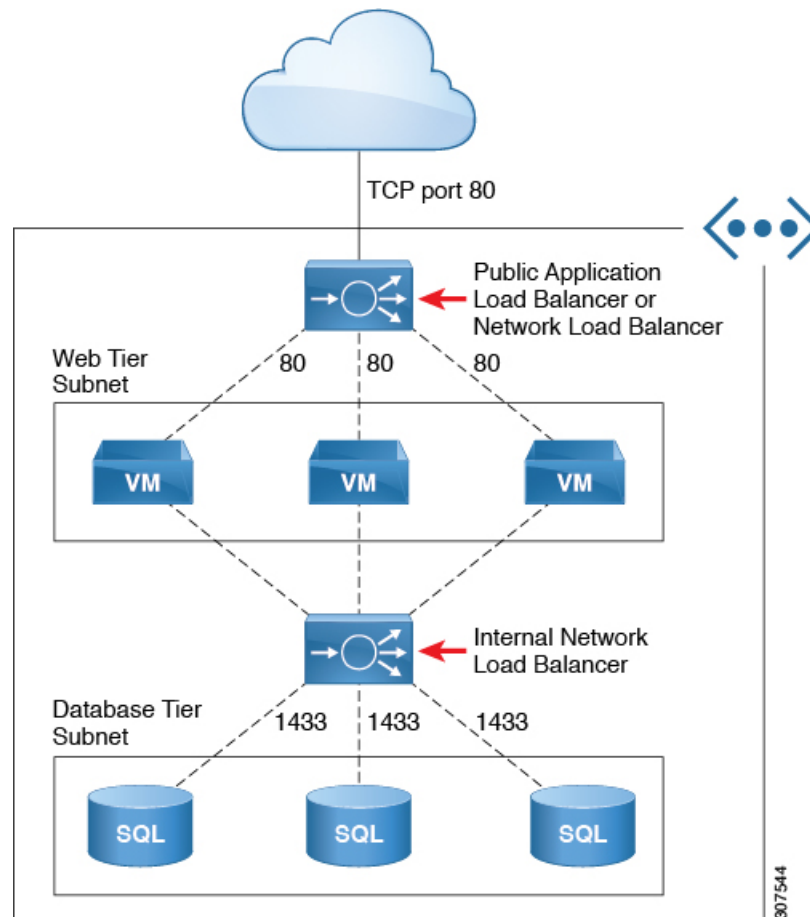
Listeners enable you to specify the ports and protocols (TCP or UDP) that the load balancer accepts and forwards traffic on. All listeners require you to configure at least one rule (a default rule, which does not have a condition). Rules enable you to specify the action that the load balancer takes when a condition is met. Unlike application gateway, here a rule can only forward traffic to specific port of the backend pool. NLB should be in a separate subnet similar to ALB. There are two modes of operation in Network load balancer:

- Forward mode: Traffic is forwarded from a specific listener port to the specified backend port.
- HA Port mode: Network load balancer will load balance TCP and UDP flows on all the ports simultaneously.

Cisco Cloud Network Controller supports Standard SKU Network Load Balancer only.

In Figure1, the frontend load balancer (ALB/NLB)- VM or firewall- backend load (ALB/NLB) balancer as a service are inserted between the consumer external EPG and the provider cloud EPG.

Figure 11: Internet-Facing and Internal-Facing Deployment



Understanding Configurations for Multiple Frontend IP Addresses on the Azure Network Load Balancer

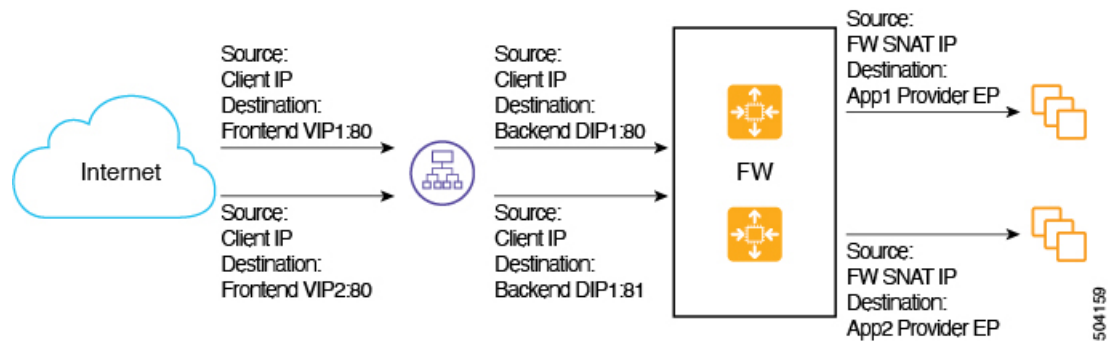
The following sections provide information on support for multiple frontend IP addresses on the Azure Network Load Balancer.

- [About Multiple Frontend IP Addresses on the Azure Network Load Balancer, on page 181](#)
- [Guidelines and Limitations, on page 182](#)

About Multiple Frontend IP Addresses on the Azure Network Load Balancer

Support is available for configuring multiple frontend IP addresses for an internet-facing Network Load Balancer. With this feature, each frontend IP address is attached to one or more rules to a specific backend pool.

The following graphic provides an example configuration where multiple frontend IP addresses are configured for an internet-facing Network Load Balancer.



This example configuration depicts the packet flow for the following listener rules:

	Listener Rule (Frontend Configuration)	Rule Action (Backend Configuration)
Rule1	<ul style="list-style-type: none"> • IP: VIP1 • Port: 80 	Port: 80
Rule2	<ul style="list-style-type: none"> • IP: VIP2 • Port: 80 	Port: 81

The service graph allows you to configure the settings for the listener rule and the rule action on services devices. When defined on a Network Load Balancer, the listener rule and rule action settings construct a mapping from a frontend configuration of the load balancer to a backend pool. With the support for configuring multiple frontend IP addresses for an internet-facing Network Load Balancer, you can configure listener rules with multiple frontends where each frontend is represented as a tuple combination of frontend IP addresses, port, and protocol.

Guidelines and Limitations

Following are the guidelines and limitations with the support for configuring multiple frontend IP addresses for an internet-facing Network Load Balancer:

- Support for multiple frontend IP addresses is available only for an internet-facing Network Load Balancer.
- Backend port reuse across multiple listener rules is not supported.

About Third-Party Load Balancers

Third-Party Load Balancer is a noncloud native Layer 4 to Layer 7 load balancer. Cisco Cloud Network Controller does not manage the configuration of the third-party load balancers. However, Cisco Cloud Network Controller automates the network stitching for connectivity to a third-party load balancer.

You can configure VIPs for a third-party load balancer from the external interface subnet. You can also configure additional VIPs for the third-party load balancers as secondary IP addresses on the external interface.

Cisco Cloud Network Controller supports third-party load balancers that are deployed in a two-arm mode (external and internal interfaces) with source NAT enabled.

Limitations for Third-Party Load Balancers:

- Cisco Cloud Network Controller does not support Direct Server Return (DSR) configurations on third-party load balancers.
- Third-party load balancers are not supported in active/standby high availability configurations.
For details about third-party load balancer VMs in active/active mode, see [Example Use Cases, on page 188](#).
- Alien VIP range is not supported for third-party load balancers.

About Allow All Traffic Option

The **Allow All Traffic** option is available for third-party firewalls and Azure network load balancers deployed as pass-through devices on a redirect-enabled service graph.




Note This option allows all inbound and outbound access to the subnet on which the interface belongs. Ensure that this does not present a security risk before enabling this option.


The following sections provide instructions for enabling the **Allow All Traffic** option.

- [Third-Party Firewall, on page 183](#)
- [Azure Network Load Balancer, on page 184](#)

Third-Party Firewall

- To enable this option when creating a new service graph type:
 1. From the **Application Management** list in the **Intent** menu, click **Services > Devices > Create Device**.
 2. Choose **Third party firewall** as the **Service Type**.
 3. Click **Add Interface**, then locate the **Allow All Traffic** area.
 4. Click the box next to the **Enabled** field in the **Allow All Traffic** area to allow all inbound and outbound access to the subnet on which the interface belongs.
 5. Click **Save** when finished.
- To enable this option when editing an existing service graph type:
 1. From the **Application Management** list in the **Intent** menu, click **Services**, then click on an existing service device with **Third-Party Firewall** shown as the **Device Type**.
A panel showing details for this service device type slides in from the right side of the window.
 2. Click the Details icon ()
Another window appears that provides more detailed information for this service device type.
 3. Locate the **Interfaces** area in the window and click the necessary interface selector under the **Interface Selectors** column.


A panel showing details for this interface slides in from the right side of the window.

4. Click the Details icon ().

Another window appears that provides more detailed information for this interface.

5. Click the pencil icon to edit the configuration settings for this interface.
6. Locate the **Allow All Traffic** area, then click the box next to the **Enabled** field in the **Allow All Traffic** area to allow all inbound and outbound access to the subnet on which the interface belongs.
7. Click **Save** when finished.

Azure Network Load Balancer

- To enable this option when creating a new service graph type:
 1. From the **Application Management** list in the **Intent** menu, click **Services > Devices > Create Device**.
 2. Choose **Network Load Balancer** as the **Service Type**.
 3. In the **Settings** area, click the box next to the **Enabled** field in the **Allow All Traffic** area to allow all inbound and outbound access to the subnet on which the interface belongs.
 4. Click **Save** when finished.
- To enable this option when editing an existing service graph type:
 1. From the **Application Management** list in the **Intent** menu, click **Services**, then click on an existing service device with **Network Load Balancer** shown as the **Device Type**.
A panel showing details for this service device type slides in from the right side of the window.
 2. Click the Details icon ().

Another window appears that provides more detailed information for this service device type.

 3. Click the pencil icon to edit the configuration settings for this service device.
 4. In the **Settings** area, locate the **Allow All Traffic** area, then click the box next to the **Enabled** field in the **Allow All Traffic** area to allow all inbound and outbound access to the subnet on which the interface belongs.
 5. Click **Save** when finished.

Dynamic Server Attachment to Server Pool

Servers in a provider EPG or service devices such as third-party firewalls behind ALB/NLB are dynamically added to the target groups. In Azure, the target groups are referenced as the backend pool. Listeners and rule configuration that define the frontend and backend protocol and port number, and load balancing action are provided by the user. When configuring a listener rule on ALB/NLB that is the last node as part of service graph configuration, you can select provider EPG for a given rule. The endpoints from that EPG are dynamically added to the target group of the load balancer. If another node, such as a third-party firewall, exists between

ALB/NLB and the provider EPG, the firewall endpoints are dynamically added to the target group of the load balancer. You do not need to specify the endpoints or FQDN for the targets.

Support is now available for VM scale sets as backend targets for load balancers.



Note When using a VM scale set for a firewall, use only a subnet-based EP selector for the firewall interfaces. Azure does not support per-NIC tagging for a VM scale set with multiple interfaces.

About Inter-VNet Services

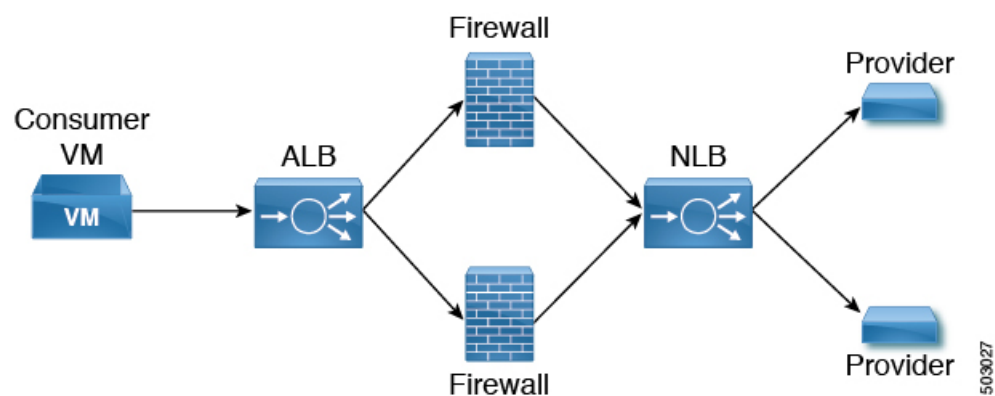
Support is available for the deployment and automation of the inter-VNet services. This is both for the East-West and North-South use cases within the cloud.

Note the following considerations for this support:

- VNet peering needs to be configured for hub-spoke topology. For more information, refer to [Configuring VNet Peering for Cisco Cloud Network Controller for Azure](#).
 - For multi-node services with redirect: The service device has to be present in the infra VNet. Service devices such as ALB fronting the provider can be present in the provider VNet.
 - For multi-node service without redirect: The service device can be in the provider VNet or spread across the hub VNet and the provider VNet.
- Inter-VNet traffic is supported with an Application load balancer or Network load balancer in the infra VNet and the provider in a non-infra VNet. The VNets should be peered together and the load balancer and the provider should be from the same region.

About Multinodes

Multinode service graph is supported. Multinodes enable multiple deployment scenarios with service graphs.



Service devices that can be deployed are Application Load Balancer, Network Load Balancer and Third Party Firewall.

Two types of nodes are admitted in a graph.

- Non-redirect: Traffic is destined to service devices (Load Balancers, Thirdparty firewalls with DNAT and SNAT, Network Load Balancer).
- Redirect: Service device is a passthrough device (Network Load Balancer or Firewall).

About Layer 4 to Layer 7 Service Redirect

The Layer 4 to Layer 7 Service Redirect feature is available for Cisco Cloud Network Controller, similar to the policy-based redirect (PBR) feature available for Cisco APIC. The Layer 4 to Layer 7 Service Redirect feature is configured using the **Redirect** option in the Cisco Cloud Network Controller.



Note Throughout this section, the term "consumer-to-provider" is sometimes used as a blanket term to describe traffic going from point A to point B, where a redirect service device might be inserted between those two points. However, this does not mean that only consumer-to-provider traffic is supported for redirect; traffic might also be from provider-to-consumer, such as in the use case described in [Spoke to Spoke, on page 190](#).

With redirect, policies are used to redirect traffic through specific service devices, where service devices can be deployed as a Network Load Balancer or a third-party firewall. This traffic isn't necessarily destined for the service device as part of the standard consumer-to-provider configuration; rather, you would configure the consumer-to-provider traffic as you normally would, and you would then configure service graphs to redirect that consumer-to-provider traffic to a specific service device.

Support for redirect for Cisco Cloud Network Controller is only available in conjunction with the VNet peering feature, taking advantage of the hub-and-spoke topology used in VNet peering. For more information on the VNet peering feature, see the [Configuring VNet Peering for Cisco Cloud Network Controller for Azure](#) document.

Passthrough Rules

When redirect is enabled, the rules in the NSGs (Network Security Groups) attached to the service devices are updated to permit traffic from consumer to provider. These rules are called "passthrough rules". In general, the passthrough rule is to permit traffic from consumer IP to provider IP. If the destination IP is an application load balancer (ALB) VIP, the rule is to permit traffic from consumer IP to the ALB VIP.

Redirect Programming

Redirect programming depends on the classification of the destination EPG (tag-based or subnet-based):

- For a subnet-based EPG, subnets of the destination EPGs are used to program redirects
- For a tag-based EPGs, CIDRs of the destination VNet are used to program redirects

As a result of this, the redirect affects traffic from other EPGs going to the same destination in the redirect, even if the EPG is not part of the service graph with the redirect. Traffic from EPGs that are not part of the redirect will also get redirected to the service device.

The following table describes how redirect is programmed in different scenarios.

Consumer	Provider	Redirect on Consumer VNet	Redirect on Provider VNet
Tag-based	Tag-based	Redirect for the provider are the CIDRs of the provider's VNet	Redirect for the consumer are the CIDRs of the consumer's VNet
Tag-based	Subnet-based	Redirect for the provider are the subnets of the provider	Redirect for the consumer are the CIDRs of the consumer's VNet
Subnet-based	Tag-based	Redirect for the provider are the CIDRs of the provider's VNet	Redirect for the consumer are the subnets of the consumer
Subnet-based	Subnet-based	Redirect for the provider are the subnets of the provider	Redirect for the consumer are the subnets of the consumer

Redirect Policy

To support the Layer 4 to Layer 7 Service Redirect feature, a new redirect flag is now available for service device connectors. The following table provides information on the existing and new flags for the service device connectors.

ConnType	Description
redir	This value means the service node is in redirect mode for that connection. This value is only available or valid for third-party firewalls and Network Load Balancers.
snat	This value tells the service graph that the service node is performing source NAT on traffic. This value is only available or valid for the provider connector of third-party firewalls and only on the provider connector of a node.
snat_dnat	This value tells the service graph that the service node is performing both source NAT and destination NAT on traffic. This value is only available or valid for the provider connector of third-party firewalls and only on the provider connector of a node.
none	Default value.

Workflow for Configuring Redirect

Following is the typical workflow for configuring redirect:

1. Create one or more service devices to use with the service graph:
 - Network load balancer (NLB)

- Application load balancer (ALB)
 - Third-party firewall
2. Create a service graph and select the appropriate service devices for this particular service graph.
You will configure redirect at this point in the procedures:
 - a. Drag and drop a network load balancer, application load balancer, or firewall icon to the **Drop Device** area to select that service device for the service graph.
 - b. To enable the redirect feature, in the **Service Node** window that appears, check the box next to the **Redirect** option under the **Consumer Connector Type** and/or under the **Provider Connector Type** areas, depending on where you want to enable the redirect function.



Note Even though you might have an application load balancer in the service graph, you cannot enable redirect on an application load balancer service device.

- c. Complete the remaining configurations in the **Service Node** window, then click **Add**.
3. Configure the EPG communication, where you create a contract between the consumer and the provider EPGs.
4. Attach the service graph to the contract.
5. Configure the service device parameters.

Example Use Cases

Following are several example use cases:

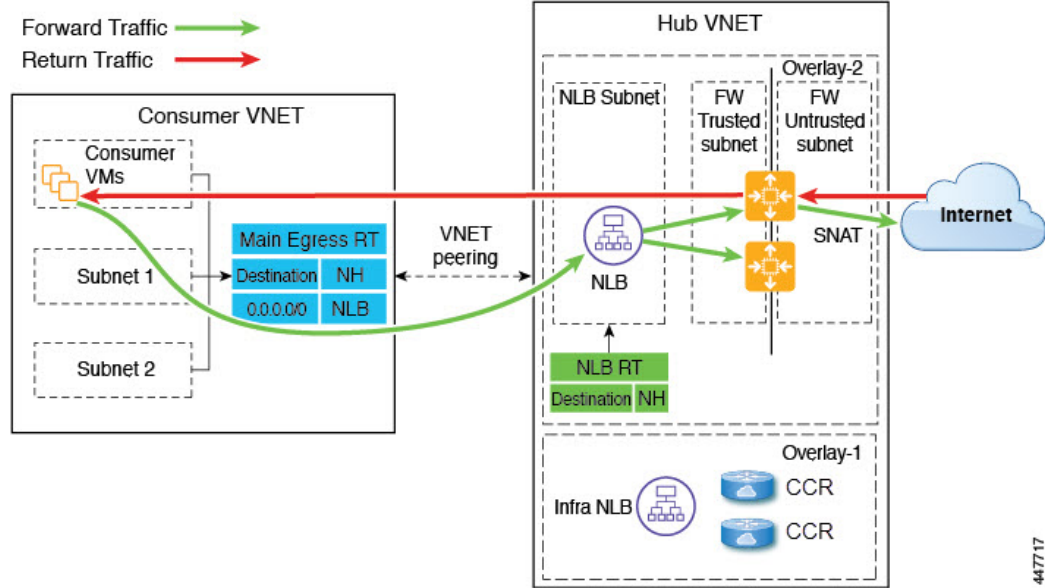
- [Spoke to Internet, on page 188](#)
- [Spoke to Spoke, on page 190](#)
- [Inter-Region Spoke to Spoke, on page 193](#)
- [Internet to Spoke \(Inter-VRF\), on page 195](#)
- [High Availability Support for Third-Party Load Balancer, on page 197](#)
- [Consumer and Provider EPGs in Two Separate VNETs, on page 199](#)
- [Hub VNET with Consumer and Provider EPGs in Two Separate VNETs, on page 201](#)

Spoke to Internet

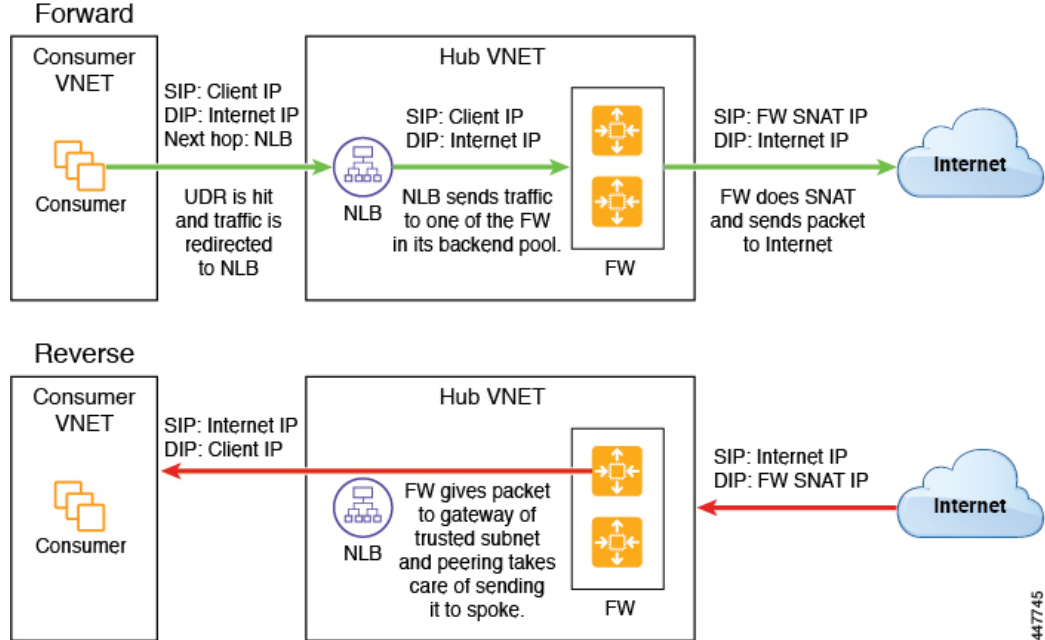
In this use case, the consumer VNet (with consumer VMs) and the hub VNet are peered using VNet peering. A network load balancer is also deployed, fronting two firewalls for scaling. In this use case, the consumer VMs need access to the internet for a certain reason, such as patch updates. In the consumer VNet, the route table is modified to include a redirect for the internet in this case, and traffic is redirected to the NLB in front of firewalls in the hub VNet. Any traffic from this consumer that is part of the service graph that is going to

the internet goes to the NLB as the next-hop. With VNet peering, traffic first goes to the NLB, then the NLB forwards the traffic to one of the firewalls in the back end. The firewalls also perform source network address translation (SNAT) when sending traffic to the internet.

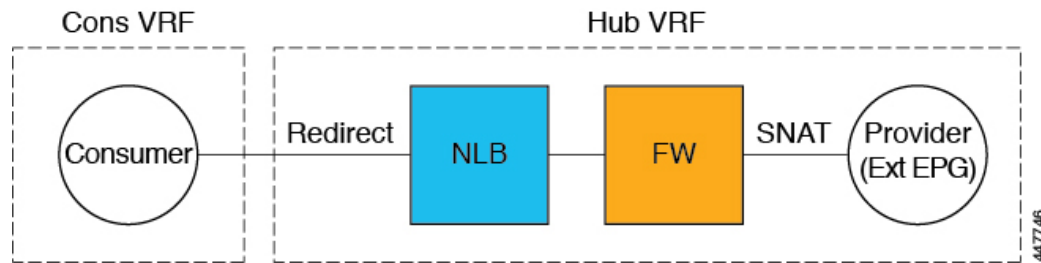
Ensure that all the Layer 4 to Layer 7 services devices used in this use case have dedicated subnets.



The following figure shows the packet flow for this use case.



The following figure shows the service graph for this use case.



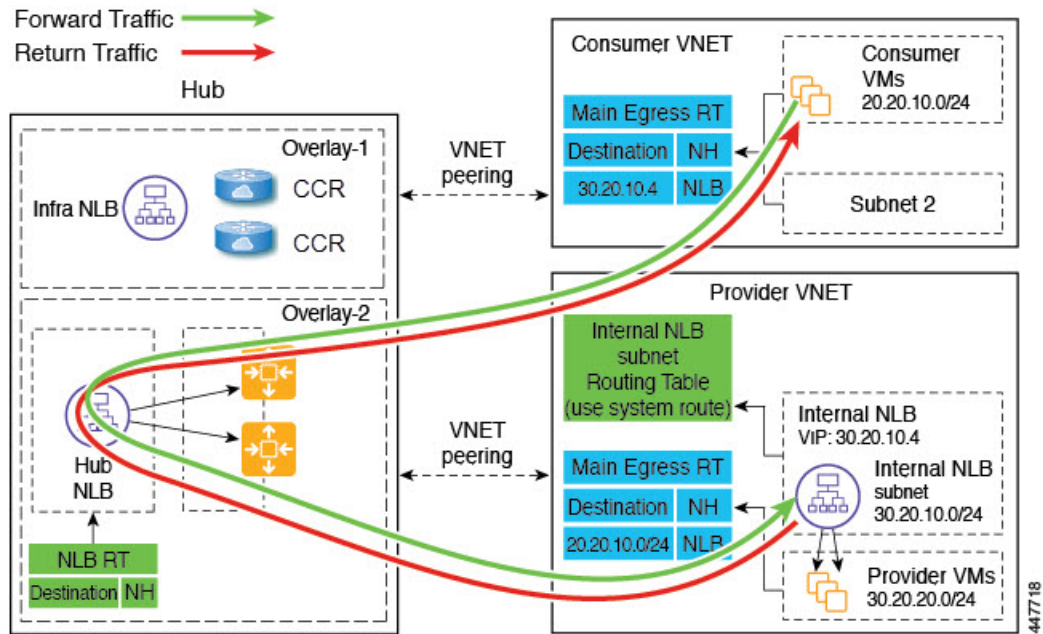
As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall
- In the **Service Node** window for the Network Load Balancer:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, leave the boxes unchecked.
- In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** option.

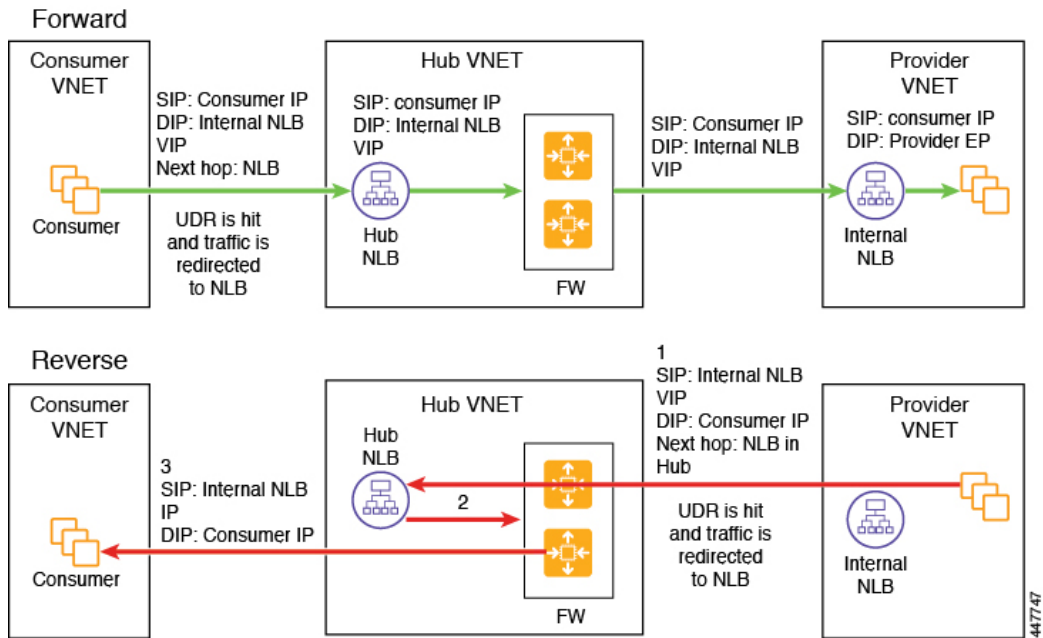
Spoke to Spoke

In this use case, traffic flows from spoke to spoke, through the hub firewall fronted by a hub NLB. Consumer endpoints are in the consumer VNet, and the provider VNet has VMs fronted by an internal NLB (or a third-party load balancer). The egress route table is modified in the consumer and provider VNet so that traffic is redirected to the firewall device fronted by the NLB. Redirect is applied in both directions in this use case.

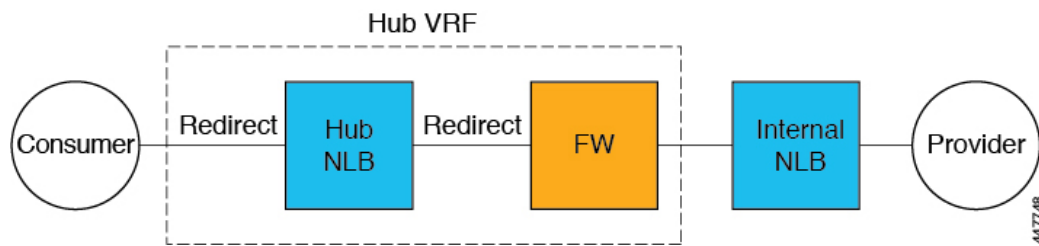
Ensure that all the Layer 4 to Layer 7 services devices used in this use case have dedicated subnets.



The following figure shows the packet flow for this use case.



The following figure shows the service graph for this use case.



As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Device** window, next create the service devices for the provider VNet:
 - In the **Tenant** field, choose the provider tenant.
 - In the **Service Type** field, choose **Network Load Balancer**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet for the provider VRF.



Note A third-party load balancer can be used in place of an internal NLB. Choose **Third-party load balancer** as the **Service Type**. Choose the **VRF** and set the interface(s) details by clicking **Add Interface**.

- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer (for the hub VNet)
 - Third-Party Firewall (for the hub VNet)
 - Network Load Balancer or Third-Party Load Balancer (for the provider VNet)
- In the **Service Node** window for the Network Load Balancer in the hub VNet:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the provider side.
- In the **Service Node** window for the Third-Party Firewall, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.

- In the **Service Node** window for the Network Load Balancer in the provider VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.

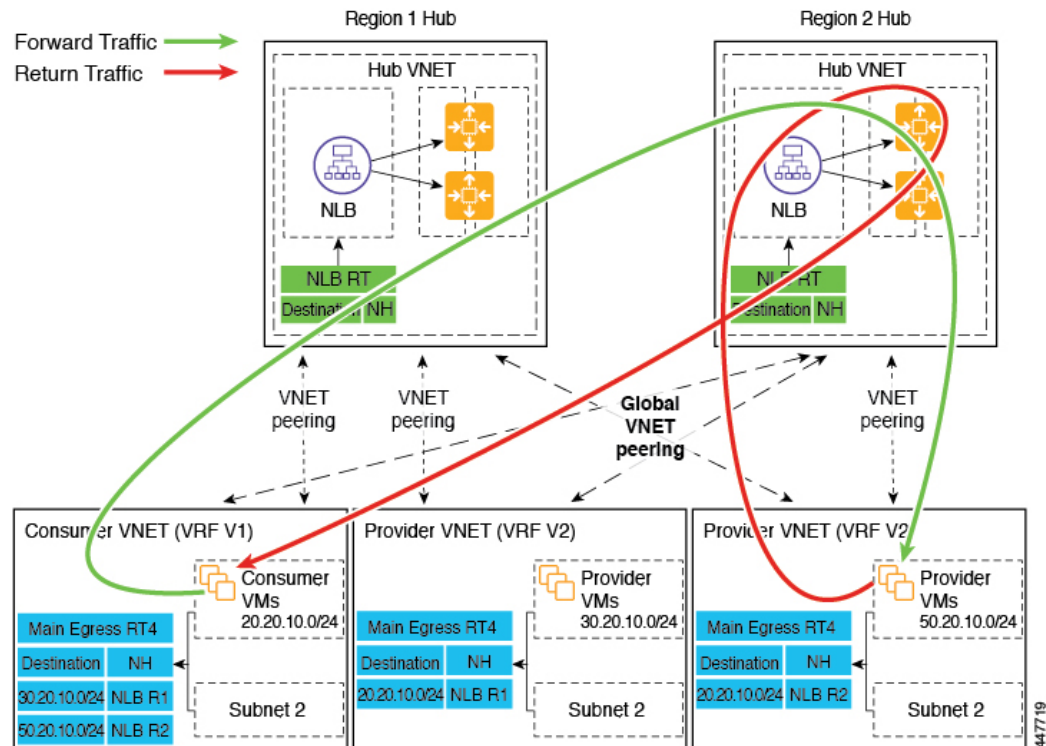


Note Ensure SNAT is configured on the third-party load balancers.

Inter-Region Spoke to Spoke

In this use case, both regions must have service devices. The consumer VNet is in region 1, the provider is stretched across both regions (regions 1 and 2), and some endpoints are in region 1 and some endpoints are in region 2. Different redirects are programmed for local provider endpoints and for remote region endpoints. In this case, the firewall that is used will be the firewall that is closest to the provider endpoint side.

Ensure that all the Layer 4 to Layer 7 services devices used in this use case have dedicated subnets.



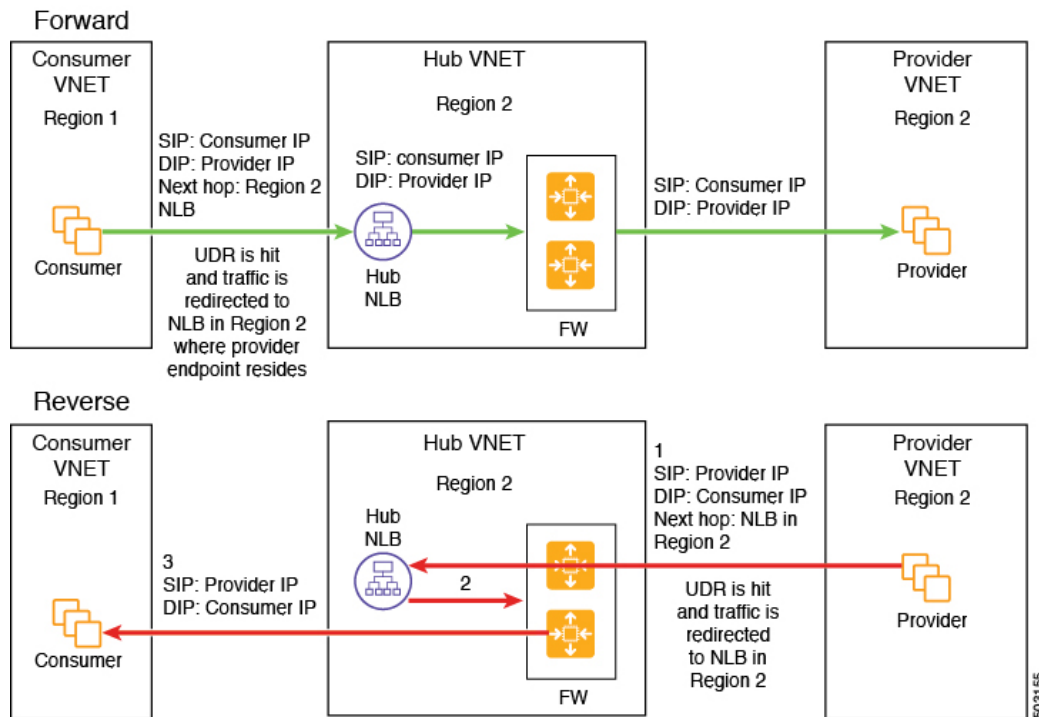
For example, consider the two subnets in the consumer VNet (VRF 1) egress route table (RT):

- 30.20.10.0/24 (NLB in region 1 [R1])
- 50.20.10.0/24 (NLB in region 2 [R2])

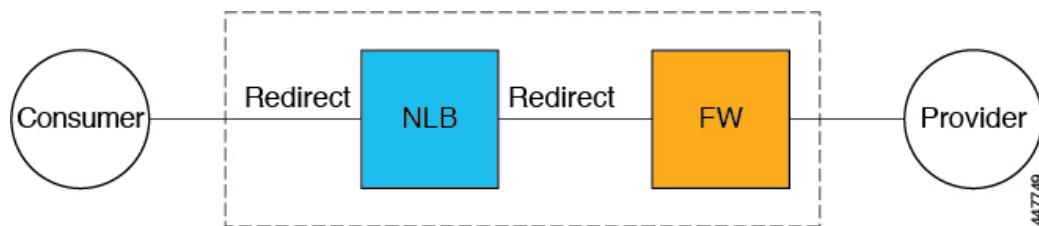
Assume the consumer wants to send traffic to the provider VMs 30.20.10.0/24, which are local to it. In that case, traffic will get redirected to the region 1 hub NLB and firewall, and will then go to the provider.

Now assume the consumer wants to send traffic to the provider VMs 50.20.10.0/24. In this case, the traffic will get redirected to the region 2 hub NLB and firewall, because that firewall is local to the provider endpoint.

The following figure shows the packet flow for this use case.



The following figure shows the service graph for this use case.



As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall

- In the **Service Node** window for the hub NLB:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the provider side.
- In the **Service Node** window for the Third-Party Firewall, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.

In the above use case, the provider VMs can also be front-ended by a cloud native or third-party load balancer.

Internet to Spoke (Inter-VRF)

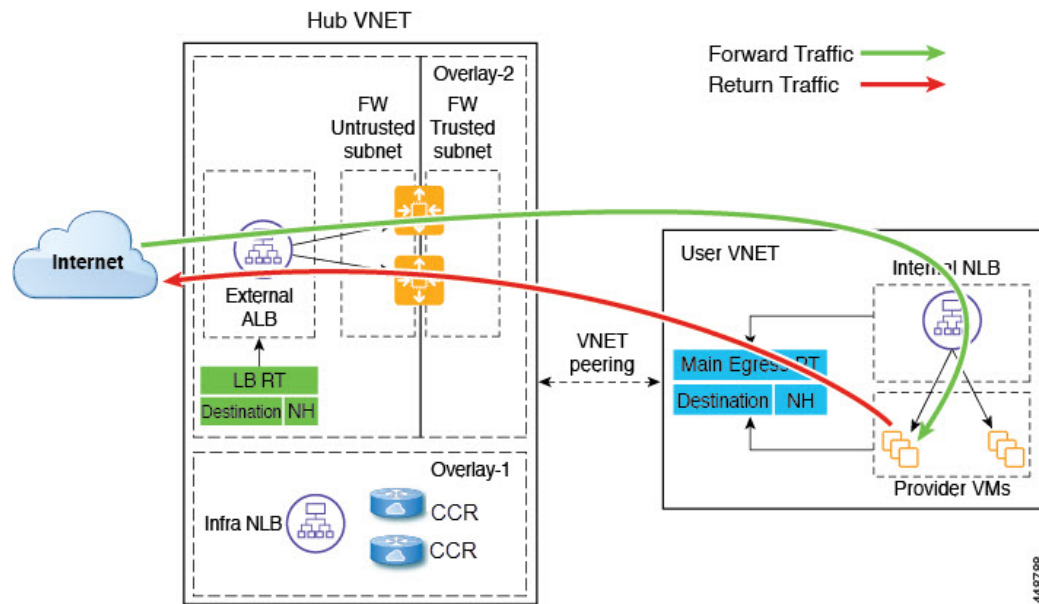
In this use case, traffic coming from the internet needs to go through the firewall before hitting the provider endpoints. Redirect is not used in this use case.



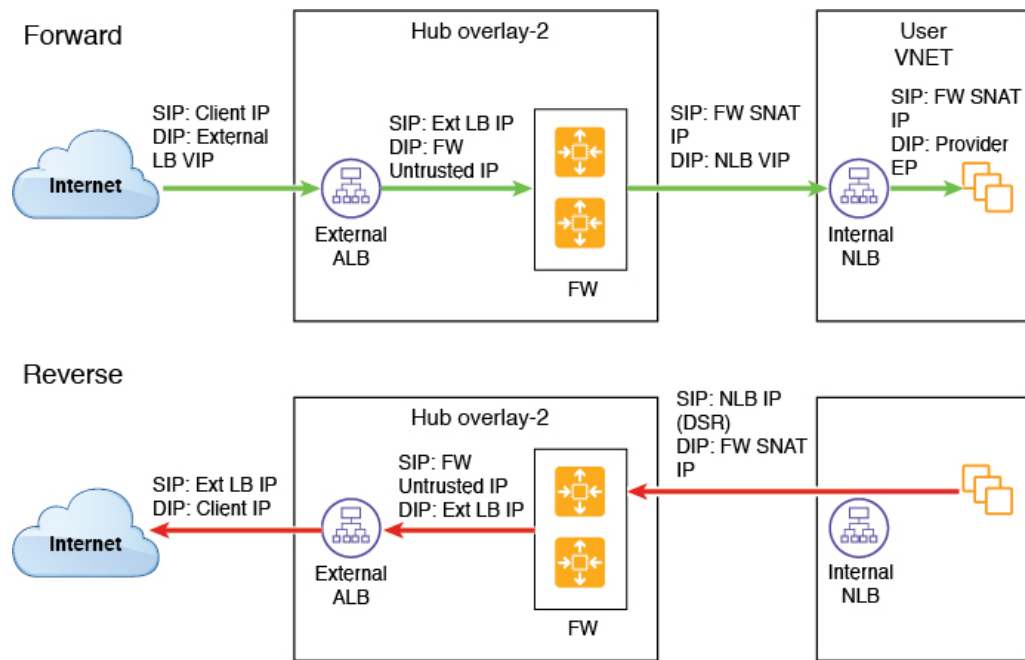
Note The general term "external load balancer" is used in this section because in this use case, the external load balancer could be either an NLB, ALB or a third-party load balancer. The following examples provide configurations using an ALB, but keep in mind that the external load balancer could be an NLB or a third-party load balancer instead.

The external load balancer exposes the service through VIP. Internet traffic is directed to that VIP, then external load balancers direct traffic to the firewalls in the backend pool (the external load balancers have the firewall's untrusted interface as its backend pool). The firewall performs SNAT and DNAT, and the traffic goes to the internal NLB VIP. The internal NLB then sends traffic to one of the provider endpoints.

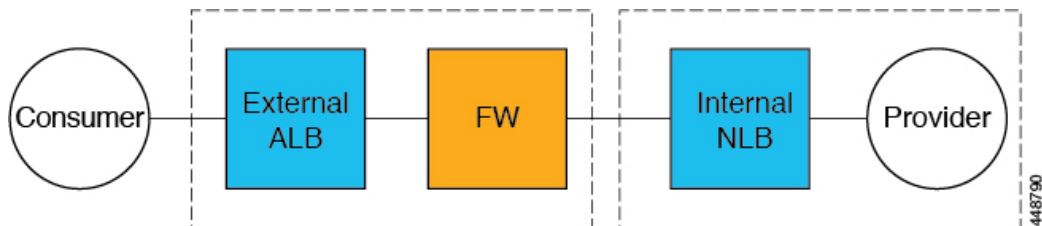
Ensure that all the Layer 4 to Layer 7 services devices used in this use case have dedicated subnets.



The following figure shows the packet flow for this use case.



The following figure shows the service graph for this use case.



As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Application Load Balancer** or **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
 - Choose **Third-Party Load Balancer** as the **Service Type**, and choose the **VRF** and set the interface(s) details by clicking **Add Interface**.
- In the **Create Device** window, next create the service devices for the provider VNet:
 - In the **Tenant** field, choose the provider tenant.

- In the **Service Type** field, choose **Network Load Balancer**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet for the provider VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer or Application Load Balancer (for the hub VNet)
 - Third-Party Firewall (for the hub VNet)
 - Network Load Balancer or Third-Party Load Balancer (for the provider VNet)
- In the **Service Node** window for the Network Load Balancer or Application Load Balancer for the hub VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.
- In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT and DNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** and **DNAT** options.
- In the **Service Node** window for the Network Load Balancer for the provider VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.



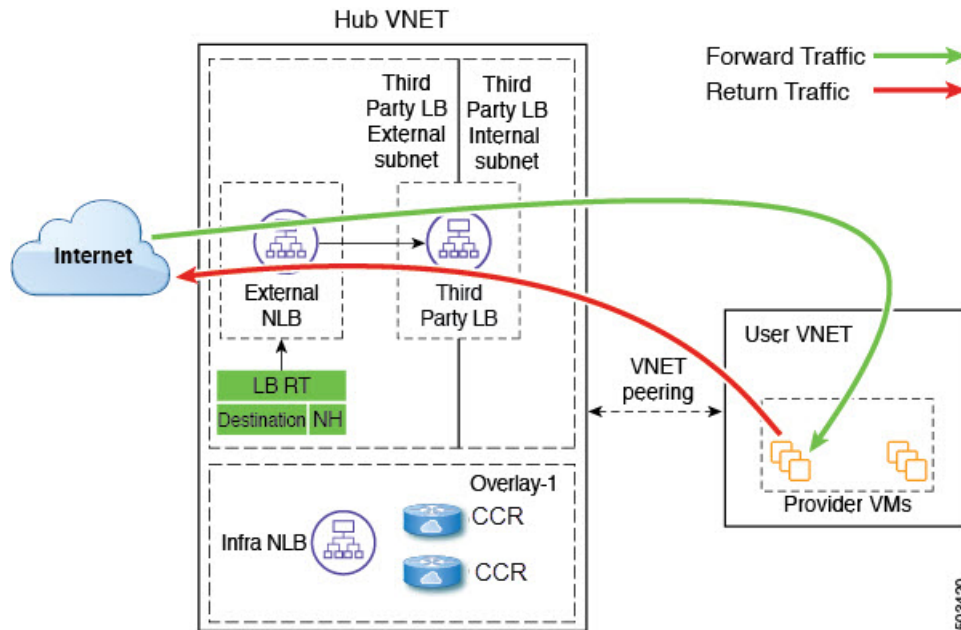
Note Ensure SNAT is configured on the third-party load balancers.

High Availability Support for Third-Party Load Balancer

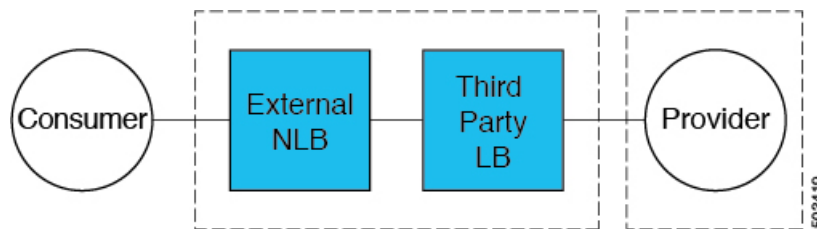
In this use case, traffic coming from the internet needs to go through the third-party load balancer before hitting the provider endpoints. Redirect is not used in this use case.

The third-party load balancer is configured as the backend pool of the NLB. Secondary IP addresses of the devices act as the target for the NLBs. You can choose to add either primary or secondary IP address (or both) as the target for the NLBs. The third-party load balancer VMs are deployed in active-active mode only. Third-party load balancers can not be used in active-standby high availability configuration.

Ensure that the third-party load balancers and the network load balancers have dedicated subnets.



The following figure shows the service graph for this use case.



As part of the configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Load Balancer** as the **Service Type**, and choose the **VRF** and set the interface(s) details by clicking **Add Interface**.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Load Balancer



Note Ensure that the Network Load Balancer and the Third-Party Load Balancer are in the same VNet.

- In the **Service Node** window for the Network Load Balancer for the hub VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.



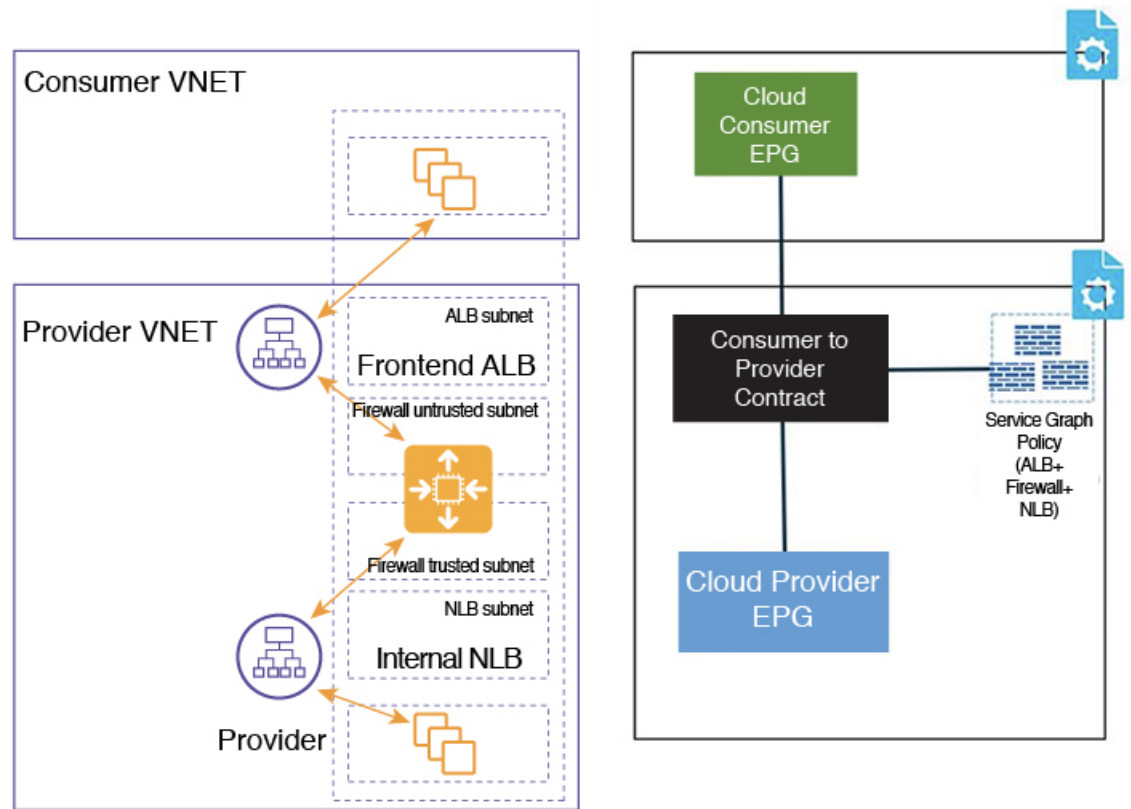
Note Ensure SNAT is configured on the third-party load balancers.

Consumer and Provider EPGs in Two Separate VNets

This use case is an example configuration with two VNets, with a consumer EPG and provider EPG in separate VNets.

- A frontend ALB, firewall, and internal NLB are inserted between the consumer and provider EPGs.
- A consumer endpoint sends traffic to the frontend ALB VIP and it is forwarded to the firewall.
- The firewall performs SNAT and DNAT, and the traffic flows to internal NLB VIP.
- The internal NLB load balances the traffic to the backend provider endpoints.

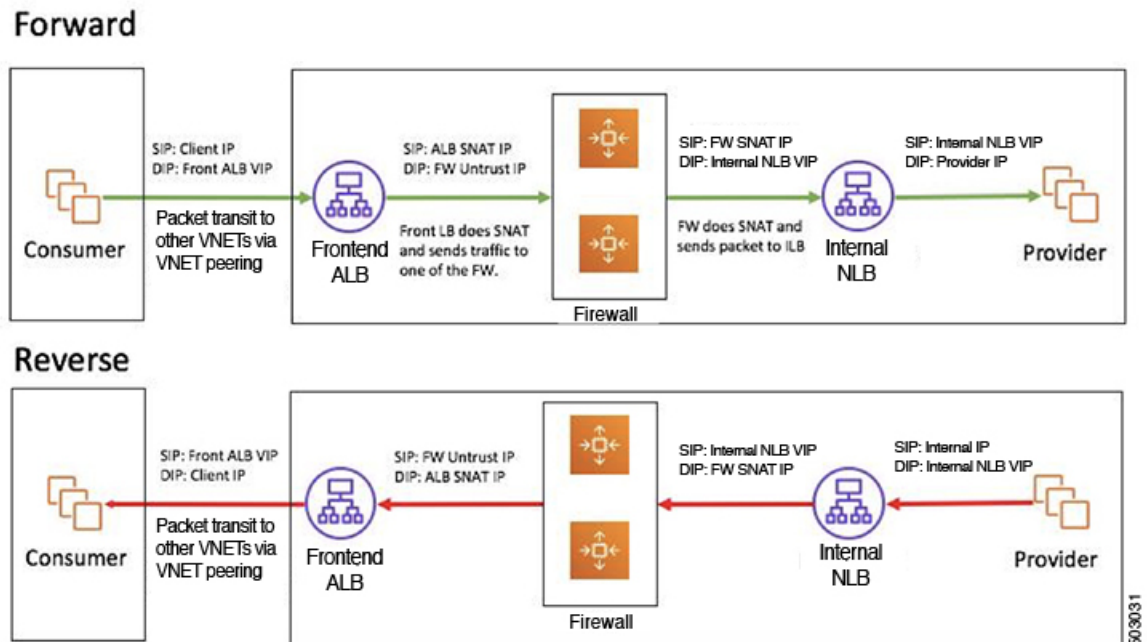
In this use case, a third-party load balancer can be used in place of the frontend ALB or an internal NLB. Ensure that all the Layer 4 to Layer 7 services devices used in this use case have dedicated subnets.



In the figure:

- The consumer EPG is in a consumer VNet.
- The provider EPG and all the service devices are in the provider VNet.
- The application load balancer, network load balancer (or third-party load balancer), and firewall need to have their own subnet in the VNet.

Packet flow for both the directions is shown in the following figure:

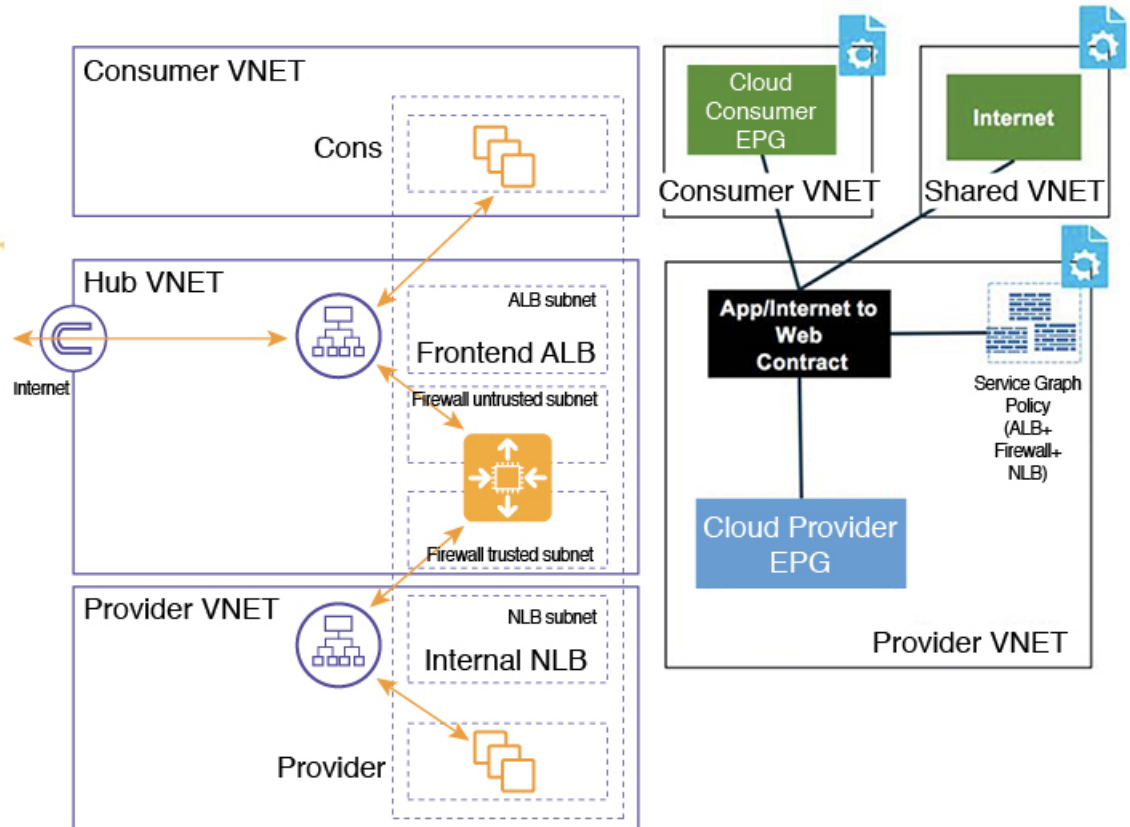


Hub VNet with Consumer and Provider EPGs in Two Separate VNets

This use case is an example configuration with three VNets: a hub VNet, and a consumer EPG and provider EPG in two separate VNets.

- A frontend ALB and firewall are inserted within the hub VNet, which is between the consumer and provider EPGs.
- An internal NLB is inserted in the provider EPG.
- A consumer endpoint sends traffic to the frontend ALB VIP and it is forwarded to the firewall.
- The firewall performs SNAT and DNAT, and the traffic flows to internal NLB VIP.
- The internal NLB load balances the traffic to the backend provider endpoints.

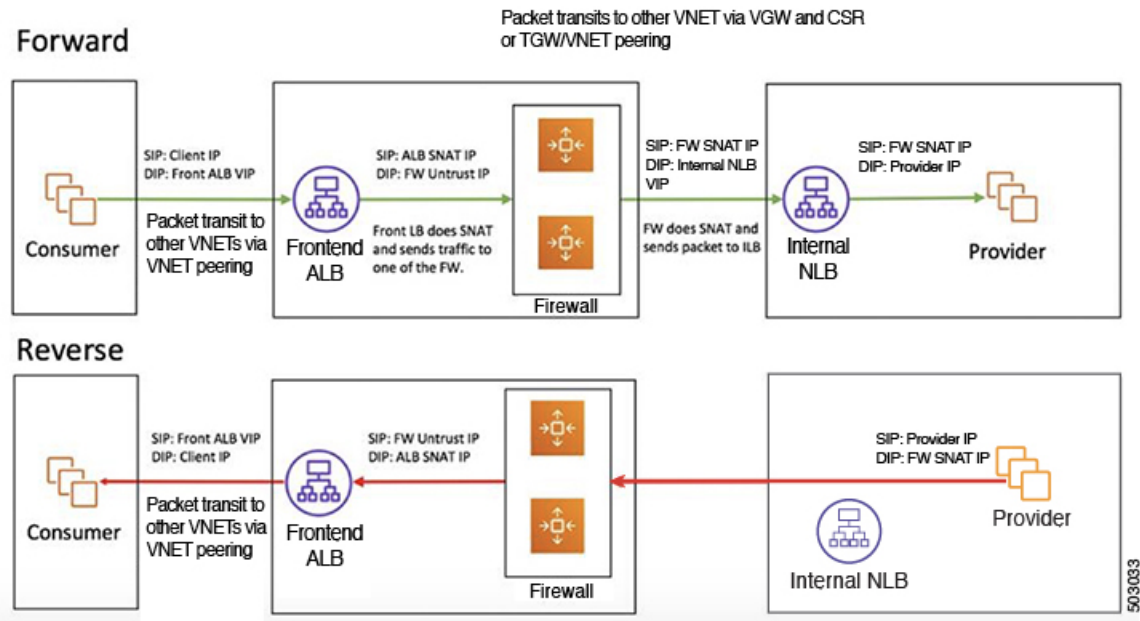
In this use case, a third-party load balancer can be used in place of the frontend ALB or an internal NLB. Ensure that all the Layer 4 to Layer 7 services devices used in this use case have dedicated subnets.



In the figure:

- The consumer EPG is in a consumer VNet.
- The provider EPG and the internal NLB are in the provider VNet.
- The frontend ALB and firewall are in the hub VNet
- The application load balancer, network load balancer (or third-party load balancer), and firewall need to have their own subnet in the VNet.

Packet flow for both the direction is shown in the following figure:



Example Use Cases for Service Graphs with Cloud Native and Third-Party Services

Following are several example use case for service graphs with cloud native and third-party services, with and without redirect. Refer to [Using Service Graphs with Cloud Native and Third-Party Services](#), on page 178 for more information and for guidelines and limitations.

Example Use Cases Without Redirect

Following are several example use case for service graphs with cloud native and third-party services without redirect.

You will be configuring cloud service EPGs as part of the process for each of these use cases. You must have the **NSG-per-subnet** configuration enabled if you are configuring cloud service EPGs. See [Security Groups](#), on page 45 and [Cloud Service Endpoint Groups](#), on page 38 for more information.

- [Single-Node Service Graph for Internet Inbound Traffic: Non-Managed Service EPG as Provider](#), on page 204
- [Single-Node Service Graph for Internet Inbound Traffic: Cloud Native Service EPG as Provider](#), on page 205
- [Two-Node Service Graph for Internet Inbound Traffic: Cloud Native Managed Service EPG as Provider](#), on page 206
- [Three-Node Service Graph for Internet Inbound Traffic: Cloud Native Managed Service EPG as Provider](#), on page 208

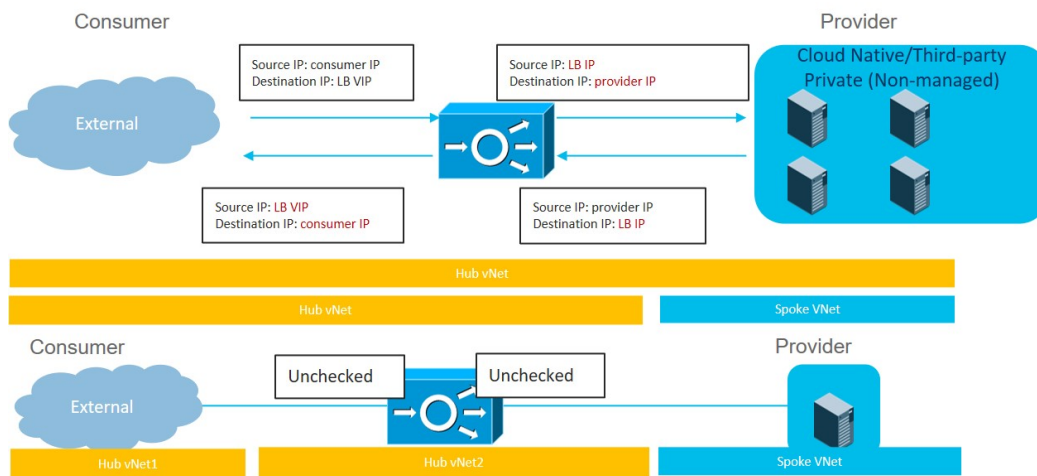


Note For each of the following use cases, a similar topology with a single node, two node and three node service graph with the service EPG as the provider can be supported for East-West traffic in the cloud. In these use cases, the consumer will be a cloud EPG and the load balancer used will be an internal load balancer.

Single-Node Service Graph for Internet Inbound Traffic: Non-Managed Service EPG as Provider

This use case has a single-node service graph, where the service node is a load balancer (application load balancer, network load balancer, or third-party load balancer).

In this use case, the service EPG is the provider, and an external EPG is configured on the consumer side. The service EPG can be in the hub or spoke VNETs. The service endpoints are learned dynamically and are added to the application load balancer or network load balancer.



To configure this use case:

1. Create the external EPG on the consumer side.
See [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#) for those procedures. Select the `infra` tenant for this external EPG.
2. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.
See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, using these settings:
 - **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure Storage` would be a supported service type with a `Cloud Native` deployment type.
 - **Deployment type:** `Cloud Native` OR `Third-Party`
 - **Access type:** `Private`
3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

Make the following selections:

- In the **Create Device** window, create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose **Application Load Balancer** or **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the Application Load Balancer or Network Load Balancer.
- In the **Service Node** window for the Application Load Balancer or Network Load Balancer for the hub VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.

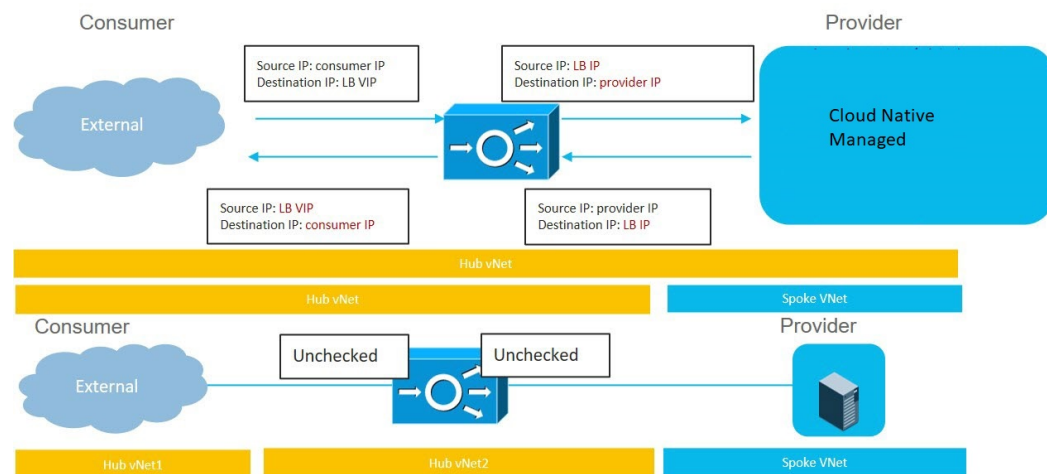
4. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

Single-Node Service Graph for Internet Inbound Traffic: Cloud Native Service EPG as Provider

This use case has a single-node service graph, where the service node is a load balancer (application load balancer, network load balancer, or third-party load balancer).

In this use case, the service EPG is the provider, and an external EPG is configured on the consumer side. The service EPG can be in the hub or spoke VNets.



To configure this use case:

1. Create the external EPG on the consumer side.

See [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#) for those procedures. Select the `infra` tenant for this external EPG.

2. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, using these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure ApiManagement Services` would be a supported service type with a `Cloud Native Managed` deployment type.
- **Deployment type:** `Cloud Native Managed`
- **Access type:** `Public` and `Private`

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

Make the following selections:

- In the **Create Device** window, create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose **Application Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the Application Load Balancer.
- In the **Service Node** window for the Application Load Balancer for the hub VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.

4. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

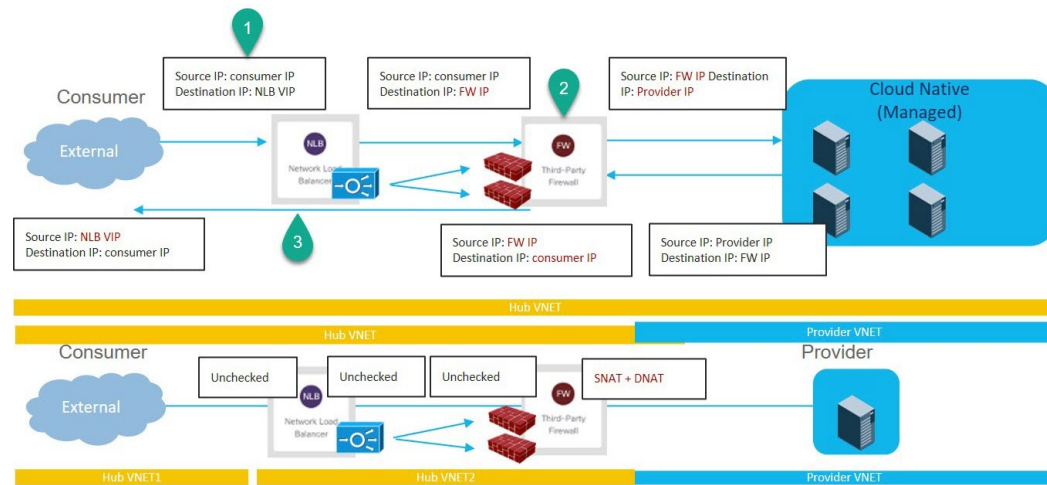
Two-Node Service Graph for Internet Inbound Traffic: Cloud Native Managed Service EPG as Provider

This use case has a two-node service graph, where the service nodes are a network load balancer and a firewall. Because this two-node service graph doesn't use redirect, SNAT+DNAT is performed on the firewall. The DNATed address is assumed to be a network load balancer or an equivalent service. For this use case, the service graph will only establish route reachability to the load balancer's subnet.

In this use case, the service EPG is the provider, and an external EPG is configured on the consumer side. The service EPG can be in the hub or spoke VNETs.

These actions take place in this use case, as shown in the following figure:

1. Traffic is destined to the network load balancer public VIP, which then load balances the traffic to the firewall (DNAT).
2. SNAT+DNAT is performed on the firewall.
3. For the return traffic, Azure translates the source IP to the network load balancer public VIP.



To configure this use case:

1. Create the external EPG on the consumer side.

See [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#) for those procedures. Select the `infra` tenant for this external EPG.

2. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, using these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, Azure Kubernetes Services (AKS) would be a supported service type with a Cloud Native Managed deployment type.
- **Deployment type:** Cloud Native Managed
- **Access type:** Private

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.

- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall
- In the **Service Node** window for the Network Load Balancer, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.
- In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT and DNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** and **DNAT** options.

4. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

Three-Node Service Graph for Internet Inbound Traffic: Cloud Native Managed Service EPG as Provider

This use case has a three-node service graph, where the service nodes are:

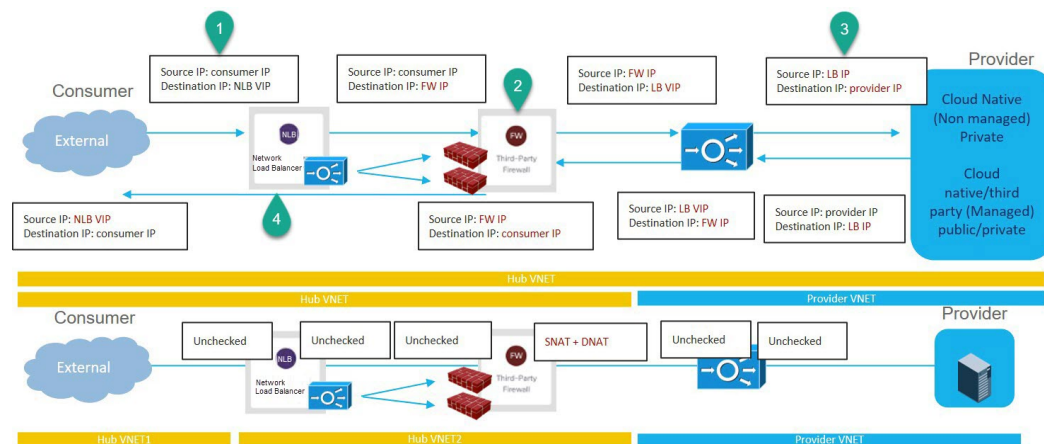
- First service device: Network load balancer in the hub VNet
- Second service device: Firewall in the hub VNet
- Third service device: Third-party load balancer in the hub VNet or spoke VNet

Because this three-node service graph doesn't use redirect, SNAT+DNAT is performed on the firewall. The DNATed address is assumed to be a load balancer or an equivalent service.

In this use case, the service EPG is the provider, and an external EPG is configured on the consumer side. The service EPG can be in the hub or spoke VNets.

These actions take place in this use case, as shown in the following figure:

1. Traffic is destined to the first service device, the network load balancer public VIP, which then load balances the traffic to the firewall (DNAT).
2. SNAT+DNAT is performed on the firewall, which is the second service device.
3. Traffic moves to the third service device, the third-party load balancer, which has SNAT configured.
4. For the return traffic, Azure translates the source IP to the network load balancer public VIP.



To configure this use case:

1. Create the external EPG on the consumer side.

See [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#) for those procedures. Select the `infra` tenant for this external EPG.

2. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure ApiManagement Services` would be a supported service type with a `Cloud Native Managed` deployment type.
- **Deployment type:** `Cloud Native Managed`
- **Access type:** `Private`

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - For the first service device, choose **Application Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - For the second service device, choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.

- If the third service device is in the hub VNet, choose **Third-Party Load Balancer** as the **Service Type**, and choose the **VRF** and set the interface(s) details by clicking **Add Interface**.
 - In the **Create Device** window, next create the service devices for the provider VNet, if necessary (if the third service device is in the provider VNet):
 - In the **Tenant** field, choose the provider tenant.
 - In the **Service Type** field, choose **Third-Party Load Balancer**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet for the provider VRF.
 - In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Application Load Balancer (for the hub VNet)
 - Third-Party Firewall (for the hub VNet)
 - Third-Party Load Balancer (for the hub or provider VNet)
 - In the **Service Node** window for the Application Load Balancer for the hub VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.
 - In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT and DNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** and **DNAT** options.
 - Ensure SNAT is configured on the third party load balancers.
4. Deploy the Layer 4 to Layer 7 services.
- See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

Example Use Cases With Redirect

Following are several example use case for service graphs with cloud native and third-party services with redirect.

You will be configuring cloud service EPGs as part of the process for each of these use cases. You must have the **NSG-per-subnet** configuration enabled if you are configuring cloud service EPGs. See [Security Groups, on page 45](#) and [Cloud Service Endpoint Groups, on page 38](#) for more information.

- [Two-Node Service Graph for Internet Outbound, on page 211](#)
- [Two-Node Service Graph for East-West, on page 213](#)
- [Two-Node Service Graph for East-West with SNAT Option, on page 215](#)

- [Two-Node Service Graph for Inbound Traffic via Express Route Gateway, on page 216](#)
- [Two-Node Service Graph for Inbound Traffic via Express Route Gateway with SNAT Option, on page 218](#)
- [Three-Node Service Graph for Inbound Traffic via Express Route Gateway, on page 220](#)

Two-Node Service Graph for Internet Outbound

This use case has a two-node service graph, where the service nodes are a network load balancer and a firewall. Redirect is enabled on the consumer side in this use case, and SNAT is enabled on the firewall.

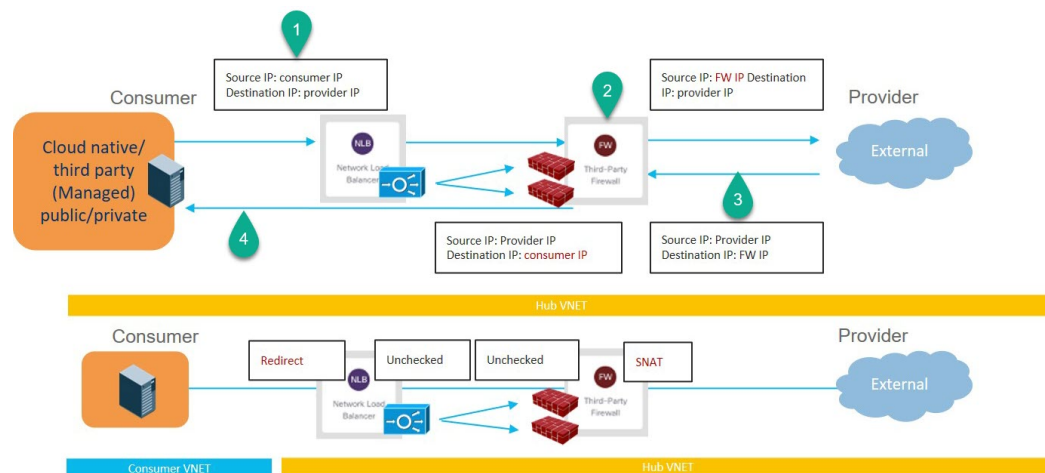
In this use case, the service EPG is the consumer, and an external EPG is configured on the provider side.



Note We recommend that you do not use 0.0.0.0/0 in an external EPG if the Layer 4 to Layer 7 service graph is used for PaaS that uses its own UDR for internet reachability.

These actions take place in this use case, as shown in the following figure:

1. Traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
2. SNAT is performed on the firewall.
3. The return traffic comes back to the firewall SNAT IP address.
4. At this point in the return direction, the return traffic doesn't go through the network load balancer.



To configure this use case:

1. Create the external EPG on the provider side.

See [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#) for those procedures.

- Select the `infra` tenant for this external EPG.
- Do not configure the external EPG with the 0.0.0.0/0 subnet.

2. Create the service EPG on the consumer side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure Kubernetes Services (AKS)` would be a supported service type with a `Cloud Native Managed` deployment type.
- **Deployment type:** `Cloud Native Managed`
- **Access type:** `Private`

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall
- In the **Service Node** window for the Network Load Balancer:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, leave the boxes unchecked.
- In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** option.

4. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

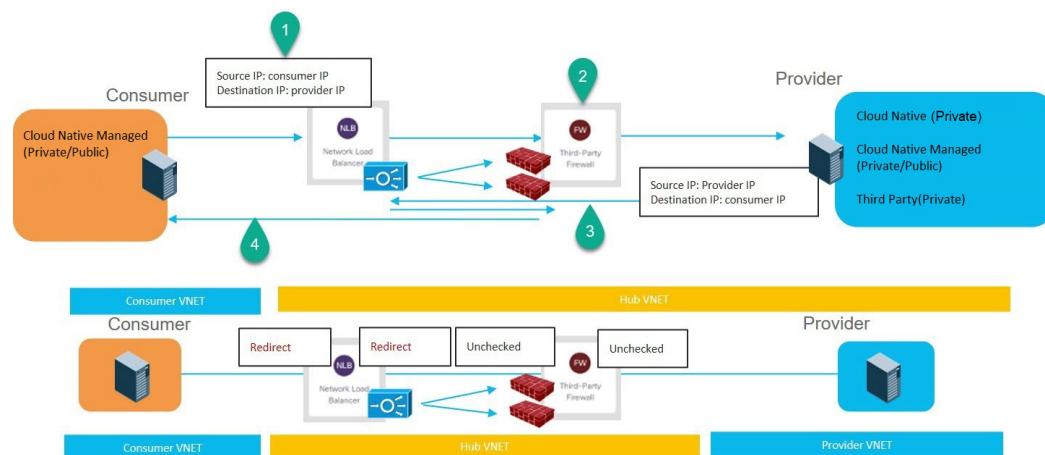
Two-Node Service Graph for East-West

This use case has a two-node service graph, where the service nodes are a network load balancer and a firewall. Redirect is enabled on both the consumer and the provider side in this use case.

In this use case, the consumer and provider could be cloud EPGs or service EPGs.

These actions take place in this use case, as shown in the following figure:

1. Traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
2. SNAT is *not* performed on the firewall in this use case.
3. The return traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
4. At this point in the return direction, the return traffic comes back to the consumer.



To configure this use case:

1. If you are using service EPGs for the consumer or provider, create the service EPG and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- The service EPG as the consumer could have the following settings:
 - **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, Azure Kubernetes Services (AKS) would be a supported service type with a Cloud Native Managed deployment type.
 - **Deployment type:** Cloud Native Managed
 - **Access type:** Private

- The service EPG as the provider could have the following settings:
 - **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure Storage File` would be a supported service type with a `Cloud Native` deployment type.
 - **Deployment type:** `Cloud Native`
 - **Access type:** `Private`

2. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall
- In the **Service Node** window for the hub NLB:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the provider side.
- In the **Service Node** window for the Third-Party Firewall, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.

3. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

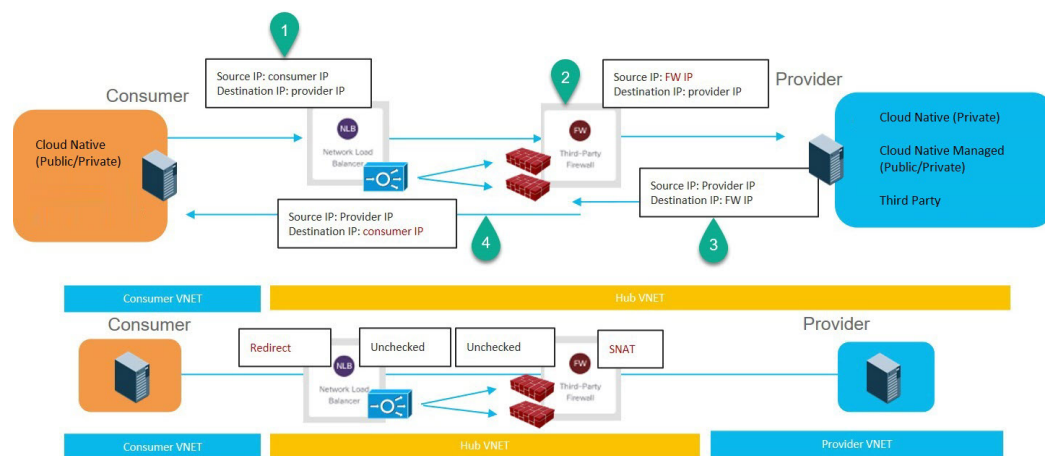
Two-Node Service Graph for East-West with SNAT Option

This use case has a two-node service graph, where the service nodes are a network load balancer and a firewall. Redirect is enabled only on the consumer side and SNAT is enabled on the firewall in this use case.

In this use case, the consumer and provider could be cloud EPGs or service EPGs.

These actions take place in this use case, as shown in the following figure:

1. Traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
2. SNAT is performed on the firewall.
3. The return traffic comes back to the firewall SNAT IP address.
4. At this point in the return direction, the return traffic doesn't go through the network load balancer.



To configure this use case:

1. If you are using service EPGs for the consumer or provider, create the service EPG and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- The service EPG as the consumer could have the following settings:
 - **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure Active Directory Domain Services` would be a supported service type with a `Cloud Native Managed` deployment type.
 - **Deployment type:** `Cloud Native Managed`
 - **Access type:** `Private`
- The service EPG as the provider could have the following settings:
 - **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure Storage File` would be a supported service type with a `Cloud Native` deployment type.
 - **Deployment type:** `Cloud Native`

- **Access type:** `Private`

2. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall
- In the **Service Node** window for the Network Load Balancer:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, leave the boxes unchecked.
- In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** option.

3. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

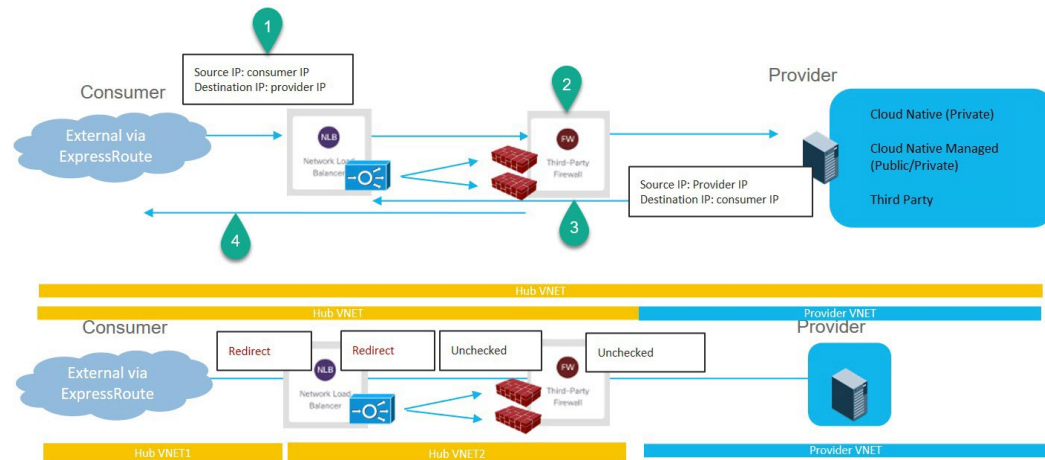
Two-Node Service Graph for Inbound Traffic via Express Route Gateway

This use case has a two-node service graph, where the service nodes are a network load balancer and a firewall. Redirect is enabled on both the consumer and the provider side in this use case.

In this use case, the service EPG is the provider, and the express route is on the consumer side.

These actions take place in this use case, as shown in the following figure:

1. Traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
2. SNAT is *not* performed on the firewall in this use case.
3. The return traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
4. At this point in the return direction, the return traffic comes back to the consumer.



To configure this use case:

1. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, Azure Active Directory Domain Services would be a supported service type with a Cloud Native Managed deployment type.
- **Deployment type:** Cloud Native Managed
- **Access type:** Private

2. Deploy the express route gateway on the consumer side.

See [Deploying Express Route Gateway Using Redirect, on page 291](#) for those procedures.

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:

- Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer
 - Third-Party Firewall
 - In the **Service Node** window for the Network Load Balancer:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Service Node** window for the Third-Party Firewall, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.
4. Deploy the Layer 4 to Layer 7 services.
- See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI](#), on page 236 for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

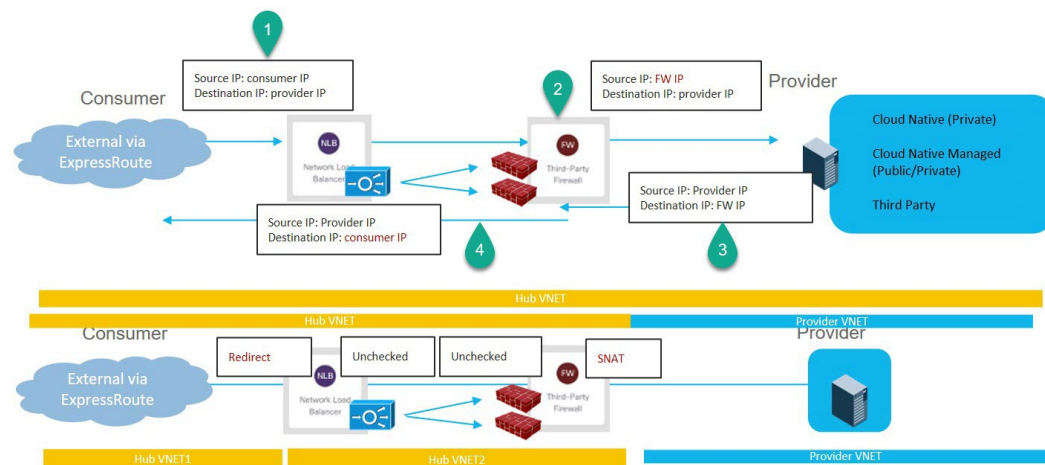
Two-Node Service Graph for Inbound Traffic via Express Route Gateway with SNAT Option

This use case has a two-node service graph, where the service nodes are a network load balancer and a firewall. Redirect is enabled only on the consumer side and SNAT is enabled on the firewall in this use case.

In this use case, the service EPG is the provider, the express route is on the consumer side.

These actions take place in this use case, as shown in the following figure:

1. Traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
2. SNAT is performed on the firewall.
3. The return traffic comes back to the firewall SNAT IP address.
4. At this point in the return direction, the return traffic doesn't go through the network load balancer.



To configure this use case:

1. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Redis Cache` would be a supported service type with a `Cloud Native Managed` deployment type.
- **Deployment type:** `Cloud Native Managed`
- **Access type:** `Private`

2. Deploy the express route gateway on the consumer side.

See [Deploying Express Route Gateway Using Redirect, on page 291](#) for those procedures.

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:

- Network Load Balancer
 - Third-Party Firewall
- In the **Service Node** window for the Network Load Balancer:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, leave the boxes unchecked.
 - In the **Service Node** window for the Third-Party Firewall:
 - In the **Consumer Connector Type** field, leave the boxes unchecked.
 - Because the firewall performs SNAT when sending traffic to the internet in this use case, in the **Provider Connector Type** field, place a check in the box next to the **SNAT** option.
4. Deploy the Layer 4 to Layer 7 services.
- See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

Three-Node Service Graph for Inbound Traffic via Express Route Gateway

This use case has a three-node service graph, where the service nodes are:

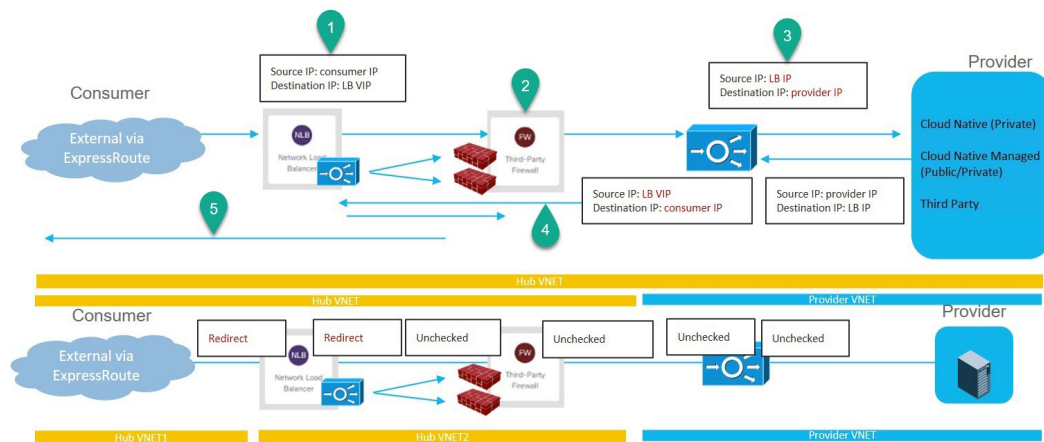
- First service device: Network load balancer in the hub VNet
- Second service device: Firewall in the hub VNet
- Third service device: Application load balancer in the hub or spoke VNet

Redirect is enabled on both the consumer and the provider side in this use case.

In this use case, the service EPG is the provider. The express route is on the consumer side, and the consumer can be a cloud EPG or a service EPG.

These actions take place in this use case, as shown in the following figure:

1. Traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
2. SNAT is *not* performed on the firewall in this use case.
3. Traffic moves to the third service device, the application load balancer, which has SNAT configured.
4. The return traffic is redirected to the network load balancer, which then load balances the traffic to the firewall.
5. At this point in the return direction, the return traffic comes back to the consumer.



To configure this use case:

1. Create the service EPG on the provider side and assign the appropriate deployment type and access type to the service EPG.

See [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#) for those procedures, with these settings:

- **Service Type:** Any supported service type, depending on the deployment type (see [Cloud Service Endpoint Groups, on page 38](#) for more information). For example, `Azure ApiManagement Services` would be a supported service type with a `Cloud Native Managed` deployment type.
- **Deployment type:** `Cloud Native Managed`
- **Access type:** `Private`

2. Deploy the express route gateway on the consumer side.

See [Deploying Express Route Gateway Using Redirect, on page 291](#) for those procedures.

3. Configure the service graph.

See [Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227](#) for those procedures.

As part of the redirect configuration for this use case, you would make the following selections:

- In the **Create Device** window, first create the service devices for the hub VNet:
 - In the **Tenant** field, choose the **infra** tenant.
 - Choose the type of service device in the **Service Type** field:
 - Choose **Network Load Balancer** as the **Service Type**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet created in the secondary VRF.
 - Choose **Third-Party Firewall** as the **Service Type**, and in the **VRF** field, choose the secondary VRF.
- In the **Create Device** window, next create the service devices for the provider VNet:
 - In the **Tenant** field, choose the provider tenant.

- In the **Service Type** field, choose **Application Load Balancer**, and in the **Subnets** area, click **Add Subnet**, then choose the appropriate region, cloud context profile, and the subnet for the provider VRF.



Note A third party load balancer can be used in place of an internal NLB. Choose **Third Party Load Balancer** as the **Service Type**. Choose the **VRF** and set the interface(s) details by clicking **Add Interface**.

- In the **Create Service Graph** window, drag-and-drop the following service devices, in this order:
 - Network Load Balancer (for the hub VNet)
 - Third-Party Firewall (for the hub VNet)
 - Application Load Balancer (for the provider VNet)
- In the **Service Node** window for the Network Load Balancer in the hub VNet:
 - In the **Consumer Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the consumer side.
 - In the **Provider Connector Type** field, place a check in the box next to the **Redirect** option to enable the redirect function on the provider side.
- In the **Service Node** window for the Third-Party Firewall, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.
- In the **Service Node** window for the Network Load Balancer in the provider VNet, leave the boxes unchecked for the **Consumer Connector Type** and the **Provider Connector Type**.



Note Ensure SNAT is configured on the third party load balancers.

4. Deploy the Layer 4 to Layer 7 services.

See [Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI, on page 236](#) for those procedures. Attach the contract that exists between the consumer and provider to the service graph.

Guidelines and Limitations for Redirect

Following are the guidelines and limitations for redirect:

- All the Layer 4 to Layer 7 services devices should have their own dedicated subnet.
- Intra VRF Layer 4 to Layer 7 services redirection within a region:
 - Layer 4 to Layer 7 services redirect is not supported for east-west deployment when the consumer EPG and provider EPG are in the same VNet.

- Layer 4 to Layer 7 services redirect is supported for north-south deployment if the external EPG is a provider EPG, regardless of whether the consumer EPG and provider EPG are in same VNet or not.
- Intra-VRF Layer 4 to Layer 7 services redirection across regions:
 - Inter-Region Layer 4 to Layer 7 services redirection are supported. However, the Consumer EPG and the Provider EPG should not stretch.
 - A region shouldn't have both a consumer EPG and a provider EPG in the same VRF. For example, if region 1 has a consumer EPG only and region 2 has a provider EPG only, this is supported, but region 1 can't have both the consumer EPG and the provider EPG.
 - Consumer and Provider EPG should be a subnet-based EPG.
- For the inter-region service graphs with Layer 4 to Layer 7 services redirection, service devices should be deployed in the provider EPG's region. If provider EPG is stretched across regions, service devices should be deployed in each region.
- For the external EPG as provider, service devices need to be deployed in the region local to consumer EPG. If the consumer EPG is stretched across regions, service devices should be deployed in each region.
- Between a consumer VNet and a provider EPG, only one redirect device can be inserted through a service graph. For example, if consumer EPG1 and consumer EPG2 are in a consumer VNet, and a provider EPG3 is in a provider VNet, you must use the same redirect device for a contract between EPG1 and EPG3, and a contract between EPG2 and EPG3.



Note The limitation is because of the cloud provider allows only one next hop for a given destination in user-defined routes.

- The following table provides information on the specific redirect configurations that are supported or unsupported, where:
 - NLB stands for network load balancer
 - ALB stands for application load balancer
 - FW stands for firewall



Note Redirection to a third party load balancer is not supported.

Service Chain Option	Spoke-to-Spoke		Spoke-to-External (consumer is spoke)		External-to-Spoke (consumer is external)	
	Intra-VNet	Inter-VNet	Intra-VNet	Inter-VNet	Intra-VNet	Inter-VNet
NLB/ALB ¹ LB(SNAT) ¹	Supported	Supported	Not supported	Not supported	Supported	Supported

Service Chain Option	Spoke-to-Spoke		Spoke-to-External (consumer is spoke)		External-to-Spoke (consumer is external)	
FW (no SNAT) ²	Not supported	Supported	Not supported	Not supported	Not supported	Not supported
FW (SNAT) ³	Supported	Supported	Supported	Supported	Not supported	Not supported
<ul style="list-style-type: none"> • NLB²-FW(no SNAT)¹ • NLB²-FW(no SNAT)¹-NLB/ALB¹ • NLB²-FW(no SNAT)¹-LB(SNAT)¹ 	Not supported	Supported	Not supported	Not supported	Not supported	Not supported
NLB ⁴ -FW(SNAT) ⁵	Not supported	Supported	Supported	Supported	Not supported	Not supported
NLB/ALB ¹ -FW(SNAT+DNAT) ⁶ -NLB/ALB ¹ NLB/ALB ¹ -FW(ANT+DNAT) ⁶ -LB(SNAT) ¹ (No redirection)	Supported	Supported	Not Supported	Not Supported	Supported	Supported
NLB ¹ -LB(SNAT) ¹ (No redirection)	Supported	Supported	Not Supported	Not Supported	Supported	Supported

¹ Unchecked on both consumer and provider connector or options are not applicable for ALB.

² Redirect is enabled on both consumer and provider connector.

³ Redirect is enabled on consumer connector. SNAT is enabled on provider connector.

⁴ Redirect is enabled on consumer connector. Unchecked on provider connector.

⁵ Unchecked on consumer connector. SNAT is enabled on provider connector.

⁶ Unchecked on consumer connector. SNAT+DNAT is enabled on provider connector.

Adding a New CIDR to a Secondary VRF Using the Cisco Cloud Network Controller GUI

In some situations, you might have to disable VNet peering before adding new CIDRs or editing existing CIDRs in a secondary VRF. This is due to a limitation in Azure, where you cannot update a CIDR on a VNet if it has active VNet peerings. To add the CIDRs, you first have to remove VNet peerings for that VNet, then you can update the CIDRs. Once you have updated the CIDRs, you can then re-enable the VNet peerings.

These procedures provide instructions for disabling Hub Network Peering, which removes all of the VNet peerings associated with a particular infra VNet.

- If you have an additional CIDR already created on the infra VNet, but you simply need to add additional subnets to that existing CIDR, you do not have to disable Hub Network Peering for that particular infra VNet before adding those subnets. To add additional subnets to an existing CIDR:

1. Navigate to the appropriate cloud context profile in that case (**Application Management > Cloud Context Profiles**).

2. Double-click the cloud context profile where you want to add a subnet to an existing CIDR, then go to [Step 10, on page 226](#) to add the new subnets to an existing CIDR.
- If you are adding a new CIDR in the infra VNet, or if you are deleting a CIDR or editing a CIDR in the infra VNet in some other way (other than adding subnets), then you must disable Hub Network Peering for that particular infra VNet. You will then re-enable Hub Network Peering again after you have added the CIDR. The following procedure provides those instructions.



Note If you are adding new CIDRs to a secondary VRF and you have a multi-site deployment where you are running on the following releases:

- Release 5.2(1) or later on the Cisco Cloud Network Controller
- Release 3.3 or later on the Nexus Dashboard Orchestrator

After you have added the new CIDRs and re-enabled Hub Network Peering, wait at least five minutes for the CIDRs to come up before refreshing the site on Nexus Dashboard Orchestrator and deploying the infra configuration from the Nexus Dashboard Orchestrator. It will take time for the CIDRs to get deployed on Azure, so newly-added CIDRs might not get propagated to the remote site through Nexus Dashboard Orchestrator if you do not wait at least minutes before refreshing the site and deploying the infra configuration from the Nexus Dashboard Orchestrator.

If you see the following error message after you deploy the infra configuration from the Nexus Dashboard Orchestrator:

```
Invalid configuration CT_Remotectx_cidr: Remote Site CIDR
```

This means that you did not wait long enough before deploying the infra configuration from the Nexus Dashboard Orchestrator and the newly-added CIDRs did not get propagated to the remote site. If this happens:

1. Disable Hub Network Peering on the Cisco Cloud Network Controller
2. Refresh the site on Nexus Dashboard Orchestrator, then deploy the infra configuration from the Nexus Dashboard Orchestrator
3. Re-enable Hub Network Peering on the Cisco Cloud Network Controller
4. Wait at least five minutes (or a longer period than you waited for previously), then refresh the site and deploy the infra configuration from the Nexus Dashboard Orchestrator again

Step 1 Log in to the Cisco Cloud Network Controller, if you are not logged in already.

Step 2 In the left navigation bar, navigate to **Application Management > Cloud Context Profiles**.

The existing cloud context profiles are displayed.

Step 3 Double-click the cloud context profile where you want to disable Hub Network Peering.

The overview window for that cloud context profile appears. You should see **Enabled** in the **Hub Network Peering** area in this overview window, which indicates that Hub Network Peering is enabled.

Step 4 Click the pencil icon to edit this cloud context profile.

The **Edit Cloud Context Profile** window appears.

Step 5 In the **Edit Cloud Context Profile** window, locate the **Hub Network Peering** field and click the check box to remove the checkmark from the **Enabled** field.

Disabling the **Hub Network Peering** option does not remove VNet peering at the global level, but rather removes all of the VNet peerings associated with this particular infra VNet.

Step 6 Click **Save**.

The overview window for that cloud context profile appears again. You should see **Disabled** in the **Hub Network Peering** area in this overview window, which indicates that Hub Network Peering is now disabled.

Step 7 To add a new CIDR, click the pencil icon to edit this cloud context profile again.

The **Edit Cloud Context Profile** window appears again.

Step 8 Click **Add CIDR**.

The **Add CIDR** dialog box appears.

Step 9 Add the new CIDR in the **CIDR Block Range** field.

Do not click the box in the **Primary** field (do not put a check in the box next to **yes** in the **Primary** field).

Step 10 Click **Add Subnet** and enter the necessary subnet addresses in the **Address** field.

Continue to click **Add Subnet** for additional subnets, if necessary.

Step 11 When you have finished adding all of the necessary information in the **Add CIDR** window, click **Add**.

The **Edit Cloud Context Profile** window appears again.

Step 12 Confirm the information in the **Edit Cloud Context Profile** window, then click **Save**.

The overview window for that cloud context profile appears. You should now see the new CIDR listed in the **CIDR Block Range** area.

Step 13 If you disabled Hub Network Peering at the beginning of these procedures, re-enable it at this time.

a) Click the pencil icon to edit this cloud context profile.

The **Edit Cloud Context Profile** window appears.

b) In the **Edit Cloud Context Profile** window, locate the **Hub Network Peering** field and click the check box to add the checkmark in the **Enabled** field to re-enable VNet peerings for this particular infra VNet.

c) Click **Save**.

The overview window for that cloud context profile appears again. You should see **Enabled** in the **Hub Network Peering** area in this overview window, which indicates that Hub Network Peering is now re-enabled again.

As described previously, if you were to go to the Azure portal at this point, you will see any additional CIDRs and subnets that you added in these procedures in the overlay-1 VNet in Azure, which is the correct and expected behavior.

Deploying a Service Graph

The service graph enables you to define how traffic flows between devices, how the traffic comes into the network, which devices the traffic passes through, and how the traffic leaves the network.

The Service graph can be deployed in two ways:

- Single node service graph: Only one device is deployed.
- Multinode service graph: Upto three nodes can be added to the service chain.

Before you can deploy a service graph in either a single node or multinode, you must configure the following:

1. A tenant
2. An application profile
3. A consumer EPG
4. A provider EPG
5. A VRF
6. A cloud context profile
7. A contract with a filter

Deploying a Service Graph Using the GUI

The following sections describe how to deploy a service graph using the GUI.

Creating Service Devices Using The Cisco Cloud Network Controller GUI

Before you begin

This section explains how to create service devices that can be used in a service graph through the Cisco Cloud Network Controller GUI.

-
- Step 1** Click the **Intent** icon. The **Intent** menu appears.
- Step 2** Click the drop-down arrow below the **Intent** search box and choose **Application Management**.
A list of **Application Management** options appear in the **Intent** menu.
- Step 3** From the **Application Management** list in the **Intent** menu, click **Services > Devices > Create Device**. The **Create Device** dialog box appears.
- Step 4** Enter the appropriate values in each field as listed in the following *Create Device Dialog Box Fields* table then continue.
Refer to the following tables for information specific to each type of service device.
- For an Application Load Balancer, see [4.a, on page 227](#).
 - For a Network Load Balancer, see [4.b, on page 229](#).
 - For a Third Party Load Balancer, see [4.c, on page 231](#).
 - For a Third Party Firewall, see [4.d, on page 232](#).
- a) Enter the necessary information for an Application Load Balancer:

Properties	Description
General	
Name	Enter the name of the device.
Tenant	To choose a tenant: <ol style="list-style-type: none"> 1. Click Select Tenant. The Select Tenant dialog appears. 2. From the column on the left, click to choose a tenant. 3. Click Select. You return to the Create Device dialog box.
Settings	
Service Type	Choose the device type: <ul style="list-style-type: none"> • Application Load Balancer
ALB SKU	Choose from: <ul style="list-style-type: none"> • Standard • Standard V2
VM Instance Count	Enter a number in the <i>VM Instance Count</i> text box. <p>Note This is applicable only for the Application Gateway.</p>
VM Instance Size	Click the radio button for the VM instance size you want to choose: large , medium , or small . <p>Note This is applicable only for the Application Gateway.</p>
Scheme	Choose Internet Facing or Internal . <ul style="list-style-type: none"> • Internet Facing— This is used for configuring a public IP for the balancer. This is assigned by Azure. • Internal—Click to choose either Dynamic or Static under IP Address Assignment. <ul style="list-style-type: none"> • Dynamic—Dynamic IP addresses are assigned by Azure. Dynamic IP addresses change each time the VMs boot up. • Static—Enter an IP address based on the CIDRs defined in Cloud Context Profile and check that the IP address is in the same subnet as the ALB. <p>ALB SKU Standard supports static and dynamic IP addresses. ALB SKU Standard V2 support static IP addresses only.</p>

Properties	Description
Subnet	<p>To choose a subnet:</p> <ol style="list-style-type: none"> 1. Click Select Region. The Select Region dialog box appears. From the Select Region dialog, click to choose a region in the left column then click Select. 2. Click Select Cloud Context Profile. The Select Cloud Context Profile dialog box appears. 3. Click Select Subnet. The Select Subnet dialog box appears. The Static IP Addresses text box is displayed. Enter the IP address of the load balancer. Click the tick mark on the right to confirm. 4. To add additional subnets, repeat steps a-c.

- b) Enter the necessary information for a Network Load Balancer:

Table 38: Create Device Dialog Box Fields for Network Load Balancer

Properties	Description
General	
Name	Enter the name of the load balancer.
Settings	
Service Type	<p>Choose the device type:</p> <ul style="list-style-type: none"> • Network Load Balancer
Allow All Traffic	<p>Determine if you want to enable the Allow All Traffic option.</p> <p>Enabling the Allow All Traffic option will allow all inbound and outbound access to the subnet on which the interface belongs. See About Allow All Traffic Option, on page 183 for more information.</p> <p>Note Ensure that this does not present a security risk before enabling this option.</p> <ul style="list-style-type: none"> • If you want to allow all traffic, in the Allow All Traffic area, click the box next to the Enabled field. • If you do not want to allow all traffic, in the Allow All Traffic area, leave the box unchecked (unselected) next to the Enabled field.

Properties	Description
Scheme	<p>Choose Internet Facing or Internal.</p> <ul style="list-style-type: none"> • Internet Facing— This is used for configuring a public IP for the balancer. This is assigned by Azure. <p>Choosing the Internet Facing option can be used to configure a single default public frontend IP address or multiple public frontend IP addresses, depending on the selection that you make in the Frontend IP Names field later in this page.</p> <ul style="list-style-type: none"> • Internal—Click to choose either Dynamic or Static under IP Address Assignment. <ul style="list-style-type: none"> • Dynamic—Dynamic IP addresses are assigned by Azure. Dynamic IP addresses change each time the VMs boot up. • Static—Enter an IP address based on the CIDRs defined in Cloud Context Profile and check that the IP address is in the same subnet as the NLB. Static IP addresses are associated to load balancers. <p>Note Cisco Cloud Network Controller creates standard SKU NLBs only.</p>
Custom Resource Group	Enter a name for a custom resource group, if necessary.
Subnets	<p>To choose a subnet:</p> <ol style="list-style-type: none"> 1. Click + Add Subnet. 2. Click Select Region. The Select Region dialog box appears. <p>From the Select Region dialog, click to choose a region in the left column, then click Select.</p> <ol style="list-style-type: none"> 3. Click Select Cloud Context Profile. The Select Cloud Context Profile dialog box appears. <p>From the Select Cloud Context Profile dialog, click to choose a cloud context profile in the left column, then click Select.</p> <ol style="list-style-type: none"> 4. Click Select Subnet. The Select Subnet dialog box appears. <p>From the Select Subnet dialog, click to choose a subnet in the left column, then click Select.</p> <ol style="list-style-type: none"> 5. Click the tick mark on the right to confirm. 6. To add additional subnets, click + Add Subnet again and repeat these steps.
Advanced Settings	<p>Click the down arrow to expand the Advanced Settings area.</p> <p>The following entries are displayed.</p>

Properties	Description
Frontend IP Names	<p>Support is available for configuring multiple frontend IP addresses for an internet-facing Network Load Balancer.</p> <ul style="list-style-type: none"> • By default, a single frontend IP address will automatically be created for the internet-facing Network Load Balancer. • If you want additional frontend IP addresses for the internet-facing Network Load Balancer, click + Add Frontend IP Name. For more information, see Understanding Configurations for Multiple Frontend IP Addresses on the Azure Network Load Balancer, on page 181. <p>Adding frontend IP names in this area tells Azure that you want multiple frontend IP addresses to be assigned to this internet-facing Network Load Balancer. Each frontend IP name that you enter in this area results in a single additional frontend IP address.</p> <p>The public frontend IP addresses in this area (the default frontend IP address and any additional frontend IP addresses) are then assigned by Azure.</p> <ol style="list-style-type: none"> 1. Click + Add Frontend IP Name to add a name for an additional frontend IP address that you want Azure to assign for the Network Load Balancer. 2. Enter a name for the additional frontend IP address, then click the tick mark on the right to confirm the new frontend IP name. 3. Click + Add Frontend IP Name again to add additional names for additional frontend IP addresses that you want Azure to assign for the Network Load Balancer. <p>For example, assume that you want a total of three frontend IP addresses for the internet-facing Network Load Balancer:</p> <ul style="list-style-type: none"> • The first of the three frontend IP addresses is automatically assigned by default. • You would then click + Add Frontend IP Name twice, entering two separate frontend IP names (for example, <code>frontend2</code> and <code>frontend3</code>), to signal to Azure that you want Azure to assign a total of three frontend IP addresses for the internet-facing Network Load Balancer. <p>To see the frontend IP addresses associated with the default and configured frontend IP names:</p> <ol style="list-style-type: none"> 1. Navigate to Application Management > Services > Devices. 2. Double-click on the configured service device to bring up the Overview page for that service device. 3. Click on Cloud Resources > Frontend IP Names. <p>The default frontend IP address will be shown with the <code>Default</code> tag in this details page.</p>

- c) Enter the necessary information for a Third Party Load Balancer:

Table 39: Create Device Dialog Box Fields for Third Party Load Balancer

Properties	Description
General	
Name	Enter the name of the device.
Tenant	To choose a tenant: <ol style="list-style-type: none"> 1. Click Select Tenant. The Select Tenant dialog appears. 2. From the column on the left, click to choose a tenant. 3. Click Select. You return to the Create Device dialog box.
Settings	
Service Type	Choose the device type: <ul style="list-style-type: none"> • Third Party Load Balancer
Creation Mode	Select Selectors . VRF and Interfaces fields are displayed.
VRF	Click Select VRF . In the Select VRF dialog box that opens, click to choose a VRF in the left column. Click Select .
Interface	Click Add Interface . The Interfaces window is displayed. <ol style="list-style-type: none"> 1. Enter a name for the external interface in the Interface Settings field. 2. Click Add Interface selector. 3. In the Interface Selector Settings page, enter the name of the interface. 4. In the Match Expressions field, click Match Expression and select <ul style="list-style-type: none"> • Key: This can be IP, region or a custom based tag selector. • Operator: This can be equal, not equals, in, not in, has key, or does not have key. • Value: IP address of the external or internal network of third party load balancer. 5. Click the tick mark to add the interface and then click Save (Interfaces window). 6. Click Save (Create Device window). Click Add Interface and repeat steps a- e to add more interfaces. <p>Note Third party load balancer interfaces should be configured with subnet-based selectors when deployed in a multi-node service graph.</p>

d) Enter the necessary information for a Third Party Firewall:

Table 40: Create Device Dialog Box Fields for Third Party Firewall

Properties	Description
General	
Name	Enter the name of the device.
Settings	
Service Type	<p>Choose the device type:</p> <ul style="list-style-type: none"> • Third party firewall <p>Note Third party firewall cannot be the first device in a multinode service graph.</p>
VRF	<p>To choose a VRF:</p> <ol style="list-style-type: none"> 1. Click Select VRF. The Select VRF dialog box appears. 2. From the Select VRF dialog, click to choose a VRF in the left column then click Select.
Interfaces	<p>Click Add Interface.</p> <p>The Settings page appears.</p> <ol style="list-style-type: none"> 1. In the Name field, enter the name of the interface. 2. Determine if you want to enable the Allow All Traffic option. <p>Enabling the Allow All Traffic option will allow all inbound and outbound access to the subnet on which the interface belongs. See About Allow All Traffic Option, on page 183 for more information.</p> <p>Note Ensure that this does not present a security risk before enabling this option.</p> <ul style="list-style-type: none"> • If you want to allow all traffic, in the Allow All Traffic area, click the box next to the Enabled field. • If you do not want to allow all traffic, in the Allow All Traffic area, leave the box unchecked (unselected) next to the Enabled field. <ol style="list-style-type: none"> 3. Click Add Interface Selector. 4. Enter the name of the interface selector. 5. Click on Match Expressions and select <ul style="list-style-type: none"> • Key: This can be IP, region or a custom based tag selector. • Operator: This can be equal, not equals, in, not in, has key, or does not have key. • Value: IP address of the app, web, internal network, management network, or external network. 6. Click Add. 7. Repeat steps a- f to add more interfaces.

Step 5 Click **Save** when finished.

Step 6 The **Create Service Graph** dialog box appears. Click on the **Create another Third Party Firewall** to create another device. The **Create Device** dialog box appears.

Note The UI usually asks to create a previously created device. However, on clicking it we return back to the **Create Device** page. Here we can choose the device that needs to be created. The first device should never be the Third Party Firewall.

Creating a Service Graph Template Using the Cisco Cloud Network Controller GUI

This section explains how to configure a service graph template for a single node or a multinode, using the Cisco Cloud Network Controller GUI.

Before you begin

You have already created the devices.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

Step 3 From the **Application Management** list in the **Intent** menu, click **Services > Service Graph > Create Service Graph**. The **Create Service Graph** pop-up appears. Click on **Let's Get Started**.

Step 4 Enter the appropriate values in each field as listed in the following *Create Service Graph Dialog Box Fields* table then continue.

Table 41: Create Service Graph Dialog Box Fields (for single node)

Properties	Description
General	
Name	Enter the name of service graph template.
Tenant	To choose a tenant: <ol style="list-style-type: none"> a. Click Select Tenant. The Select Tenant dialog appears. b. From the column on the left, click to choose a tenant. c. Click Select. You return to the Create Service Graph dialog box.
Description	Enter a description of the service graph template.
Settings	

Properties	Description
Select a Device	<p>To choose a device:</p> <ol style="list-style-type: none"> Click Select Device. The Select Device dialog appears. From the column on the left, click to choose a device. Drag and drop the device in the Drop Device space below. This will open a small window where the actual device for this device type can be selected. Click Select. You return to the Create Service Graph dialog box.

Table 42: Create Service Graph Dialog Box Fields (for multinode)

Properties	Description
General	
Name	Enter the name of service graph template.
Tenant	<p>To choose a tenant:</p> <ol style="list-style-type: none"> Click Select Tenant. The Select Tenant dialog appears. From the column on the left, click to choose a tenant. Click Select. You return to the Create Service Graph dialog box.
Description	Enter a description of the service graph template.
Settings: Based on the required topology, drag and drop the devices in the box below	
Application Load Balancer	<ol style="list-style-type: none"> Drag and drop the Application load balancer device into the box below. In the Service node dialog box, click on the Select Application Load Balancer and click to choose a Application Load Balancer in the left column then click Add.
Third Party Firewall	<ol style="list-style-type: none"> Drag and drop the Third Party Firewall next to the device in the box below. In the Service node dialog box, click on the Third Party Firewall and click to choose a Third Party Firewall in the left column then click Add. <p>Note Third Party Firewall cannot be the first node on the service graph.</p> If you want to enable the user-based redirect function on the <i>consumer</i> side of the Third Party Firewall, in the Consumer Connector Type field, place a check in the box next to the Redirect option. If you want to enable the user-based redirect function on the <i>provider</i> side of the Third Party Firewall, in the Provider Connector Type field, place a check in the box next to the Redirect option. In the Provider Connector Type, place a check next to the applicable option. Refer to About Layer 4 to Layer 7 Service Redirect for information. Click Add.

Properties	Description
Network Load Balancer	<ol style="list-style-type: none"> Drag and drop the Network load balancer device into the box below. In the Service node dialog box, click on the Select Network Load Balancer and click to choose a Network Load Balancer in the left column then click Add. If you want to enable the user-based redirect function on the <i>consumer</i> side of the network load balancer, in the Consumer Connector Type field, place a check in the box next to the Redirect option. If you want to enable the user-based redirect function on the <i>provider</i> side of the network load balancer, in the Provider Connector Type field, place a check in the box next to the Redirect option. Click Add.
Third Party Load Balancer	<ol style="list-style-type: none"> Drag and drop the third party load balancer device into the box below. In the Service node dialog box, click Select Third Party Load Balancer and click to choose a third party load balancer in the left column. Click Select Consumer Interface. Select the interface marked as external. Click Select Provider Interface. Select the interface marked as internal. Click Add.

Step 5 Click **Save** when finished.

Step 6 The **EPG Communication** dialog box appears. Click on the **Go to details** to verify the Service Graph template.

Deploying Layer 4 to Layer 7 Services Using the Cisco Cloud Network Controller GUI

This section explains how to deploy Layer 4 to Layer 7 services. This procedure is applicable for single node as well as multinode deployments.

Before you begin

- You have configured the devices.
- You have configured a service graph.

Step 1 Click the **Intent** icon. The **Intent** menu appears.

Step 2 Click the drop-down arrow below the **Intent** search box and choose **Configuration**.

A list of **Configuration** options appear in the **Intent** menu.

Step 3 From the **Configuration** list in the **Intent** menu, click **EPG Communication**. The **EPG Communication** dialog box appears with the **Consumer EPGs**, **Contract**, and **Provider EPGs** information.

Step 4 To choose a contract:

- Click **Select Contract**. The **Select Contract** dialog appears.

- b) In the pane on the left side of the **Select Contract** dialog, click to choose a contract then click **Select**. The **Select Contract** dialog box closes.

Step 5

To add a consumer EPG:

- a) Click **Add Consumer EPGs**. The **Select Consumer EPGs** dialog appears.
- b) In the pane on the left side of the **Select Consumer EPGs** dialog, click the check box to choose a cloud EPG (for an internal facing load balancer) or a cloud external EPG (for an internet facing load balancer) then click **Select**. The **Select Consumer EPGs** dialog box closes.

Step 6

To add a provider EPG:

- a) Click **Add Provider EPGs**. The **Select Provider EPGs** dialog appears.
- b) In the pane on the left side of the **Select Provider EPGs** dialog, click the check box to choose a provider EPG then click **Select**. The **Select Provider EPGs** dialog box closes.

Step 7

To choose a service graph:

- a) From the **EPG Communication Configuration** dialog, click **Select Service Graph**. The **Select Service Graph** dialog box appears.
- b) In the pane on the left side of the **Select Service Graph** dialog, click to choose a service graph then click **Select**. The **Select Service Graph** dialog box closes.

Step 8

Under **Service Graph Preview**, click **Add Cloud Load Balancer Listener**. The **Add Cloud Load Balancer Listener** dialog appears that enables you to add listeners.

Listeners are the ports and protocols that the device will work on.

Step 9

Enter the appropriate values in each field as listed in the following *Add Cloud Load Balancer Listener Dialog Box Fields* table then continue.

Table 43: Add Cloud Load Balancer Listener Dialog Box Fields For Application Gateway

Properties	Description
Name	Enter the name of the listener.
Port	Enter the port that the device will accept traffic on.
Protocol	For Application Gateway, click to choose HTTP or HTTPS .
Security Policy	Click the drop-down list and choose a security policy (only available when HTTPS is chosen).
SSL Certificate	<p>To choose an SSL certificate(only available when HTTPS is chosen):</p> <ol style="list-style-type: none"> Click Add SSL Certificates. Click to place a check mark in the check box of the certificates you want to add. Choose a key ring: <ol style="list-style-type: none"> Click Select Key Ring. The Select Key Ring dialog appears. From the Select Key Ring dialog, click to choose a key ring in the left column then click Select. The Select Key Ring dialog box closes. Click the Certificate Store drop-down list and choose a certificate. <p>Note A listener can have multiple certificates.</p>

Properties	Description
Add Rule	To add rule settings to the device listener, click Add Rule . A new row appears in the Rules list an the Rules Settings fields are enabled.
Rule Settings	<p>The Rule Settings pane contains the following options:</p> <ul style="list-style-type: none"> • Name—Enter a name for the rule. • Host—Enter a hostname to create a host-based condition. When a request is made for this hostname, the action you specify is taken. • Path—Enter a path to create a path-based condition. When a request is made for this path, the action you specify is taken. • Type—The action type tells the device which action to take. The action type options: <ul style="list-style-type: none"> • Return fixed response—Returns a response using the following options: <ul style="list-style-type: none"> • Fixed Response Body—Enter a response message. • Fixed Response Code—Enter a response code. • Fixed response Content-Type—Choose a content type. • Forward—Forwards traffic using the following options: <ul style="list-style-type: none"> • Port—Enter the port that the device will accept traffic on. • Protocol—Click to choose HTTP or HTTPS. • Provider EPG—The EPG with the web server that handles the traffic. • EPG—To choose an EPG: <ol style="list-style-type: none"> a. Click Select EPG. The Select EPG dialog box appears. b. From the Select EPG dialog ox, click to choose an EPG in the left column then click Select. The Select EPG dialog box closes. • Redirect—Redirects requests to another location using the following options: <ul style="list-style-type: none"> • Redirect Code—Click the Redirect Code drop-down list and choose a code. • Redirect Hostname—Enter a hostname for the redirect. • Redirect Path—Enter a redirect path. • Redirect Port—Enter the port that the device will accept traffic on. • Redirect Protocol—Click to the Redirect Protocol drop-down list and choose HTTP, HTTPS, or Inherit. • Redirect Query—Enter a redirect query.

Properties	Description
Health Checks	<p>The Application load balancer performs health checks on its backend pool targets for high availability. This can be configured under health checks:</p> <ul style="list-style-type: none"> • Protocol-Click to choose HTTP or HTTPS. • Path- Enter the path. Default is / • Port-Enter a port on which health checks should be performed. • Advanced Settings- <ul style="list-style-type: none"> Unhealthy Threshold-Configure this threshold to determine when a backend target is advertised as unhealthy. • Timeout- Enter the value for health check timeout. • Interval-Enter a time in seconds to determine at what intervals checks should be performed. • Success Code- Enter the success code. Default is 200-399. • Use host from rule- Click on the checkbox if the hostname needs to be picked from the rule. • Host- If Use host from rule is not checked, provide the hostname to be used for health check. <p>Click Add Rule when finished.</p>

Table 44: Add Cloud Load Balancer Listener Dialog Box Fields for Network Load Balancer

Properties	Description
Name	Enter the name of the listener.
Protocol	Click to choose TCP or UDP .
Port	Enter the port that the device will accept traffic on.
Frontend IP Name	<p>Select the frontend IP address where you will be configuring the cloud load balancer listener.</p> <ul style="list-style-type: none"> • Use default IP: Select this option to configure the cloud load balancer listener on the default frontend IP address. • <frontend_ip_name>: Select this option to configure the cloud load balancer listener on the additional frontend IP address associated with the frontend IP name that you configured when you created the service device for this internet-facing Network Load Balancer in Creating Service Devices Using The Cisco Cloud Network Controller GUI, on page 227. <p>For more information, see Understanding Configurations for Multiple Frontend IP Addresses on the Azure Network Load Balancer, on page 181.</p>

Properties	Description
Rule Settings	<p>The Rule Settings pane contains the following options:</p> <ul style="list-style-type: none"> • Name—Enter a name for the rule. • Action Type—Set to <code>Forward to</code> by default. The traffic is forwarded to the port for the EPG selected using the protocol chosen below. • Port—Enter the port on which the backend pool servers will accept traffic from the load balancer. • Protocol—Click to choose TCP or UDP. • EPG—The EPG with the web servers handling traffic.
Health Checks	<p>The load balancer performs health checks on its backend pool targets for high availability. This can be configured here.</p> <ul style="list-style-type: none"> • Protocol—Click to choose TCP, HTTP or HTTPS. • Port—Enter a port on which health checks should be performed. • Advanced Settings— <ul style="list-style-type: none"> • Unhealthy Threshold—Configure this threshold to determine when a backend target is advertised as unhealthy. • Interval—Enter a time in seconds to determine at what intervals checks should be performed. <p>Click Add when finished.</p>

Step 10 Click **Add** when finished.
The service graph is deployed.

Deploying a Service Graph Using the REST API

The following sections describe how to deploy a service graph using the REST API.

Creating an Internal-Facing Load Balancer Using the REST API

This example demonstrates how to create an internal-facing load balancer using the REST API.

Step 1 To create an internal-facing load balancer for Application Gateway (Application Load Balancer):

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[<subscription id>]-vendor-azure" />
```

```

    <cloudLB scheme="internal" type="application" name="alb-151-15" status="">
      <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn15/ctxprofile-cProfilewestus15/cidr-[15.151.0.0/16]/subnet-[15.151.2.0/24]"/>
      </cloudLB>
    </fvTenant>
  </polUni>

```

Step 2 To create an internal-facing load balancer for Azure Load Balancing (Network Load Balancer):

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[subscription id]-vendor-azure" />
    <cloudLB scheme="internal" type="network" name="nlb-151-15" status="">
      <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn15/ctxprofile-cProfilewestus15/cidr-[15.151.0.0/16]/subnet-[15.151.2.0/24]" />
      </cloudLB>
    </fvTenant>
  </polUni>

```

Step 3 To create an internal-facing load balancer for Azure Load Balancing (Network Load Balancer) using the **Allow All Traffic** option described in [About Allow All Traffic Option, on page 183](#):

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[subscription id]-vendor-azure" />
    <cloudLB scheme="internal" type="network" name="nlb-151-15" allowAll="true" status="">
      <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn15/ctxprofile-cProfilewestus15/cidr-[15.151.0.0/16]/subnet-[15.151.2.0/24]" />
      </cloudLB>
    </fvTenant>
  </polUni>

```

Configuring an Internet-Facing Load Balancer Using the REST API

This example demonstrates how to create an internet-facing load balancer using the REST API.

Step 1 To create an internet-facing load balancer for Application Gateway:

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[<subscription id>]-vendor-azure" />
    <cloudLB scheme="internet" type="application" name="alb-151-15" status="">

```

Creating a Third-Party Firewall Using the REST API

```

        <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn15/ctxprofile-cProfilewestus151/cidr-[15.151.0.0/16]/subnet-[15.151.2.0/24]" />
        </cloudLB>

    </fvTenant>
</polUni>

```

Step 2 To create an internet-facing Network Load Balancer:

Example:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->

<polUni>
    <fvTenant name="tn15">
        <fvRsCloudAccount tDn="uni/tn-infra/act-[<subscription id>]-vendor-azure" />
        <cloudLB scheme="internet" type="network" name="nlb-151-15" status="">
            <cloudVip name="IP1" />
            <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn15/ctxprofile-cProfilewestus151/cidr-[15.151.0.0/16]/subnet-[15.151.2.0/24]" />
            </cloudLB>
        </fvTenant>
    </polUni>

```

In this example:

- The `cloudLB` line is used to create an internet-facing Network Load Balancer with a single default IP address.
- The `cloudVip` line is used to create any additional frontend IP addresses for the internet-facing Network Load Balancer. For more information, see [Understanding Configurations for Multiple Frontend IP Addresses on the Azure Network Load Balancer, on page 181](#).

Creating a Third-Party Firewall Using the REST API

This example demonstrates how to create a third-party firewall using the REST API.

Step 1 To create a third-party firewall:

Example:

```

<cloudLDev name="HubFW" svcType="FW" status="">
    <cloudRsLDevToCtx tDn="uni/tn-infra/ctx-overlay-2" />
    <cloudLIf name="provider">
        <cloudEPSelector name="east" matchExpression="IP=='{{eastus_FwUntrustSubnet}}'" status="" />
    </cloudLIf>
    <cloudLIf name="consumer">
        <cloudEPSelector name="east" matchExpression="IP=='{{eastus_FwTrustSubnet}}'" status="" />
    </cloudLIf>
</cloudLDev>

```

Step 2 To create a third-party firewall using the **Allow All Traffic** option described in [About Allow All Traffic Option, on page 183](#):

Example:

```
<cloudLDev name="HubFW" svcType="FW" status="">
  <cloudRsLDevToCtx tDn="uni/tn-infra/ctx-overlay-2"/>
  <cloudLIf name="provider" allowAll="true" status="">
    <cloudEPSelector name="1" matchExpression="IP=='10.1.1.0/28'" status=""/>
  </cloudLIf>
  <cloudLIf name="consumer" allowAll="true" status="">
    <cloudEPSelector name="east" matchExpression="IP=='10.1.2.0/28'" status=""/>
  </cloudLIf>
</cloudLDev>
```

Creating a Third Party Load Balancer Using the REST API

This example demonstrates how to create a third party load balancer using the REST API.

This example demonstrates how to create a third party load balancer using the REST API:

Example:

```
<cloudLDev name="ThirdPartyLB" svcType="ADC" status="">
  <cloudRsLDevToCtx tDn="uni/tn-infra/ctx-overlay-2"/>
  <cloudLIf name="external">
    <cloudEPSelector name="ExtInterfaceSelector" matchExpression="IP=='{{ExtInterfaceSubnet}}'"
status=""/>
  </cloudLIf>
  <cloudLIf name="internal">
    <cloudEPSelector name="IntInterfaceSelector" matchExpression="IP=='{{IntInterfaceSubnet}}'"
status=""/>
  </cloudLIf>
</cloudLDev>
```

Creating a Service Graph Using the REST API for an Application Gateway

This example demonstrates how to create a service graph using the REST API.

To create a service graph for an application gateway:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
  <fvTenant name="tn15">
    <vnsAbsGraph name="c15_g1" type="cloud" status="">
      <vnsAbsTermNodeProv name="p1">
        <vnsAbsTermConn/>
      </vnsAbsTermNodeProv>
      <vnsAbsTermNodeCon name="c1">
        <vnsAbsTermConn/>
      </vnsAbsTermNodeCon>
      <vnsAbsNode managed="yes" name="N1" funcType="GoTo">
        <vnsRsNodeToCloudLDev tDn="uni/tn-tn15/clb-alb-151-15"/>
      </vnsAbsNode>
    </vnsAbsGraph>
  </fvTenant>
</polUni>
```

```

    <vnsAbsFuncConn name="provider"/>
    <vnsAbsFuncConn name="consumer"/>
  </vnsAbsNode>
  <vnsAbsConnection connDir="consumer" connType="external" name="con1">
    <vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsTermNodeCon-c1/AbsTConn"/>
    <vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsNode-N1/AbsFConn-consumer"/>
  </vnsAbsConnection>
  <vnsAbsConnection connDir="provider" connType="internal" name="con2">
    <vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsTermNodeProv-p1/AbsTConn"/>
    <vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsNode-N1/AbsFConn-provider"/>
  </vnsAbsConnection>
</vnsAbsGraph>

</fvTenant>
</polUni>

```

Creating a Service Graph Using the REST API for Azure Load Balancer

To create a service graph for an Azure load balancer:

```

<?xml version="1.0" encoding="UTF-8"?>

<!-- api/node/mo/uni/.xml -->

<polUni>

  <fvTenant name="tn15">

    <vnsAbsGraph name="c15_g1" type="cloud" status="">

      <vnsAbsTermNodeProv name="p1">

        <vnsAbsTermConn />

      </vnsAbsTermNodeProv>

      <vnsAbsTermNodeCon name="c1">

        <vnsAbsTermConn />

      </vnsAbsTermNodeCon>

      <vnsAbsNode managed="yes" name="N1" funcType="GoTo">

        <vnsRsNodeToCloudLDev tDn="uni/tn-tn15/clb-nlb-151-15" />

        <vnsAbsFuncConn name="provider" />

        <vnsAbsFuncConn name="consumer" />

      </vnsAbsNode>

      <vnsAbsConnection connDir="consumer" connType="external" name="con1">

        <vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsTermNodeCon-c1/AbsTConn" />

        <vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsNode-N1/AbsFConn-consumer" />

      </vnsAbsConnection>

      <vnsAbsConnection connDir="provider" connType="internal" name="con2">

```



```

<vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsTermNodeProv-p1/AbsTConn" />
<vnsRsAbsConnectionConns tDn="uni/tn-tn15/AbsGraph-c15_g1/AbsNode-N1/AbsFConn-provider" />
</vnsAbsConnection>
</vnsAbsGraph>
</fvTenant>
</polUni>

```

Creating a Service Graph Using the REST API for a Third Party Load Balancer

To create a service graph for a third party load balancer:

```

<polUni>
<fvTenant name="infra" >
<!-- Abs Graph Creation -->
<vnsAbsGraph name="{{graphName}}" uiTemplateType="UNSPECIFIED" type="cloud" status="">
<vnsAbsTermNodeProv name="T2">
<vnsOutTerm/>
<vnsInTerm />
<vnsAbsTermConn attNotify="no" name="1" />
</vnsAbsTermNodeProv>
<vnsAbsTermNodeCon name="T1" >
<vnsOutTerm/>
<vnsInTerm />
<vnsAbsTermConn attNotify="no" name="1" />
</vnsAbsTermNodeCon>
<vnsAbsNode funcTemplateType="ADC_TWO_ARM" name="{{F5Name}}" managed="no">
<vnsRsNodeToCloudLDev tDn="uni/tn-infra/cld-{{F5Name}}" />
<vnsAbsFuncConn attNotify="no" name="consumer" deviceLIIfName="external"/>
<vnsAbsFuncConn attNotify="no" name="provider" deviceLIIfName="internal"/>
</vnsAbsNode>
<vnsAbsConnection adjType="L3" connDir="provider" connType="external" directConnect="no"
name="ConsTermToF5">
<vnsRsAbsConnectionConns
tDn="uni/tn-infra/AbsGraph-{{graphName}}/AbsTermNodeCon-T1/AbsTConn"/>
<vnsRsAbsConnectionConns
tDn="uni/tn-infra/AbsGraph-{{graphName}}/AbsNode-{{F5Name}}/AbsFConn-consumer"/>
</vnsAbsConnection>
<vnsAbsConnection adjType="L3" connDir="provider" connType="external" directConnect="no"
name="F5ToProv">
<vnsRsAbsConnectionConns
tDn="uni/tn-infra/AbsGraph-{{graphName}}/AbsNode-{{F5Name}}/AbsFConn-provider" />
<vnsRsAbsConnectionConns
tDn="uni/tn-infra/AbsGraph-{{graphName}}/AbsTermNodeProv-T2/AbsTConn"/>
</vnsAbsConnection>
</vnsAbsGraph>
</fvTenant>
</polUni>

```

Creating a Multi-Node Service Graph Using the REST API

This example demonstrates how to create a multi-node service graph using the REST API.

To create a multi-node service graph, enter a post such as the following example;

```

<polUni>
<fvTenant name="tn12_iar_iavpc" status="">
<fvRsCloudAccount tDn="uni/tn-infra/[SubscriptionId]-vendor-azure"/>
<fvCtx name="vrf50" status=""/>
<fvCtx name="vrf60" status=""/>
<cloudVpnGwPol name="VgwPol0"/>
<cloudCtxProfile name="c50" status="">
<cloudRsCtxProfileToRegion tDn="uni/clouddomp/provp-azure/region-westus"/>
<cloudRsToCtx tnFvCtxName="vrf50"/>
<cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status=""/>
<cloudCidr addr="12.3.0.0/16" primary="true" status="">
<cloudSubnet ip="12.3.30.0/24" status="" name="GatewaySubnet" usage="gateway">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.3.2.0/24" status="" name="ALBSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.3.1.0/24" status="" name="FwMgmtSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.3.3.0/24" status="" name="FwUntrustSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.3.4.0/24" status="" name="FwTrustSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.3.5.0/24" status="" name="ConsumerSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus/zone-default"/>
</cloudSubnet>
</cloudCidr>
</cloudCtxProfile>
<cloudCtxProfile name="c60" status="">
<cloudRsCtxProfileToRegion tDn="uni/clouddomp/provp-azure/region-westus2"/>
<cloudRsToCtx tnFvCtxName="vrf60"/>
<cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status=""/>
<cloudCidr addr="12.4.0.0/16" primary="true" status="">
<cloudSubnet ip="12.4.1.0/24" status="" name="ProviderSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus2/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.4.2.0/24" status="" name="NLBSubnet">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus2/zone-default"/>
</cloudSubnet>
<cloudSubnet ip="12.4.30.0/24" status="" name="GatewaySubnet" usage="gateway">
<cloudRsZoneAttach tDn="uni/clouddomp/provp-azure/region-westus2/zone-default"/>
</cloudSubnet>
</cloudCidr>
</cloudCtxProfile>
<cloudApp name="ap50" status="">
<cloudEPg name="ap50vrf50epg1" status="">
<cloudRsCloudEPgCtx tnFvCtxName="vrf50"/>
<fvRsCons tnVzBrCPName="con50"/>
<fvRsProv tnVzBrCPName="con60"/>
<cloudEPSelector matchExpression="IP=='12.3.5.0/24'" name="100"/>
</cloudEPg>
<cloudEPg name="ap50vrf50epg2" status="">
<cloudRsCloudEPgCtx tnFvCtxName="vrf50"/>
<fvRsProv tnVzBrCPName="con60"/>
<cloudEPSelector matchExpression="IP=='12.3.1.0/24'" name="100"/>
</cloudEPg>
<cloudExtEPg routeReachability="internet" name="ap50extepg1">
<cloudExtEPSelector name="1" subnet="0.0.0.0/0"/>
<cloudRsCloudEPgCtx tnFvCtxName="vrf50"/>
<fvRsCons tnVzBrCPName="con60"/>

```

```

    </cloudExtEPg>
  </cloudApp>
  <cloudApp name="ap60" status="">
    <cloudEPg name="ap60vrf60epg1" status="">
      <cloudRsCloudEPgCtx tnFvCtxName="vrf60"/>
      <fvRsProv tnVzBrCPName="con50"/>
      <fvRsProv tnVzBrCPName="con70"/>
      <cloudEPSelector matchExpression="IP=='12.4.1.0/24'" name="100"/>
    </cloudEPg>
    <cloudExtEPg routeReachability="internet" name="ap60extepg1">
      <cloudExtEPSelector name="1" subnet="0.0.0.0/0"/>
      <cloudRsCloudEPgCtx tnFvCtxName="vrf60"/>
      <fvRsCons tnVzBrCPName="con70"/>
    </cloudExtEPg>
  </cloudApp>
  <vzBrCP name="con50" scope="tenant" status="">
    <vzSubj name="con50">
      <vzRsSubjFiltAtt tnVzFilterName="f10"/>
      <vzRsSubjGraphAtt tnVnsAbsGraphName="g1" status=""/>
    </vzSubj>
  </vzBrCP>
  <vzBrCP name="con60" scope="tenant" status="">
    <vzSubj name="con60">
      <vzRsSubjFiltAtt tnVzFilterName="f20"/>
    </vzSubj>
  </vzBrCP>
  <vzBrCP name="con70" scope="context" status="">
    <vzSubj name="con70">
      <vzRsSubjFiltAtt tnVzFilterName="f20"/>
    </vzSubj>
  </vzBrCP>
  <vzFilter name="f10" status="">
    <vzEntry etherT="ip" prot="icmp" name="f10entry1" status=""/>
    <vzEntry etherT="ip" prot="udp" dFromPort="1" dToPort="65535" name="f10entry2" status=""/>
    <vzEntry etherT="ip" prot="tcp" dFromPort="1" dToPort="65535" name="f10entry3" status=""/>
  </vzFilter>
  <vzFilter name="f20" status="">
    <vzEntry etherT="ip" prot="tcp" dFromPort="http" dToPort="http" name="f20entry1" status=""/>
    <vzEntry etherT="ip" prot="tcp" dFromPort="https" dToPort="https" name="f20entry2" status=""/>
    <vzEntry etherT="ip" prot="tcp" dFromPort="22" dToPort="22" name="f20entry3" status=""/>
  </vzFilter>
  <cloudLB name="FrontALB" type="application" scheme="internal" >
    <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn12_iar_iavpc/ctxprofile-c50/cidr-[12.3.0.0/16]/subnet-[12.3.2.0/24]"/>
  </cloudLB>
  <cloudLDev name="FW" svcType="FW" status="">
    <cloudRsLDevToCtx tDn="uni/tn-tn12_iar_iavpc/ctx-vrf50" />
    <cloudLIf name="provider" >
      <cloudEPSelector name="1" matchExpression="custom:tagp=='trustFW'"/>
    </cloudLIf>
    <cloudLIf name="consumer" >
      <cloudEPSelector name="1" matchExpression="custom:tagp=='untrustFW'"/>
    </cloudLIf>
  </cloudLDev>
  <cloudLB name="BackNLB" type="network" scheme="internal" >
    <cloudRsLDevToCloudSubnet
tDn="uni/tn-tn12_iar_iavpc/ctxprofile-c60/cidr-[12.4.0.0/16]/subnet-[12.4.2.0/24]"/>
  </cloudLB>
  <vnsAbsGraph name="g1" type="cloud" status="" >
    <vnsAbsTermNodeProv name="Input1" >
      <vnsAbsTermConn name="C1"/>
    </vnsAbsTermNodeProv>
    <vnsAbsTermNodeCon descr="" name="Output1" nameAlias="" ownerKey="" ownerTag="">
      <vnsAbsTermConn name="C2" />
    </vnsAbsTermNodeCon>
  </vnsAbsGraph>

```

```

</vnsAbsTermNodeCon>
<vnsAbsNode funcType="GoTo" name="N1" managed="yes" funcTemplateType="ADC_ONE_ARM" >
  <vnsRsNodeToCloudLDev tDn="uni/tn-tn12_iar_iavpc/clb-FrontALB" />
  <vnsAbsFuncConn attNotify="no" descr="" name="provider" nameAlias="" ownerKey="" ownerTag="" />
  <vnsAbsFuncConn attNotify="no" descr="" name="consumer" nameAlias="" ownerKey="" ownerTag="" />
  <cloudSvcPolicy tenantName="tn12_iar_iavpc" contractName="con50" subjectName="con50" >
    <cloudListener name="http_listener1" port="80" protocol="http">
      <cloudListenerRule name="rule1" priority="20" default="yes" >
        <cloudRuleAction type="forward" port="80" protocol="http"/>
      </cloudListenerRule>
    </cloudListener>
  </cloudSvcPolicy>
</vnsAbsNode>
<vnsAbsNode funcType="GoTo" name="N2" managed="no" funcTemplateType="ADC_TWO_ARM" >
  <vnsRsNodeToCloudLDev tDn="uni/tn-tn12_iar_iavpc/cld-FW" />
  <vnsAbsFuncConn attNotify="no" descr="" connType="snat_dnat" name="provider" nameAlias=""
ownerKey="" ownerTag="" />
  <vnsAbsFuncConn attNotify="no" descr="" connType="none" name="consumer" nameAlias="" ownerKey=""
ownerTag="" />
</vnsAbsNode>
<vnsAbsNode funcType="GoTo" name="N3" managed="yes" funcTemplateType="ADC_ONE_ARM" >
  <vnsRsNodeToCloudLDev tDn="uni/tn-tn12_iar_iavpc/clb-BackNLB" />
  <vnsAbsFuncConn attNotify="no" descr="" name="provider" nameAlias="" ownerKey="" ownerTag="" />
  <vnsAbsFuncConn attNotify="no" descr="" name="consumer" nameAlias="" ownerKey="" ownerTag="" />
  <cloudSvcPolicy tenantName="tn12_iar_iavpc" contractName="con50" subjectName="con50" >
    <cloudListener name="http_listener1" port="80" protocol="tcp">
      <cloudListenerRule name="rule1" priority="20" default="yes" >
        <cloudRuleAction type="forward" port="80" protocol="tcp"
epgdn="uni/tn-tn12_iar_iavpc/cloudapp-ap60/clouddep-ap60vrf60epg1"/>
      </cloudListenerRule>
    </cloudListener>
  </cloudSvcPolicy>
</vnsAbsNode>
<vnsAbsConnection connDir="provider" connType="external" name="CON4">
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsNode-N3/AbsFConn-provider"/>
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
</vnsAbsConnection>
<vnsAbsConnection connDir="consumer" connType="external" name="CON1">
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsNode-N1/AbsFConn-consumer"/>
</vnsAbsConnection>
<vnsAbsConnection connDir="consumer" connType="external" name="CON2">
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsNode-N1/AbsFConn-provider"/>
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsNode-N2/AbsFConn-consumer"/>
</vnsAbsConnection>
<vnsAbsConnection connDir="consumer" connType="external" name="CON3">
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsNode-N2/AbsFConn-provider"/>
  <vnsRsAbsConnectionConns tDn="uni/tn-tn12_iar_iavpc/AbsGraph-g1/AbsNode-N3/AbsFConn-consumer"/>
</vnsAbsConnection>
</vnsAbsGraph>
</fvTenant>
</polUni>

```

Creating a Multi-Node Service Graph With Redirect Using the REST API

This example demonstrates how to create a multi-node service graph with redirect using the REST API.

Step 1 To set up the infra tenant:

```

<polUni>
  <fabricInst>
    <commPol name="default">
      <commSsh name="ssh" adminSt="enabled" passwordAuth="enabled" />
    </commPol>
    <dnsProfile name="default">
      <dnsProv addr="172.23.136.143" preferred="yes" status="" />
    </dnsProfile>
  </fabricInst>
  <fvTenant name="infra">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[[{subscriptionId}]]-vendor-azure"/>
    <cloudAccount name="insbu" id=[[{subscriptionId}]] vendor="azure" accessType="credentials"
status="">
      <cloudRsCredentials tDn="uni/tn-infra/credentials-CNC-App"/>
    </cloudAccount>
    <cloudCredentials name="CNC-App" keyId=[[{accessKeyId}]] key=[[{accessKey}]] httpProxy="">
      <cloudRsAD tDn="uni/tn-infra/ad-[[{adId}]]"/>
    </cloudCredentials>
    <cloudAD name="CiscoINSBUAd" id=[[{adId}]] />
    <cloudApicSubnetPool subnet="10.10.1.0/24" />
    <cloudtemplateInfraNetwork name="default" numRoutersPerRegion="2" vrfName="overlay-1"
numRemoteSiteSubnetPool="1" status="">
      <cloudtemplateProfile name="default" routerUsername="cisco" routerPassword="ins3965" />
      <cloudtemplateExtSubnetPool subnetpool="11.11.0.0/16" status="" />
      <cloudtemplateExtNetwork name="default" status="">
        <cloudRegionName provider="azure" region=[[{region}]] />
        <cloudtemplateVpnNetwork name="default">
          <cloudtemplateIpSecTunnel peeraddr=[[{peerAddress}]] />
          <cloudtemplateOspf area="0.0.0.1" />
        </cloudtemplateVpnNetwork>
      </cloudtemplateExtNetwork>
      <cloudtemplateIntNetwork name="default">
        <cloudRegionName provider="azure" region=[[{region}]] />
      </cloudtemplateIntNetwork>
    </cloudtemplateInfraNetwork>
  </fvTenant>
  <cloudDomP>
    <cloudBgpAsP asn="1111"/>
    <cloudProvP vendor="azure">
      <cloudRegion adminSt="managed" name=[[{region}]]>
        <cloudZone name="default"/>
      </cloudRegion>
    </cloudProvP>
  </cloudDomP>
</polUni>

```

Step 2 To configure the service device in the hub VNet:

```

<polUni>
  <fvTenant name="infra">
    <fvRsCloudAccount tDn="uni/tn-infra/act-[[{subscriptionId}]]-vendor-azure"/>
    <cloudCtxProfile name="ct_ctxprofile_{{region}}" status="modified">
      <cloudRsCtxProfileToRegion status="" tDn="uni/clouddomp/provp-azure/region-{{region}}"/>

      <cloudCidr name="cidr1" addr=[[{HubCidrSvc}]] primary="no" status="">
        <cloudSubnet ip=[[{HubNLBSubnet}]] name="HubNLBSubnet" status="">
          <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
        </cloudSubnet>
      </cloudCidr>
    </cloudCtxProfile>
  </fvTenant>

```

```

        <cloudSubnet ip="{{HubFWSubnetInt}}" name="HubFWSubnetInt" status="">
            <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
        </cloudSubnet>
        <cloudSubnet ip="{{HubFWSubnetExt}}" name="HubFWSubnetExt" status="">
            <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
        </cloudSubnet>
        <cloudSubnet ip="{{HubFWMgmtSubnet}}" name="HubFWMgmtSubnet" status="">
            <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
        </cloudSubnet>
        <cloudSubnet ip="{{ConsHubEPgSubnet}}" name="ConsHubEPgSubnet" status="">
            <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
        </cloudSubnet>
    </cloudCidr>
</cloudCtxProfile>
<cloudLDev name="{{FWName}}" status="">
    <cloudRsLDevToCtx tDn="uni/tn-infra/ctx-{{ServiceVNetName}}"/>
    <cloudLIf name="external" >
        <cloudEPSelector matchExpression="custom:EPG=='FwExt'" name="1"/>
    </cloudLIf>
    <cloudLIf name="internal" >
        <cloudEPSelector matchExpression="custom:EPG=='FwInt'" name="1"/>
    </cloudLIf>
</cloudLDev>
<cloudLB name="{{NLBName}}" type="network" scheme="internal" size="small" instanceCount="2"
status="">
    <cloudRsLDevToCloudSubnet
tDn="uni/tn-infra/ctxprofile-ct_ctxprofile_{{region}}/cidr-{{HubCidrSvc}}/subnet-{{HubNLBSubnet}}"/>
    </cloudLB>
</fvTenant>
</polUni>

```

Step 3 To configure a provider and the graph in a spoke:

```

<polUni>
    <fvTenant name="{{tnNameProv}}" status="" >
        <fvRsCloudAccount tDn="uni/tn-infra/act-{{subscriptionId}}-vendor-azure"/>
        <fvCtx name="{{ProviderVNetName}}"/>
        <cloudCtxProfile name="{{ProviderVNetName}}" status="">
            <cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status="">
            <cloudRsCtxProfileToRegion status="" tDn="uni/clouddomp/provp-azure/region-{{region}}"/>

            <cloudRsToCtx tnFvCtxName="{{ProviderVNetName}}"/>
            <cloudCidr name="cidr1" addr="{{VnetCidrProv}}" primary="yes" status="">
                <cloudSubnet ip="{{ProviderSubnet}}" name="ProviderSubnet" status="">
                    <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
                </cloudSubnet>
                <cloudSubnet ip="{{BackALBSubnet}}" name="BackALBSubnet" status="">
                    <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
                </cloudSubnet>
            </cloudCidr>
        </cloudCtxProfile>
        <!-- contract-->
        <vzFilter descr="" name="HttpsFilter" ownerKey="" ownerTag="">
            <vzEntry dFromPort="443" dToPort="443" etherT="ip" name="https" prot="tcp" status="">
            <vzEntry dFromPort="80" dToPort="80" etherT="ip" name="http" prot="tcp" status="">
            <vzEntry dFromPort="22" dToPort="22" etherT="ip" name="ssh" prot="tcp" status="">

```

```

</vzFilter>
<vzBrCP name="{contractName}" scope="global" status="">
  <vzSubj name="Sub1" revFltPorts="yes">
    <vzRsSubjGraphAtt directives="" tnVnsAbsGraphName="{graphName}"/>
    <vzRsSubjFiltAtt tnVzFilterName="HttpsFilter"/>
  </vzSubj>
</vzBrCP>
<!-- cloud App Profile-->
<cloudApp name="provApp" status="">
  <cloudEPg name="App" status="">
    <cloudRsCloudEPgCtx tnFvCtxName="{ProviderVNetName}"/>
    <cloudEPSelector matchExpression="custom:EPG=='App'" name="1"/>
    <fvRsProv status="" tnVzBrCPName="{contractName}"/>
    <fvRsProv tnVzBrCPName="mgmt_common"/>
  </cloudEPg>
</cloudApp>
<!-- Abs Graph Creation -->
<vnsAbsGraph name="{graphName}" uiTemplateType="UNSPECIFIED" type="cloud">
  <vnsAbsTermNodeProv name="T2">
    <vnsOutTerm/>
    <vnsInTerm />
    <vnsAbsTermConn attNotify="no" name="1" />
  </vnsAbsTermNodeProv>
  <vnsAbsTermNodeCon name="T1" >
    <vnsOutTerm/>
    <vnsInTerm />
    <vnsAbsTermConn attNotify="no" name="1" />
  </vnsAbsTermNodeCon>
  <vnsAbsNode name="{NLBName}" managed="yes" >
    <vnsRsNodeToCloudLDev tDn="uni/tn-infra/clb-{{NLBName}}" status=""/>
    <cloudSvcPolicy tenantName="{tnNameProv}" contractName="{contractName}"
subjectName="Sub1" status="">
      <cloudHealthProbe name="http_listener1-rule1" protocol="tcp" port=22 interval=15
unhealthyThreshold=2/>
      <cloudListener name="http_listener1" port="80" protocol="tcp" status="">
        <cloudListenerRule name="rule1" default="true">
          <cloudRuleAction type="haPort" port="80" protocol="tcp"
healthProbe="http_listener1-rule1"/>
        </cloudListenerRule>
      </cloudListener>
    </cloudSvcPolicy>
    <vnsAbsFuncConn attNotify="no" name="provider" connType="redir"/>
    <vnsAbsFuncConn attNotify="no" name="consumer" connType="redir"/>
  </vnsAbsNode>
  <vnsAbsNode funcTemplateType="FW_ROUTED" name="{FWName}" managed="no">
    <vnsRsNodeToCloudLDev tDn="uni/tn-infra/cld-{{FWName}}"/>
    <vnsAbsFuncConn attNotify="no" name="consumer" deviceLIIfName="internal"/>
    <vnsAbsFuncConn attNotify="no" name="provider" deviceLIIfName="internal"/>
  </vnsAbsNode>
  <vnsAbsNode name="{BackALBName}" managed="yes">
    <vnsRsNodeToCloudLDev tDn="uni/tn-{{tnNameProv}}/clb-{{BackALBName}}"/>
    <cloudSvcPolicy tenantName="{tnNameProv}" contractName="{contractName}"
subjectName="Sub1" status="">
      <cloudListener name="http_listener1" port="80" protocol="http" status="">
        <cloudListenerRule name="rule1" default="true">
          <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-{{tnNameProv}}/cloudapp-provApp/cloudepg-App"/>
        </cloudListenerRule>
      </cloudListener>
    </cloudSvcPolicy>
    <vnsAbsFuncConn attNotify="no" name="provider"/>
    <vnsAbsFuncConn attNotify="no" name="consumer"/>
  </vnsAbsNode>
  <vnsAbsConnection adjType="L3" connDir="provider" connType="external" directConnect="no"

```

```

    name="ConsTermToNLB">
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsTermNodeCon-T1/AbsTConn"/>
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsNode-{{NLBName}}/AbsFConn-consumer"/>
      </vnsAbsConnection>
      <vnsAbsConnection adjType="L3" connDir="provider" connType="external" directConnect="no"
    name="NLBToFW">
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsNode-{{NLBName}}/AbsFConn-provider" />
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsNode-{{FWName}}/AbsFConn-consumer"/>
      </vnsAbsConnection>
      <vnsAbsConnection adjType="L3" connDir="provider" connType="external" directConnect="no"
    name="FWToBackALB">
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsNode-{{FWName}}/AbsFConn-provider" />
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsNode-{{BackALBName}}/AbsFConn-consumer"/>
      </vnsAbsConnection>
      <vnsAbsConnection adjType="L3" connDir="provider" connType="external" directConnect="no"
    name="BackALBToProv">
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsNode-{{BackALBName}}/AbsFConn-provider" />
      <vnsRsAbsConnectionConns
tDn="uni/tn-{{tnNameProv}}/AbsGraph-{{graphName}}/AbsTermNodeProv-T2/AbsTConn"/>
      </vnsAbsConnection>
    </vnsAbsGraph>
    <cloudLB name="{{BackALBName}}" type="application" scheme="internal" size="small"
instanceCount="2">
      <cloudRsLDevToCloudSubnet
tDn="uni/tn-{{tnNameProv}}/ctxprofile-{{ProviderVNetName}}/cidr-{{VNetCidrProv}}/subnet-{{BackALBSubnet}}"/>
      status=""/>
    </cloudLB>
  </fvTenant>
</polUni>

```

Step 4 To configure the consumer and import the contract defined in the provider:

```

<polUni>
  <fvTenant name="{{tnNameCons}}" >
    <fvRsCloudAccount tDn="uni/tn-infra/act-{{subscriptionId}}-vendor-azure"/>
    <fvCtx name="{{ConsumerVNetName}}"/>
    <cloudCtxProfile name="{{ConsumerVNetName}}" status="">
      <cloudRsCtxProfileToGatewayRouterP tDn="uni/tn-infra/gwrouterp-default" status=""/>
      <cloudRsCtxProfileToRegion status="" tDn="uni/clouddomp/provp-azure/region-{{region}}"/>

      <cloudRsToCtx tnFvCtxName="{{ConsumerVNetName}}"/>
      <cloudCidr name="cidr1" addr="{{VNetCidrCons}}" primary="yes" status="">
        <cloudSubnet ip="{{ConsumerSubnet}}" name="ConsumerSubnet" status="">
          <cloudRsZoneAttach status=""
tDn="uni/clouddomp/provp-azure/region-{{region}}/zone-default"/>
        </cloudSubnet>
      </cloudCidr>
    </cloudCtxProfile>
    <vzCPIf name="imported_{{contractName}}">
      <vzRsIf tDn="uni/tn-{{tnNameProv}}/brc-{{contractName}}"/>
    </vzCPIf>
    <!-- cloud App Profile-->
    <cloudApp name="consApp" status="">
      <cloudEPg name="Web" status="">
        <cloudRsCloudEPgCtx tnFvCtxName="{{ConsumerVNetName}}"/>
        <cloudEPSelector matchExpression="custom:EPG=='Web'" name="1"/>
      </cloudEPg>
    </cloudApp>
  </fvTenant>
</polUni>

```



```

        <fvRsConsIf tnVzCPIfName="imported_{{contractName}}"/>
        <fvRsProv tnVzBrCPName="mgmt_common"/>
    </cloudEPg>
</cloudApp>
</fvTenant>
</polUni>

```

Attaching a Service Graph Using the REST API

This example demonstrates how to attach a service graph using the REST API.

Step 1 To attach a service graph for Application Gateways:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>
    <fvTenant name="tn15">

        <vzBrCP name="c1">
            <vzSubj name="c1">
                <vzRsSubjGraphAtt tnVnsAbsGraphName="c15_g1"/>
            </vzSubj>
        </vzBrCP>

    </fvTenant>
</polUni>

```

Step 2 To attach a service graph for Azure Load Balancing:

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- api/node/mo/uni/.xml -->
<polUni>

<fvTenant name="tn15">

<vzBrCP name="c1">

<vzSubj name="c1">

<vzRsSubjGraphAtt tnVnsAbsGraphName="c15_g1" />

</vzSubj>

</vzBrCP>

</fvTenant>

</polUni>

```

Configuring an HTTP Service Policy Using the REST API

This example demonstrates how to create an HTTP service policy using the REST API.

Step 1 To create an HTTP service policy for Application Gateways:

```

<polUni>
  <fvTenant name="t2">
    <vnsAbsGraph name="CloudGraph" type="cloud" status="">
      <vnsAbsNode funcType="GoTo" name="N1" managed="yes">
        <cloudSvcPolicy tenantName="t2" contractName="httpFamily" subjectName="consubj">
          <cloudListener name="http_listener1" port="80" protocol="http" status="">
            <cloudListenerRule name="rule1" priority="10" default="yes" status="">
              <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-t2/cloudapp-ap/cloudepg-provEPG"/>
            </cloudListenerRule>
            <cloudListenerRule name="redirectRule" priority="20">
              <cloudRuleCondition type="path" value="/img/*"/>
              <cloudRuleAction type="redirect" RedirectPort="8080"/>
            </cloudListenerRule>
            <cloudListenerRule name="FixedRspRule" priority="30">
              <cloudRuleCondition type="host" value="example.com"/>
              <cloudRuleAction type="fixedResponse" FixedResponseCode="200"/>
            </cloudListenerRule>
            <cloudListenerRule name="redirectHPRule" priority="40" status="">
              <cloudRuleCondition type="host" value="example.com"/>
              <cloudRuleCondition type="path" value="/img/*"/>
              <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-t2/cloudapp-ap/cloudepg-provEPG"/>
            </cloudListenerRule>
          </cloudListener>
        </cloudSvcPolicy>
      </vnsAbsNode>
    </vnsAbsGraph>
  </fvTenant>
</polUni>

```

Step 2 To create an HTTP service policy for a Network Load Balancer:

```

<?xml version="1.0" encoding="UTF-8"?>
<polUni>
  <fvTenant name="tn15">
    <vnsAbsGraph name="CloudGraph" type="cloud" status="">
      <vnsAbsNode funcType="GoTo" name="N1" managed="yes">
        <cloudSvcPolicy tenantName="tn15" contractName="httpFamily" subjectName="consubj">
          <cloudListener name="tcp_listener" port="80" protocol="tcp" status="">
            <cloudListenerRule name="rule1" priority="10" default="yes" status="">
              <cloudRsListenerToVip tDn="uni/tn-infra/clb-NLB/vip-IP1" status=""/>
              <cloudRuleAction type="forward" port="80" protocol="tcp" epgdn="uni/tn-
tn15/cloudapp-ap/cloudepg-provEPG" />
            </cloudListenerRule>
          </cloudListener>
          <cloudListener name="udp_listener" port="55" protocol="udp" status="">
            <cloudListenerRule name="rule1" priority="10" default="yes" status="">
              <cloudRsListenerToVip tDn="uni/tn-infra/clb-NLB/vip-IP1" status=""/>
              <cloudRuleAction type="forward" port="55" protocol="udp" epgdn="uni/tn-
tn15/cloudapp-ap/cloudepg-provEPG" />
            </cloudListenerRule>
          </cloudListener>
        </cloudSvcPolicy>
      </vnsAbsNode>
    </vnsAbsGraph>
  </fvTenant>
</polUni>

```

As described previously in this document, when defined on a Network Load Balancer, the listener rule and rule action settings construct a mapping from a frontend configuration of the load balancer to a backend pool. In this example:

- The `cloudListenerRule` lines are used to configure listeners with a single frontend IP address but with different port and protocol combinations for an internet-facing Network Load Balancer on Cisco Cloud Network Controller.
- The `cloudRsListenerToVip` lines are used to configure listener rules with multiple frontends for an internet-facing Network Load Balancer on Cisco Cloud Network Controller, where each frontend is represented as a tuple combination of frontend IP addresses, port, and protocol. For more information, see [Understanding Configurations for Multiple Frontend IP Addresses on the Azure Network Load Balancer](#), on page 181.

Configuring a Key Ring Using the REST API

This example demonstrates how to configure a key ring using the REST API. For more information about key ring configuration, see the *Cisco APIC Basic Configuration Guide*.



Note This procedure is applicable only for Application Gateways.

To configure a key ring:

```
<polUni>
  <fvTenant name="tn15" >
    <cloudCertStore>
      <pkiKeyRing status="" name="lbCert" tp="lbTP" key="-----BEGIN RSA PRIVATE KEY-----
MIIEpQIBAAKCAQEAA4DGxaK+RHv/nToHLnmDBq2BfLimgX/zNJQC9bGuzr8Mj7dm0
XuHfQYGv0h1PtL4Pdx5f5qjB0NbhjAVB1Gw8cDiErEgAXy9Km27ySo2foKryNqCRE
Ginn/CgF75QPfIed568eScNDZPt/eMeHAuRX/PykKUatWwncGanjvHqc+SOLPF6TD
gQ5nwOHFFvym2DY8bfdYWrWmGsO7JqZzbPMptA2QWblILsSoIrdkIIgf6ZfYy/EN
bH+nYN2rJt8lZysxz0YmR0oRQHTiN2NiDY/ZV63yxCXfLg9qpNZCuD8K0fdCZPEq
8takiWBxiR5/HRPscWAdWQsoiKgG1k4NEbFA9QIDAQABAoIBAQQDQqA9IslyrdtqN
q6mZ3s2BNfF/4kgb7gn0DWS+9EJLJCJNZVhFEo2ZxxyfPp6HRnjYS50W83/E1and
+GD1bSucTuxqFWIQVh7r1ebYZIWk+NYSjr5yNVxux8U2hCNNV8WWVqkJKcUqICB
Bm47FKj53LV46zE0gyCaibFrYxZJ9+farGneyBdnoV+3thmez7534KCi0t3J3Eri
lgSY3ql6hPXB2ZXAP4jdAoLgWDU4IIM6OqOiWopZM/QYIE/WtPYyJ0QzNCXObtc5
FboDcvedsgd4x5G1fV2A4xTBQMCTZUZJ9fYAcFogTZXD+UVqxorh47tf/mz+1fjq
f1XphEDlAoGBAPVlvKfGW46qqRnYovfryxxxz4OM1sVSgcJpQTQtBQi2koJ8OwEZJ
2s+CX0r+oDqwP23go/QEVYVkcic9RGkJBNGel+dm/bTjzgmQYtqSCntecTsZD5JN
y1jkciiiznDkjcjReSZ2kh3dGXIBRiYk7ezp2z7EKfDrHe5x5ouGMgCnAoGBAOnh
buDeohv8KJaB+DiUfhtoa3aKNPBO+zWPCHP0HFGjPXshJcIYZc1GcycmuDKVnDd
MxhE/yOnQHowi4T9FMLpz5yh5zuCUVqOBgB1P6MzbC5t5MtLrEYr/AqFN11CqyXQ
cVcT6iCWL0AFJRW3c/OiESwLmZchs18RnbwOi6kDAoGBANVlzmPb07zB3eGTCU0t
KGiqwFLncUkVadZRRFZYPPnwiRkoe73j9brkNbgCqxW+Nlp5UjoeFry0N6y106q/
ZA4I7FnXryLBw2HYuw41Vixl+XOZ/HeO3RmFN1z717dGmaGbv43aKIB9x+X5n8wF
6z1NtBHMbk7yNwom1IRag1sbAoGAX0p4cJ/tJNXSe7AswHDQCL68uimJdDfZ5nKG
k83nE+Qc0qQozDJAmCiSFmuSNRnSep3FiafjBFXK0X4h+mdBJCC7bagRnI92Mh0X
mOwsp4P2GdywkZwdbuHQ6UBp1Ferf9aztzTn+as6xKOUATEezy9DK9zMWzQhhtaY
m9yZTp0CgYEA1UtcpWjAzQbXODJGmxGdAAakPpeiKw/Da3MccrTdgJt88ezM1Oej
Pdoab0G2PcFgJz0TSGk7N4XArVKeq7pgZ0kwcYAsh06A2Hal+D1z/bGoZP+kmD/x
Ny82phxYOXCnEc5Vv921U59+j7e067UFLAYJe6fu+oFImvofRnP4DIQ=
-----END RSA PRIVATE KEY-----" cert="-----BEGIN CERTIFICATE-----
MIIElTCCA32gAwIBAgIJAKWNjp//arBsMA0GCSqGSIb3DQEBCwUAMIGNMQswCQYD
VQQGEwJVUzELMAkGA1UECBMCQ0EwETAPBgNVBACtCFNhbibiBkb3NlMRlweAYDVQQK
EwlNeUNvbXBhbnkxXjAMBgNVBAsTBU15T3JnMRgwFgYDVQQDDFA8qLmFtYXpvc3F3
cy5jb20xIDAeBgkqhkiG9w0BCQEWEXJhbXNoY21yZ28uY29tMB4XDTE4MTAw
```

```

MjIwNTMwNVoXDE5MTAwMjIwNTMwNVowgY0xCzAJBgNVBAYTA1VTMQswCQYDVQQL
EwJQTERMA8GA1UEBxMIU2FuIEpvc2UxEjAQBGNVBAOTCU15Q29tcGFueTEOMAwG
A1UECmFTX1PcmcxGDAwBgNVBAMUDyouYw1hem9uYXdzLmNvbTEgMB4GCSqGSIb3
DQEJARYRcmFtc2hhaEBjaXNjbY5jb20wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQQDgMbFor5Ee/+dOgcueYMGryF8uKaBf/M0lAL1sa7OvwyPt2bRe4d9B
ga/SHU+0vg93F/mqMHQ1seMBUHUbDxwOISsSABfL0qbbvJKjZ+ggvI2oJF4aKef8
KAXv1A8h53nrx5Jw0Nk+394x4cC5Ff8/KQpRq1ZadwZqe08epz5I4s8XpMOBDMfA
4ccW/IzYnjxt9lhataYaw7smpnNs8ym0DZBZuUguxKgit2QgiB/pl9jL8Q1sf6dg
3as1PyXNizHPRiZHSFAdOI3Y2INj9lXrflEJd8u2dq1kK4Pwo590Jk8Sry1qSJ
YHGJHn8dE+xxYB1ZCyIqAbWTg0RsUD1AgMBAAGjgUwgfIwHQYDVR0OBByEFBvYq
K3b39+1oOr4IBSsePwcOpML7MIHCBgNVHSMEgbowgbeAFBYqK3b39+1oOr4IBSse
PwcOpML7oYGTpIGQMIGNMQswCQYDVQGEwJVUzELMAKGA1UECBMCQ0EwETAPBgNV
BACTCFNhbiBkb3NlMRlweAYDVQKQEWlNeUNvbXBhbnkxDjAMBGNVBAsTBU15T3Jn
MRgwFgYDVQDFA8qLmFtYXpvbmF3cy5jb20xIDAeBgkqhkiG9w0BCQEWEXJhbXNo
YWhAY2l2Y28uY29tggkApY2On/9qsGwwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAE/RuzCheLibHbrurGet6eaVx9DPYydNiKVBSAKO+5iuR84mQzhoT
nx5CN109xu5ml5baCYZsSnn6D7usC092bPA/kRCGxt29gkjpWA74tJHqIhVWgBM
mOrLiShoelwv+wRl0vRchlTfktXO68Tuk6vrqp76hKfOHia7b2h1IIMdq6VA/
+A5FQ0xqYfKdVd2RaINpzI8mqZiszw+7E6j1PL5k4tftWEaYpfGPlVesFEyJEL
gHBUiPt8TIbaMYI8qUqMB/emnLXeKQ5PRxdRnleA3h8jfq3D1CQRTLjmlDL3tpFwg
qopM6et5ZKqShX4T87BsgZIoiquzXqsuHg==
-----END CERTIFICATE-----"
  </pkiKeyRing>

  <pkiTP status="" name="lbTP" certChain="-----BEGIN CERTIFICATE-----
MIIElTCCA32gAwIBAgIJAKWNjp//arBsMA0GCSqGSIb3DQEBCwUAMIGNMQswCQYD
VQGEwJVUzELMAKGA1UECBMCQ0EwETAPBgNVBACTCFNhbiBkb3NlMRlweAYDVQKQ
EWlNeUNvbXBhbnkxDjAMBGNVBAsTBU15T3JnMRgwFgYDVQDFA8qLmFtYXpvbmF3
cy5jb20xIDAeBgkqhkiG9w0BCQEWEXJhbXNoYWhAY2l2Y28uY29tMB4XDTE4MTAw
MjIwNTMwNVoXDE5MTAwMjIwNTMwNVowgY0xCzAJBgNVBAYTA1VTMQswCQYDVQQL
EwJQTERMA8GA1UEBxMIU2FuIEpvc2UxEjAQBGNVBAOTCU15Q29tcGFueTEOMAwG
A1UECmFTX1PcmcxGDAwBgNVBAMUDyouYw1hem9uYXdzLmNvbTEgMB4GCSqGSIb3
DQEJARYRcmFtc2hhaEBjaXNjbY5jb20wggeiMA0GCSqGSIb3DQEBAQUAA4IBDwAw
ggEKAoIBAQQDgMbFor5Ee/+dOgcueYMGryF8uKaBf/M0lAL1sa7OvwyPt2bRe4d9B
ga/SHU+0vg93F/mqMHQ1seMBUHUbDxwOISsSABfL0qbbvJKjZ+ggvI2oJF4aKef8
KAXv1A8h53nrx5Jw0Nk+394x4cC5Ff8/KQpRq1ZadwZqe08epz5I4s8XpMOBDMfA
4ccW/IzYnjxt9lhataYaw7smpnNs8ym0DZBZuUguxKgit2QgiB/pl9jL8Q1sf6dg
3as1PyXNizHPRiZHSFAdOI3Y2INj9lXrflEJd8u2dq1kK4Pwo590Jk8Sry1qSJ
YHGJHn8dE+xxYB1ZCyIqAbWTg0RsUD1AgMBAAGjgUwgfIwHQYDVR0OBByEFBvYq
K3b39+1oOr4IBSsePwcOpML7MIHCBgNVHSMEgbowgbeAFBYqK3b39+1oOr4IBSse
PwcOpML7oYGTpIGQMIGNMQswCQYDVQGEwJVUzELMAKGA1UECBMCQ0EwETAPBgNV
BACTCFNhbiBkb3NlMRlweAYDVQKQEWlNeUNvbXBhbnkxDjAMBGNVBAsTBU15T3Jn
MRgwFgYDVQDFA8qLmFtYXpvbmF3cy5jb20xIDAeBgkqhkiG9w0BCQEWEXJhbXNo
YWhAY2l2Y28uY29tggkApY2On/9qsGwwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAE/RuzCheLibHbrurGet6eaVx9DPYydNiKVBSAKO+5iuR84mQzhoT
nx5CN109xu5ml5baCYZsSnn6D7usC092bPA/kRCGxt29gkjpWA74tJHqIhVWgBM
mOrLiShoelwv+wRl0vRchlTfktXO68Tuk6vrqp76hKfOHia7b2h1IIMdq6VA/
+A5FQ0xqYfKdVd2RaINpzI8mqZiszw+7E6j1PL5k4tftWEaYpfGPlVesFEyJEL
gHBUiPt8TIbaMYI8qUqMB/emnLXeKQ5PRxdRnleA3h8jfq3D1CQRTLjmlDL3tpFwg
qopM6et5ZKqShX4T87BsgZIoiquzXqsuHg==
-----END CERTIFICATE-----"
  </pkiTP>
</cloudCertStore>
  </fvTenant>
</polUni>

```

Creating an HTTPS Service Policy Using the REST API

This section demonstrates how to create an HTTPS service policy using the REST API.



Note A listener can have multiple certificates. The certificate options are:

- ELBSecurityPolicy-2016-08 – The default when no security policy is chosen.
- ELBSecurityPolicy-FS-2018-06
- ELBSecurityPolicy-TLS-1-2-2017-01
- ELBSecurityPolicy-TLS-1-2-Ext-2018-06
- ELBSecurityPolicy-TLS-1-1-2017-01
- ELBSecurityPolicy-2015-05
- ELBSecurityPolicy-TLS-1-0-2015-04

If you use multiple certificates, you must specify the default certificate. The default is specified using the **defaultCert** property in **cloudRsListenerToCert**.

Before you begin

You have already configured a key ring certificate.



Note This is applicable only for the Application Gateways.

To create an HTTPS service policy:

```
<polUni>
  <fvTenant name="t2">
    <vnsAbsGraph name="CloudGraph" type="cloud" status="">
      <vnsAbsNode funcType="GoTo" name="N1" managed="yes">
        <cloudSvcPolicy tenantName="t2" contractName="httpFamily" subjectName="consubj">
          <cloudListener name="https_listener" port="443" protocol="https"
secPolicy="eLBSecurityPolicy-2016-08" status="">
            <cloudRsListenerToCert defaultCert="yes" certStore="default"
tDn="uni/tn-t2/certstore/keyring-lbCert" status=""/>
            <cloudListenerRule name="defaultRule" default="yes" priority="100" status="">
              <cloudRuleAction type="forward" port="80" protocol="http"
epgdn="uni/tn-t1/cloudapp-ap/cloudepg-ep1">
                </cloudRuleAction>
              </cloudListenerRule>
            </cloudListener>
          </cloudSvcPolicy>
        </vnsAbsNode>
      </vnsAbsGraph>
    </fvTenant>
  </polUni>
```




CHAPTER 7

Cisco Cloud Network Controller Security

This chapter contains the following sections:

- [Access, Authentication, and Accounting, on page 259](#)
- [Configuring TACACS+, RADIUS, LDAP and SAML Access, on page 260](#)
- [Configuring HTTPS Access, on page 267](#)

Access, Authentication, and Accounting

Cisco Cloud Network Controller policies manage the authentication, authorization, and accounting (AAA) functions. The combination of user privileges, roles, and domains with access rights inheritance enables administrators to configure AAA functions at the managed object level in a granular fashion. These configurations can be implemented using the REST API or the GUI.



Note There is a known limitation where you cannot have more than 32 characters for the login domain name. In addition, the combined number of characters for the login domain name and the user name cannot exceed 64 characters.

For more access, authentication, and accounting configuration information, see [Cisco Cloud Network Controller Security Configuration Guide](#).

Configuration

The admin account is configured in the initial configuration script, and the admin is the only user when the system starts.

Configuring a Local User

Refer to [Creating a Local User Using the Cisco Cloud Network Controller GUI, on page 136](#) to configure a Local User and associate it to the OTP, SSH Public Key, and X.509 User Certificate using the Cisco Cloud Network Controller GUI.

Configuring TACACS+, RADIUS, LDAP and SAML Access

The following topics describe how to configure TACACS+, RADIUS, LDAP and SAML access for the Cisco Cloud Network Controller.

Overview

This topic provides step-by-step instructions on how to enable access to the Cisco Cloud Network Controller for RADIUS, TACACS+, LDAP, and SAML users, including ADFS, Okta, and PingID.

For additional TACACS+, RADIUS, LDAP, and SAML information, see [Cisco Cloud Network Controller Security Configuration Guide](#).

Configuring Cisco Cloud Network Controller for TACACS+ Access

Before you begin

- The Cisco Cloud Network Controller is online.
- The TACACS+ server host name or IP address, port, and key are available.
- The Cisco Cloud Network Controller management endpoint group is available.

Step 1 In the Cisco Cloud Network Controller, create the **TACACS+ Provider**.

- On the menu bar, choose **Administrative > Authentication**.
- In the Work pane, click on **Providers** tab and then click on the **Actions** drop-down and select **Create Provider**.
The **Create Provider** dialog box appears.
- In the **Host Name/IP Address** field, enter the Host Name/IP Address of the provider.
- In the **Description** field, enter a description of the provider.
- Click the **Type** drop-down list and choose **TACACS+**.
- In **Settings** section, specify the **Key**, **Port**, **Authentication Protocol**, **Timeout**, **Retries**, **Management EPG**. Select either **Enabled** or **Disabled** for **Server Monitoring**.

Step 2 Create the **Login Domain** for TACACS+.

- Click the **Intent** icon.
The **Intent** menu appears.
- Click the drop-down arrow below the **Intent** search box and choose **Administrative**.
A list of **Administrative** options appear in the **Intent** menu.
- From the **Administrative** list in the **Intent** menu, click **Create Login Domain**.
The **Create Login Domain** dialog box appears.
- Enter the appropriate values in each field as listed in the following *Create Login Domain Dialog Box Fields* table then continue.

Properties	Description
General	
Name	Enter the name of the Login Domain
Description	Enter the description of the Login Domain.
Settings	
Realm	Choose TACACS+ from the dropdown menu
Providers	To choose a Provider(s): <ol style="list-style-type: none"> 1. Click Add Providers. The Select Providers dialog appears. 2. Click to choose a provider(s) in the column on the left. 3. Click Select. You return to the Create Login Domain dialog box.

- e) Click **Save** to save the configuration.

What to do next

This completes the APIC TACACS+ configuration steps. Next, if a RADIUS server will also be used, configure the APIC for RADIUS.

Configuring Cisco Cloud Network Controller for RADIUS Access

Before you begin

- The Cisco Cloud Network Controller is online.
- The RADIUS server host name or IP address, port, and key are available.
- The Cisco Cloud Network Controller management endpoint group is available.

Step 1

In the Cisco Cloud Network Controller, create the **RADIUS Provider**.

- On the menu bar, choose **Administrative > Authentication**.
- In the Work pane, click on **Providers** tab and then click on the **Actions** drop-down and select **Create Provider**.

The **Create Provider** dialog box appears.

- In the **Host Name/IP Address** field, enter the Host Name/IP Address of the provider.
- In the **Description** field, enter a description of the provider.
- Click the **Type** drop-down list and choose **RADIUS**.

- f) In the **Settings** section, specify the **Key, Port, Authentication Protocol, Timeout, Retries, Management EPG**. Select either **Enabled** or **Disabled** for **Server Monitoring**.

Step 2 Create the **Login Domain** for **RADIUS**.

- a) Click the **Intent** icon.

The **Intent** menu appears.

- b) Click the drop-down arrow below the **Intent** search box and choose **Administrative**

A list of **Administrative** options appear in the **Intent** menu.

- c) From the **Administrative** list in the **Intent** menu, click **Create Login Domain**.

The **Create Login Domain** dialog box appears.

- d) Enter the appropriate values in each field as listed in the following *Create Login Domain Dialog Box Fields* table then continue.

Properties	Description
General	
Name	Enter the name of the Login Domain
Description	Enter the description of the Login Domain.
Settings	
Realm	Choose RADIUS from the dropdown menu
Providers	To choose a Provider(s): <ol style="list-style-type: none"> 1. Click Add Providers. The Select Providers dialog appears. 2. Click to choose a provider(s) in the column on the left. 3. Click Select. You return to the Create Login Domain dialog box.

- e) Click **Save** to save the configuration.

What to do next

This completes the Cisco Cloud Network Controller RADIUS configuration steps. Next, configure the RADIUS server.

Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the Cisco Cloud Network Controller

Refer to the section *Configuring a Cisco Secure Access Control Server for RADIUS and TACACS+ Access to the APIC* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Configuring LDAP Access

There are two options for LDAP configurations:

- Configure a Cisco AVPair
- Configure LDAP group maps in the Cisco Cloud Network Controller

The following sections contain instructions for both configuration options.

Configuring Windows Server 2008 LDAP for APIC Access with Cisco AVPair

Refer to the section *Configuring Windows Server 2008 LDAP for APIC Access with Cisco AVPair* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Configuring Cisco Cloud Network Controller for LDAP Access

Before you begin

- The Cisco Cloud Network Controller is online.
- The LDAP server host name or IP address, port, bind DN, Base DN, and password are available.
- The Cisco Cloud Network Controller management endpoint group is available.

Step 1

In the Cisco Cloud Network Controller, create the **LDAP Provider**.

- a) On the menu bar, choose **Administrative > Authentication**.
- b) In the Work pane, click on **Providers** tab and then click on the **Actions** drop-down and select **Create Provider**.

The **Create Provider** dialog box appears.

- c) In the **Host Name/IP Address** field, enter the Host Name/IP Address of the provider.
- d) In the **Description** field, enter a description of the provider.
- e) Click the **Type** drop-down list and choose **LDAP**.
- f) Specify the **Bind DN**, **Base DN**, **Password**, **Port**, **Attribute**, **Filter Type** and **Management EPG**.

Note

- The bind DN is the string that the Cisco Cloud Network Controller uses to log in to the LDAP server. The Cisco Cloud Network Controller uses this account to validate the remote user attempting to log in. The base DN is the container name and path in the LDAP server where the Cisco Cloud Network Controller searches for the remote user account. This is where the password is validated. Filter is used to locate the attribute that the Cisco Cloud Network Controller requests to use for the *cisco-av-pair*. This contains the user authorization and assigned RBAC roles for use on the Cisco Cloud Network Controller. The Cisco Cloud Network Controller requests the attribute from the LDAP server.
- **Attribute** field—Enter one of the following:
 - For LDAP server configurations with a Cisco AVPair, enter **CiscoAVPair**.
 - For LDAP server configurations with an LDAP group map, enter **memberOf**.

Step 2 Create the **Login Domain** for LDAP.

- a) On the menu bar, choose **Administrative > Authentication**.
- b) In the Work pane, click on **Login Domains** tab and then click on the **Actions** drop-down and select **Create Login Domain**.
- c) Enter the appropriate values in each field as listed in the following *Create Login Domain Dialog Box Fields* table then continue.

Properties	Description
General	
Name	Enter the name of the Login Domain
Description	Enter the description of the Login Domain.
Settings	
Realm	Choose LDAP from the dropdown menu
Providers	<p>To choose a Provider(s):</p> <ol style="list-style-type: none"> 1. Click Add Providers. The Select Providers dialog appears. 2. Click to choose a provider(s) in the column on the left. 3. Click Select. You return to the Create Login Domain dialog box.

Properties	Description
Authentication Type	<ol style="list-style-type: none"> 1. Select Cisco AV Pairs, if provider(s) was configured with CiscoAVPair as the Attribute. 2. Select LDAP Group Map Rules, if provider(s) was configured with memberOf as the Attribute. <ol style="list-style-type: none"> a. Click Add LDAP Group Map Rule. The dialog box appears. b. Specify the map rule Name, Description (optional), and Group DN. c. Click the + next to Add Security Domain. The dialog box appears. d. Click the + to access the Role name and Role Privilege Type (Read or Write) fields. Click check mark. e. Repeat step 4 to add more roles. Then click Add. f. Repeat step 3 to add more security domains. Then click Add.

d) Click **Save** on Create Login Domain dialog box.

Configuring Cisco Cloud Network Controller for SAML Access

The following sections provide detailed information on configuring Cisco Cloud Network Controller for SAML access.

About SAML

Refer to the section *About SAML* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Basic Elements of SAML

Refer to the section *Basic Elements of SAML* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Supported IdPs and SAML Components

Refer to the section *Supported IdPs and SAML Components* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Configuring Cisco Cloud Network Controller for SAML Access



Note SAML based Authentication is only for Cisco Cloud Network Controller GUI and not for REST.

Before you begin

- The SAML server host name or IP address, and the IdP's metadata URL are available.
- The Cisco Cloud Network Controller management endpoint group is available.
- Set up the following:
 - Time Synchronization and NTP
 - Configuring a DNS Provider Using the GUI
 - Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI

Step 1 In the Cisco Cloud Network Controller, create the **SAML Provider**.

- a) On the menu bar, choose **Administrative > Authentication**.
- b) In the **Work** pane, click on **Providers** tab and then click on the **Actions** drop-down and select **Create Provider**.
- c) In the **Host Name/IP Address** field, enter the Host Name/IP Address of the provider.
- d) In the **Description** field, enter a description of the provider.
- e) Click the **Type** drop-down list and choose **SAML**.
- f) In **Settings** pane, perform following:
 - Specify the IdP metadata URL:
 - In case of AD FS, IdP Metadata URL is of the format *https://<FQDN ofADFS>/FederationMetadata/2007-06/FederationMetadata.xml*.
 - In case of Okta, to get the IdP Metadata URL, copy the link for **Identity Provider Metadata** in the **Sign On** section of the corresponding SAML Application from the Okta server.
 - Specify the **Entity ID** for the SAML-based service.
 - Configure the **HTTPS Proxy for Metadata URL** if it is needed to access the IdP metadata URL.
 - Select the **Certificate Authority** if IdP is signed by a Private CA.
 - Select the **Signature Algorithm Authentication User Requests** from the drop-down.
 - Select checkbox to enable **Sign SAML Authentication Requests**, **Sign SAML Response Message**, **Sign Assertions in SAML Response**, **Encrypt SAML Assertions**.
- g) Click **Save** to save the configuration.

Step 2 Create the login domain for SAML.

- a) On the menu bar, choose **Administrative > Authentication**.
- b) In the **Work** pane, click on the **Login Domains** tab and then click on the **Actions** drop-down and select **Create Login Domain**.

- c) Enter the appropriate values in each field as listed in the following Create Login Domain Dialog Box Fields table then continue.

Properties	Description
General	
Name	Enter the name of the Login Domain
Description	Enter the description of the Login Domain.
Settings	
Realm	Choose SAML from the dropdown menu
Providers	To choose a Provider(s): <ol style="list-style-type: none"> 1. Click Add Providers. The Select Providers dialog appears. 2. Click to choose a provider(s) in the column on the left. 3. Click Select. You return to the Create Login Domain dialog box.

- d) Click **Save** to save the configuration.

Setting Up a SAML Application in Okta

Refer to the section *Setting Up a SAML Application in Okta* of [Cisco Cloud Network Controller Security Configuration Guide](#).

Setting Up a Relying Party Trust in AD FS

Refer to the section *Setting Up a Relying Party Trust in AD FS* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Configuring HTTPS Access

The following sections describe how to configure HTTPS access.

About HTTPS Access

This article provides an example of how to configure a custom certificate for HTTPS access when using Cisco ACI.

For more information, see the section *HTTPS Access* in the [Cisco Cloud Network Controller Security Configuration Guide](#).

Guidelines for Configuring Custom Certificates

- Wild card certificates (such as *.cisco.com, which is used across multiple devices) and its associated private key generated elsewhere are not supported on the Cisco Cloud Network Controller as there is no support to input the private key or password in the Cisco Cloud Network Controller. Also, exporting private keys for any certificates, including wild card certificates, is not supported.
- You must download and install the public intermediate and root CA certificates before generating a Certificate Signing Request (CSR). Although a root CA Certificate is not technically required to generate a CSR, Cisco requires the root CA certificate before generating the CSR to prevent mismatches between the intended CA authority and the actual one used to sign the CSR. The Cisco Cloud Network Controller verifies that the certificate submitted is signed by the configured CA.
- To use the same public and private keys for a renewed certificate generation, you must satisfy the following guidelines:
 - You must preserve the originating CSR as it contains the public key that pairs with the private key in the key ring.
 - The same CSR used for the originating certificate must be resubmitted for the renewed certificate if you want to re-use the public and private keys on the Cisco Cloud Network Controller.
 - Do not delete the original key ring when using the same public and private keys for the renewed certificate. Deleting the key ring will automatically delete the associated private key used with CSRs.
- Only one Certificate Based Root can be active per pod.
- Client Certificate based authentication is not supported for this release.

Configuring a Custom Certificate for Cisco ACI HTTPS Access Using the GUI

Determine from which authority you will obtain the trusted certification so that you can create the appropriate Certificate Authority.

Before you begin

CAUTION: PERFORM THIS TASK ONLY DURING A MAINTENANCE WINDOW AS THERE IS A POTENTIAL FOR DOWNTIME. Expect a restart of all web servers on Cisco Cloud Network Controller during this operation.

Step 1 On the menu bar, choose **Administrative > Security**.

- Step 2** In the Work pane, click on **Certificate Authorities** tab and then click on the **Actions** drop-down and select **Create Certificate Authority**.
- Step 3** In the **Create Certificate Authority** dialog box, in the **Name** field, enter a name for the certificate authority and in the **Description** field, enter a description.
- Step 4** Select **System** in the **Used for** field.
- Step 5** In the **Certificate Chain** field, copy the intermediate and root certificates for the certificate authority that will sign the Certificate Signing Request (CSR) for the Cloud Network Controller. The certificate should be in Base64 encoded X.509 (CER) format. The intermediate certificate is placed before the root CA certificate. It should look similar to the following example:
- ```
-----BEGIN CERTIFICATE-----
<Intermediate Certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA Certificate>
-----END CERTIFICATE-----
```
- Step 6** Click **Save**.
- Step 7** On the menu bar, choose **Administrative > Security**.
- Step 8** In the Work pane, click on the **Key Rings** tab, then click on the **Actions** drop-down and select **Create Key Ring**.
- Step 9** In the **Create Key Ring** dialog box, enter a name for the key ring in the **Name** field and a description in the **Description** field.
- Step 10** Select **System** in the **Used for** field.
- Step 11** For the **Certificate Authority** field, click on **Select Certificate Authority** and select the Certificate Authority that you created earlier.
- Step 12** Select either **Generate New Key** or **Import Existing Key** for the field **Private Key**. If you select **Import Existing Key**, enter a private key in the **Private Key** text box.
- Step 13** Select modulus from the **Modulus** drop-down menu.
- Step 14** In the **Certificate** field, do not add any content.
- Step 15** Click **Save**.
- In the **Work** pane, in the **Key Rings** area, the **Admin State** for the key ring created displays **Started**.
- Step 16** Double-click on the created Key Ring to open **Key Ring** *key\_ring\_name* dialog box from the **Work** pane.
- Step 17** In the **Work** pane, click on **Create Certificate Request**.
- Step 18** In the **Subject** field, enter the fully qualified domain name (FQDN) of the Cisco Cloud Network Controller.
- Step 19** Fill in the remaining fields as appropriate.
- Step 20** Click **Save**.
- The **Key Ring** *key\_ring\_name* dialog box appears.
- Step 21** Copy the contents from the field Request to submit to the **Certificate Authority** for signing.
- Step 22** From the **Key Ring** *key\_ring\_name* dialog box, click on edit icon to display the **Key Ring** *key\_ring\_name* dialog box.
- Step 23** In the **Certificate** field, paste the signed certificate that you received from the certificate authority.
- Step 24** Click **Save** to return to the **Key Rings** work pane.
- The key is verified, and in the **Work** pane, the **Admin State** changes to **Completed** and is now ready for use in the HTTPS policy.
- Step 25** Navigate to **Infrastructure > System Configuration**, then click the **Management Access** tab.

**Step 26** Click the edit icon on the **HTTPS** work pane to display the **HTTPS Settings** dialog box.

**Step 27** Click on **Admin Key Ring** and associate the Key Ring that you created earlier.

**Step 28** Click **Save**.

All web servers restart. The certificate is activated, and the non-default key ring is associated with HTTPS access.

---

### What to do next

You must remain aware of the expiration date of the certificate and take action before it expires. To preserve the same key pair for the renewed certificate, you must preserve the CSR, as it contains the public key that pairs with the private key in the key ring. Before the certificate expires, the same CSR must be resubmitted. Do not delete or create a new key ring, as deleting the key ring will delete the private key stored internally on the Cisco Cloud Network Controller.



## CHAPTER 8

# Restricting Access

- [Restricting Access by Domains, on page 271](#)
- [RBAC Roles, on page 271](#)
- [RBAC Rules, on page 275](#)
- [Guidelines and Limitations for Restricted Domains , on page 276](#)
- [Creating an RBAC Rule Using the Cisco Cloud Network Controller GUI, on page 276](#)

## Restricting Access by Domains

A restricted security domain allows a fabric administrator to prevent a group of users, such as Tenant A, from viewing or modifying any objects created by a group of users in a different security domain, such as Tenant B, when users in both groups have the same assigned privileges. For example, a tenant administrator in Tenant A's restricted security domain will not be able to see policies, profiles, or users configured in Tenant B's security domain. Unless Tenant B's security domain is also restricted, Tenant B will be able to see policies, profiles, or users configured in Tenant A. Note that a user will always have read-only visibility to system-created configurations for which the user has proper privileges.

For example, consider a user in a restricted security domain is associated to Tenant A. Tenant A includes two application profiles, application profile 1 created by the user and application profile 2 created by an administrator. The user can view only application profile 1 although application profile 2 is also with the same tenant. When a user is in a restricted security domain, even profiles created by administrators are not visible.

In the above example, an unrestricted user (user not in a restricted security domain), can view both, application profile 1 and application profile 2, although application profile 2 was created by another user (administrator).

## RBAC Roles

The Cloud Network Controller provides access according to a user's role through role-based access control (RBAC). A fabric user is associated with the following:

- A set of roles
- For each role, a privilege type: no access, read-only, or read-write
- One or more security domain tags that identify the portions of the management information tree (MIT) that a user can access

The Cisco Cloud Network Controller manages access privileges at the managed object (MO) level. A privilege is an MO that enables or restricts access to a particular function within the system. For example, fabric-equipment is a privilege bit. This bit is set by the Cisco Cloud Network Controller on all objects that correspond to equipment in the physical fabric.

A role is a collection of privilege bits. For example, because an “admin” role is configured with privilege bits for “fabric-equipment” and “tenant-security,” the “admin” role has access to all objects that correspond to equipment of the fabric and tenant security.

A security domain is a tag associated with a certain subtree in the Cisco Cloud Network Controller object hierarchy. For example, the default tenant “common” has a domain tag `common`. Similarly, the special domain tag `all` includes the entire MIT object tree. An administrator can assign custom domain tags to the MIT object hierarchy.

Creating a user and assigning a role to that user does not enable access rights. It is necessary to also assign the user to one or more security domains. By default, the Cisco Cloud Network Controller fabric includes two special pre-created domains:

- All—allows access to the entire MIT
- Infra—allows access to fabric infrastructure objects/subtrees, such as fabric access policies

Cisco Cloud Network Controller supports the following AAA roles and privileges:

| Privilege                 | Description                                                                                                                                                                                                 |
|---------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Role: admin</b>        |                                                                                                                                                                                                             |
| admin                     | Provides full access to all of the features of the fabric. The admin privilege can be considered to be a union of all other privileges.                                                                     |
| <b>Role: aaa</b>          |                                                                                                                                                                                                             |
| aaa                       | Used for configuring authentication, authorization, accounting, and import/export policies.                                                                                                                 |
| <b>Role: access-admin</b> |                                                                                                                                                                                                             |
| access-connectivity       | Used for Layer 1-3 configuration under infra, static route configurations under a tenant's L3Out, management infra policies, and tenant ERSPAN policies.                                                    |
| access-equipment          | Used for access port configuration.                                                                                                                                                                         |
| access-protocol           | Used for Layer 1-3 protocol configurations under infra, fabric-wide policies for NTP, SNMP, DNS, and image management, and operations-related access policies such as cluster policy and firmware policies. |
| access-qos                | Used for changing CoPP and QoS-related policies.                                                                                                                                                            |
| <b>Role: fabric-admin</b> |                                                                                                                                                                                                             |

| Privilege                  | Description                                                                                                                                                                                                                                          |
|----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| fabric-connectivity        | Used for Layer 1-3 configuration under the fabric, firmware and deployment policies for raising warnings for estimating policy deployment impact, and atomic counter, diagnostic, and image management policies on leaf switches and spine switches. |
| fabric-equipment           | Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.                                                                                                                                              |
| fabric-protocol            | Used for Layer 1-3 protocol configurations under the fabric, fabric-wide policies for NTP, SNMP, DNS, and image management, ERSPAN and health score policies, and firmware management traceroute and endpoint tracking policies.                     |
| <b>Role: nw-svc-admin</b>  |                                                                                                                                                                                                                                                      |
| nw-svc-policy              | Used for managing Layer 4 to Layer 7 service devices and network service orchestration.                                                                                                                                                              |
| <b>Role: nw-svc-params</b> |                                                                                                                                                                                                                                                      |
| nw-svc-params              | Used for managing Layer 4 to Layer 7 service policies.                                                                                                                                                                                               |
| <b>Role: ops</b>           |                                                                                                                                                                                                                                                      |
| ops                        | Used for viewing the policies configured including troubleshooting policies.                                                                                                                                                                         |
| <b>Role: port-mgmt</b>     |                                                                                                                                                                                                                                                      |
| port-mgmt                  | Used for assigning a node to a security domain. A user in a security domain with a Node Rule must also be assigned to domain <code>all</code> with the role of <code>port-mgmt</code> .                                                              |
| <b>Role: tenant-admin</b>  |                                                                                                                                                                                                                                                      |
| aaa                        | Used for configuring authentication, authorization, accounting and import/export policies.                                                                                                                                                           |
| access-connectivity        | Used for Layer 1-3 configuration under infra, static route configurations under a tenant's L3Out, management infra policies, and tenant ERSPAN policies.                                                                                             |
| access-equipment           | Used for access port configuration.                                                                                                                                                                                                                  |
| access-protocol            | Used for Layer 1-3 protocol configurations under infra, fabric-wide policies for NTP, SNMP, DNS, and image management, and operations-related access policies such as cluster policy and firmware policies.                                          |

| Privilege                     | Description                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| access-qos                    | Used for changing CoPP and QoS-related policies.                                                                                                                                                                                                                                                                                            |
| fabric-connectivity           | Used for Layer 1-3 configuration under the fabric, firmware and deployment policies for raising warnings for estimating policy deployment impact, and atomic counter, diagnostic, and image management policies on leaf switches and spine switches.                                                                                        |
| fabric-equipment              | Used for atomic counter, diagnostic, and image management policies on leaf switches and spine switches.                                                                                                                                                                                                                                     |
| fabric-protocol               | Used for Layer 1-3 protocol configurations under the fabric, fabric-wide policies for NTP, SNMP, DNS, and image management, ERSPAN and health score policies, and firmware management traceroute and endpoint tracking policies.                                                                                                            |
| nw-svc-policy                 | Used for managing Layer 4 to Layer 7 service devices and network service orchestration.                                                                                                                                                                                                                                                     |
| tenant-network-profile        | Used for managing tenant configurations, such as deleting and creating network profiles, and deleting and creating endpoint groups.                                                                                                                                                                                                         |
| tenant-protocol               | Used for managing configurations for Layer 1-3 protocols under a tenant, for tenant traceroute policies, and as write access for firmware policies.                                                                                                                                                                                         |
| tenant-qos                    | Used for QoS-related configurations for a tenant.                                                                                                                                                                                                                                                                                           |
| tenant-security               | Used for contract-related configurations for a tenant.                                                                                                                                                                                                                                                                                      |
| <b>Role: tenant-ext-admin</b> |                                                                                                                                                                                                                                                                                                                                             |
| tenant-connectivity           | Used for Layer 1-3 connectivity changes, including bridge domains, subnets, and VRFs; for atomic counter, diagnostic, and image management policies on leaf switches and spine switches; tenant in-band and out-of-band management connectivity configurations; and debugging/monitoring policies such as atomic counters and health score. |
| tenant-epg                    | Used for managing tenant configurations such as deleting/creating endpoint groups, VRFs, and bridge domains.                                                                                                                                                                                                                                |
| tenant-ext-connectivity       | Used for write access firmware policies; managing tenant L2Out and L3Out configurations; and debugging/monitoring/observer policies.                                                                                                                                                                                                        |

| Privilege              | Description                                                                                                                                                                                                                                     |
|------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| tenant-ext-protocol    | Used for managing tenant external Layer 1-3 protocols, including BGP, OSPF, PIM, and IGMP, and for debugging/monitoring/observer policies such as traceroute, ping, oam, and eptrk. Generally only used for write access for firmware policies. |
| tenant-network-profile | Used for managing tenant configurations, such as deleting and creating network profiles, and deleting and creating endpoint groups.                                                                                                             |
| tenant-protocol        | Used for managing configurations for Layer 1-3 protocols under a tenant, for tenant traceroute policies, and as write access for firmware policies.                                                                                             |
| tenant-qos             | Used for QoS-related configurations for a tenant.                                                                                                                                                                                               |
| tenant-security        | Used for contract-related configurations for a tenant.                                                                                                                                                                                          |

Custom privileges can be assigned to any MO Class. Twenty-two custom privileges are displayed in the Cisco Cloud Network Controller GUI. If one of these custom privileges is assigned to any class, that MO's access will include the newly added custom privilege. One custom privilege can be associated with one or more MO classes.



**Note** Although custom privileges are displayed by the Cisco Cloud Network Controller GUI, they are currently not supported.

A set of predefined managed object classes can be associated with domains. These classes should not have overlapping containment. Examples of classes that support domain association are as follows:

- Layer 2 and Layer 3 network managed objects
- Network profiles (such as physical, Layer 2, Layer 3, management)
- QoS policies

When an object that can be associated with a domain is created, the user must assign domain(s) to the object within the limits of the user's access rights. Domain assignment can be modified at any time.

## RBAC Rules

RBAC rules selectively expose resources (such as, application profiles, EPGs, contracts) to users that are otherwise inaccessible because they are in a different security domain. An RBAC rule comprises two parts: the distinguished name (DN) that locates the object to be accessed and the name of the security domain that contains the user who will access the object.

There are two types of RBAC rules:

- Implicit—a user inherits a rule or permission based on RBAC hierarchy

- Explicit—a rule is directly assigned to the user based on certain policies

Both restricted and unrestricted security domains are supported.



---

**Note** While an RBAC rule exposes an object to a user in a different part of the management information tree, it is not possible to use the CLI to navigate to such an object by traversing the structure of the tree. However, as long as the user knows the DN of the object included in the RBAC rule, the user can use the CLI to locate it via an MO find command.

---

## Guidelines and Limitations for Restricted Domains

The following are the guidelines and restrictions for the users of restricted domains:

- If a user from one security domain is assigned another security domain, the user gets access to the configurations associated with the new domain.
- A user can be part of one or more security domains which are marked “restricted”.
- Restricted domain users have read-only access to system created configurations.
- For a user with multiple security domains, the combined length of all security domains cannot exceed 1024 characters. If the length exceeds 1024, the user will have policy creation issues.
- Restricted domain on the Cisco Cloud Network Controller is not supported on the cloud resources. This means, a user of one restricted domain will be able to see cloud resources created by a user of another restricted domain.

## Creating an RBAC Rule Using the Cisco Cloud Network Controller GUI

This section explains how to create an RBAC rule using the GUI.



---

**Note** You can configure RBAC rules, however the Cisco Cloud Network Controller GUI does not support the configurations. The DN configured using this procedure (step 4) can be queried using API.

---

### Before you begin

Create a security domain. For the detailed task, see [Creating a Security Domain Using the Cisco Cloud Network Controller GUI](#).

---

**Step 1** Click the **Intent** icon. The **Intent** menu appears.

**Step 2** Click the drop-down arrow below the **Intent** search box and choose **Administrative**.

A list of **Administrative** options appears in the **Intent** menu.



**Step 3** From the **Administrative** list in the **Intent** menu, click **Security > RBAC Rules > Create RBAC Rule**. The **Create RBAC Rule** dialog box appears.

**Step 4** In the **DN** field, enter the DN for the rule.

For creating an explicit RBAC rule, locate the DN for the application in ObjectStore. Use that DN value here.

**Step 5** Choose a security domain:

- a) Click **Select Security Domain**. The **Select Security Domain** dialog box appears.
- b) From the **Select Security Domain** dialog box, click to choose a security domain from the column on the left then click **Select**. You return to the **Create RBAC Rule** dialog box.

**Step 6** From the **Allow Writes** field, click **Yes** to allow writes or **No** to not allow writes.

**Step 7** Click **Save** when finished.

**Note** After creating an explicit RBAC rule, a user assigned to a security domain will be able to only see the application and its children that were defined earlier (from ObjectStore).

---





## CHAPTER 9

# Configuration Drifts

---

- [Configuration Drift Notifications and Faults, on page 279](#)
- [Accessing the Main Configuration Drift Page, on page 280](#)
- [Checking for Missing Contracts Configuration, on page 283](#)
- [Checking for Missing EPGs Configuration, on page 284](#)
- [Checking for Missing VRFs Configuration, on page 285](#)
- [Configuration Drift Troubleshooting, on page 287](#)

## Configuration Drift Notifications and Faults

Once Cloud Network Controller is deployed, you will perform most of the configuration either via its GUI or REST API interface. However, there may be cases where you or another cloud administrator changes the deployed configuration directly in the cloud provider's GUI using the tools provided by AWS or Azure. In these cases, the intended configuration you deployed from the Cloud Network Controller and the actual configuration in the cloud site may become out of sync, we call this a configuration drift.

Cloud Network Controller provides visibility into any security policy (contracts) configuration discrepancy between what you deploy from the Cloud Network Controller and what is actually configured in the cloud site. Configuration drift is enabled by default, and configuration drift information is available for EPGs, VRFs, and contracts with or without Layer 4 to Layer 7 service graphs attached.

Configuration drift information is now consolidated under a single page, located at **Cloud Resources > Drifts**.

**Detection Summary**

| Unmanaged Objects | Objects with Drifts | Last Drift Check              |
|-------------------|---------------------|-------------------------------|
| 236               | 3                   | Feb 16 2023 04:14:55pm +08:00 |

Filter by attributes

| Object                              | Status    | Drift Type   | Last Configuration Update     |
|-------------------------------------|-----------|--------------|-------------------------------|
| brown2                              | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| brown3                              | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| context-[vr12]-addr-[10.119.0.0/16] | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| ctxcidr-[10.119.0.0/16]             | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| ctxcidr-[10.129.0.0/16]             | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| ctxcidr-[10.198.2.0/24]             | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| ctxcidr-[10.198.3.0/24]             | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |

See [Accessing the Main Configuration Drift Page, on page 280](#) for more information.

There are two aspects to analyzing configuration drift:

- Have all the fabric elements configured in the Cloud Network Controller and intended to be deployed in the cloud fabric been properly deployed?

This scenario can occur due to user configuration errors in Cloud Network Controller that could not be deployed in the cloud, connection or API issues on the cloud provider end, or if a cloud administrator manually deletes or modifies security rules directly in the cloud provider's UI. Any intended but missing configurations may present an issue for the Cloud Network Controller fabric.

- Are there any additional configurations that exist in the cloud but were not intended to be deployed from the Cloud Network Controller ?

Similarly to the previous scenario, this can occur if there are connection or API issues or if a cloud administrator manually creates additional security rules directly in the cloud provider's UI. Any existing but not intended configuration may present issues.

## Accessing the Main Configuration Drift Page

Configuration drift information is consolidated under a single **Drifts** page.

The **Drifts** page is used to provide the following pieces of information:

- To identify if something has been deleted
- To ensure that anything that should be present is correctly shown as present

**Step 1** Log in to your Cloud Network Controller GUI.

**Step 2** Navigate to the main configuration drift page:

## Cloud Resources > Drifts

The consolidated **Drifts** page appears.

The screenshot displays the 'Drifts' page in the Cisco Cloud Network Controller interface. It features a sidebar with navigation options like Dashboard, Topology, Cloud Resources, Application Management, Operations, Infrastructure, and Administrative. The main content area is titled 'Drifts' and includes a 'Detection Summary' box with the following data:

| Unmanaged Objects | Objects with Drifts | Last Drift Check              |
|-------------------|---------------------|-------------------------------|
| 236               | 3                   | Feb 16 2023 04:14:55pm +08:00 |

Below the summary is a table with a search filter and columns for Object, Status, Drift Type, and Last Configuration Update. The table lists several objects, all with a status of 'Unmanaged' and a drift type of 'Extra Object'. The last configuration update times are consistently from Feb 15 2023.

| Object                              | Status    | Drift Type   | Last Configuration Update     |
|-------------------------------------|-----------|--------------|-------------------------------|
| brown2                              | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| brown3                              | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| context-[vrf2]-addr-[10.119.0.0/16] | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| ctxcidr-[10.119.0.0/16]             | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| ctxcidr-[10.129.0.0/16]             | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| ctxcidr-[10.198.2.0/24]             | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| ctxcidr-[10.198.3.0/24]             | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:55:39pm +08:00 |
| egress-[1]-[0]-[0]-[0.0.0.0/0]      | Unmanaged | Extra Object | Feb 15 2023 04:57:13pm +08:00 |

In the **Drifts** page, you can see a summary of any configuration issues in your fabric.

The **Detection Summary** area provides an overview of how many configuration drifts were detected with managed or unmanaged objects, and the time when this information was last updated. If the inventory update timestamp is out of date, you can refresh the information by clicking the **Refresh** icon in the top right corner of this screen.

**Step 3** Use the information in the table below the **Detection Summary** area to find any configuration drifts.

- **Object:** Provides information on the object associated with the configuration drift.
- **Status:** Following are the different values that might appear in the **Status** column:
  - **Transient (low):** Drifts that are likely due to recent configuration changes. We recommend waiting for the fabric to stabilize and the drift will likely resolve on its own after the next configuration refresh.
  - **Presumed (medium):** Drifts that may or may not be transient. We recommend monitoring the status and troubleshoot the configuration should the drift persist.
  - **Raised (high):** Critical drifts. We recommend verifying the configuration on Cloud Network Controller and checking for any associated faults. Redeploying the configuration may help resolve communication issues between the Cloud Network Controller and cloud services. If the issue persists, check the tech-support logs.
  - **Unmanaged:** Configuration drifts related to extra inventory objects not created through the Cisco Cloud Network Controller.
- **Drift Type:** Following are the different values that might appear in the **Drift Type** column:
  - **Configuration:** External changes on the cloud providers site that could result in the intended configuration and the actual configuration being out of sync. Used for configuration drifts related to *EPGs* or *VRFs*.

- **Rule:** External changes on the cloud providers site that could result in the intended security rules and the expected rules that are established through a contract being out of sync. Used for configuration drifts related to *contracts*.
- **Extra Object:** Used to show extra inventory objects that were not created through the Cisco Cloud Network Controller. Cisco Cloud Network Controller does not perform drift detection on these objects.
- **Last Configuration Update:** Provides information on when the last configuration update occurred.

**Step 4** Enter information in the filter line to filter the configuration drifts provided in the table, if necessary.

a) Click in the filter line below the **Detection Summary** area. The following filter types appear:

- Object
- Status
- Drift Type
- Last Configuration Update
- Parent Path

Select the appropriate type for your filter.

b) Click the necessary operator.

The options are:


- **==:** The equal-to operator
- **!=:** The not-equal-to operator

c) Click the necessary drift type.

The options are **Extra Object**, **Rule**, and **Configuration**. See the explanations for the **Drift Type** field above for more information.

The entries in the table are filtered based on your selections above.

**Step 5** View additional information on a specific configuration drift, if necessary.

For any object listed in this page, you can bring up additional configuration drift information by clicking the appropriate line in the **Configuration Drifts** table. A side panel appears with additional information on this particular configuration drift; clicking the Details icon () automatically brings you to the appropriate **Cloud Mapping** page for this particular object.

Refer to the following sections for additional configuration drift information for specific objects:

- [Checking for Missing Contracts Configuration, on page 283](#)
- [Checking for Missing EPGs Configuration, on page 284](#)
- [Checking for Missing VRFs Configuration, on page 285](#)

# Checking for Missing Contracts Configuration

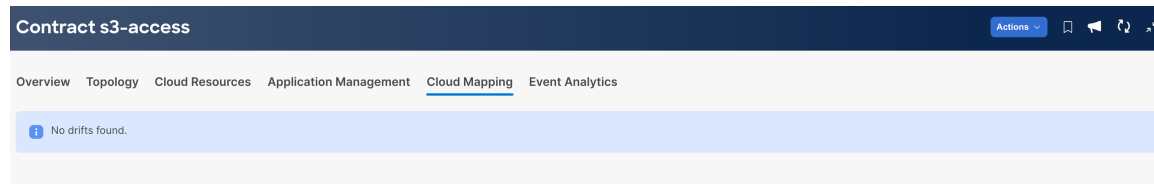
This section describes how to check for any contract settings you have configured from the Cloud Network Controller, but which have not been properly deployed to the cloud fabric.

- Step 1** Log in to your Cloud Network Controller GUI.
- Step 2** Click **Application Management > Contracts**.
- Step 3** Double-click the appropriate contract to bring up the **Overview** page for that contract.
- Step 4** Note the service graph information provided in the **Service Graph** area, if applicable.

Contract drift information is available for contracts with or without Layer 4 to Layer 7 service graphs attached. See [Deploying Layer 4 to Layer 7 Services, on page 177](#) for more information.

- Step 5** Click the **Cloud Mapping** tab.

The **Cloud Mapping** view displays all the information about the contract and the cloud resources it uses.



**Note** You can also navigate to this page by navigating to **Cloud Resources > Drifts**, then clicking the appropriate line in the **Configuration Drifts** table. A side panel appears with additional information on this particular configuration drift; clicking the Details icon (🔍) automatically brings you to the appropriate **Cloud Mapping** page for this particular object. See [Accessing the Main Configuration Drift Page, on page 280](#) for more information.

The screen is divided into four sections: **Detection Summary**, **Related Objects**, **Configuration Drifts**, and **Presented Cloud Resources**. Each section contains a table that lists the respective information about the contract you selected.

- The **Detection Summary** area provides an overview of how many configuration drifts were detected, number of intended and actual cloud resources configured, and the time when this information was last updated. If the inventory update timestamp is out of date, you can refresh the information by clicking the **Refresh** icon in the top right corner of this screen.
- The **Related Objects** area shows any other objects that have a relation to the contract, such as consumer and provider EPGs, and filters.
- The **Configuration Drifts** table lists all the issues with the contract rules. Specifically, all the contract rules that were intended to be deployed but are missing in the actual fabric configuration.

The table contains detailed information, such as the protocol used, port ranges, source and destination IP or group, consumer and provider EPGs, description of the issue, and the recommended action to resolve it. For each configuration drift, the **Status** field will indicate the severity and recommended action:

- **Transient (low)**: Drifts that are likely due to recent configuration changes. We recommend waiting for the fabric to stabilize and the drift will likely resolve on its own after the next configuration refresh.

- **Presumed (medium):** Drifts that may or may not be transient. We recommend monitoring the status and troubleshoot the configuration should the drift persist.
- **Raised (high):** Critical drifts. We recommend verifying the configuration on Cloud Network Controller and checking for any associated faults. Redeploying the configuration may help resolve communication issues between the Cloud Network Controller and cloud services. If the issue persists, check the tech-support logs.
- The **Presented Cloud Resources** table shows the information about all the resources that were properly configured in your cloud. This table is designed to provide you with better visibility into what rules are configured in your cloud for a specific contract.

## Checking for Missing EPGs Configuration

This section describes how to check for any EPG settings you have configured from the Cloud Network Controller, but which have not been properly deployed to the cloud fabric.

- Step 1** Log in to your Cloud Network Controller GUI.
- Step 2** Click **Application Management > EPGs**.
- Step 3** Double-click the appropriate EPG to bring up the **Overview** page for that EPG.
- Step 4** Click the **Cloud Mapping** tab.

The **Cloud Mapping** view displays all the information about the EPG and the cloud resources it uses.

The screenshot shows the EPG web interface with the following data:

| Detection Summary          |                            |                          |                               |
|----------------------------|----------------------------|--------------------------|-------------------------------|
| Configuration Drift Status | Configured Cloud Resources | Expected Cloud Resources | Last Drift Check              |
| 0 Drifts Found             | 1                          | 1                        | Feb 16 2023 04:39:55pm +08:00 |

| Related Objects |                    |                    |
|-----------------|--------------------|--------------------|
| Security Groups | Provider Contracts | Consumer Contracts |
| 1               | 2                  | 6                  |

Configuration Drifts section includes a filter by attributes input field.

**Note** You can also navigate to this page by navigating to **Cloud Resources > Drifts**, then clicking the appropriate line in the **Configuration Drifts** table. A side panel appears with additional information on this particular configuration drift; clicking the Details icon (🔍) automatically brings you to the appropriate **Cloud Mapping** page for this particular object. See [Accessing the Main Configuration Drift Page, on page 280](#) for more information.

The screen is divided into four sections: **Detection Summary**, **Related Objects**, **Configuration Drifts**, and **Presented Cloud Resources**. Each section contains a table that lists the respective information about the EPG you selected.



- The **Detection Summary** area provides an overview of how many configuration drifts were detected, number of intended and actual cloud resources configured, and the time when this information was last updated. If the inventory update timestamp is out of date, you can refresh the information by clicking the **Refresh** icon in the top right corner of this screen.
- The **Related Objects** area shows any other objects that have a relation to the EPG, such as security groups and contracts.
- The **Configuration Drifts** table lists all the issues with the security groups associated with the EPG. Specifically, all the security groups that were intended to be deployed but are missing in the actual fabric configuration.

The table contains detailed information, such as the logical DN, cloud provider ID, drift type, description of the issue, and the recommended action to resolve it. For each configuration drift, the **Status** field will indicate the severity and recommended action:

- **Transient (low)**: Drifts that are likely due to recent configuration changes. We recommend waiting for the fabric to stabilize and the drift will likely resolve on its own after the next configuration refresh.
  - **Presumed (medium)**: Drifts that may or may not be transient. We recommend monitoring the status and troubleshoot the configuration should the drift persist.
  - **Raised (high)**: Critical drifts. We recommend verifying the configuration on Cloud Network Controller and checking for any associated faults. Redeploying the configuration may help resolve communication issues between the Cloud Network Controller and cloud services. If the issue persists, check the tech-support logs.
- The **Presented Cloud Resources** table shows the information about all the resources that were properly configured in your cloud. This table is designed to provide you with better visibility into what security groups are associated with a specific EPG in your cloud.

---

## Checking for Missing VRFs Configuration

This section describes how to check for any VRF settings you have configured from the Cloud Network Controller, but which have not been properly deployed to the cloud fabric.

- 
- Step 1** Log in to your Cloud Network Controller GUI.
  - Step 2** Click **Application Management > VRFs**.
  - Step 3** Double-click the appropriate VRF to bring up the **Overview** page for that VRF.
  - Step 4** Click the **Cloud Mapping** tab.

The **Cloud Mapping** view displays all the information about the VRF and the cloud resources it uses.

The screenshot shows the VRF vrf1 configuration page. At the top, there is a navigation bar with tabs for Overview, Topology, Cloud Resources, Application Management, Cloud Mapping (selected), Statistics, and Event Analytics. Below the navigation bar, the page is divided into three main sections:

- Detection Summary:** A table with four columns: Configuration Drift Status (0 Drifts Found), Configured Cloud Resources (8), Expected Cloud Resources (8), and Last Drift Check (Feb 16 2023 04:14:55pm +08:00).
- Related Objects:** A table with three columns: Security Groups (3), CIDRs (1), and Subnets (6).
- Configuration Drifts:** A section with a filter box labeled "Filter by attributes" and a list of categories: VPCs, CIDRs, and Subnets.

**Note** You can also navigate to this page by navigating to **Cloud Resources > Drifts**, then clicking the appropriate line in the **Configuration Drifts** table. A side panel appears with additional information on this particular configuration drift; clicking the Details icon (🔍) automatically brings you to the appropriate **Cloud Mapping** page for this particular object. See [Accessing the Main Configuration Drift Page, on page 280](#) for more information.

The screen is divided into four sections: **Detection Summary**, **Related Objects**, **Configuration Drifts**, and **Presented Cloud Resources**. Each section contains a table that lists the respective information about the VRF you selected.

- The **Detection Summary** area provides an overview of how many configuration drifts were detected, number of intended and actual cloud resources configured, and the time when this information was last updated. If the inventory update timestamp is out of date, you can refresh the information by clicking the **Refresh** icon in the top right corner of this screen.
- The **Related Objects** area shows any other objects that have a relation to the VRF, such as security groups, CIDRs, and subnets.
- The **Configuration Drifts** table lists all the issues with the virtual networks, the CIDRs that are associated with the virtual networks, and the subnets within those CIDRs. Specifically, all the virtual networks, CIDRs, and subnets that were intended to be deployed but are missing in the actual fabric configuration.

Note that if there are configuration drifts at any one level, the table will show the configuration drift at that level and not any configuration drifts at the levels below it. For example, if a configuration drift occurs at a CIDR level and the corresponding subnets within that CIDR, the table will display the configuration drifts in the CIDR area but not the configuration drifts for the corresponding subnets within that CIDR.

The table contains detailed information in these areas:

- **Virtual Networks:** Provides information on logical DN, region, primary CIDR, drift type, description of the issue, and the recommended action to resolve it.
- **CIDRs:** Provides information on logical DN, region, CIDR block range, whether it is a primary CIDR or not, the subnets within the CIDR, drift type, description of the issue, and the recommended action to resolve it.
- **Subnets:** Provides information on logical DN, region, IP address, drift type, description of the issue, and the recommended action to resolve it.

For each configuration drift, the **Status** field will indicate the severity and recommended action:

- **Transient (low)**: Drifts that are likely due to recent configuration changes. We recommend waiting for the fabric to stabilize and the drift will likely resolve on its own after the next configuration refresh.
  - **Presumed (medium)**: Drifts that may or may not be transient. We recommend monitoring the status and troubleshoot the configuration should the drift persist.
  - **Raised (high)**: Critical drifts. We recommend verifying the configuration on Cloud Network Controller and checking for any associated faults. Redeploying the configuration may help resolve communication issues between the Cloud Network Controller and cloud services. If the issue persists, check the tech-support logs.
- The **Presented Cloud Resources** table shows the information about all the resources that were properly configured in your cloud, split into the same hierarchies that is shown in the **Configuration Drifts** table (Virtual Networks, CIDRs, and Subnets). This table is designed to provide you with better visibility into what virtual networks, CIDRs, and subnets are associated with a specific VRF in your cloud.

## Configuration Drift Troubleshooting

This section provides a few useful command to verify that the configuration drift processes are up and running on your Cisco Cloud Network Controller, check the application logs, and if necessary generate tech support information.

**Step 1** Log in to the Cisco Cloud Network Controller via console as a `root` user.

**Step 2** Check the status of the configuration drift application.

```
ACI-Cloud-Fabric-1# moquery -d pluginContr/plugin-Cisco_CApicDrift | egrep "dn |pluginSt |operSt |version"
dn: pluginContr/plugin-Cisco_CApicDrift
operSt: active
pluginSt: active
Version: 5.1.0
```

**Step 3** Check the status of the application container.

```
ACI-Cloud-Fabric-1# docker ps | grep drift
CONTAINER ID IMAGE COMMAND CREATED STATUS
NAMES
649af6feb72c a5ea08bbf541 "/opt/bin/conit.bi..." 13 hours ago Up 13
hours drift-api-b703e569-0aa6-859f-c538-a5fecbc5708f
```

**Step 4** Check memory consumed by all Docker containers.

Total amount of memory consumed must be under 12GB.

```
ACI-Cloud-Fabric-1# systemctl status ifc-scheduler_allocations.slice | grep Memory
```

**Step 5** If necessary, collect the tech support logs.

Logs will be saved in the `/data/techsupport` directory on the controller.

```
ACI-Cloud-Fabric-1# trigger techsupport controllers application CApicDrift
ACI-Cloud-Fabric-1# trigger techsupport controllers application CApicDrift vendorName Cisco
```

**Step 6** Check the application logs.

The logs for configuration drift process are stored in the `/data2/logs/Cisco_CApicDrift` directory.

The `runhist.log` file provides information about each time the application was started, for example:

```
cat runhist.log
1- Thu Jun 11 23:55:59 UTC 2020
2- Fri Jun 12 01:19:41 UTC 2020
```

The `drift.log` file is the application log file and can be used to view the number of times configuration drift was updated and how long each update took.

```
cat drift.log | grep ITER
{"file":"online_snapshot.go:178","func":"Wait","level":"info","msg":"ITER# 109
ENDED === RDFGEN TIME: 1m40.383751649s, MODEL UPLOAD TIME 5m54.245550374s; TOTAL
TIME:: 7m34.629447083s","time":"2020-06-12T19:53:13Z"}
```

---



## CHAPTER 10

# Configuring Connectivity Between Cisco Cloud Network Controller-Managed Cloud Site and Non-ACI Remote Site

---

The sections in this chapter describe how to configure connectivity between a Cisco Cloud Network Controller-managed cloud site and a non-ACI remote site, either by using express route gateway or without using express route gateway.

- [Configuring Connectivity Using Express Route Gateway, on page 289](#)
- [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\), on page 295](#)

## Configuring Connectivity Using Express Route Gateway

Express route gateway is supported with which you can deploy an express route gateway in the hub VNet using redirect or without using redirect. The express route gateway is used to provide connectivity between a Cisco Cloud Network Controller-managed cloud site and a non-ACI remote site. The external EPG for the non-ACI remote site (in this case, connected by an express route gateway) has a contract with the cloud EPG in the hub or spoke VNet.

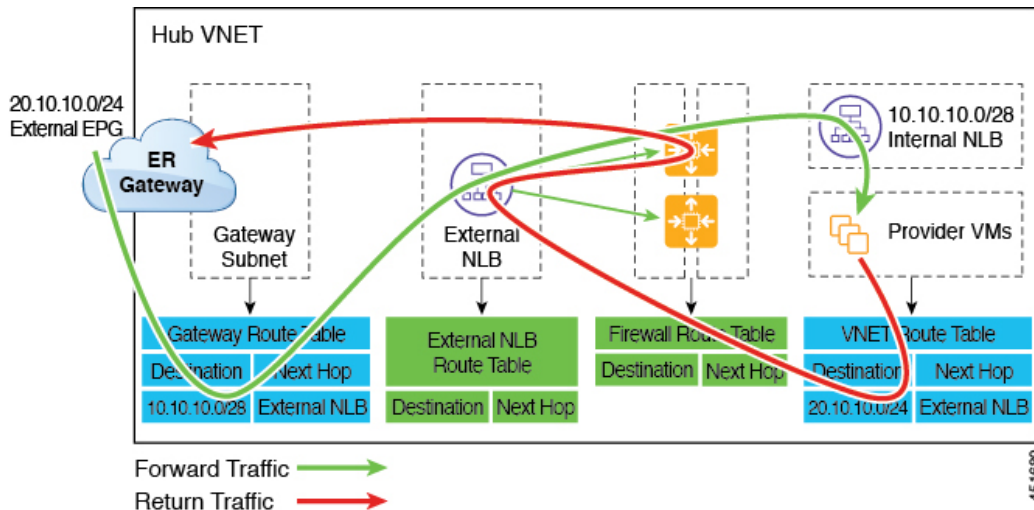
## About Deploying Express Route Gateway Using Redirect

In situations where you are deploying a connection between a cloud endpoint and an external network through an express route gateway, you can insert a service device between them using redirect.

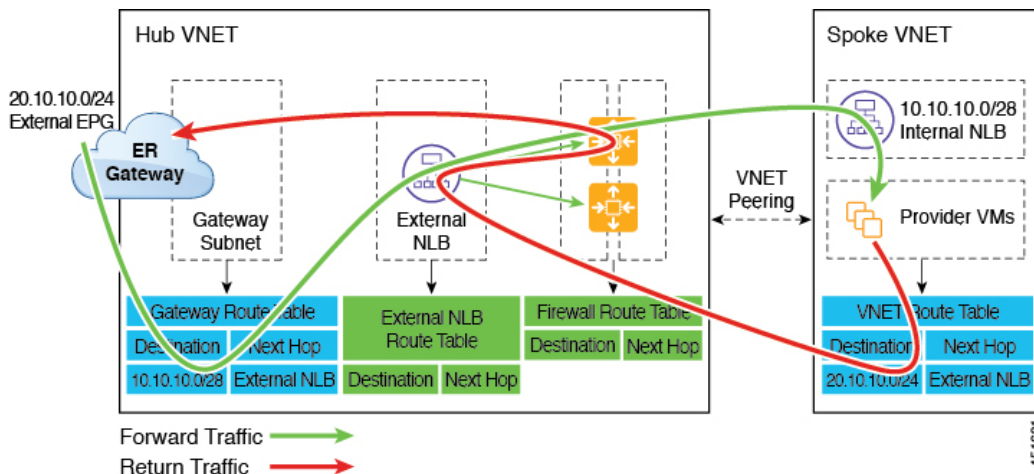
For this use case, the external EPG connected by the express route gateway has a contract with the cloud EPG in either the hub or the spoke VNet. In this situation:

- The redirect is configured on the gateway subnet route table by the Cisco Cloud Network Controller. The traffic destined to the provider cloud EPG is redirected to the service device deployed in the hub VNet as the next hop.
- You should have the service device that is used in the redirect in the same VNet as the external EPG connected by the express route gateway (in this case, in the hub VNet).
- Having the provider cloud EPG stretched across regions is supported in this case.

The following figure shows an example of a redirect for express route gateway to the provider EPG in the hub VNet.



The following figure shows an example of a redirect for express route gateway to the provider EPG in the spoke VNet.



The following table describes how redirect is programmed.

| Consumer                                            | Provider                                      | Redirect on Gateway Subnet Route Table                                          | Redirect on Provider VNet                                                           |
|-----------------------------------------------------|-----------------------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| External EPG connected by the express route gateway | Cloud EPG with subnet-based endpoint selector | Redirect for the consumer-to-provider traffic using the subnets of the provider | Redirect for the provider-to-consumer traffic using the subnets of the external EPG |

## Deploying Express Route Gateway Using Redirect

### Before you begin

Review the information provided in [About Deploying Express Route Gateway Using Redirect, on page 289](#) before proceeding with these procedures.

**Step 1** Enable VNet peering on your Cisco Cloud Network Controller.

Refer to [Configuring VNET Peering for Cisco Cloud Network Controller for Azure](#) for those instructions.

The gateway subnet in the hub VNet that is required for the express route gateway is deployed by the Cisco Cloud Network Controller when VNet peering is enabled. This is done to prepare the hub VNet for the deployment of the express route gateway.

**Step 2** Create an external EPG in the hub VNet that represents the network for the non-ACI remote site.

- To create an external EPG using the GUI, see [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#).

In the **Route Reachability** field for the external EPG, select **External-Site**.

- To create an external EPG using the REST API, see [Creating an External Cloud EPG Using the REST API, on page 154](#).

Create an external cloud EPG with the type **site-external**.

**Step 3** Through the Azure portal, deploy the express route gateway in the hub VNet using the gateway subnet that you configured in [Step 1, on page 291](#).

Depending on the number of regions that you selected when you enabled VNet peering in [Step 1, on page 291](#), if you need express route gateway access on multiple regions that the Cisco Cloud Network Controller will manage, deploy express route gateways in each of those regions separately.

- a) In the Azure portal, navigate to the Resource Manager virtual network where you want to create a virtual network gateway.
- b) On the left side, select **Create a resource**, and type **Virtual Network Gateway** in search.
- c) Locate **Virtual network gateway** in the search return and click the entry.
- d) On the **Virtual network gateway** page, choose **Create**.
- e) On the **Create virtual network gateway** page, enter the appropriate information for these fields:
  - **Subscription**: Verify that the correct subscription is selected.
  - **Resource Group**: The resource group will automatically be chosen once you choose the virtual network.
  - **Name**: The name of your express route gateway.
  - **Region**: Change the **Region** field to point to the location where your virtual network is located. If the location isn't pointing to the region where your virtual network is, the virtual network won't appear in the **Choose a virtual network** dropdown.
  - **Gateway type**: Choose **ExpressRoute**.
  - **SKU**: Choose the gateway SKU from the dropdown.

- **Virtual network:** Choose the virtual network that was created by the Cisco Cloud Network Controller in [Step 1, on page 291](#).
- **Public IP address:** Choose **Create new**.
- **Public IP address name:** Provide a name for the public IP address.

f) Select **Review + Create**, and then **Create** to begin creating the gateway.

The settings are validated and the gateway deploys. Creating virtual network gateway can take up to 45 minutes to complete.

To verify that the express route gateway was deployed successfully, navigate to the network gateways page in the Azure portal and verify that a network gateway with the type **express route** was created.

If you need express route gateway access on additional regions, repeat these steps for each of those regions.

**Step 4** Configure the service device for the redirect.

To configure a service device for redirect using the GUI or REST API, see [Deploying Layer 4 to Layer 7 Services, on page 177](#).

**Step 5** Configure a contract between the cloud EPG and the external EPG connected by the express route gateway.

- To create a contract using the GUI, see [Creating a Contract Using the Cisco Cloud Network Controller GUI, on page 103](#).
- To configure a contract using the REST API, see [Creating a Contract Using the REST API, on page 149](#).

---

## About Deploying Express Route Gateway Without Redirect

For this type of deployment, route propagation to the spoke VNet is automatically enabled by the Cisco Cloud Network Controller. This allows your non-ACI remote site subnet routes to be available to the spoke VNet through the hub VNet using VNet peering with gateway transit (also referred to as transit peering). VNet peering with gateway transit is also automatically enabled by the Cisco Cloud Network Controller in this situation.

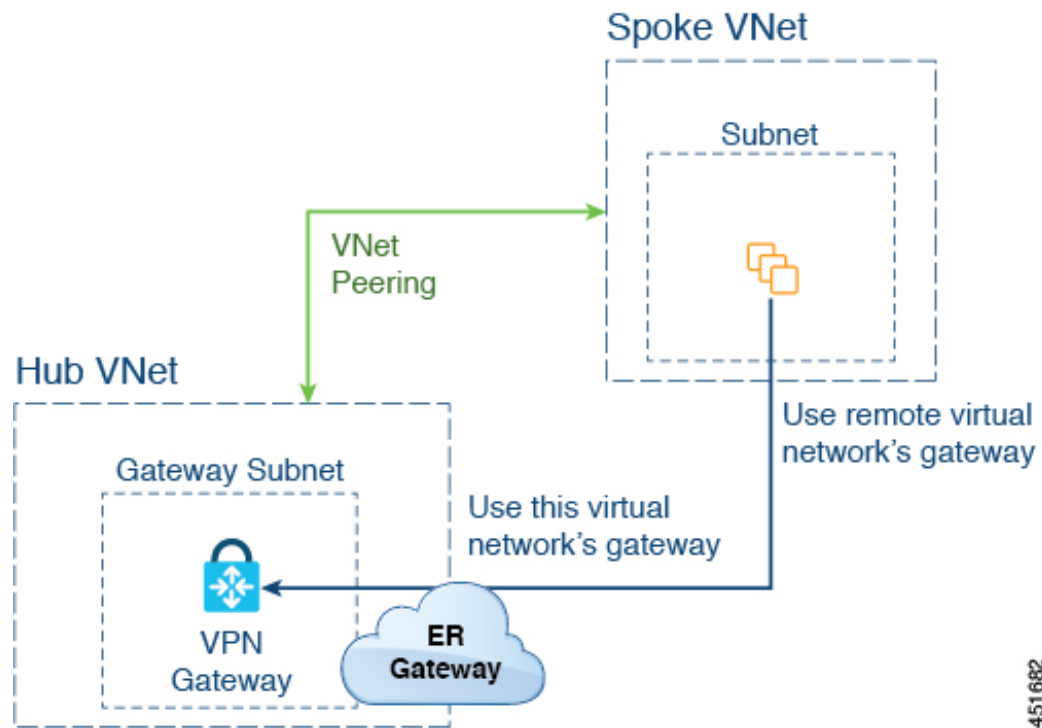
As part of this configuration, you will deploy the express route gateway in the hub VNet. When the Cisco Cloud Network Controller detects that the express route gateway has been configured in the hub VNet, it automatically sets the transit peering properties, one for the hub → spoke peering and the other for the spoke → hub peering, in the Azure portal:

- **Hub VNet:** Automatically set to **Use this virtual network's gateway**
- **Spoke VNet:** Automatically set to **Use remote virtual network's gateway** in the spoke VNet that is managed by the Cisco Cloud Network Controller

In order to have the route propagation enabled for the egress route table of the spoke VNet, you must configure a contract between the cloud EPG in the spoke VNet and the external EPG connecting to the non-ACI remote site.

The following figure shows an example of this type of deployment.





In this example:

- The following configurations are done automatically by the Cisco Cloud Network Controller:
  - The spoke VNet uses VNet peering with gateway transit (transit peering)
  - The VPN gateway in the hub VNet is connected to an on-premises non-ACI remote site
  - When the Cisco Cloud Network Controller detects that the express route gateway is deployed in the hub VNet, the transit peering properties are automatically set on each side of the peering (hub → spoke and spoke → hub):
    - **Hub VNet:** Automatically set to **Use this virtual network's gateway**
    - **Spoke VNet:** Automatically set to **Use remote virtual network's gateway** in the spoke VNet that is managed by the Cisco Cloud Network Controller
- The on-premises non-ACI routes learned by the VPN gateway are available to the spoke VNet if the EPG in the spoke VNet has a contract with the external EPG
- The hub VNet allows traffic from the EPG in the spoke VNet destined to the on-premises non-ACI remote site through the VPN gateway

## Deploying Express Route Gateway Without Redirect

### Before you begin

Review the information provided in [About Deploying Express Route Gateway Without Redirect](#), on page 292 before proceeding with these procedures.

**Step 1** Enable VNet peering on your Cisco Cloud Network Controller.

Refer to [Configuring VNET Peering for Cisco Cloud Network Controller for Azure](#) for those instructions.

The gateway subnet in the hub VNet that is required for the express route gateway is deployed by the Cisco Cloud Network Controller when VNet peering is enabled. This is done to prepare the hub VNet for the deployment of the express route gateway.

**Step 2** Create an external EPG in the hub VNet that represents the network for the non-ACI remote site.

- To create an external EPG using the GUI, see [Creating an External EPG Using the Cisco Cloud Network Controller GUI, on page 85](#).

In the **Route Reachability** field for the external EPG, select **External-Site**.

- To create an external EPG using the REST API, see [Creating an External Cloud EPG Using the REST API, on page 154](#).

Create an external cloud EPG with the type **site-external**.

**Step 3** Through the Azure portal, deploy the express route gateway in the hub VNet using the gateway subnet that you configured in [Step 1, on page 294](#).

Depending on the number of regions that you selected when you enabled VNet peering in [Step 1, on page 294](#), if you need express route gateway access on multiple regions that the Cisco Cloud Network Controller will manage, deploy express route gateways in each of those regions separately.

- In the Azure portal, navigate to the Resource Manager virtual network where you want to create a virtual network gateway.
- On the left side, select **Create a resource**, and type **Virtual Network Gateway** in search.
- Locate **Virtual network gateway** in the search return and click the entry.
- On the **Virtual network gateway** page, choose **Create**.
- On the **Create virtual network gateway** page, enter the appropriate information for these fields:
  - **Subscription**: Verify that the correct subscription is selected.
  - **Resource Group**: The resource group will automatically be chosen once you choose the virtual network.
  - **Name**: The name of your express route gateway.
  - **Region**: Change the **Region** field to point to the location where your virtual network is located. If the location isn't pointing to the region where your virtual network is, the virtual network won't appear in the **Choose a virtual network** dropdown.
  - **Gateway type**: Choose **ExpressRoute**.
  - **SKU**: Choose the gateway SKU from the dropdown.
  - **Virtual network**: Choose the virtual network that was created by the Cisco Cloud Network Controller in [Step 1, on page 294](#).
  - **Public IP address**: Choose **Create new**.
  - **Public IP address name**: Provide a name for the public IP address.
- Select **Review + Create**, and then **Create** to begin creating the gateway.

The settings are validated and the gateway deploys. Creating virtual network gateway can take up to 45 minutes to complete.

To verify that the express route gateway was deployed successfully, navigate to the network gateways page in the Azure portal and verify that a network gateway with the type **express route** was created.

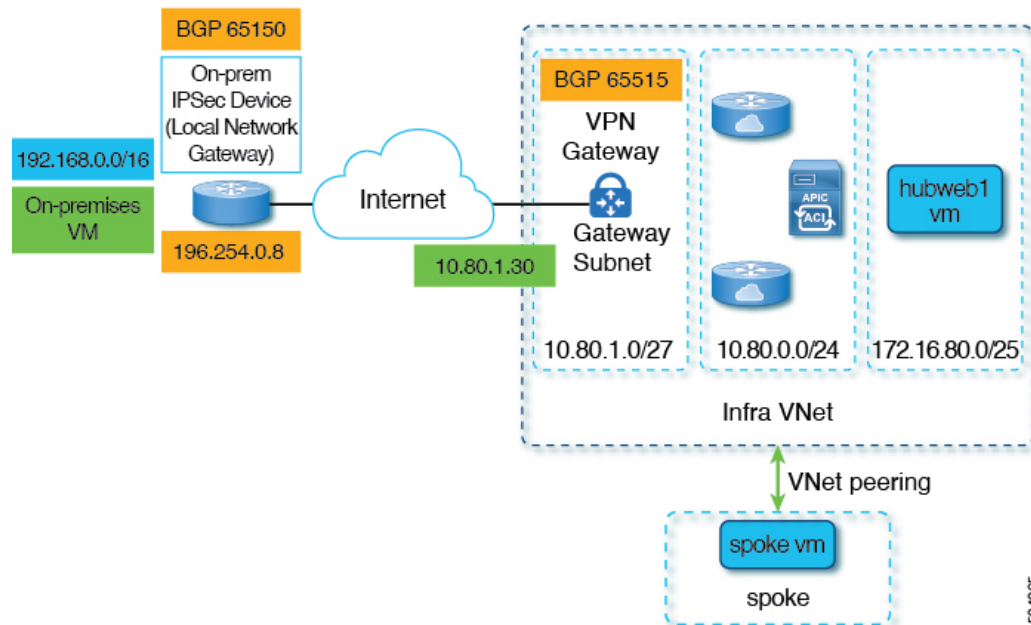
If you need express route gateway access on additional regions, repeat these steps for each of those regions.

- Step 4** Configure a contract between the cloud EPG and the external EPG connected by the express route gateway.
- To create a contract using the GUI, see [Creating a Contract Using the Cisco Cloud Network Controller GUI, on page 103](#).
  - To configure a contract using the REST API, see [Creating a Contract Using the REST API, on page 149](#).

## Configuring Connectivity Using VPN Gateway (Virtual Network Gateway)

Support is available for providing connectivity between a Cisco Cloud Network Controller-managed cloud site and a non-ACI remote site using VPN gateway. For this type of connectivity, a virtual network gateway (VNG) is deployed in the infra (hub) VNet, allowing you to connect from the Cisco Cloud Network Controller-managed cloud site to a non-ACI remote branch site. BGP runs over the IPsec tunnels as the routing protocol between the CCR routers and VNG in the infra VNet and the on-premises IPsec device (local network gateway) in the non-ACI remote branch site.

The following figure shows an example configuration with this type of connectivity.



The following procedures describe how to configure this type of connectivity, where the end result is to have reachability between the on-premises virtual machine residing in the 192.168.20.0/24 subnet and the `hubweb` virtual machine residing in the 172.16.80.0/25 subnet.

## Configuring Connectivity Using VPN Gateway


### Before you begin

Review the information provided in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295 before proceeding with these procedures.

**Step 1** Enable VNet peering on your Cisco Cloud Network Controller, if necessary.

Refer to [Configuring VNET Peering for Cisco Cloud Network Controller for Azure](#) for those instructions.

**Step 2** Add the second subnet for the VPN gateway subnet.

- a) In the Cisco Cloud Network Controller GUI, click the Intent icon (  ) and select **Cisco Cloud Network Controller Setup**.
- b) In the **Region Management** area, click **Edit Configuration**.
- c) In the **Regions to Manage** window, click **Next**.

The **General Connectivity** window appears.

- d) Under the **General** area, in the **Subnet Pools for Cloud Routers** field, click **Add Subnet Pool for Cloud Routers**.
- e) Enter the information for the second subnet for the VPN gateway router.

For example, using the example configuration in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295, you would add 10.80.1.0/24 for the second subnet for the VPN gateway router in this field.

- f) Click **Next**, then enter the necessary information in the following page and click **Save and Continue**.

Cisco Cloud Network Controller will create the subnet for the VPN gateway router after you have completed the **Cisco Cloud Network Controller Setup** process. You can verify that the configuration for the subnet for the VPN gateway router was pushed to Azure successfully by navigating to the **Subnets** page in the Azure portal and locating the **GatewaySubnet** entry.

**Step 3** Create an infra-hosted VRF and use that VRF for the site-external EPG.

You will create an infra-hosted VRF, where you have a VRF that is hosted within the parent infra VNet, and you will use that VRF for the site-external EPG that you will create in the next step.

- a) In the Cisco Cloud Network Controller GUI, navigate to **Application Management > VRFs**.
- b) Click **Actions > Create VRF**.  
The **Create VRF** window appears.
- c) Enter a name for this infra-hosted VRF, then click **Select Tenant** and select **infra** for the tenant and click **Select**.
- d) Enter a description if necessary, then click **Save**.

**Step 4** Create an external EPG in the hub VNet that represents the network for the non-ACI remote site.

- To create an external EPG using the GUI, see [Creating an External EPG Using the Cisco Cloud Network Controller GUI](#), on page 85.

- In the **VRF** field for the external EPG, select the infra-hosted VRF that you just created for this external EPG.
- In the **Route Reachability** field for the external EPG, select **External-Site**.
- To create an external EPG using the REST API, see [Creating an External Cloud EPG Using the REST API, on page 154](#).
  - Use the infra-hosted VRF for this site-external EPG.
  - Create an external cloud EPG with the type **site-external**.

**Step 5**

Through the Azure portal, create the virtual network gateway in the infra VNet for the VPN gateway subnet that you configured in [Step 2, on page 296](#).

In these steps, you will build the IPsec and BGP connections from the on-premises site to the Azure VPN gateway. For more information, see the following article in the Azure site:

<https://docs.microsoft.com/en-gb/azure/virtual-network/virtual-network-configure-vnet-connections>

- a) In the Azure portal, create the virtual network gateways by navigating to the Resource Manager virtual network where you want to create a virtual network gateway.
- b) On the left side, select **Create a resource**, and type **Virtual Network Gateway** in search.
- c) Locate **Virtual network gateway** in the search return and click the entry.
- d) On the **Virtual network gateway** page, choose **Create**.
- e) On the **Create virtual network gateway** page, enter the appropriate information for these fields:
  - **Subscription**: Verify that the correct subscription is selected.
  - **Resource Group**: The resource group will automatically be chosen once you choose the virtual network.
  - **Name**: The name of your virtual network gateway.
  - **Region**: Change the **Region** field to point to the location where your virtual network is located. If the location isn't pointing to the region where your virtual network is, the virtual network won't appear in the **Choose a virtual network** dropdown.
  - **Gateway type**: Choose **VPN**.
  - **VPN type**: Choose **Route-based**.
  - **SKU**: Choose **VpnGw1**.
  - **Generation**: Choose **Generation1**.
  - **Virtual network**: Choose **overlay-1**.
  - **Public IP address**: Choose **Create new**.
  - **Public IP address name**: Provide a name for the public IP address.
  - **Enable active-active mode**: Set to **Disabled**.
  - **Configure BGP**: Set to **Enabled**.
  - **Autonomous system number (ASN)**: Enter the appropriate BGP ASN value for the VPN gateway. By default, Azure uses an ASN value of 65515.
- f) Select **Review + Create**, and then **Create** to begin creating the gateway.

The settings are validated and the gateway deploys. Creating a virtual network gateway can take up to 45 minutes to complete.

To verify that the virtual network gateway was deployed successfully, navigate to the virtual network gateways page and select the virtual network gateway that you just created, then click on **Settings: Configuration** to view and verify the configuration settings for the virtual network gateway.

### Step 6 Create the local network gateway.

For this configuration, the local network gateway is an object that represents the on-premises IPsec device. Prepare the following parameters before creating the local network gateway:

- BGP autonomous system number (ASN)
  - Public IP address
  - An appropriate address space for the on-premises subnet that needs to be advertised to the virtual network gateway
- a) In the Azure portal, create the local network gateway by navigating to the Resource Manager local network where you want to create a local network gateway.
  - b) On the left side, select **Create a resource**, and type **Local Network Gateway** in search.
  - c) Locate **Local network gateway** in the search return and click the entry.
  - d) On the **Local network gateway** page, choose **Create**.
  - e) On the **Create local network gateway** page, enter the appropriate information for these fields:
    - **Name:** The name of your local network gateway.
    - **Endpoint:** Choose **IP address**.
    - **IP address:** Enter the appropriate IP address for the local network gateway.
    - **Address space:** Enter the appropriate value for the address space. For example, using the example configuration in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295, you would add 192.168.0.0/16 in this field.
    - **Configure BGP settings:** Click the checkbox to enable this setting.
    - **Autonomous system number (ASN):** Enter the appropriate BGP ASN value for the local network gateway. This is the ASN value of the remote device. For example, using the example configuration in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295, you would add 65150 in this field.
    - **BGP peer IP address:** Enter the BGP peer IP address that you will use for the on-premises device in this field (not the Azure virtual network gateway). For example, using the example configuration in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295, you would add 196.254.0.8 in this field.
    - **Subscription:** Choose the same subscription that you used for the virtual network gateway in [Step 5, on page 297](#).
    - **Resource group:** Choose the same resource group that you used for the virtual network gateway in [Step 5, on page 297](#).
    - **Location:** Choose the same location (region) that you used for the virtual network gateway in [Step 5, on page 297](#).
  - f) Select **Review + Create**, and then **Create** to begin creating the gateway.

The settings are validated and the gateway deploys.

To verify that the local network gateway was deployed successfully, navigate to the local network gateways page and select the local network gateway that you just created, then click on **Settings: Configuration** to view and verify the configuration settings for the local network gateway.

### Step 7

Create the VPN connection from the Azure virtual network gateway to the local network gateway (the on-premises IPsec device).

- In the Azure portal, navigate to the virtual network gateway page and locate the Azure virtual network gateway that you created in [Step 5, on page 297](#).
- Select the virtual network gateway that you created and click on **Settings: Connections**.
- Click **Add**.

The **Add connection** window appears.

- Fill in the necessary information to add this VPN connection from the Azure virtual network gateway to the local network gateway (the on-premises IPsec device).
  - In the **Connection type** field, select `Site-to-site (IPsec)`.
  - In the **Virtual network gateway** field, select the Azure virtual network gateway that you created in [Step 5, on page 297](#).
  - In the **Local network gateway** field, select the local network gateway that you created in [Step 6, on page 298](#).
  - In the **Enable BGP** field, click the checkbox to enable BGP for this connection.
  - In the **IKE Protocol** field, select `IKEv2`.
- Click **OK** when you have finished entering the configuration information for this VPN connection.

### Step 8

Download the VPN configuration template from Azure.

- In the Azure portal, navigate to the virtual network gateway page and locate the Azure virtual network gateway that you created in [Step 5, on page 297](#).
- Select the virtual network gateway that you created and click on **Settings: Connections**.
- Select the name of the VPN connection that you just configured.

The overview page for that VPN connection appears.

- Click **Download configuration**.

The **Download configuration** page appears.

- Make the following selections in the **Download configuration** page:

- In the **Device vendor** field, select `Cisco`.
- In the **Device family** field, select `IOS (ISR, ASR)`.
- In the **Firmware version** field, select `15.x (IKEv2)`.

- Click **Download configuration**.

### Step 9

Open the downloaded configuration template file in a text editor and make the necessary edits using the instructions in the configuration template.

Typically, the only changes needed in the configuration template are the following fields in the BGP configuration:

- **LOCAL\_ROUTE:** Must be the network that needs to be advertised to Azure. For example, using the example configuration in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295, you would enter 192.168.0.0 in this field.
- **LOCAL\_MASK:** Must be 255.255.255.0

**Step 10** Save and close the edited configuration template.

**Step 11** Apply the edited configuration template to the on-premises IPsec device.

Following is an example edited configuration template based on the example configuration in [Configuring Connectivity Using VPN Gateway \(Virtual Network Gateway\)](#), on page 295:

```
access-list 101 permit ip 192.168.0.0 0.0.255.255 10.80.0.0 0.0.0.127
access-list 101 permit ip 192.168.0.0 0.0.255.255 10.80.0.128 0.0.0.127
access-list 101 permit ip 192.168.0.0 0.0.255.255 10.80.1.0 0.0.0.127
access-list 101 permit esp host 52.152.235.192 host 173.39.125.130
access-list 101 permit udp host 52.152.235.192 eq isakmp host 173.39.125.130
access-list 101 permit udp host 52.152.235.192 eq non500-isakmp host 173.39.125.130
!
crypto ikev2 proposal Azure-Ikev2-Proposal
 encryption aes-cbc-256
 integrity sha1
 group 2
 exit
!
crypto ikev2 policy Azure-Ikev2-Policy
 proposal Azure-Ikev2-Proposal
 match address local 173.39.125.130
 exit
!
crypto ikev2 keyring singaporeisr-keyring
 peer 52.152.235.192
 address 52.152.235.192
 pre-shared-key 0123456789cisco
 exit
 exit
!
crypto ikev2 profile Azure-Ikev2-Profile
 match address local 173.39.125.130
 match identity remote address 52.152.235.192 255.255.255.255
 authentication remote pre-share
 authentication local pre-share
 lifetime 28800
 dpd 10 5 on-demand
 keyring local singaporeisr-keyring
 exit
!
crypto ipsec transform-set Azure-TransformSet esp-aes 256 esp-sha256-hmac
 mode tunnel
 exit
!
crypto ipsec profile Azure-IPsecProfile
 set transform-set Azure-TransformSet
 set ikev2-profile Azure-Ikev2-Profile
 set security-association lifetime seconds 3600
 ! Note: PFS (perfect-forward-secrecy) is an optional feature (commented out)
 !set pfs None
 exit
!
int tunnel 11
 ip address 169.254.0.1 255.255.255.255
```



```
tunnel mode ipsec ipv4
ip tcp adjust-mss 1350
tunnel source 173.39.125.130
tunnel destination 52.152.235.192
tunnel protection ipsec profile Azure-IPsecProfile
exit

interface Loopback 11
ip address 196.254.0.8 255.255.255.255
exit
!
router bgp 65150
bgp log-neighbor-changes
neighbor 10.80.1.30 remote-as 65515
neighbor 10.80.1.30 ebgp-multihop 255
neighbor 10.80.1.30 update-source loopback 11

address-family ipv4
network 192.168.0.0 mask 255.255.0.0
neighbor 10.80.1.30 activate
exit
exit
!
ip route 10.80.0.0 255.255.255.128 Tunnel 11
ip route 10.80.0.128 255.255.255.128 Tunnel 11
ip route 10.80.1.0 255.255.255.128 Tunnel 11
ip route 10.80.1.30 255.255.255.255 Tunnel 11
```

**Step 12** Verify the VPN connections.

- In the Azure portal, navigate to the virtual network gateway page and locate the Azure virtual network gateway that you created in [Step 5, on page 297](#).
- Select the virtual network gateway that you created and click on **Settings: Connections**.
- Verify that the VPN connection that you created is shown as `Connected` in the **Status** column.

**Step 13** Determine if you are deploying the virtual network gateway with or without redirect.

- If you are deploying the virtual network gateway without redirect, go to [Step 14, on page 301](#).
- If you are deploying the virtual network gateway with redirect, configure the service device for the redirect. To configure a service device for redirect using the GUI or REST API, see [Deploying Layer 4 to Layer 7 Services, on page 177](#).

**Step 14** Configure a contract between the cloud EPG and the external EPG connected by the virtual network gateway.

- To create a contract using the GUI, see [Creating a Contract Using the Cisco Cloud Network Controller GUI, on page 103](#).
  - To configure a contract using the REST API, see [Creating a Contract Using the REST API, on page 149](#).
-





# APPENDIX A

## Cisco Cloud Network Controller Error Codes

- [Cisco Cloud Network Controller Error Codes, on page 303](#)

### Cisco Cloud Network Controller Error Codes

This section describes the Cisco Cloud Network Controller error codes.

**Table 45: Cisco Cloud Network Controller Error Codes**

| Component      | Error Code                                  | Constraint                                                                                                                            |
|----------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| cloud-template | CT_INFRANETWORK_COUNT                       | The count of <code>cloudtemplateInfraNetwork</code> MO is at most 1                                                                   |
| cloud-template | CT_INFRANETWORK_VRF                         | In the <code>cloudtemplateInfraNetwork</code> MO, the <code>vrfName</code> must be <code>overlay-1</code>                             |
| cloud-template | CT_INFRANETWORK_PARENT                      | For the <code>cloudtemplateInfraNetworkMO</code> , the parent MO must be <code>uni/tn-infra</code>                                    |
| cloud-template | CT_INFRANETWORK_NUMROUTERSPERREGION_MINIMUM | In the <code>cloudtemplateInfraNetwork</code> MO, for the attribute <code>numRoutersPerRegion</code> , the minimum allowed value is 2 |
| cloud-template | CT_INFRANETWORK_NUMROUTERSPERREGION_MAXIMUM | In the <code>cloudtemplateInfraNetwork</code> MO, for the attribute <code>numRoutersPerRegion</code> , the maximum allowed value is 4 |
| cloud-template | CT_INTNETWORK_COUNT                         | The count of <code>cloudtemplateIntNetwork</code> MO is at most 1                                                                     |

| Component      | Error Code                           | Constraint                                                                                                                                                                                                            |
|----------------|--------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cloud-template | CT_EXTNETWORK_COUNT                  | The count of <code>cloudtemplateExtNetwork MO</code> is at most 1                                                                                                                                                     |
| cloud-template | CT_VPNNETWORK_COUNT                  | The count of <code>cloudtemplateVpnNetwork MO</code> is at most 1                                                                                                                                                     |
| cloud-template | CT_OSPF_COUNT                        | The count of <code>cloudtemplateOspf MO</code> is at most 1                                                                                                                                                           |
| cloud-template | CT_INTNETWORK_REGION_MATCH           | The regions specified by <code>cloudRegionName</code> under <code>cloudtemplateIntNetwork</code> must have a corresponding <code>cloudRegion</code> under <code>cloudProvP</code>                                     |
| cloud-template | CT_INTNETWORK_REGION_MANAGED         | The regions specified by the <code>cloudRegionName</code> children of <code>cloudtemplateIntNetwork</code> must have corresponding <code>cloudRegion</code> with <code>adminSt</code> as managed                      |
| cloud-template | CT_INTNETWORK_REGION_MAXIMUM         | The maximum number of regions ( <code>cloudRegionName</code> ) specified under <code>cloudtemplateIntNetwork</code> is 4                                                                                              |
| cloud-template | CT_EXTNETWORK_REGION_SUBSET          | The regions specified by the <code>cloudRegionName</code> children of <code>cloudtemplateExtNetwork</code> must also be specified by <code>cloudRegionName</code> children under <code>cloudtemplateIntNetwork</code> |
| cloud-template | CT_EXTNETWORK_REQUIRES_EXTSUBNETPOOL | The presence of <code>cloudtemplateExtNetwork</code> requires presence of <code>cloudtemplateExtSubnetPool</code>                                                                                                     |
| cloud-template | CT_EXTSUBNETPOOL_COUNT               | The count of <code>cloudtemplateExtSubnetPool</code> is at most 1                                                                                                                                                     |
| cloud-template | CT_EXTSUBNETPOOL_SUBNETPOOL_ADDRESS  | In <code>cloudtemplateExtSubnetPool</code> , the <code>subnetpool</code> must contain a network address.                                                                                                              |

| Component      | Error Code                               | Constraint                                                                                                                                                                                                                                                                                                                                                                                                               |
|----------------|------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cloud-template | CT_EXTSUBNETPOOL_SUBNETPOOL_IP_VERSION   | In <code>cloudtemplateExtSubnetPool</code> , the subnetpool must contain an IPv4 address.                                                                                                                                                                                                                                                                                                                                |
| cloud-template | CT_EXTSUBNETPOOL_SUBNETPOOL_ADDRESS_TYPE | In <code>cloudtemplateExtSubnetPool</code> , the subnetpool IP address must not from multicast or loopback address space                                                                                                                                                                                                                                                                                                 |
| cloud-template | CT_EXTSUBNETPOOL_SUBNETPOOL_MINIMUM_SIZE | In <code>cloudtemplateExtSubnetPool</code> , the subnetpool must be at least /22 (i.e. the netmask must be 22 or less).                                                                                                                                                                                                                                                                                                  |
| cloud-template | CT_EXTSUBNETPOOL_AND_REMOTESITE          | The <code>cloudtemplateExtSubnetPool</code> needs to be big enough to have at least one <code>cloudtemplateRemoteSiteSubnetPool</code> for each <code>cloudtemplateRemoteSite</code> .                                                                                                                                                                                                                                   |
| cloud-template | CT_INTNETWORK_MISSING_HOME               | If there are any <code>cloudRegionName</code> under <code>cloudtemplateIntNetwork</code> , then one of the <code>cloudRegionName</code> must be associated to a region that is the home region of the Cisco Cloud Network Controller ( <code>capicDeployed</code> ).                                                                                                                                                     |
| cloud-template | CT_CLOUD_APICSUBNETPOOL_INSUFFICIENT     | There must be enough <code>cloudApicSubnetPool</code> MOs to generate <code>cloudApicSubnet</code> MOs so that all the <code>cloudRegionName</code> MOs specified under <code>cloudtemplateIntNetwork</code> can be associated to a unique <code>cloudApicSubnet</code> MO. The subnets from the <code>cloudApicSubnet</code> MOs are used as the CIDRs in the <code>cloudCtxProfile</code> of the corresponding region. |
| cloud-template | CT_IPSECTUNNEL_PEERADDR_IP_VERSION       | In <code>cloudtemplateIpSecTunnel</code> , the <code>peeraddr</code> must contain a IPv4 address.                                                                                                                                                                                                                                                                                                                        |

| Component      | Error Code                                      | Constraint                                                                                                                                                                                                                                                                                                      |
|----------------|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cloud-template | CT_IPSECTUNNEL_PEERADDR_IS_HOST                 | In <code>cloudtemplateIpSecTunnel</code> , the <code>peeraddr</code> must be host address (i.e. /32)                                                                                                                                                                                                            |
| cloud-template | CT_PROFILE_COUNT                                | The count of <code>cloudtemplateProfile</code> MO is at most 1                                                                                                                                                                                                                                                  |
| cloud-template | CT_PROFILE_DELETE                               | The <code>cloudtemplateProfile</code> MO cannot be deleted unless its parent <code>cloudtemplateInfraNetwork</code> is also deleted.                                                                                                                                                                            |
| cloud-template | CT_AZURE_PROFILE_ROUTERUSERNAME_INVALID         | In Azure, some usernames are not valid (admin, root, ...) and must not end with a period.                                                                                                                                                                                                                       |
| cloud-template | CT_AZURE_PROFILE_ROUTERUSERNAME_TOO_LONG        | In Azure, username is restricted to max 20 characters.                                                                                                                                                                                                                                                          |
| cloud-template | CT_PROFILE_ROUTERUSERNAME_NONEMPTY              | In <code>cloudtemplateProfile</code> , the <code>routerUsername</code> must be non-empty.                                                                                                                                                                                                                       |
| cloud-template | CT_PROFILE_ROUTERPASSWORD_NONEMPTY              | In <code>cloudtemplateProfile</code> , the <code>routerLicenseToken</code> must not contain invalid characters.                                                                                                                                                                                                 |
| cloud-template | CT_PROFILE_ROUTERTHROUGHPUT_MODIFY              | In <code>cloudtemplateProfile</code> , the <code>routerThroughput</code> cannot be modified when there are routers deployed in any region, i.e. any <code>cloudRegionName</code> under <code>cloudtemplateIntNetwork</code> . (The modification is allowed when there are no router deployments in any region.) |
| cloud-template | CT_PROFILE_ROUTERLICENSETOKEN_INVALID_CHARACTER | In <code>cloudtemplateProfile</code> , the <code>routerPassword</code> must be non-empty.                                                                                                                                                                                                                       |
| cloud-template | CT_APICSUBNET_INVALID_HOME_REGION               | In a <code>cloudApicSubnet</code> MO, the region marked for <code>capicDeployed</code> must be a valid region                                                                                                                                                                                                   |
| cloud-template | CT_APICSUBNET_REPEATED_REGION                   | In a <code>cloudApicSubnet</code> MO, a region can be associated with at most 1 subnet                                                                                                                                                                                                                          |

| Component      | Error Code                              | Constraint                                                                                                                                    |
|----------------|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| cloud-template | CT_APICSUBNET_MULTIPLE_HOME_REGION      | In <code>cloudApicSubnet</code> MOs, at most 1 region may have <code>capicDeployed</code> as true.                                            |
| cloud-template | CT_HUBNETWORK_COUNT                     | The count of <code>cloudtemplateHubNetwork</code> MO is at most 1                                                                             |
| cloud          | CLOUD_APICSUBNETPOOL_CREATEDBY_USER     | In <code>cloudApicSubnetPool</code> , the <code>createdBy</code> attribute must be USER                                                       |
| cloud          | CLOUD_APICSUBNETPOOL_SUBNET_IP_VERSION  | In <code>cloudApicSubnetPool</code> , the subnet must contain a IPv4 address.                                                                 |
| cloud          | CLOUD_APICSUBNETPOOL_SUBNET_SIZE        | In <code>cloudApicSubnetPool</code> , the subnet must be /24.                                                                                 |
| cloud          | CLOUD_APICSUBNETPOOL_DELETE_USAGE       | A <code>cloudApicSubnetPool</code> cannot be deleted if at least one of its <code>cloudApicSubnet</code> child is in use by a region.         |
| cloud          | CLOUD_APICSUBNETPOOL_DELETE_CREATEDBY   | A <code>cloudApicSubnetPool</code> whose <code>createdBy</code> attribute is not USER cannot be deleted.                                      |
| cloud          | CLOUD_AZURE_CTXPROFILE_SUBNET_RENAME    | <code>cloudSubnet</code> name cannot be modified                                                                                              |
| cloud          | CLOUD_AZURE_CTXPROFILE_SUBNET_DUPLICATE | Two <code>cloudSubnet</code> in the same <code>cloudCtxProfile</code> cannot have the same name                                               |
| cloud          | CLOUD_CAPIC_IP_EXT_EPG_SELECTOR_MAXIMUM | There can be maximum 1 <code>cloudExtEpSelector</code> in the <code>cloudExtEPg</code> corresponding to Cisco Cloud Network Controller IP     |
| cloud          | CLOUD_AZURE_ACCOUNT_IN_USE              | The association between the account and the tenant cannot be updated or deleted while the account is in use and context profiles are deployed |
| cloud          | CLOUD_AZURE_INFRA_ACCOUNT_CHANGE        | The account for the tenant infra cannot be modified or deleted                                                                                |
| cloud          | CLOUD_SOURCE_PORT_NOT_SUPPORTED         | Source port range is not allowed on Cisco Cloud Network Controller                                                                            |

| Component      | Error Code                                      | Constraint                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------------|-------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| cloud          | CLOUD_ONLY_PERMIT_ACTION_SUPPORTED              | Actions different from 'permit' are not supported on Cisco Cloud Network Controller                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| cloud          | CLOUD_CIDR_OVERLAP                              | The subnets of <code>cloudCidrs</code> cannot overlap                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| cloud          | CLOUD_SUBNET_USAGE                              | There can be at most 1 gateway subnet for a given zone and each user subnet should have exactly 1 gateway subnet in the same user subnet's zone                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| cloud          | CLOUD_AZURE_ACCOUNT_CRED_CROSS_TENANT           | The <code>cloudCredentials</code> used by the <code>cloudAccount</code> must be in the same tenant                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| cloud          | CLOUD_AZURE_ACCOUNT_AD_CROSS_TENANT             | The <code>cloudAd</code> used by the <code>cloudAccount</code> must be in the same tenant                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| cloud-template | CT_CLOUD_APICSUBNETPOOL_INSUFFICIENT_HUBNETWORK | There must be enough <code>cloudApicSubnetPool</code> MOs to generate <code>cloudApicSubnet</code> MOs so that all the <code>cloudRegionName</code> MOs specified under <code>cloudtemplateIntNetwork</code> can be associated to a unique <code>cloudApicSubnet</code> MO. The subnets from the <code>cloudApicSubnet</code> MOs are used as the CIDRs in the <code>cloudCtxProfile</code> of the corresponding region. With <code>HubNetworking</code> enabled, there must be as many <code>cloudApicSubnetPool</code> as the <code>cloudRegionName</code> under <code>cloudtemplateIntNetwork</code> . |
| cloud          | CLOUD_SYSTEM_MO_IS_IMMUTABLE                    | Instances created by the system are immutable                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| cloud-template | CT_BGPEVPN_PEERADDR_IP_VERSION                  | In <code>cloudtemplateBgpEvpn</code> , the <code>peeraddr</code> must contain a IPv4 address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| cloud-template | CT_BGPEVPN_PEERADDR_ADDRESS_TYPE                | In <code>cloudtemplateBgpEvpn</code> , the <code>peeraddr</code> IP address must be a host address                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| cloud          | CLOUD_APICSUBNETPOOL_SUBNET_HOST_PART           | In <code>cloudApicSubnetPool</code> subnet, the host part must be 0.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |



| Component      | Error Code                                    | Constraint                                                                                                       |
|----------------|-----------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| cloud-template | CT_EXTSUBNETPOOL_CLOUD_APICSUBNETPOOL_OVERLAP | There is a subnet overlap between <code>cloudtemplateExtSubnetPool</code> and <code>cloudApicSubnetPool</code> . |





## APPENDIX **B**

# Service EPG Configuration Examples

---

For more information on service EPGs, see:

- [Cloud Service Endpoint Groups, on page 38](#)
- [Creating a Service EPG Using the Cisco Cloud Network Controller GUI, on page 92](#)
- [Creating a Service EPG Using the REST API, on page 155](#)

The following sections provide configuration examples for service EPGs.

- [Azure Kubernetes Services \(AKS\) Service EPG Configuration Example, on page 311](#)

## Azure Kubernetes Services (AKS) Service EPG Configuration Example

This section provides procedures for configuring an example service EPG with the following settings:

- **Service Type:** Azure Kubernetes Services (AKS)
  - Azure Kubernetes Services (AKS) requires access to other services.
  - Cisco Cloud Network Controller automates the [programming of the rules](#).
- **Deployment type:** Cloud Native Managed. In this type of deployment, the service is instantiated in your VNet or subnet (created through the Cisco Cloud Network Controller ). For example, an Azure Kubernetes Services (AKS) service could be deployed in a subnet that is managed by the Cisco Cloud Network Controller.
- **Access type:** Private

The procedures to configure this example service EPG for AKS are provided in the following sections.

### Creating a Subnet in the Cloud Context Profile

These procedures describe how to create a subnet in a cloud context profile to be used by the Azure Kubernetes Services (AKS) service EPG. You will be making configurations through the Cisco Cloud Network Controller GUI in these procedures.

### Before you begin

- In one browser window, log into your Cisco Cloud Network Controller GUI.
- In another browser window, log into your Azure account for the Cisco Cloud Network Controller infra tenant and go to the Azure management portal:

<https://portal.azure.com/#home>

**Step 1** In the Cisco Cloud Network Controller GUI, click the **Intent** icon.

The **Intent** menu appears.

**Step 2** Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

**Step 3** From the **Application Management** list in the **Intent** menu, click **Create Cloud Context Profile**.

The **Create Cloud Context Profile** window appears.

**Step 4** Enter the following information in the **Create Cloud Context Profile** window.

- **Name:** Enter the name of the cloud context profile. For example, **ct\_ctxprofile\_eastus**.
- **Tenant:** Click **Select Tenant**, choose a tenant for the cloud context profile for this use case, then click **Select**.
- **Region:** Click **Select Region**, choose the region (for example, **eastus**), then click **Select**.
- **VRF:** Click **Select VRF**, select the appropriate VRF, then click **Select**.
- **Add CIDR:** Enter the CIDR information.
  - a. Click **Add CIDR**.
  - b. Enter the address in the **CIDR Block Range** field.  
For example, **30.1.0.0/16**.
  - c. Uncheck (disable) the **Primary** check box.
  - d. Click **Add Subnet** and enter the subnet address in the **Address** field.

For example, 30.1.0.0/17. Note that AKS cluster requires 338 addresses.

e. Click **Add**.

- **VNet Gateway Router**: Leave the box unchecked (unselected) for this field.
- **VNet Peering**: Check this box to enable VNet peering.

**Step 5** Click **Save** when finished.

---

#### What to do next

Go to [Creating the Cloud Service EPG for AKS, on page 313](#).

## Creating the Cloud Service EPG for AKS

These procedures describe how to create the cloud service EPG with the Azure Kubernetes Services (AKS) service type. You will be making configurations through the Cisco Cloud Network Controller GUI in these procedures.

#### Before you begin

Complete the procedures in [Creating a Subnet in the Cloud Context Profile, on page 311](#) before proceeding with these procedures.

---

**Step 1** In the Cisco Cloud Network Controller GUI, click the **Intent** icon.

The **Intent** menu appears.

**Step 2** Click the drop-down arrow below the **Intent** search box and choose **Application Management**.

A list of **Application Management** options appear in the **Intent** menu.

**Step 3** From the **Application Management** list in the **Intent** menu, click **Create EPG**.

The **Create EPG** window appears.

**Step 4** Enter the following information in the **Create EPG** window.

- **Name:** Enter the name of the cloud service EPG. For example, **svc-Hub-AzureAKS**.
- **Tenant:** Click **Select Tenant**, choose a tenant for the cloud service EPG for this use case, then click **Select**.
- **Application Profile:** Click **Select Application Profile**, choose the application profile, then click **Select**.
- **Type:** Choose **Service** as the EPG type.
- **VRF:** Click **Select VRF**, select the appropriate VRF, then click **Select**.
- **Service Type :** Choose the **Azure Kubernetes Services (AKS)** service type.
- **Deployment Type:** Choose the **Cloud Native Managed** deployment type.
- **Access Type:** Choose the **Private** access type.

**Step 5** Click **Add Endpoint Selector**.

The **Add Endpoint Selector** window appears.

For this use case, we will be creating an endpoint selector where the IP address matches the subnet information configured in the previous step, `30.1.0.0/17`. Having the IP address in the endpoint selector match the subnet in the previous step allows the Cisco Cloud Network Controller to program the NSG to allow all of the required rules for this service type.

**Step 6** In the **Add Endpoint Selector** window, enter a name in the **Name** field.

**Step 7** Click the **Key** drop-down list to choose a key.

At this time, **IP** is the only option available as a key for this access type.

**Step 8** Click the **Operator** drop-down list and choose **equals**.

**Step 9** In the **Value** field, enter `30.1.0.0/17`, then click the check mark to validate the entry.

**Step 10** Click **Add**.

**Step 11** Click **Save** when finished.

**What to do next**

Go to [Verifying the Outbound Security Rules](#), on page 315.

## Verifying the Outbound Security Rules

These procedures describe how to verify that the necessary outbound security rules are getting configured correctly. The Cisco Cloud Network Controller configures all of the outbound security rules in Azure that are needed for AKS to be deployed in the Azure portal.

**Before you begin**

Complete the procedures in [Creating the Cloud Service EPG for AKS](#), on page 313 before proceeding with these procedures.

**Step 1**

In the Azure portal, navigate to the network security group for the subnet that was automatically created:

- a) Navigate to appropriate resource group.
- b) Select the subnet that was used for the AKS service EPG.
- c) Locate the necessary outbound security group.

**Step 2**

Locate the **Outbound security rules** area in the page and verify that the outbound security rules for the NSG are configured correctly.

For more information on the outbound security rules, see:

<https://docs.microsoft.com/en-us/azure/aks/limit-egress-traffic>

**What to do next**

Go to [Creating a Kubernetes Service](#), on page 315.

## Creating a Kubernetes Service

These procedures describes how to create a Kubernetes service. You will be making configurations through the Azure portal in these procedures.



**Note** The following procedure describes how to create a Kubernetes service through the Azure portal. An alternative method for creating a Kubernetes service is also provided in the [Using Azure Kubernetes Service with Cisco Cloud Network Controller](#) document.

**Before you begin**

Complete the procedures in [Verifying the Outbound Security Rules](#), on page 315 before proceeding with these procedures.

**Step 1**

In the Azure portal, search for the term `Kubernetes Service` by Microsoft and click on the search result.

The **Kubernetes Service** page appears.

**Step 2** Click **Create** in the **Kubernetes Service** page.

The **Create Kubernetes cluster** page appears.

[Home](#) > [Kubernetes services](#) >

## Create Kubernetes cluster

Select a subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription \* ⓘ

Resource group \* ⓘ  [Create new](#)

### Cluster details

Kubernetes cluster name \* ⓘ

Region \* ⓘ

Availability zones ⓘ

Kubernetes version \* ⓘ

### Primary node pool

The number and size of nodes in the primary node pool in your cluster. For production workloads, at least 3 nodes are recommended for resiliency. For development or test workloads, only one node is required. If you would like to add additional node pools or to see additional configuration options for this node pool, go to the 'Node pools' tab above. You will be able to add additional node pools after creating your cluster. [Learn more about node pools in Azure Kubernetes Service](#)

Node size \* ⓘ  [Change size](#)

Node count \* ⓘ

**Step 3** In the **Basics** tab, configure the following areas:

- **Subscription:** Select the appropriate subscription.
- **Resource Group:** Select the appropriate resource group.
- **Kubernetes cluster name:** Enter a unique name for this Kubernetes cluster.
- **Region:** Select the appropriate region.
- **Kubernetes version:** Leave the default selection as-is.
- **Node size:** Leave the default selection as-is.
- **Node count:** Verify that the scroll bar is all the way to the left so that the entry for this field is 1.

**Step 4** Click **Next: Node pools**. Leave the default entries as-is and click **Next: Authentication** to advance to the **Authentication** tab.



The image shows two screenshots from the Azure portal. The left screenshot is the 'Create Kubernetes cluster' page, 'Authentication' tab. It shows the 'Authentication method' set to 'Service principal' and the 'Service principal' dropdown menu open, with 'Configure service principal' selected. Below this, there are sections for 'Kubernetes authentication and authorization', 'Role-based access control (RBAC)' (set to 'Enabled'), 'AKS-managed Azure Active Directory' (set to 'Disabled'), and 'Node pool OS disk encryption' (set to '(Default) Encryption at-rest with a platform-managed key'). The right screenshot is the 'Configure service principal' dialog box. It has two radio buttons: 'Create new' (unselected) and 'Use existing' (selected). Below are two text input fields: 'Service principal client ID' and 'Service principal client secret', both containing masked text. An 'OK' button is at the bottom right.

**Step 5** In the **Authentication** tab, configure the following areas:

- **Authentication method:** Choose **Service principal**.  
The **Service principal** field appears.
- **Service principal:** Click **Configure service principal**.

In the **Configure service principal** window, configure the following areas:

- **Service principal:** Choose either **Create new** or **Use existing**.

If you choose **Use existing**, enter the following information for the existing service principal:

- **Service principal client ID**
- **Service principal client secret**

**Note** Make a note of the entries that you enter in these two fields. You will be using the entries in these fields later in these procedures.

Click **OK** to return to the **Authentication** tab in the **Create Kubernetes cluster** window.

- **Role-based access control (RBAC):** Choose **Enabled**.
- **AKS-managed Azure Active Directory:** Choose **Disabled**.
- **Encryption type:** Leave the default selection as-is.

**Step 6** Click **Next: Networking** to advance to the **Networking** tab.

[Home](#) > [Kubernetes services](#) >

## Create Kubernetes cluster

Basics Node pools Authentication **Networking** Integrations Tags Review + create

You can change networking settings for your cluster, including enabling HTTP application routing and configuring your network using either the 'Kubenet' or 'Azure CNI' options:

- The **kubenet** networking plug-in creates a new VNet for your cluster using default values.
- The **Azure CNI** networking plug-in allows clusters to use a new or existing VNet with customizable addresses. Application pods are connected directly to the VNet, which allows for native integration with VNet features.

[Learn more about networking in Azure Kubernetes Service](#)

Network configuration ⓘ

Kubenet

Azure CNI

ⓘ The Azure CNI plugin requires an IP address from the subnet below for each pod on a node, which can more quickly exhaust available IP addresses if a high value is set for pods per node. Consider modifying the default values for pods per node for each node pool on the "Node pools" tab. [Learn more](#) ↗

Virtual network \* ⓘ

Cluster subnet \* ⓘ

Kubernetes service address range \* ⓘ

Kubernetes DNS service IP address \* ⓘ

Docker Bridge address \* ⓘ

DNS name prefix \* ⓘ

Traffic routing

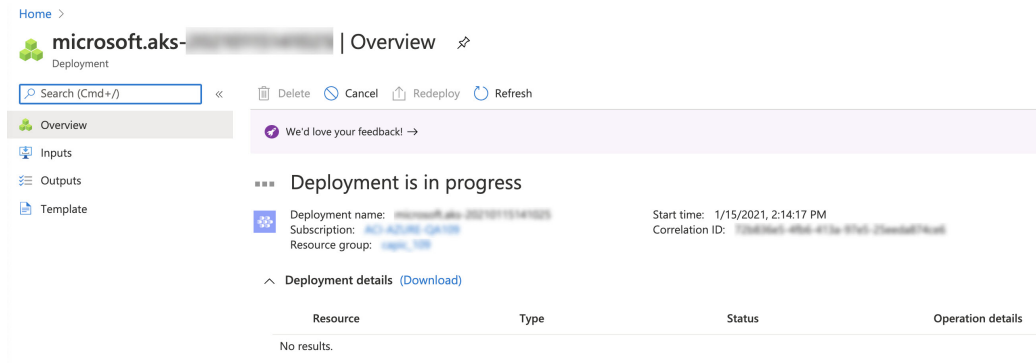
**Step 7** In the **Networking** tab, configure the following areas:

- **Network configuration:** Choose **Azure CNI**.
- **Virtual network:** Choose the corresponding virtual network.
- **Cluster subnet:** Choose the Cisco Cloud Network Controller -managed subnet.
- **Kubernetes service address range:** Leave the default selection as-is, or change the entry, if necessary.
- **Kubernetes DNS service IP address:** Leave the default selection as-is, or change the entry, if necessary.
- **Docker Bridge address:** Leave the default selection as-is, or change the entry, if necessary.
- **DNS name prefix:** Leave the default selection as-is, or change the entry, if necessary.
- **Load balancer:** Standard
- **Enable HTTP application routing:** Leave the default selection as-is (not enabled), or change the entry, if necessary.
- **Enable Private cluster:** Leave the default selection as-is (not enabled), or change the entry, if necessary.

**Step 8** Click **Next: Integration**, then **Next: Tags**, to advance through those screens without changing any of the default entries, then click **Next: Review+Create**.

**Step 9** In the **Review+Create** window, click **Create**, then click **Create** again after the validations pass to create the Kubernetes cluster.

You will see the message `Deployment is in progress` and the Overview screen for the Kubernetes service will appear.



Wait until the Kubernetes service is deployed successfully before proceeding (the amount of time it takes to deploy varies). Once this process is completed, the main AKS service will be in your original resource group. Azure will also create an additional resource group specifically for the Kubernetes service, with all of the agentpools VM scales set.

### What to do next

Go to [Verifying the New Kubernetes Service, on page 319](#).

## Verifying the New Kubernetes Service

These procedures describe how to verify that the new Kubernetes service is in the resource group that was created specifically for the Kubernetes service.

### Before you begin

Complete the procedures in [Creating a Kubernetes Service, on page 315](#) before proceeding with these procedures.

**Step 1** In the Azure portal, click on **Resource groups** in the left nav bar to navigate to the resource groups page.

**Step 2** In the **Resource groups** page, locate the resource group that was created specifically for the Kubernetes service and click the link for that resource group.

The resource group created specifically for the Kubernetes service will have the following format:

`MC_resourcegroupname_clustername_region`

Where:

- `resourcegroupname` is the name of the resource group created specifically for the Kubernetes service (`MC_aks` is the resource group name used by default by Azure)
- `clustername` is the Kubernetes cluster name that you provided in [Step 3, on page 316](#) in [Creating a Kubernetes Service, on page 315](#).
- `region` is the region that you selected in [Step 3, on page 316](#) in [Creating a Kubernetes Service, on page 315](#).

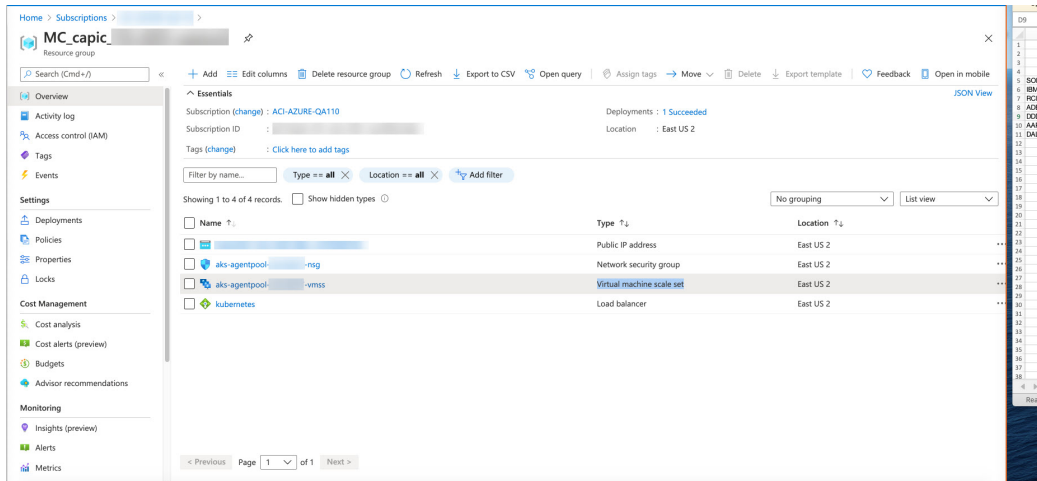
For example:

MC\_aks\_acme-aks-cluster\_centralus

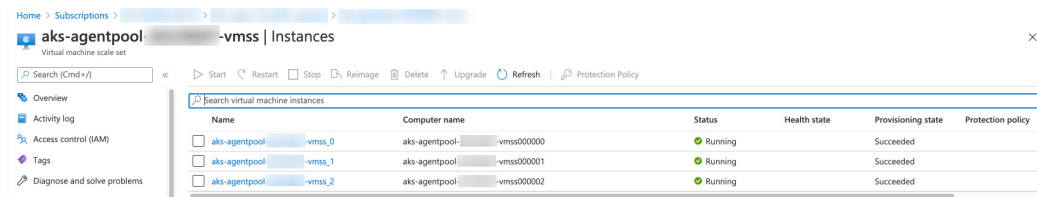
The Overview page for the Kubernetes service resource group appears.

**Step 3** Locate the line for the **Virtual machine scale set** and click on that link.

This is where the AKS agent is running.

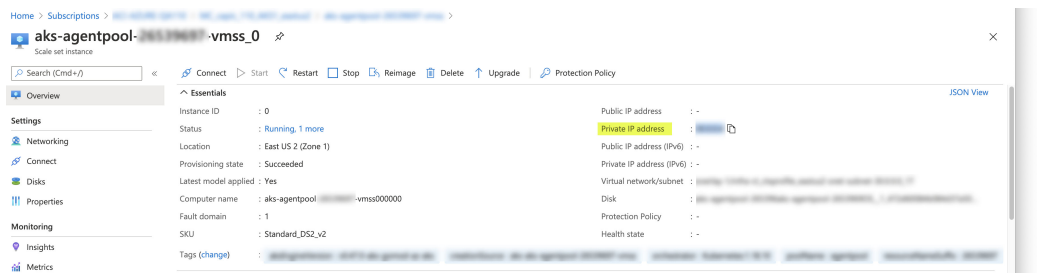


**Step 4** In the left nav bar, click on **Instances** to display the virtual machine instances for this Kubernetes service resource group.



**Step 5** Click on any of the three instances in this window, then verify that the IP address shown in the **Private IP address** field matches your hub subnet IP address.

All three instances shown in this window should have an IP address from the subnet that you selected in [Step 7, on page 318 in Creating a Kubernetes Service, on page 315](#).



**Step 6** Navigate back to the Overview page for the Kubernetes service resource group, then locate the `kubernetes` entry, shown with `Load balancer` as the Type, and click that link.

The Overview page for the Kubernetes load balancer appears.

**Step 7** In the left nav bar, click **Backend pools** to view the AKS agents.

| Backend pool                                | Virtual machine            | Virtual machine status | Network interface  | Private IP address | Availability zone |
|---------------------------------------------|----------------------------|------------------------|--------------------|--------------------|-------------------|
| aksOutboundBackendPool (3 virtual machines) | aks-agentpool-vmss (in...) | Running                | aks-agentpool-vmss |                    | 1                 |
|                                             | aks-agentpool-vmss (in...) | Running                | aks-agentpool-vmss |                    | 2                 |
|                                             | aks-agentpool-vmss (in...) | Running                | aks-agentpool-vmss |                    | 3                 |
| kubernetes (3 virtual machines)             | aks-agentpool-vmss (in...) | Running                | aks-agentpool-vmss |                    | 1                 |
|                                             | aks-agentpool-vmss (in...) | Running                | aks-agentpool-vmss |                    | 2                 |
|                                             | aks-agentpool-vmss (in...) | Running                | aks-agentpool-vmss |                    | 3                 |

**Step 8** If a virtual machine was created as part of the process of configuring the contract (for example, if a virtual machine was created for the consumer), and if you have AKS as the provider, verify that the rules were configured correctly.

- In the Azure portal, navigate back to the infra resource group.
- Choose **Group by type** for the records shown in the Overview page for the infra resource group.
- Scroll down until you see the **Virtual machine** area and click on the virtual machine for the consumer in your contract.

The Overview window for that virtual machine appears.

- In the left nav bar, under **Settings**, click **Networking**.

The Networking window for that virtual machine appears, with information on the inbound and outbound port rules.

- Click on the **Outbound port rules** tab, then click on one of the outbound port rules listed in the table.

A window slides in from the right, displaying additional information on these outbound port rules. For example, the entry in the **Destination IP addresses/CIDR ranges** area provides information on the addresses that are associated with the AKS cluster.

### What to do next

Go to [Installing the Azure and AKS CLI, on page 321](#).

## Installing the Azure and AKS CLI

These procedures describe how to install the Azure and AKS CLI.

### Before you begin

Complete the procedures in [Verifying the New Kubernetes Service, on page 319](#) before proceeding with these procedures.

**Step 1** On the consumer VM that has internet access, install the Azure CLI.

For more information, see:

<https://docs.microsoft.com/en-us/cli/azure/install-azure-cli-linux>

For example, to install the Azure CLI in an Ubuntu Linux VM in Azure:

```
curl -sL https://aka.ms/InstallAzureCliDeb | sudo bash
```

**Step 2** Download and install **kubectrl**, the Kubernetes command-line tool, and **kubelogin**, a client-go credential (exec) plugin implementing azure authentication:

```
az aks install-cli
```

**Step 3** Log in with the service principle information that you entered in [Step 5, on page 317](#) in [Creating a Kubernetes Service, on page 315](#) in these procedures:

```
az login --service-principal --username <service_principal_client_id>
--password '<service_principal_client_secret>' --tenant <tenant_ID>
```

Where:

- *<service\_principal\_client\_id>* is the entry from the **Service principal client ID** field in [Step 5, on page 317](#) in [Creating a Kubernetes Service, on page 315](#).
- *<service\_principal\_client\_secret>* is the entry from the **Service principal client secret** field in [Step 5, on page 317](#) in [Creating a Kubernetes Service, on page 315](#).
- *<tenant\_ID>* is the tenant associated with the service principal (the Azure Active Directory tenant ID). To locate the tenant ID information for this command:
  - a. Sign in to the Azure portal.
  - b. Select **Azure Active Directory**.
  - c. Select **Properties**.
  - d. Scroll down to the **Tenant ID** field. Your tenant ID is displayed in the box.

For more information, see:

<https://docs.microsoft.com/en-us/azure/active-directory/fundamentals/active-directory-how-to-find-tenant>

For example:

```
az login --service-principal --username 12a3b456-7c89-1234-5de6-7f89012gh3i4
--password 'secretkey12341234!' --tenant 98765zy4-xwv-3ut2-1uts-rq0pon98m765
```

**Step 4** Set a subscription to be the current active subscription.

```
az account set --subscription <AKS_rg_subscription_ID>
```

Where *<AKS\_rg\_subscription\_ID>* is the subscription ID of the resource group that Azure created for the Kubernetes service in [Verifying the New Kubernetes Service, on page 319](#).

For example:

```
az account set --subscription 56klm789n-o0p1-234q-5r6s-7t890123u4v5
```

**Step 5** From the consumer VM, enter the following to log in and connect to AKS.

```
root@hub-vm: /home/CNC# az aks get-credentials --resource-group <resource_group> --name
<AKS_cluster_name> --admin
```

Where:

- *<resource\_group>* is the name of the infra resource group
- *<AKS\_cluster\_name>* is the name for the Kubernetes cluster that was entered in [Step 3, on page 316](#) in [Creating a Kubernetes Service, on page 315](#)

For example:

```
root@hub-vm:/home/CNC# az aks get-credentials --resource-group CNC_infra_westus --name azureaksclus --admin
```

A message similar to the following appears:

```
Merged "azureaksclus-admin" as current context in /root/.kube/config
```

**Step 6** Check the internal IP addresses of each of the nodes.

```
root@hub-vm:/home/CNC# kubectl get nodes -o wide
```

Output similar to the following appears:

| NAME                              | STATUS | ROLES | AGE | VERSION | INTERNAL-IP | EXTERNAL-IP | OS-IMAGE           | KERNEL-VERSION    | CONTAINER-RUNTIME |
|-----------------------------------|--------|-------|-----|---------|-------------|-------------|--------------------|-------------------|-------------------|
| aks-agentpool-12345678-vmss000000 | Ready  | agent | 14h | v1.17.9 | 30.1.1.1    | <none>      | Ubuntu 16.04.7 LTS | 4.15.0-1092-azure | docker://19.3.12  |
| aks-agentpool-12345678-vmss000001 | Ready  | agent | 14h | v1.17.9 | 30.1.1.21   | <none>      | Ubuntu 16.04.7 LTS | 4.15.0-1092-azure | docker://19.3.12  |
| aks-agentpool-12345678-vmss000002 | Ready  | agent | 14h | v1.17.9 | 30.1.1.31   | <none>      | Ubuntu 16.04.7 LTS | 4.15.0-1092-azure | docker://19.3.12  |

The IP addresses listed in the `INTERNAL-IP` column are in your hub subnet.

**Note** In the example output above, the entries in the `EXTERNAL-IP` column are shown as `<none>` because the **Access Type** was set to `Private` in [Step 4, on page 314](#) in [Creating the Cloud Service EPG for AKS, on page 313](#). IP addresses would be displayed in the `EXTERNAL-IP` column if the **Access Type** is set to `Public` and `Private`.

**Step 7** (Optional) Assign an admin role to a new user, if necessary.

- In the Azure portal, navigate back to the infra resource group.
- In the records area in the page, scroll down until you find the **Kubernetes service** entries.
- Click on the Kubernetes service that you configured.

The Overview page for that Kubernetes service is displayed.

- In the left nav bar, click on **Access Control (IAM)**.

The Access Control (IAM) for that Kubernetes service is displayed.

- Click + **Add**, then select **Add role assignment** from the drop-down menu.
- In the **Add role assignment** page, make the following selections:
  - In the **Role** field, select **Azure Kubernetes Service Cluster Admin Role** from the drop-down menu.
  - In the **Assign access to** field, select **User, group, or service principal**.
  - Select the appropriate key.
- Click **Save** at the bottom of the screen.

