



Cisco Cloud APIC Policy Model

- [About the ACI Policy Model, on page 1](#)
- [Policy Model Key Characteristics, on page 1](#)
- [Logical Constructs, on page 2](#)
- [The Cisco ACI Policy Management Information Model, on page 3](#)
- [Tenants, on page 5](#)
- [Cloud Context Profile, on page 5](#)
- [VRFs, on page 8](#)
- [Cloud Application Profiles, on page 10](#)
- [Cloud Endpoint Groups, on page 10](#)
- [Security Groups, on page 18](#)
- [Contracts, on page 23](#)
- [About the Cloud Template, on page 26](#)
- [Managed Object Relations and Policy Resolution, on page 29](#)
- [Default Policies, on page 30](#)
- [Shared Services, on page 31](#)

About the ACI Policy Model

The ACI policy model enables the specification of application requirements policies. The Cisco Cloud APIC 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 APIC 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.

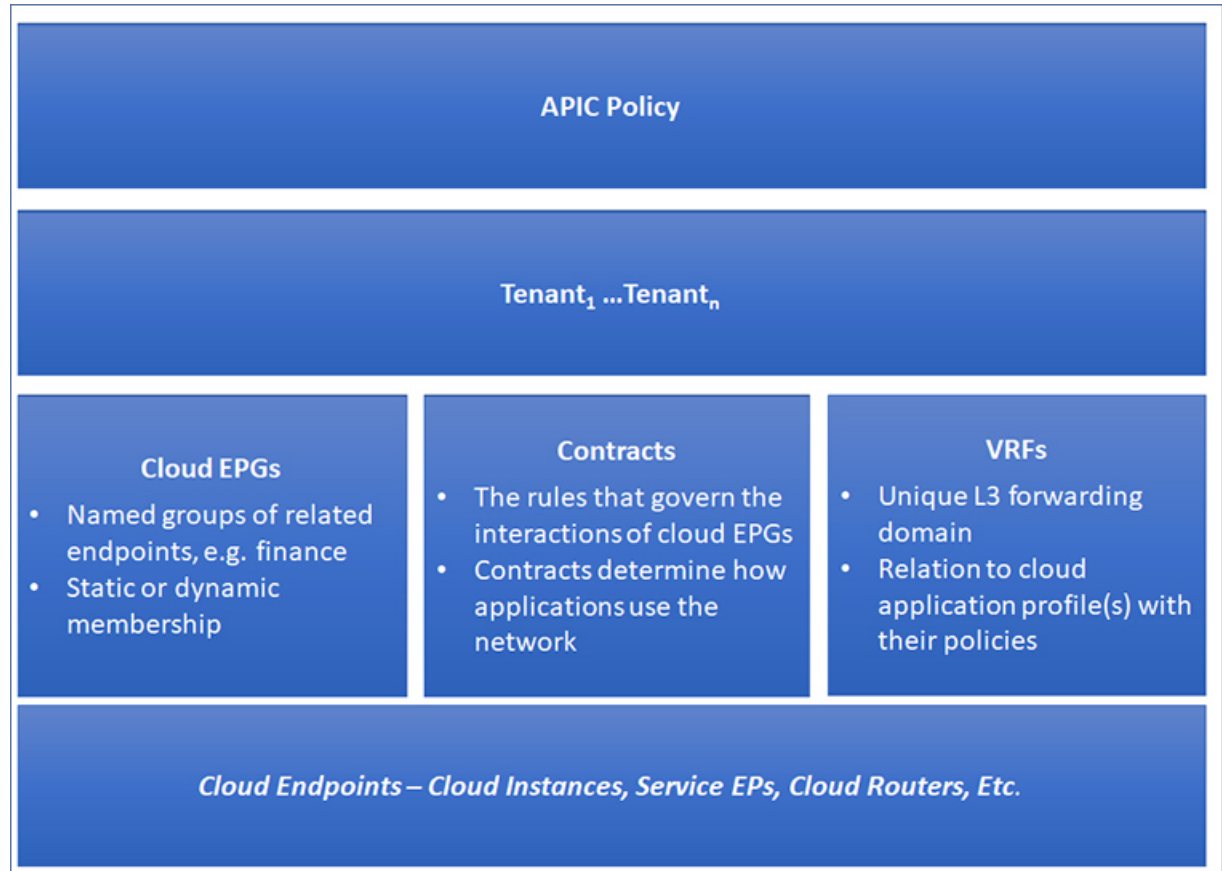
- The system prohibits communications with newly connected endpoints until the policy model is updated to include the new endpoint.
- Network administrators do not configure logical system resources directly. Instead, they define logical (hardware-independent) configurations and the Cisco Cloud APIC 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 APIC 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 ACI policy model logical constructs.

Figure 1: ACI Policy Model Logical Constructs Overview



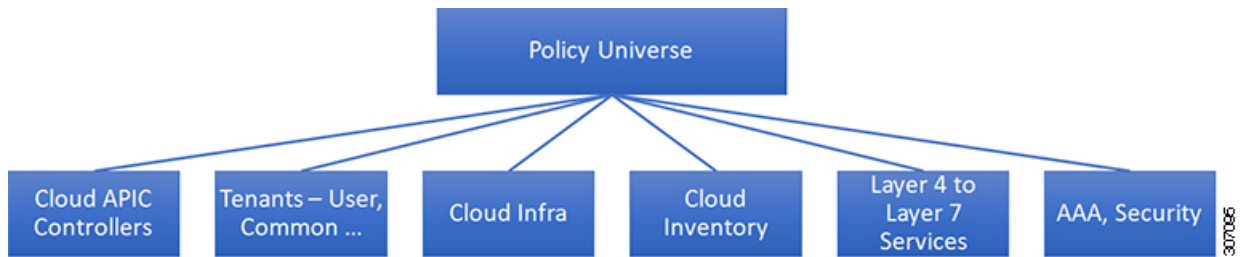
cloud infrastructure-wide or tenant administrators create predefined policies that contain application or shared resource requirements. These policies automate the provisioning of applications, network-attached services, security policies, and tenant subnets, which puts administrators in the position 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 ACI 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 APIC 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 APIC 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 ACI 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 As of the Cisco Application Policy Infrastructure Controller (APIC) Release 4.1(1), the Cisco Cloud APIC only supports load balancers as a Layer 4 to Layer 7 service.

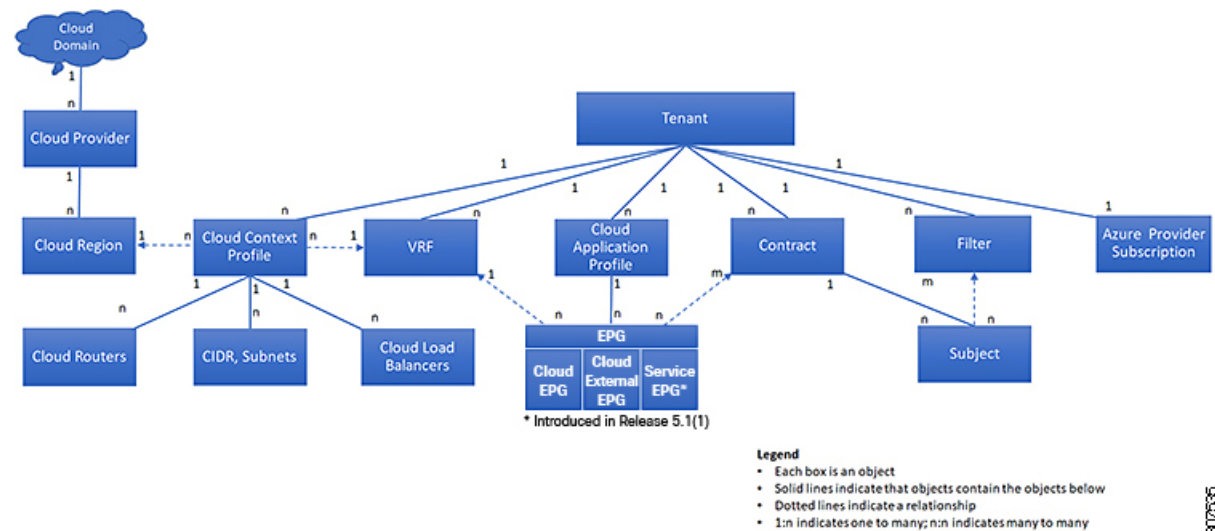
- The infrastructure tenant is provided by the system but can be configured by the cloud infrastructure administrator. It contains policies that govern the operation of infrastructure resources. It also enables a cloud infrastructure provider to selectively deploy resources to one or more user tenants. 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 APIC. For more information, see the *Cisco Cloud APIC 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](#)
- Access, authentication, and accounting (AAA) policies govern user privileges, roles, and security domains of the Cisco Cloud ACI cloud infrastructure. For more information, see [Cisco Cloud APIC Security](#)

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 ACI cloud infrastructure supports IPv4 and dual-stack configurations for tenant networking.

Cloud Context Profile

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

- CIDRs

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

Cloud Service Routers

The Cisco Cloud Services Router 1000V (CSR 1000V) is a virtual router that delivers comprehensive WAN gateway and network services into virtual and cloud environments. The CSR 1000V enables enterprises to extend their WANs into provider-hosted clouds. Two CSR 1000Vs are required for Cisco Cloud ACI solution.

For more information, see the [Cisco CSR 1000v documentation](#).

Changing the Number of CSRs

Beginning with Release 5.1(2), the maximum number of CSRs supported per region increased from 4 to 8. These procedures provide instructions for increasing the number of CSRs above 4, or for reducing the number of CSRs 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 CSRs in a range between 2-4 CSRs. Use these instructions only if you are increasing the number of CSRs above 4, or if you are decreasing the number of CSRs from a range of 5-8 CSRs.
- Changing the number of CSRs can impact traffic for up to 30 minutes.

Step 1 Disable Azure VNet peering 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) 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 CSRs above 4, add additional subnet pools for the additional CSRs, if necessary. You will see an error message is you attempt to increase the number of CSRs above 4 and the system determines that additional subnet pools are required.


- a) In the Cloud APIC GUI, click the Intent icon () and select **cAPIC 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 CSRs.

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 Cloud APIC. This must be a valid IPv4 subnet with mask /24.

Step 3 Increase the number of CSRs above 4, or decrease the number of CSRs from a range of 5-8 CSRs.

- a) In your Cloud APIC GUI, click the Intent icon () and choose **cAPIC 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 CSRs as-is.

The **General Connectivity** window appears.

- d) Locate the **CSRs** 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 CSRs.
- e) Click **Next**, then enter the necessary information in the following page and click **Save and Continue**.

The process of adding or removing the CSRs 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 APIC and Cisco Cloud Services Router

Prior to Release 5.1(2), Cisco Cloud Router (CSR) interfaces were assigned both public and private IP address by Cloud APIC. Beginning with Release 5.1(2), CSR interfaces are assigned private IP addresses only and assignment of public IP addresses to CSR interfaces is optional. Private IP addresses are always assigned to all the interfaces of a CSR. The private IP of GigabitEthernet1 of a CSR is used as BGP and OSPF router IDs. Hcloud with on-premise ACI sites over express route is supported when CSRs are assigned private IP addresses. To enable private IP for a CSR, see [Managing Regions \(Configuring a Cloud Template\) Using the Cisco Cloud APIC GUI](#) procedure.

Prior to Release 5.1(2), the management interface of the Cloud APIC was assigned a public IP address and a private IP address. Beginning with Release 5.1(2), a private IP address is assigned to the management interface of the Cisco Cloud APIC and assigning a public IP address is optional. To enable private IP for Cloud APIC, see *Deploying the Cloud APIC in Azure* procedure in the *Cisco Cloud APIC for Azure Installation Guide*.

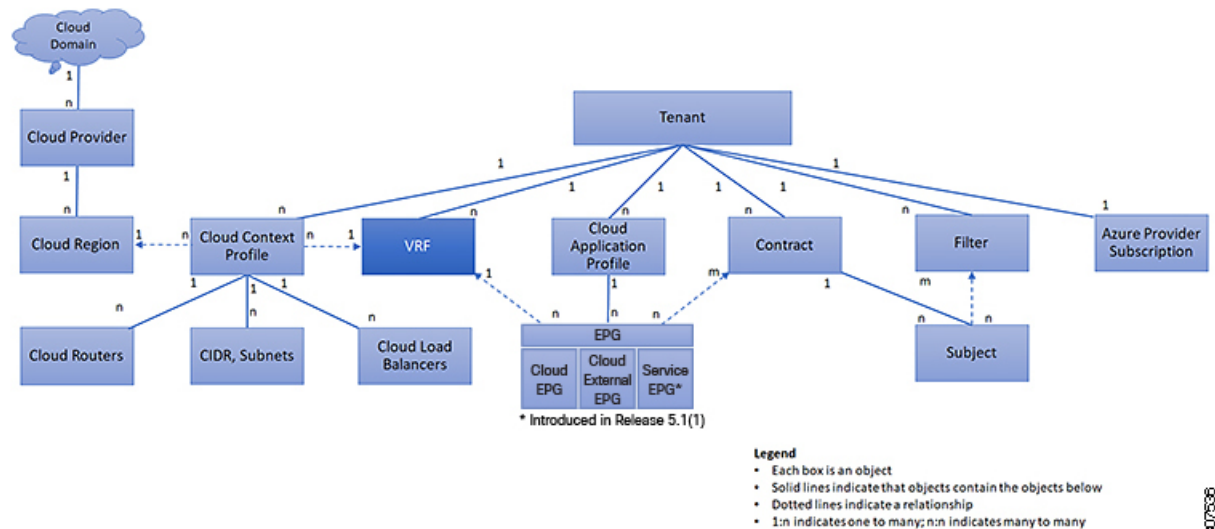
Restrictions for CSR with private IP address:

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

VRFs

A Virtual Routing and Forwarding (VRF) object (`fVCtx`) or context is a tenant network (called a VRF in the Cisco Cloud APIC 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

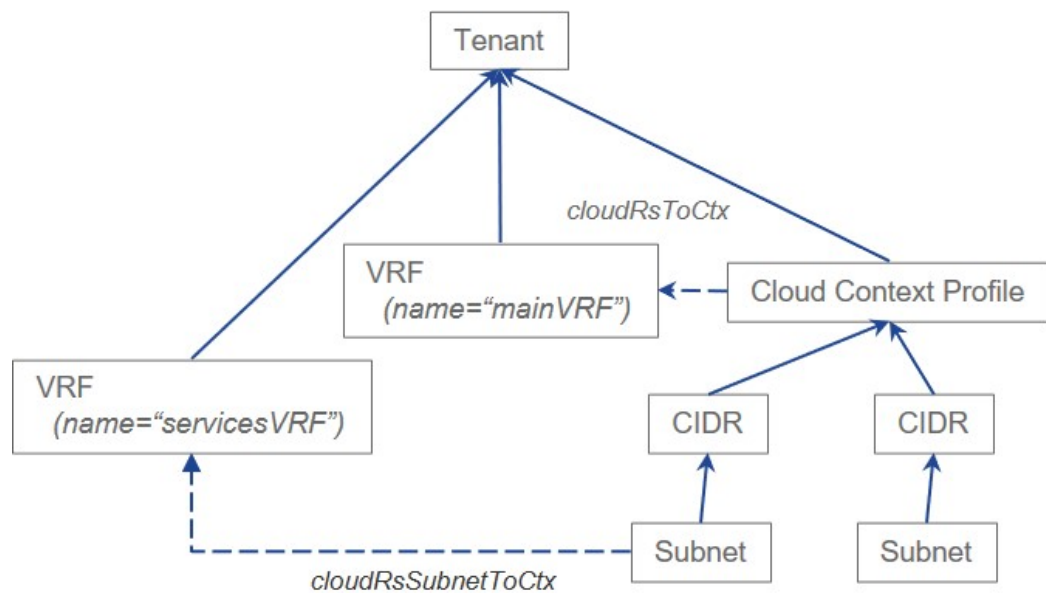
for a device cluster. A logical device can be selected based on a contract name, a graph name, or the function node name inside the graph.

Support for Multiple VRFs Under Single VNet

Support is now available for multiple VRFs under a single VNet, depending on the APIC release:

- Beginning with Release 5.0(2), you can have an infra (hub) VNet (a `cloudCtxProfile` in the infra tenant) that can be carved out into multiple VRFs. The cloud template subnets will be mapped to the overlay-1 VRF, but the user-created subnets will be implicitly mapped to the overlay-2 VRF in the same infra VNet. All subnets in the respective VRFs will have separate route tables in the cloud for VRF segregation.
- Beginning with Release 5.1(2), the ability to carve out multiple VRFs has been extended 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):

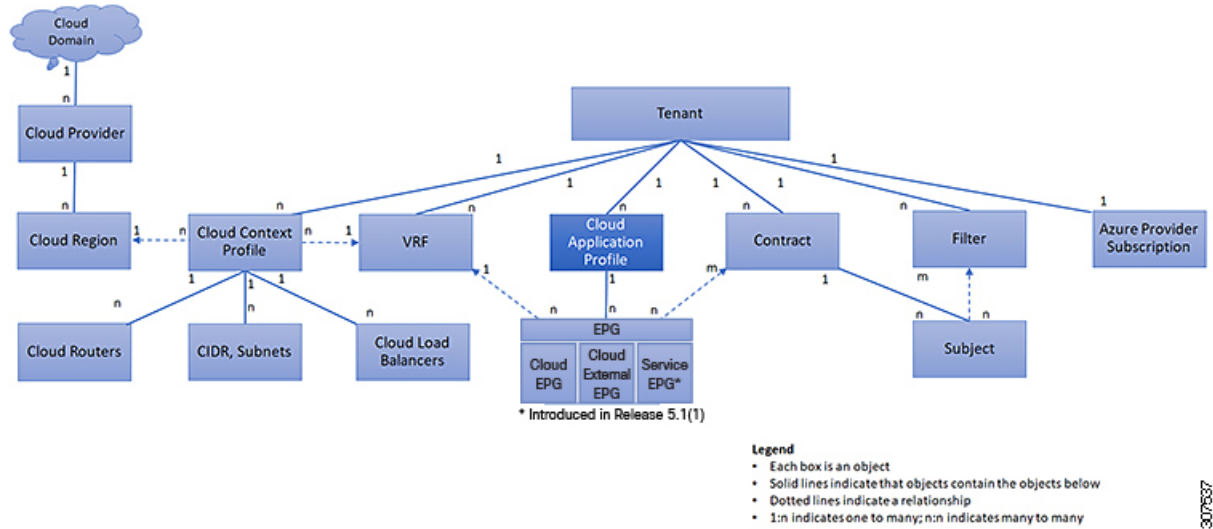
- The primary VRF with the name `mainVRF`
- A secondary VRF with the name `servicesVRF`

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.

Cloud Application Profiles

A cloud application profile (cloudAp) 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.

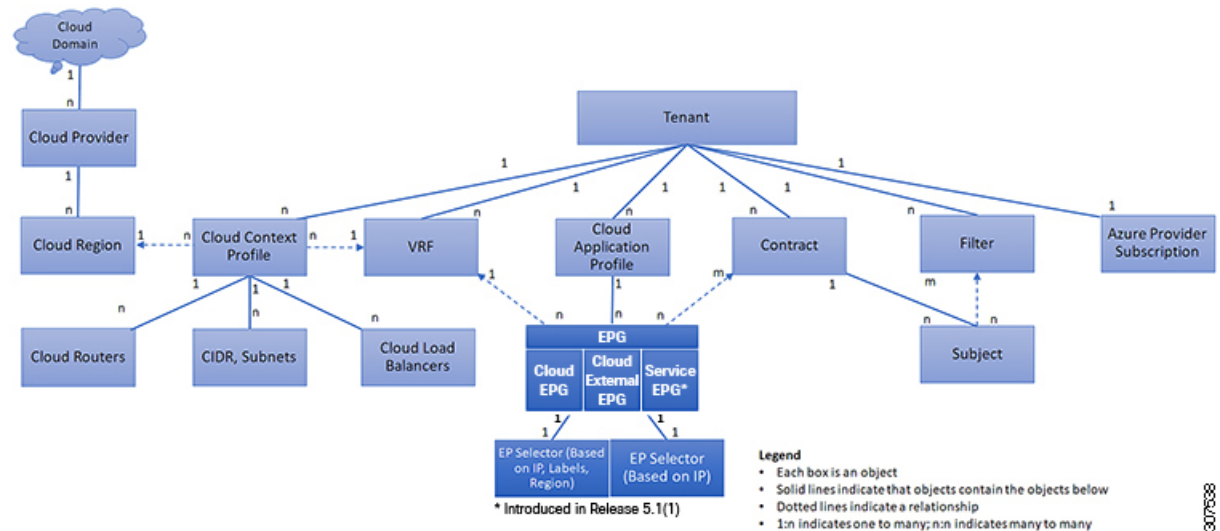
Cloud EPGs can be organized according to one of the following:

- 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 ACI 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 12 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 APIC 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 ACI. 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 APIC). For example, an Azure Kubernetes cluster (AKS) could be deployed in a subnet that is managed by the Cisco Cloud APIC.
 - **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

Service Type	Provider	Deployment Type/Access Type		
		Cloud Native	Cloud Native Managed	Third-Party
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
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 18](#) for more information.

About Service Types

Additional information specific to certain service types are provided below:

- [Azure Storage, on page 14](#)
- [Azure ApiManagement Services, on page 15](#)
- [Azure Databricks Services, on page 15](#)
- [Azure Active Directory Domain Services, on page 16](#)
- [Azure Kubernetes Services, on page 16](#)
- [Azure Redis Cache, on page 16](#)

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 APIC 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 APIC 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 APIC 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 APIC 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](#) for an example configuration of the AKS service EPG.

Azure Redis Cache

Azure Redis cache requires access to other services.

Cisco Cloud APIC 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 17](#)
- [Cloud Native Managed, on page 18](#)

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 APIC 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 APIC GUI](#). Configure a private link label to be used by the Azure SQL service EPG (for example, `SQL-PLL`).

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

Follow the procedures in [Creating a Service EPG Using the Cisco Cloud APIC GUI](#), 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 APIC). For example, an Azure ApiManagement Services could be deployed in a subnet that is managed by the Cisco Cloud APIC.

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 APIC 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 APIC GUI](#). 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 APIC GUI, create a service EPG of the service type Azure ApiManagement Services.

Follow the procedures in [Creating a Service EPG Using the Cisco Cloud APIC GUI](#), 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 Cloud APIC, 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 Cloud APIC, 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 19](#)
- [Release 5.1\(2\) and Later: NSG-Per-Subnet Configurations, on page 19](#)
- [Release 5.1\(2g\) and Later: IP-Based Rules for Inter-VRF Contracts in the Same VNet, on page 20](#)

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 APIC (these configurations are also referred to as **NSG-per-EPG** configurations throughout this document). These NSGs for the Cloud APIC 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 Cloud APIC 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 Cloud APIC (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 Cloud APIC 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 Cloud APIC-managed VNet

Looked at from another perspective:

- Every EPG in a Cloud APIC-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 Cloud APIC-managed VNet will have an NSG associated with it.

The default setting for a Greenfield or a fresh Cloud APIC 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 12](#) for more information.

See [Configuring Network Security Groups Using the Cloud APIC GUI](#) 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 20](#)
- [Guidelines and Limitations for Release 5.1\(2\) or Later, on page 20](#)

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 Cloud APIC.

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 Cloud APIC. However, you can have *either* the newer NSG-per-subnet configuration or the NSG-per-EPG configuration, but not both in the same Cloud APIC system.
- You can configure one NSG per subnet in a Cloud APIC-managed VNET. Having one NSG per a group of subnets is not supported for Cloud APIC 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 21](#)
- [NSG-Per-Subnet Security Rules, on page 21](#)

NSG-Per-EPG Security Rules

- Once the EPGs and the contract are defined on the Cloud APIC, 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 22](#)
- [NSG Behavior With Software Downgrades, on page 22](#)

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 18](#) for more information on the different NSG configurations, and [Configuring Network Security Groups Using the Cloud APIC GUI](#) 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 Cloud APIC system. See [Guidelines and Limitations for ASGs and NSGs, on page 20](#) for more information.

However, if you backed up your existing Cloud APIC configuration using the procedures in [Creating a Backup Configuration Using the Cisco Cloud APIC GUI](#), 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 Cloud APIC GUI](#). The Cloud APIC software begins the transition from NSG-per-subnet mapping to NSG-per-EPG mapping.
2. Wait until the transition is complete, where the Cloud APIC 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 Cloud APIC 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

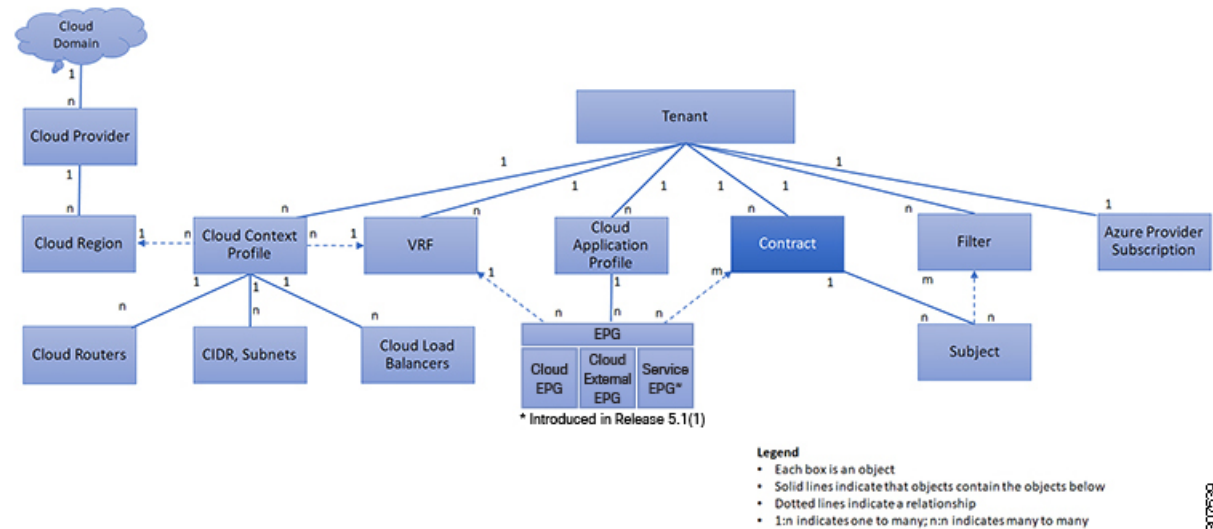
Verify that the property `sgForSubnetModeConverged` is set to `yes` before proceeding with the downgrade through the REST API.

- When you have confirmation that the system has successfully completed the transition back to the NSG-per-EPG mapping, you can downgrade the Cloud APIC software using the instructions provided in the *Cisco Cloud APIC 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):

```

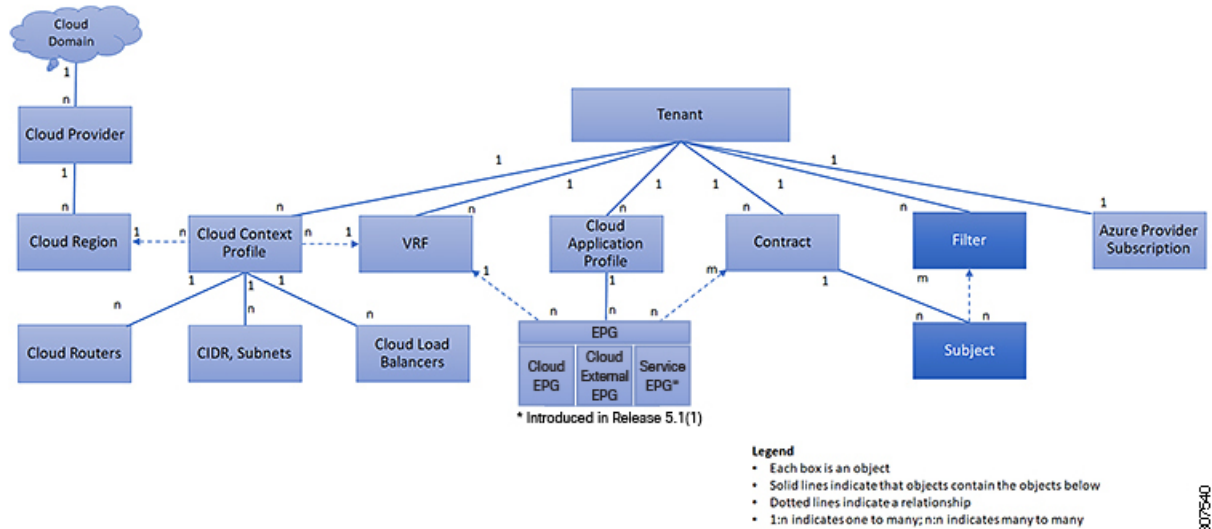
EPG1 -> EPG2   TCP  80,8080
EPG1 -> EPG2   UDP  53,5000
EPG1 -> EPG2   TCP   53
EPG1 -> 1.1.1.1/32, 2.2.2.2/32   TCP  80,8080
EPG1 -> 1.1.1.1/32, 2.2.2.2/32   UDP  53,5000
EPG1 -> 1.1.1.1/32, 2.2.2.2/32   TCP   53

```

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 APIC 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.
- ACI 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 APIC 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 APIC 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 APIC-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 APIC, 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`.

In releases prior to Release 5.0(2), you are not allowed to add more CIDRs to the infra VNet. Beginning with Release 5.0(2), multiple CIDR and subnet blocks can now 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 the user-created subnets will be implicitly mapped to the overlay-2 VRF in the same infra VNet. All subnets in the respective VRFs will have separate route tables in the cloud for VRF segregation.

In addition, beginning with Release 5.0(2), 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 overlay-2 VRF in the infra tenant. A cloud EPG in the overlay-2 VRF can communicate with other cloud EPGs and cloud external EPGs in the overlay-2 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 overlay-2 VRF.

For more information, see [Creating an Application EPG Using the Cisco Cloud APIC GUI](#) and [About the Overlay-1 and Overlay-2 VRFs](#).

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 1: 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 APIC 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.

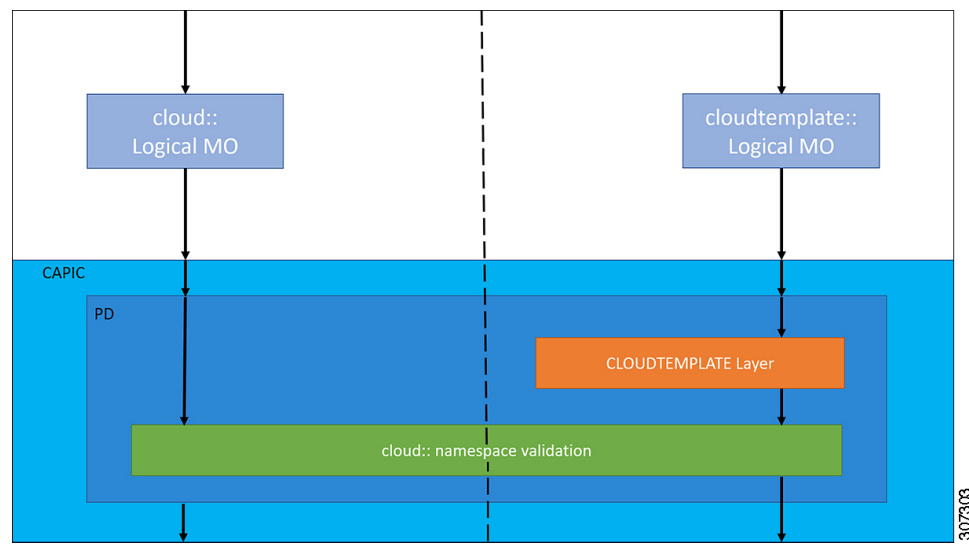
MO	Purpose
cloudtemplateBgpEvpn	Captures the peer IP address, ASN, and so forth, for setting up the BGP session with the on-premises site.

In Cisco Cloud APIC, 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 APIC, 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 APIC Components](#)

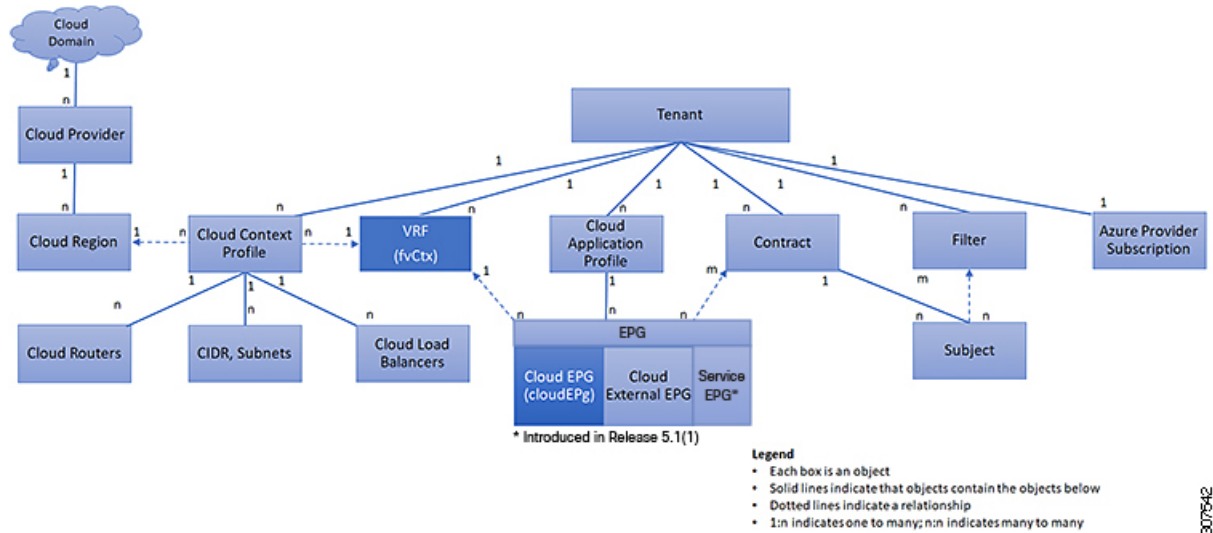
Managed Object Relations and Policy Resolution

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 ACI 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 ACI 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 ACI 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 ACI 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 CloudAPIC applies the default policy. An administrator can create a default policy and the Cisco Cloud APIC 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.