Cisco Cloud APIC Policy Model

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, 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.

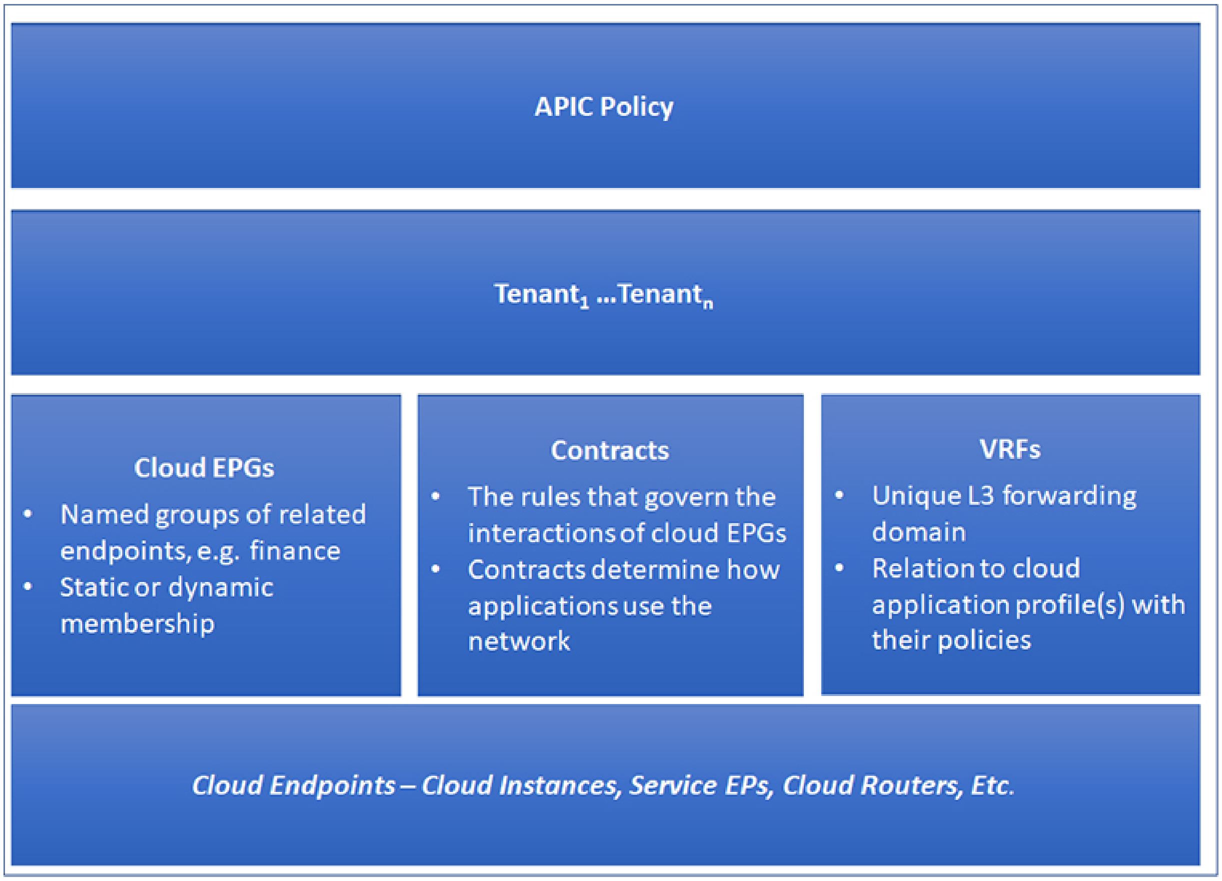
- The system prohibits communications with newly connected endpoints until the policy model is updated to include the new endpoint.
- Network administrators do not configure logical system resources directly. Instead, they define logical (hardware-independent) configurations and the Cisco Cloud APIC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



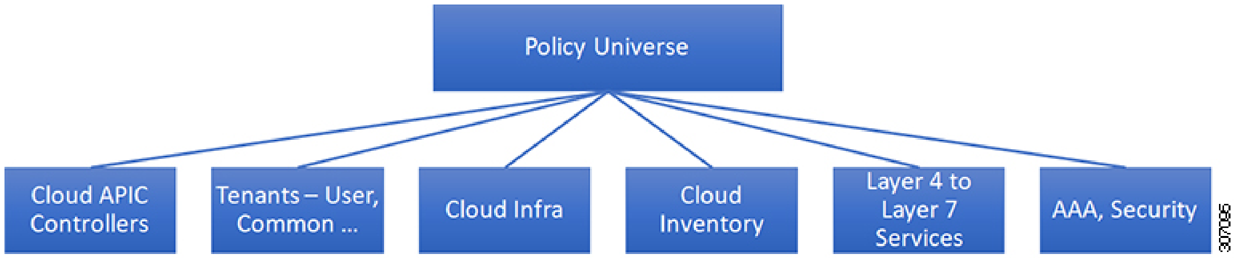
cloud infrastructure-wide or tenant administrators create predefined policies that contain application or shared resource requirements. These policies automate the provisioning of applications, network-attached services, security policies, and tenant subnets, which puts administrators in the position 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.

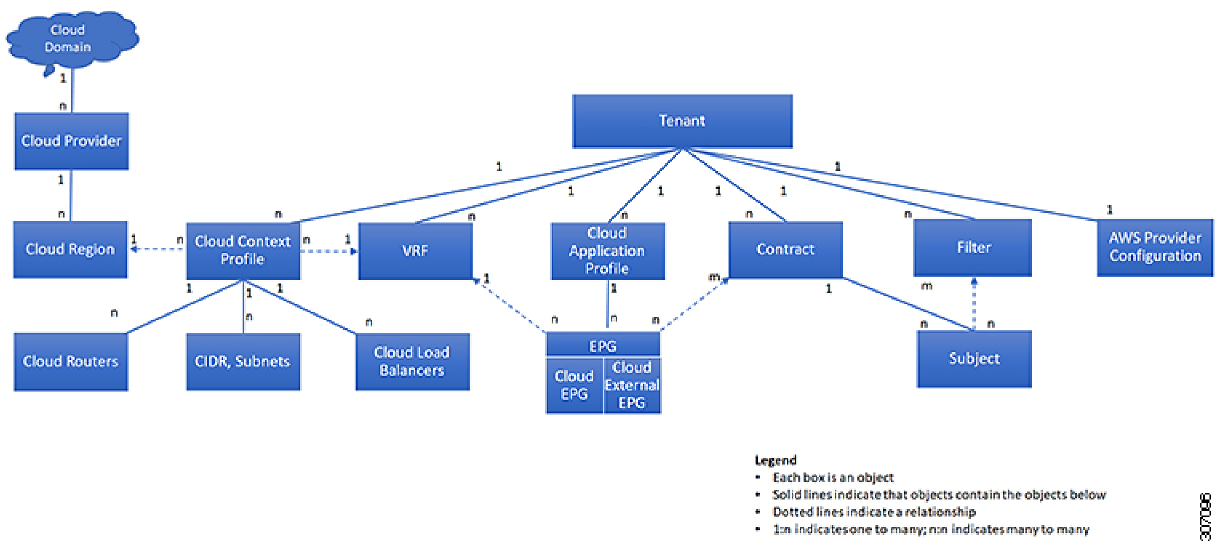
- The cloud infra policies enable you to manage on-premises and inter-region connectivity when setting up the Cisco Cloud APIC. For more information, see the *Cisco Cloud APIC Installation Guide*.
- Cloud inventory is a service that enables you to view different aspects of the system using the GUI. For 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, AWS 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 and a region represents the AWS VPC in that region.

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
- CSRs

The following sections provide additional information on some of the components that are part of the cloud context profile.

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.

Private IP Address Support for Cisco Cloud APIC and Cisco Cloud Services Router in AWS

Note For Azure, support for private IP addresses for Cisco Cloud APIC and CSRs became available in release 5.1(2). For AWS, this support is available beginning with release 5.2(1).

For AWS, prior to release 5.2(1), Cisco Cloud Router (CSR) interfaces were assigned both public and private IP address by Cisco Cloud APIC. Beginning with release 5.2(1), 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 address of GigabitEthernet1 of a CSR is used as BGP and OSPF router IDs.

To enable CSR private IP addresses for inter-site connectivity, where you are disabling public IP addresses for the CSR interfaces, see the Managing Regions (Configuring a Cloud Template) Using the Cisco Cloud APIC GUI procedure.

For AWS, prior to release 5.2(1), the management interface of the Cisco Cloud APIC was assigned a public IP address and a private IP address. Beginning with release 5.2(1), a private IP address is assigned to the management interface of the Cisco Cloud APIC and assigning a public IP address is optional. To disable public IP to the Cisco Cloud APIC so that a private IP address is used for connectivity, see the *Deploying the Cloud APIC in AWS* procedure in the *Cisco Cloud APIC for AWS Installation Guide*, Release 5.2(1) or later.

Restrictions for CSR with Private IP Address

When public IP is disabled, the underlay inter-site connectivity cannot be Public internet because Public Internet requires a public IP address. The underlay inter-site connectivity can only be one of the following:

- Private connection for connecting to an on-premise ACI site, which is through AWS Direct Connect or Azure Express Route

- Cloud backbone for connecting to a Cisco Cloud APIC site of the same cloud provider, which is through AWS VPC Peering or Azure Vnet Peering

Communicating to External Sites From Regions Without a CSR

Prior to release 5.2(1), for traffic to pass through to an external site, the region where the traffic is passing through must have a CSR deployed. The CSR advertises the CIDRs that are local to that region. If an EPG in a region has a contract with an external site, then that region must have a CSR deployed in order to communicate with that external site.

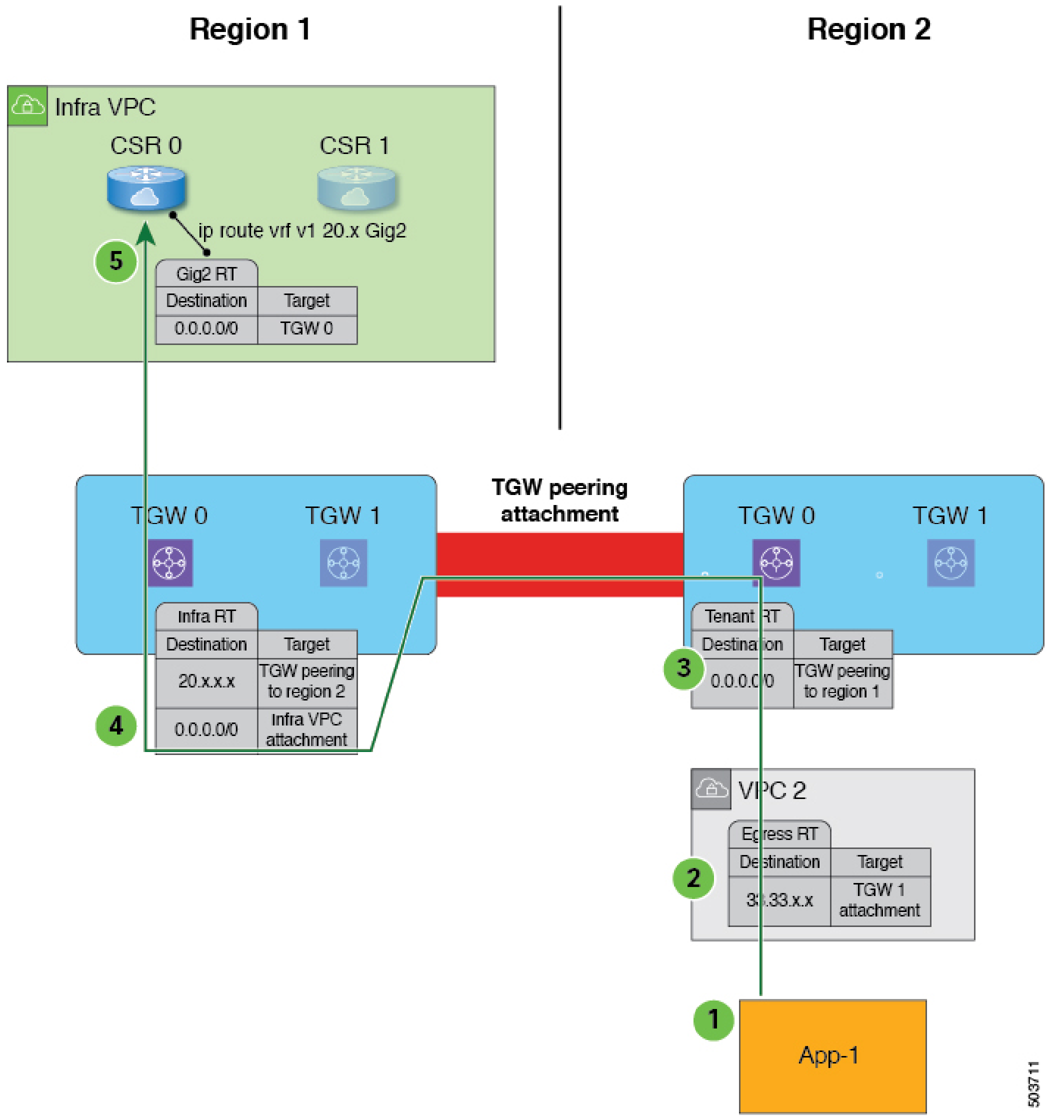
Beginning with release 5.2(1), you can now have communication with an external site from regions without a CSR. This is accomplished by making use of the AWS Transit Gateway feature, which became available for Cisco Cloud APIC in release 5.0(1). When you enable the AWS Transit Gateway feature on Cisco Cloud APIC, you also enable peering between all managed regions on AWS. In this way, the AWS Transit Gateway peering feature allows the Cisco Cloud APIC to address the issue of communicating with external sites from regions without a CSR. It addresses this issue by having traffic rerouted to a region with a CSR.

Using the AWS Transit Gateway feature, when traffic from a region without a CSR tries to egress out of a site, this traffic will be routed to the infra VPC for the closest region with a CSR. After the traffic is rerouted to that region's infra VPC, that CSR is used to egress out the packet. For ingress traffic, packets coming from an external site can reach any region's CSR and then be routed to the destination region using the AWS Transit Gateway peering in the ingress data path.

In these situations, traffic is rerouted automatically when the system recognizes that external traffic is egressing or ingressing through a region without a CSR. However, you must have the following components configured in order for the system to automatically perform this rerouting task:

- AWS Transit Gateway must be configured. If AWS Transit Gateway is not already configured, see Increasing Bandwidth Between VPCs by Using AWS Transit Gateway for those instructions.
- CSRs must be deployed in at least one region. Even though this enhancement allows you to communicate with an external site from a region that *does not* contain a CSR, in order to do this, you must have another separate region that *does* contain a CSR so that traffic can be rerouted from the region without a CSR to the region with a CSR.

The following figure shows an example scenario where traffic is rerouted automatically when the system recognizes that external traffic is egressing from a region without a CSR.



The following occurs when the Cisco Cloud APIC recognizes that Region 2 does not have a CSR, but traffic is egressing out to an external site (shown with the green arrow and circles):

1. Traffic flow begins egressing out from the CIDR in App-1 in Region 2 to a remote site. Note that the endpoint is configured with the appropriate outbound rule to allow the remote site CIDR.

2. The VPC 2 egress route table has the remote site CIDR, which then has the AWS Transit Gateway as the next hop. The AWS Transit Gateway information is programmed automatically based on the configured contracts.

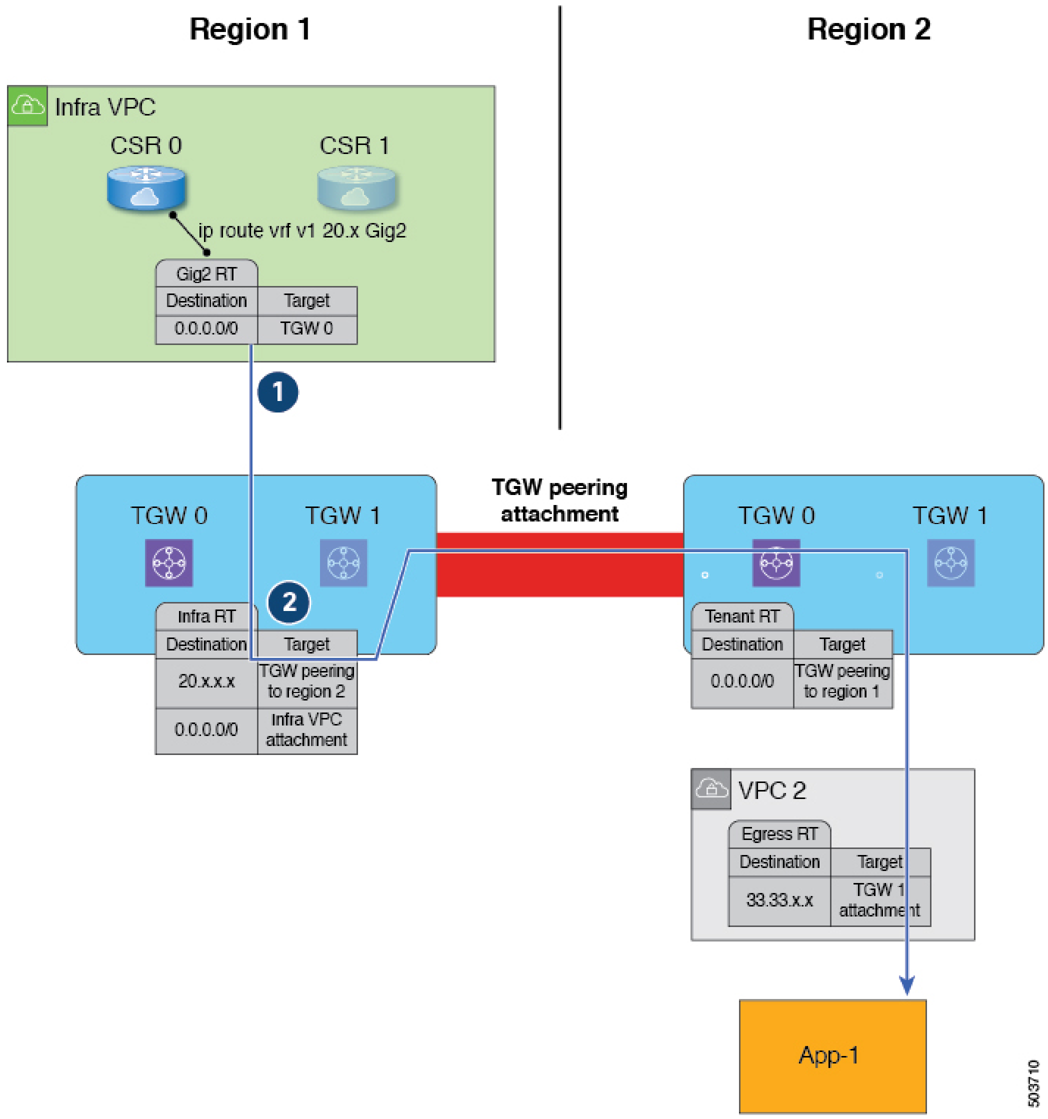
3. A 0.0.0.0/0 route is inserted in the AWS Transit Gateway route table, which essentially tells the system to take the AWS Transit Gateway peering attachment if traffic is egressing out to an external site but there is no CSR in this region. In this situation, the AWS Transit Gateway peering attachment goes to another region that does have a CSR (Region 1 in the example scenario). The region with a CSR that will be used is chosen based on geographical proximity to the region that does not have a CSR.

4. The packet is first forwarded to the infra VPC in the region with the CSR (Region 1), and is then forwarded to the gigabit ethernet network interface on the CSR that is associated with the appropriate VRF group.
5. The gigabit 2 inbound security group on the CSR in Region 1 is configured to allow traffic from the remote region VPC.

It's useful to note that in the egress example shown above:

- For steps 1 and 2, there is no change from pre-release 5.2(1) behavior.
- Step 3 is behavior that is new and unique to this feature in release 5.2(1), where the redirect occurs to the TGW peering attachment from the region without a CSR to the region with a CSR. In addition, step 3 occurs on the AWS cloud.
- Steps 4 and 5 would normally occur in Region 2 before release 5.2(1), but only if Region 2 had a CSR. However, because Region 2 does not have a CSR, with this feature in release 5.2(1), these steps are taking place in Region 1 where a CSR is present.

The following figure shows an example scenario where traffic is rerouted automatically when the system recognizes that external traffic is ingressing to a region without a CSR.



The following occurs when the Cisco Cloud APIC recognizes that Region 2 does not have a CSR, but traffic is ingressing in from an external site to a CIDR in App-1 in Region 2 (shown with the blue arrow and circles):

1. Normally, the CSR in Region 1 would only advertise the CIDRs that are local to that region. However, with this enhancement that is part of release 5.2(1), all CSRs in all regions now advertise CIDRs from all remote regions. Therefore, in this example, the CSR in Region 1 will also advertise the CIDRs that are in Region 2 (assuming AWS Transit Gateway peering is configured between both regions and the contracts are configured correctly). In this situation, the traffic ingresses in from an external site to the CSR in Region 1, where the CSR in Region 1 advertises the static route for the remote region VPC CIDRs.

2. The infra route table (the AWS Transit Gateway route table in Region 1) has the next hop to the AWS Transit Gateway peering attachment to Region 2.

It's useful to note that in the ingress example shown above:

- Both steps (steps 1 and 2) in the ingress example shown above are new and unique to this feature in release 5.2(1).

- Step 1 in the ingress example shows configurations programmed on the CSR.
- Step 2 in the ingress example occurs on the AWS cloud.

Support for ECMP Forwarding from Remote Sites for CSRs

CSRs in a cloud will typically receive more than one path for a prefix. Prior to release 5.2(1), there was no support for data forwarding from CSRs using Equal Cost Multiple Path (ECMP), even though the CSR receives multiple paths.

Beginning with release 5.2(1), support is now available for ECMP with CSRs, where traffic from CSRs will be forwarded to all ECMP paths received from a destination site. This support is automatically enabled with release 5.2(1) and requires no manual configuration to enable this support.

Preference For Routes to CSRs in Regions with Local CIDRs

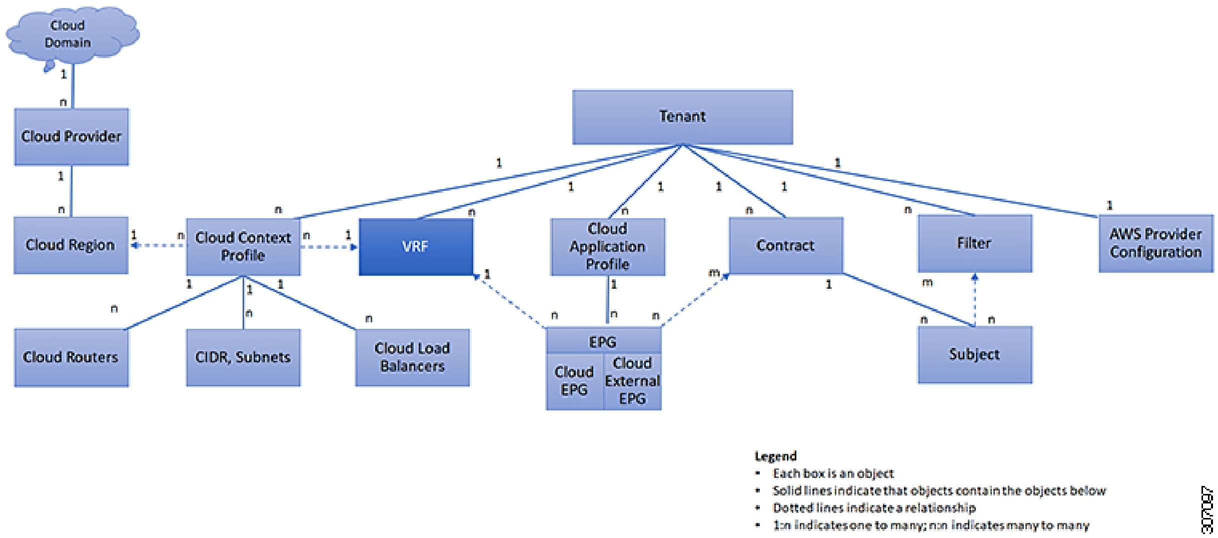
Every CIDR that is configured is local to a specific region. With multiple regions in a cloud, CSRs from all regions advertise the CIDRs for redundancy. Prior to release 5.2(1), CSRs from all regions advertised the CIDRs with the same preference. This can cause a remote cloud site or an on-prem site to install the path to a CIDR through a region where the CIDR is not local. This, in turn, could result in a packet taking a longer route than necessary.

Beginning with release 5.2(1), CSRs from multiple regions will continue to advertise the CIDRs, but CSRs from the region where the CIDR is local will advertise with a higher preference. This causes the on-premises site or the remote cloud site to direct traffic directly to the region where the CIDR is local. If the CSRs in the local region fail, the paths from the other regions can be used for data forwarding.

VRFs

A Virtual Routing and Forwarding (VRF) object (`fvCtx`) or context is a tenant network (called a private network 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

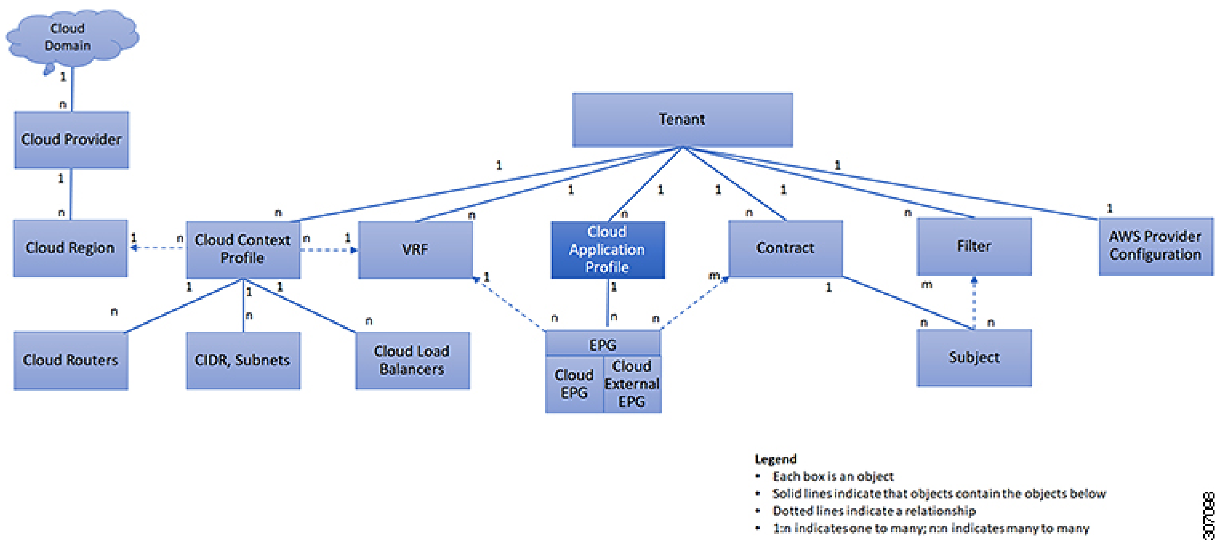


A VRF defines a Layer 3 address domain. One or more cloud context profiles are associated with a VRF. You can only associate one cloud context profile with a VRF in a given region. All the endpoints within the Layer 3 domain must have unique IP addresses because it is possible to forward packets directly 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.

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.

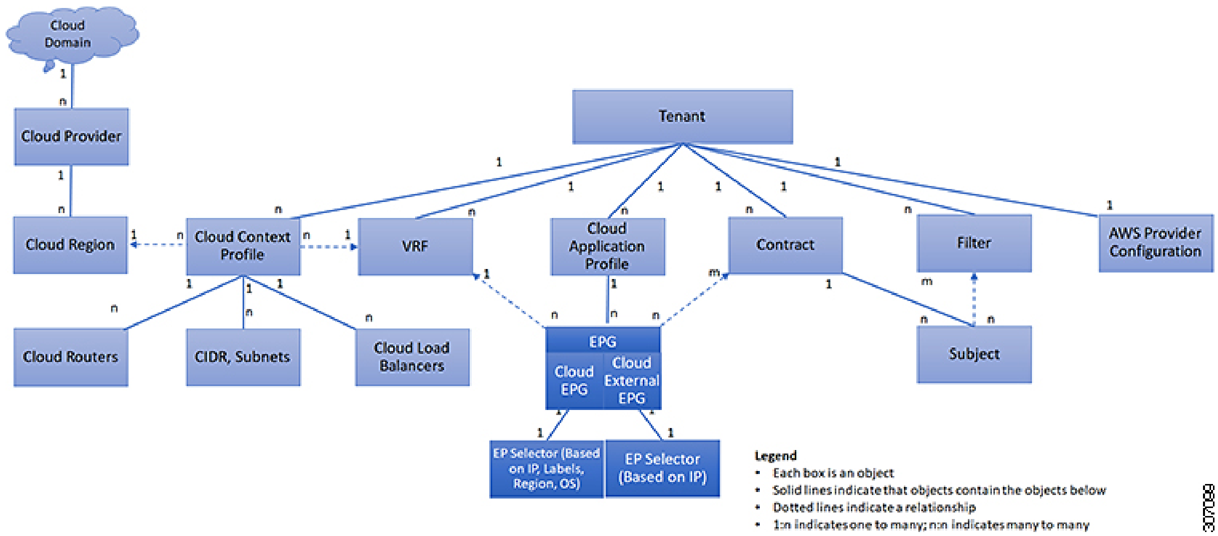
Cloud EPGs can be organized according to one of the following:

- 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 directly or indirectly. 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 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.

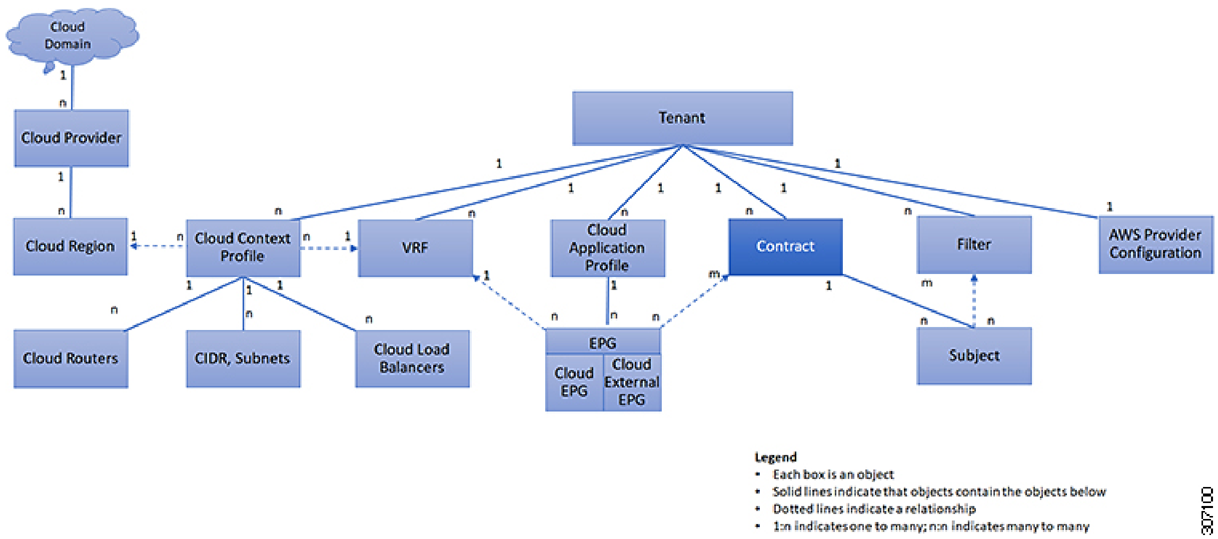
WAN router connectivity to the cloud infrastructure is an example of a configuration that uses a static cloud EPG. To configure WAN router connectivity to the cloud infrastructure, an administrator configures a `cloudExtEPg` cloud EPG that includes any endpoints within an associated WAN subnet. The cloud 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 (`cloudExtEPg`) 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 AWS VPC 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.

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`)

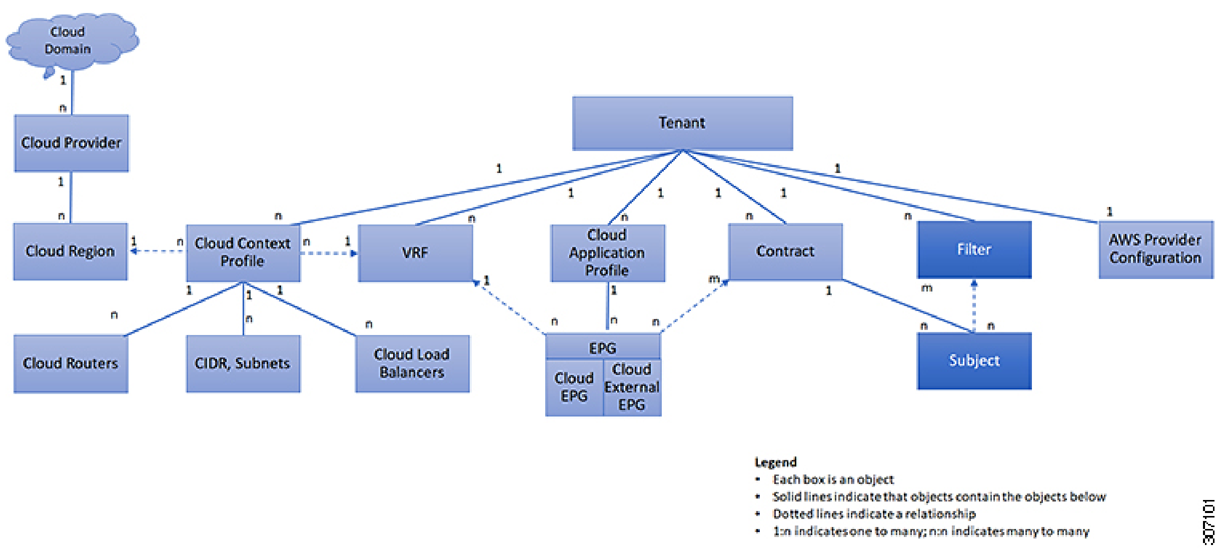
Contracts govern the communication between cloud EPGs that are labeled providers, consumers, or both. The relationship between a cloud EPG and a contract can be either a provider or consumer. When a cloud EPG provides a contract, communication with that cloud EPG can be initiated from other cloud EPGs 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.

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 AWS, source port provided in the filter entry s not taken into account.

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

- Filters are Layer 2 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.

Note When a contract filter match type is `All`, best practice is to use the VRF unenforced mode. Under certain circumstances, failure to follow these guidelines results in the contract not allowing traffic among cloud EPGs in the VRF.

- Subjects are contained in contracts. One or more subjects within a contract use 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.

Note For rules that are installed in AWS, the source port provided in the filter entry is not taken into account.

- ACI contracts rendered in AWS 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 AWS network configuration is the Virtual Private Cloud (VPC). AWS supports many regions worldwide and one VPC is specific to one region.

The cloud template accepts one or more region names and generates the entire configuration for the infra VPCs in those regions. They are the infra VPCs. The Cisco Cloud APIC-managed object (MO) corresponding to the AWS VPC 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. A `cloudCtxProfile` MO generated by the cloud template carries `ctxProfileOwner == 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 AWS VPC is the CIDR. Every region needs a unique CIDR. In Cisco Cloud APIC, you can provide the CIDRs for the infra VPCs. The CIDRs for the first two regions come from the Cloud Formation Template (CFT) that deploys the Cisco Cloud APIC AMI on the AWS. The `cloudApicSubnetPool` MO provides CIDRs for the additional regions directly to the Cisco Cloud APIC. In the Cisco Cloud APIC configuration, the `cloudCidr` MO, which is a child of `cloudCtxProfile`, models the CIDR.

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

- Subnets
- Association of subnets to AWS availability zones
- 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: • `routerUsername` • `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.

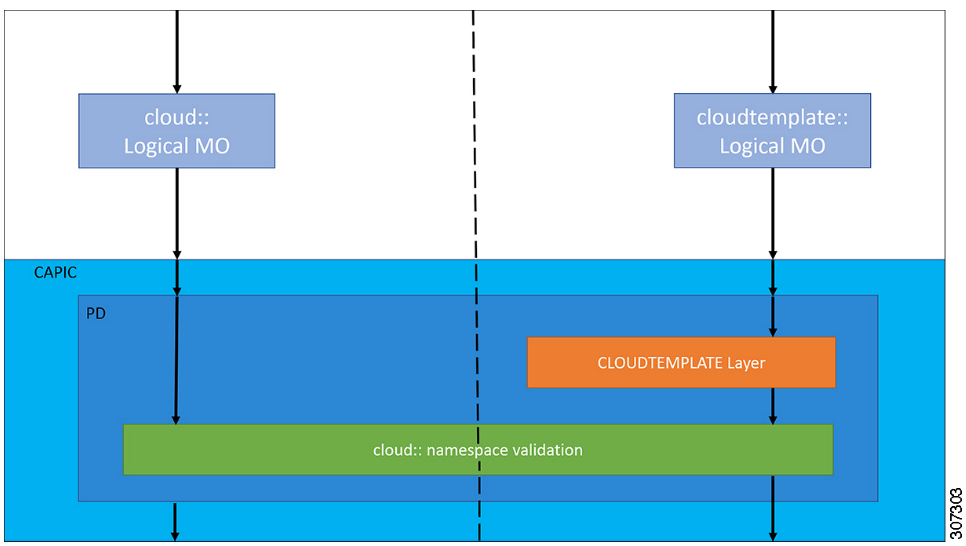
MO	Purpose
`cloudtemplateIpSecTunnel`	Captures the IP address of the IPSec peer in the ACI on-premises site.
`cloudtemplateOspf`	Captures the OSPF area to be used for the VPN connections.
`cloudtemplateBgpEvpn`	Captures the peer IP address, ASN, and so forth, for setting up the BGP session with the on-premises site.

In Cisco Cloud 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

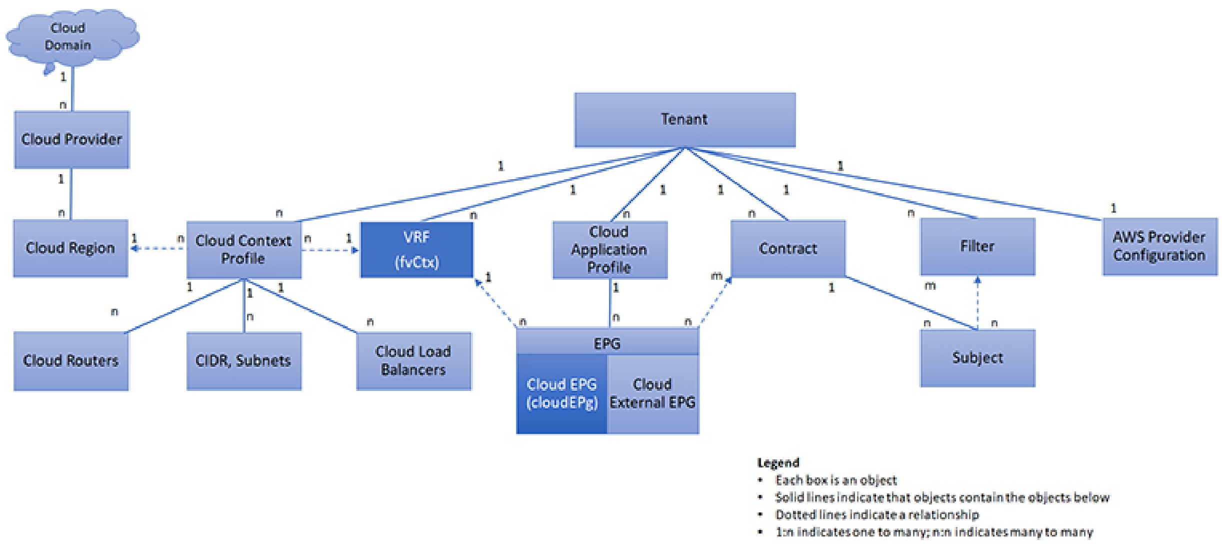
Managed Object Relations and Policy Resolution

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 `cloudRsZoneAttach` and `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 AWS 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.