About Cisco Cloud Network Controller

Overview

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

When deployed, the Cisco Cloud Network Controller:

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

Note

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

- Provides an automated connection between on-premises data centers and the public cloud with easy provisioning and monitoring

- Policies are pushed by Cisco Nexus Dashboard Orchestrator to the on-premises and cloud sites, and Cisco Cloud Network Controller translates the policies to the cloud to keep the policies consistent with the on-premises site

For more information about extending Cisco ACI to the public cloud, see the *Cisco Cloud Network Controller Installation Guide*.

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

External Network Connectivity

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

The following sections provide more information on the components that allow for external network connectivity.

External VRF

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

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

Connections to Non-ACI External Devices

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

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

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

Guidelines and Limitations

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

Understanding Supported Routing and Security Policies

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

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

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

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

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

Routing and Security Policies: Release 25.0(1)

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

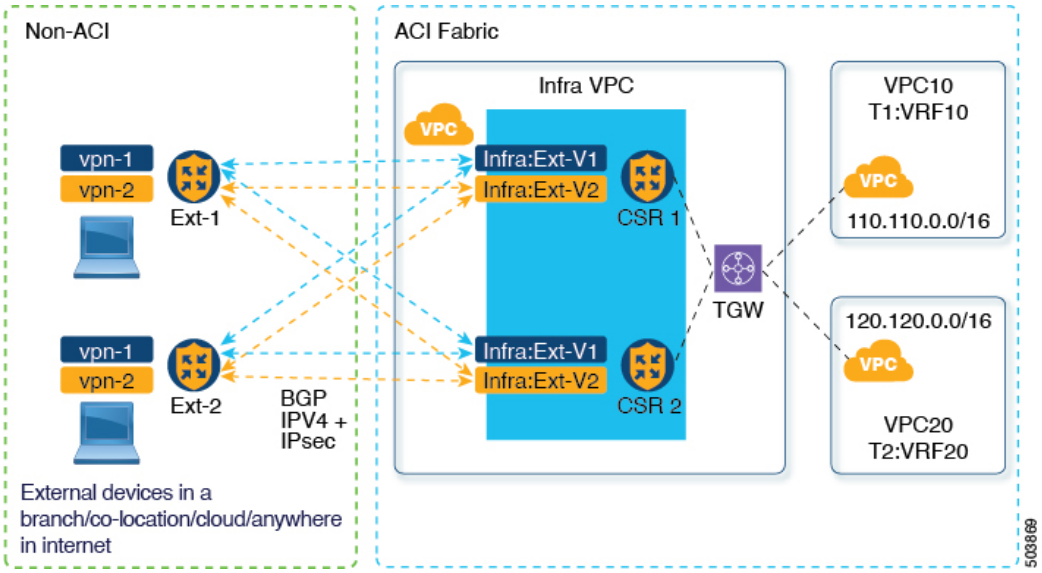
Note The routing and security policies described in this section are specifically for the 25.0(1) release and apply only between internal and external VRFs. For changes in the routing and security policies in the 25.0(2) release, see Routing Policies: Release 25.0(2), on page 5.

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

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

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

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



In this example, the infra:Ext-V1 is the external VRF on the CCRs in the infra VPC, with BGP IPv4 sessions over IPSec tunnels to the remote devices. The remote endpoint routes are received over these sessions in the infra:Ext-V1 VRF, which are then leaked into the internal VRFs displayed on the right side of the graphic (for example, the T1:VRF10 in VPC10). The reverse leaking routes are also configured.

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

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

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

Guidelines and Restrictions for Security and Routing Policies in Release 25.0(1)

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

- Routes are always leaked bi-directionally between an internal VRF and the external VRF.

  For example, assume there is a user tenant (t1) with an internal VRF (V1) and external VRF (Ext-V1). The route leak must be configured for both of these VRFs bi-directionally.

- You cannot configure "smaller" prefixes to be leaked while a "larger" prefix is already being leaked. For example, configuring the 10.10.10.0/24 prefix will be rejected if you already have the 10.10.0.0/16 prefix configured to be leaked. Similarly, if you configure the 0.0.0.0/0 (leak all) prefix, no other prefix will be allowed to be configured.
- Contracts are not allowed between cloud external EPGs (cloudExtEpgs).
- An external VRF cannot be used for creating cloud EPGs.
- An external VRF always belongs to the infra tenant.
- Leak routing is not supported between external VRFs.

Routing Policies: Release 25.0(2)

Note The routing and security policies described in this section are specifically for the 25.0(2) release. For changes in the routing and security policies in the previous release, see Routing and Security Policies: Release 25.0(1), on page 3.

For release 25.0(2), the routing and security policies continue to be split as described in Routing and Security Policies: Release 25.0(1), on page 3, but with these additional changes specifically for the routing policies:

- Route Leaking Between Internal VRFs, on page 5
- Global Inter-VRF Route Leak Policy, on page 6
- Guidelines and Limitations, on page 7

Route Leaking Between Internal VRFs

In the previous 25.0(1) release, the inter-VRF route map-based routing feature was introduced, where you can configure an independent routing policy to specify which routes to leak between a pair of internal and external VRFs. This route map-based routing feature applied specifically between internal and external VRFs; when configuring routing between a pair of internal VRFs, you could only use contract-based routing in that situation, as described in Routing and Security Policies: Release 25.0(1), on page 3.

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

- Leak all CIDRS or specific subnet IP addresses associated with the VRF by using:
  - **Leak All** option through the GUI
  - `leakInternalPrefix` field through the REST API

- Leak between a pair of VRFs by using:
  - **Subnet IP** option through the GUI
  - `leakInternalSubnet` field through the REST API

Global Inter-VRF Route Leak Policy

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

This policy has the following characteristics:

- This policy is associated with every internal VRF.
- This is a Cisco Cloud Network Controller-created policy.
- Contracts-based routing is disabled by default (turned off) for greenfield cases (when you are configuring a Cisco Cloud Network Controller for the first time). For upgrades, where you have a Cisco Cloud Network Controller that was already configured prior to release 25.0(2), contract-based routing is enabled (turned on).

The internal VRF route leak policy is a global policy that is configured in the First Time Setup screen under the infra tenant, where a Boolean flag is used to indicate whether contracts can drive routes in the absence of route maps:

- **Off**: Default setting. Routes are not leaked based on contracts, and are leaked based on route maps instead.
- **On**: Routes are leaked based on contracts in the absence of route maps. When enabled, contracts drive routing when route maps are not configured. When route maps exist, route maps always drives routing.

You can toggle this Boolen flag back and forth. Following are the general recommended steps for toggling this global VRF route leak policy, with more detailed instructions provided in Configuring Leak Routes for Internal VRFs Using the Cisco Cloud Network Controller GUI.

- You should enable contract-based routing in Cisco Cloud Network Controller for multi-cloud and hybrid-cloud deployments with EVPN.
- For multi-cloud and hybrid-cloud deployments without EVPN, routing is driven through route maps only and not through contracts.
- If you want to disable contract-based routing by toggling from contract-based routing to route map-based routing (toggling to the **Off** setting), this action can be disruptive if route map-based routing is not configured before you've toggled this setting to **Off**.

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

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

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

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

Guidelines and Limitations

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

- Routing between external and internal VRFs continues to use route map-based routing only.
- The `leakExternalPrefix` should not overlap with the route to the internet gateway (the external endpoint selector configured for external EPG to perform SSH), otherwise SSH will be broken.

Source Interface Selection for Tunnels

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

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

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

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

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

Guidelines and Limitations

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

General Guidelines and Limitations for Cisco Cloud Network Controller

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

- Inter-site (VRF-to-VRF) traffic is not supported if one of the VRFs is present as an attachment in a different VRF group (hub network). For example, consider the following scenario:

- VRF-1 is stretched across different sites (Azure and AWS). In the AWS site, VRF-1 is in VRF group 1.
- VRF-2 is present in a different VRF group (VRF group 2).

In this scenario, traffic from VRF-2 to VRF-1 across sites is not supported, since the contracts between the VRFs will be implicitly allowing traffic between different VRF groups as well. Traffic across different VRF groups (hub networks) is not supported.

- You cannot stretch more than one VRF between on-prem and the cloud while using inter-VRF route leaking in the CCRs (cloud routers). For example, in a situation where VRF1 with EPG1 is stretched and VRF2 with EPG2 is also stretched, EPG1 cannot have a contract with EPG2. However, you can have multiple VRFs in the cloud, sharing one or more contracts with one on-premises VRF.
- Set the BD subnet for on-premises sites as advertised externally to advertise to the CSR1kv on the cloud.
- The default AWS security group (SG) rules limit only permits 2 CCRs per region and only 2 regions can deploy CCRs (a total maximum of 4 CCRs). To deploy more CCRs, increase the AWS SG rule limit to 120 or more. We recommend increasing the rule limit to 500.
- When configuring an object for a tenant, first check for any stale cloud resources in AWS. A stale configuration might be present if it was not cleaned properly from the previous Cisco Cloud Network Controller instances that managed the account.

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

To check for and clean up stale cloud resources:

1. Click the **Navigation menu** > **Application Management** > **Tenants**. The **Tenants** summary table appears in the work pane with a list of tenants as rows in a summary table.
2. Double click the tenant you are creating objects for. The **Overview**, **Cloud Resources**, **Application Management**, **Statistics**, and **Event Analytics** tabs appear.
3. Click the **Cloud Resources** > **Actions** > **View Stale Cloud Objects**. The **Stale Cloud Objects** dialog box appears.
4. If you see any stale objects, click to place a check mark in the **Automatically Clean Up Stale Cloud Objects** check box.
5. Click **Save**. The Cisco Cloud Network Controller automatically cleans up stale cloud objects.

Note To disable the automatic cleanup, follow steps 1 - 4 and click the **Automatically Clean Up Stale Cloud Objects** check box to remove the check mark.

- Cisco Cloud Network Controller tries to manage the AWS resources that it created. It does not attempt to manage resources created by other applications, other than listing existing resources as inventory. At the same time, it is also expected that AWS IAM users in the AWS infra tenant account, and the other tenant accounts, do not disturb the resources that Cisco Cloud Network Controller creates. For this

purpose, all resources Cisco Cloud Network Controller creates on AWS has at least one of these two tags:

- AciDnTag
- AciOwnerTag

Cisco Cloud Network Controller must prevent AWS IAM users who have access to create, delete, or update EC2, or any other resources, from accessing or modifying the resources that Cisco Cloud Network Controller created and manages. Such restrictions should apply on both the infra tenant and other user tenant accounts. AWS account administrators should utilize the above two tags to prevent their unintentional access and modifications. For example, you can have an access policy like the following to prevent access to resources managed by Cisco Cloud Network Controller:

```
{
  "Effect": "Deny",
  "Action": [
    "ec2:*"
  ],
  "Resource": "*",
  "Condition": {
    "StringLike": {"ec2:ResourceTag/AciDnTag": "*"}
  }
}
```

- When configuring shared L3Out:
  - An on-premises L3Out and cloud EPGs cannot be in tenant common.
  - If an on-premises L3Out and a cloud EPG are in different tenants, define a contract in tenant common. The contract cannot be in the on-premises site or the cloud tenant.
  - Specify the CIDR for the cloud EPG in the on-premises L3Out external EPGs (l3extInstP).
  - When an on-premises L3Out has a contract with a cloud EPG in a different VRF, the VRF in which the cloud EPG resides cannot be stretched to the on-premises site and cannot have a contract with any other VRF in the on-premises site.
  - When configuring an external subnet in an on-premises external EPG:
    - Specify the external subnet as a non-zero subnet.
    - The external subnet cannot overlap with another external subnet.
    - Mark the external subnet with a shared route-control flag to have a contract with a cloud EPG.
  - The external subnet that is marked in the on-premises external EPG should have been learned through the routing protocol in the L3Out or created as a static route.
- When mapping availability zones, choose only a or b in Cisco Cloud Network Controller. Internally, the zone-mapping function maps this to actual availability zones in AWS.

Note The mapping works in alphabetical order. The availability zones are sorted alphabetically and then the function picks the first two and associates them to a zone a and b in Cisco Cloud Network Controller.

- Configuring ASN 64512 for cloud routers causes BGP sessions to not work between cloud routers and AWS virtual private gateways.
- For the total supported scale, see the following *Scale Supported* table:

Note With the scale that is specified in the *Scale Supported* table:

- You can have only 4 total managed regions.
- You can have only CCRs in 2 regions, 2 * 2 CCRs. This is irrespective of AWS SG rule limit.

Table 1: Scale Supported

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

About the Cisco Cloud Network Controller GUI

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

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

The **Intent** menu bar icon enables you to create a component from anywhere in the GUI. For example, to create a tenant while viewing the **Availability Zones** window, click the **Intent** icon. A dialog appears with

a search box and a drop-down list. When you click the drop-down list and choose **Application Management**, a list of options, including the **Tenant** option, appears. When you click the **Tenant** option, the **Create Tenant** dialog appears displaying a group of fields that are required for creating the tenant.

For more information about configuring Cisco Cloud Network Controller components, see Configuring Cisco Cloud Network Controller Components