Defining a Logical Device

- About Device Clusters, on page 1
- About Concrete Devices, on page 2
- About Trunking, on page 2
- About Layer 4 to Layer 7 Services Endpoint Groups, on page 3
- Using Static Encapsulation for a Graph Connector, on page 3
- Configuring a Layer 4 to Layer 7 Services Device Using the GUI, on page 3
- Creating a Layer 4 to Layer 7 Device Using the NX-OS-Style CLI, on page 6
- Creating a High Availablity Cluster Using the NX-OS-Style CLI, on page 10
- Creating a Virtual Device Using the NX-OS-Style CLI, on page 11
- Example XML for Creating a Logical Device, on page 12
- Modifying a Device Using the GUI, on page 14
- Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the GUI, on page 15
- Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the REST APIs, on page 15
- Using an Imported Device with the REST APIs, on page 16
- Importing a Device From Another Tenant Using the NX-OS-Style CLI, on page 16
- Verifying the Import of a Device Using the GUI, on page 17

About Device Clusters

A device cluster (also known as a logical device) is one or more concrete devices that act as a single device. A device cluster has cluster (logical) interfaces, which describe the interface information for the device cluster. During service graph template rendering, function node connectors are associated with cluster (logical) interfaces. The Application Policy Infrastructure Controller (APIC) allocates the network resources (VLAN or Virtual Extensible Local Area Network [VXLAN]) for a function node connector during service graph template instantiation and rendering and programs the network resources onto the cluster (logical) interfaces.

The Cisco APIC allocates only the network resources for the service graph and programs only on the fabric side during graph instantiation. This behavior is useful if your environment already has an existing orchestrator or a dev-op tool that programs the devices in a device cluster.

The Cisco APIC needs to know the topology information (logical interface and concrete interface) for the device cluster and devices. This information enables the Cisco APIC to program the appropriate ports on the leaf switch, and the Cisco APIC can also use this information for troubleshooting wizard purposes. The Cisco APIC also needs to know the relation to `DomP`, which is used for allocating the encapsulation.

A device cluster or logical device can be either physical or virtual. A device cluster is considered virtual when the virtual machines that are part of that cluster reside on an hypervisor that is integrated with Cisco APIC using VMM domains. If these virtual machines are not part of a VMM domain, then they are treated as physical devices even though they are virtual machine instances.

Note You can use only a VMware VMM domain or SCVMM VMM domain for a logical device

The following settings are required:

- Connectivity information for the logical device (`vnsLDevViP`) and devices (`CDev`)
- Information about supported function type (go-through, go-to, L1, L2)

The service graph template uses a specific device that is based on a device selection policy (called a *logical device context*) that an administrator defines.

An administrator can set up a maximum of two concrete devices in active-standby mode.

To set up a device cluster, you must perform the following tasks:

1. Connect the concrete devices to the fabric.
2. Configure the device cluster with the Cisco APIC.

Note The Cisco APIC does not validate a duplicate IP address that is assigned to two device clusters. The Cisco APIC can provision the wrong device cluster when two device clusters have the same management IP address. If you have duplicate IP addresses for your device clusters, delete the IP address configuration on one of the devices and ensure there are no duplicate IP addresses that are provisioned for the management IP address configuration.

About Concrete Devices

A concrete device can be either physical or virtual. If the device is virtual, you must choose the controller (vCenter or SCVMM controller) and the virtual machine name. A concrete device has concrete interfaces. When a concrete device is added to a logical device, the concrete interfaces are mapped to the logical interfaces. During service graph template instantiation, VLANs and VXLANs are programmed on concrete interfaces that are based on their association with logical interfaces.

About Trunking

You can enable trunking for a Layer 4 to Layer 7 virtual ASA device, which uses trunk port groups to aggregate the traffic of endpoint groups. Without trunking, a virtual service device can have only 1 VLAN per interface and up to 10 service graphs. With trunking enabled, the virtual service device can have an unlimited number of service graphs.

For more information about trunk port groups, see the *Cisco ACI Virtualization Guide*.

About Layer 4 to Layer 7 Services Endpoint Groups

The Application Policy Infrastructure Controller (APIC) enables you to specify an endpoint group to be used for the graph connector during graph instantiation. This enables you to better troubleshoot the graph deployment. The APIC uses the Layer 4 to Layer 7 services endpoint group that you specified to download encapsulation information to a leaf switch. The APIC also uses the endpoint group to create port groups on the distributed virtual switch for virtual devices. A Layer 4 to Layer 7 services endpoint group is also used to aggregate faults and statistics information for a graph connector.

In addition to the enhanced visibility into the deployed graph resources, a Layer 4 to Layer 7 services endpoint group can also be used to specify static encapsulation to be used for a specific graph instance. This encapsulation can also be shared across multiple graph instances by sharing the Layer 4 to Layer 7 services endpoint group across multiple graph instances.

For example XML code that shows how you can use a Layer 4 to Layer 7 services endpoint group with a graph connector, see Example XML of Associating a Layer 4 to Layer 7 Service Endpoint Group with a Connector, on page 13.

Using Static Encapsulation for a Graph Connector

The Application Policy Infrastructure Controller (APIC) allocates the encapsulation for various service graphs during processing. In some use cases, you want to be able explicitly to specify the encapsulation to be used for a specific connector in the service graph. This is known as static encapsulation. Static encapsulations are only supported for the service graph connector that has a service device cluster with physical services. Service device clusters with virtual service devices use the VLANs from the VMware or SCVMM domain that is associated with the service device cluster.

A static encapsulation can be used with a graph connector by specifying the encapsulation value as part of the Layer 4 to Layer 7 service endpoint group. For example XML code that shows how you can use a static encapsulation with a Layer 4 to Layer 7 services endpoint group, see Example XML of Using Static Encapsulation with a Layer 4 to Layer 7 Service Endpoint Group, on page 14.

Configuring a Layer 4 to Layer 7 Services Device Using the GUI

When you create a Layer 4 to Layer 7 services device, you can connect to either a physical device or a virtual machine. The fields are slightly different depending on the type to which you are connecting. When you connect to a physical device, you specify the physical interface. When you connect to a virtual machine, you specify the VMM domain, the virtual machine, and the virtual interfaces. Additionally, you can select an unknown model, which allows you to configure the connections manually.

Before you begin

- You must have configured a tenant.

Procedure

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices**.

Step 4 In the Work pane, choose **Actions** > **Create L4-L7 Devices**.

Step 5 In the **Create L4-L7 Devices** dialog box, in the **General** section, complete the following fields:

Name	Description
Name field	Enter a name for the device.
Service Type drop-down list	Choose the service type. The types are: • ADC • Firewall • Other **Note** For a Layer 1/Layer 2 firewall configuration, choose **Other**.
Device Type buttons	Choose the device type.
Physical Domain or **VMM Domain** drop-down list	Choose the physical domain or VMM domain.
Switching Mode (Cisco ACI Virtual Edge only)	For a Cisco ACI Virtual Edge virtual domain, choose one of the following modes: • **AVE**: Traffic is switched through the Cisco ACI Virtual Edge. • **native**: Traffic is switched through the VMware DVS.
View radio buttons	Choose the view for the device. The view can be: • **Single Node**: Only one node • **HA Node**: High availability nodes (two nodes) • **Cluster**: 3 or more nodes

Name	Description
Context Aware	The context awareness of the device. The awareness can be: • **Single**: The device cluster cannot be shared across multiple tenants of a given type that are hosted on the provider network. You must give the device cluster to a specific tenant for a given user. • **Multiple**: The device cluster can be shared across multiple tenants of a given type that you are hosting on the provider network. For example, there could be two hosting companies that share the same device. The default is **Single**. **Note** When you create a Layer 4 to Layer 7 services device that is a load balancer, the **Context Aware** parameter is not used and can be ignored. Beginning with the 5.2(1) release, this parameter is deprecated and the Cisco APIC ignores the value.
Function Type	Function types are: • **GoThrough**: Transparent mode • **GoTo**: Routed mode • **L1**: Layer 1 firewall mode • **L2**: Layer 2 firewall mode The default is **GoTo**. **Note** For Layer 1 or Layer 2 mode, check the check box to enable Active-Active mode. When enabled, active-active deployment/ECMP paths for Layer 1/Layer 2 PBR devices are supported.

Step 6 In the **Device 1** section, complete the following fields:

Name	Description
VM drop-down list	(Only for the virtual device type) Choose a virtual machine.

Step 7 In the **Device Interfaces** table, click the **+** button to add an interface and complete the following fields:

Name	Description
Name drop-down list	Choose the interface name.
VNIC drop-down list	(Only for the virtual device type) Choose a vNIC.
Path drop-down list	(Only for the physical device type or for an interface in L3Out) Choose a port, port channel, or virtual port channel to which the interface will connect.

Step 8 Click **Update**.

Step 9 (Only for an HA cluster) Complete the fields for each device.

Step 10 Complete the fields for the **Cluster Interfaces** section.

Click (+) to add a cluster interface and complete the following details:

Name	Description
Name drop-down list	Enter a name for the cluster interface.
Concrete Interfaces drop-down list	Select a concrete interface. The interfaces in the drop-down list are based on the device interfaces created in step 7.
Enhanced Lag Policy drop-down list	(Optional) Choose the Lag policy configured for the VMM domain of the device. This option is available, only if you have selected the **Device Type** (discussed in Step 5) as **Virtual**.

For an HA cluster, make sure that the cluster interfaces are mapped to the corresponding interfaces on both concrete devices in the cluster.

Step 11 Click **Finish**.

Creating a Layer 4 to Layer 7 Device Using the NX-OS-Style CLI

When you create a Layer 4 to Layer 7 device, you can connect to either a physical device or a virtual machine. When you connecting to a physical device, you specify the physical interface. When you connect to a virtual machine, you specify the VMM domain, the virtual machine, and the virtual interfaces.

Note When you configure a Layer 4 to Layer 7 device that is a load balancer, the context aware parameter is not used. The context aware parameter has a default value of `single context`, which can be ignored.

Before you begin

- You must have configured a tenant.

Procedure

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

```
apic1(config)# tenant t1
```

Step 3 Add a Layer 4 to Layer 7 device cluster.

```
l4l7 cluster name cluster_name type cluster_type vlan-domain domain_name
  [function function_type] [service service_type]
```

Parameter	Description
name	The name of the device cluster.
type	The type of the device cluster. Possible values are: • virtual • physical
vlan-domain	The domain to use for allocating the VLANs. The domain must be a VMM domain for virtual device and physical domain for physical device.
switching-mode (Cisco ACI Virtual Edge only)	(Optional) Choose one of the following modes: • **AVE**—Switches traffic through the Cisco ACI Virtual Edge. • **native**—Switches traffic through the VMware DVS. This is the default value.
function	(Optional) The function type. Possible values are: • go-to • go-through • L1 • L2
service	(Optional) The service type. This is used by the GUI to show the ADC- or firewall-specific icons and GUI. Possible values are: • ADC • FW • OTHERS

Example:

For a physical device, enter:

```
apic1(config-tenant)# l4l7 cluster name D1 type physical vlan-domain phys
  function go-through service ADC
```

For a virtual device, enter:

```
apic1(config-tenant)# l4l7 cluster name ADCCluster1 type virtual vlan-domain mininet
```

Step 4 Add one or more cluster devices in the device cluster.

```
cluster-device device_name [vcenter vcenter_name] [vm vm_name]
```

Parameter	Description
vcenter	(Only for a virtual device) The name of VCenter that hosts the virtual machine for the virtual device.
vm	(Only for a virtual device) The name of the virtual machine for the virtual device.

Example:

For a physical device, enter:

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

For a virtual device, enter:

```
apic1(config-cluster)# cluster-device C1 vcenter vcenter1 vm VM1
apic1(config-cluster)# cluster-device C2 vcenter vcenter1 vm VM2
```

Step 5 Add one or more cluster interfaces in the device cluster.

```
cluster-interface interface_name [vlan static_encap]
```

Parameter	Description
vlan	(Only for a physical device) The static encapsulation for the cluster interface. VLAN value must be between 1 to 4094.

Example:

For a physical device, enter:

```
apic1(config-cluster)# cluster-interface consumer vlan 1001
```

For a virtual device, enter:

```
apic1(config-cluster)# cluster-interface consumer
```

Step 6 Add one or more members in the cluster interface.

```
member device device_name device-interface interface_name
```

Parameter	Description
device	The name of the cluster device that must have been already added to this device cluster using **cluster-device** command.
device-interface	The name of the interface on the cluster device.

Example:

```
apic1(config-cluster-interface)# member device C1 device-interface 1.1
```

Step 7 Add an interface to a member.

```
interface {ethernet ethernet_port | port-channel port_channel_name [fex fex_ID] |
  vpc vpc_name [fex fex_ID]} leaf leaf_ID
```

If you want to add a vNIC instead of an interface, then skip this step.

Parameter	Description
ethernet	(Only for an Ethernet or FEX Ethernet interface) The Ethernet port on the leaf where the cluster device is connected to Cisco Application Centric Infrastructure (ACI) fabric. If you are adding a FEX Ethernet member, specify both the FEX ID and the FEX port in the following format: *FEX_ID/FEX_port* For example: 101/1/23 The FEX ID specifies where the cluster device is connected to Fabric extender.
port-channel	(Only for a port channel or FEX port channel interface) The port channel name where the cluster device is connected to ACI fabric.
vpc	(Only for a virtual port channel or FEX virtual port channel interface) The virtual port channel name where the cluster device is connected to ACI fabric.
fex	(Only for a port channel, FEX port channel, virtual port channel, or FEX virtual port channel interface) The FEX IDs in a space-separated list that are used to form the port channel or virtual port channel.
leaf	The leaf IDs in a space-separated list where the cluster device is connected.

Example:

For an Ethernet interface, enter:

```
apic1(config-member)# interface ethernet 1/23 leaf 101
apic1(config-member)# exit
```

For a FEX Ethernet interface, enter:

```
apic1(config-member)# interface ethernet 101/1/23 leaf 101
apic1(config-member)# exit
```

For a port channel interface, enter:

```
apic1(config-member)# interface port-channel pc1 leaf 101
apic1(config-member)# exit
```

For a FEX port channel interface, enter:

```
apic1(config-member)# interface port-channel pc1 leaf 101 fex 101
apic1(config-member)# exit
```

For a virtual port channel interface, enter:

```
apic1(config-member)# interface vpc vpc1 leaf 101 102
apic1(config-member)# exit
```

For a FEX virtual port channel interface, enter:

```
apic1(config-member)# interface vpc vpc1 leaf 101 102 fex 101 102
apic1(config-member)# exit
```

Step 8 Add a vNIC to a member.

```
vnic "vnic_name"
```

If you want to add an interface instead of a vNIC, then see the previous step.

Parameter	Description
vnic	The name of the vNIC adapter on the virtual machine for the cluster-device. Enclose the name in double quotes.

Example:

```
apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
```

Step 9 If you are done creating the device, exit the configuration mode.

Example:

```
apic1(config-cluster-interface)# exit
apic1(config-cluster)# exit
apic1(config-tenant)# exit
apic1(config)# exit
```

Creating a High Availablity Cluster Using the NX-OS-Style CLI

This example procedure creates a high availability cluster using the NX-OS-style CLI.

Procedure

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

```
apic1(config)# tenant t1
```

Step 3 Create a cluster:

Example:

```
apic1(config-tenant)# l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 servicetype FW
```

Step 4 Add the cluster devices:

Example:

```
apic1(config-cluster)# cluster-device C1
apic1(config-cluster)# cluster-device C2
```

Step 5 Add a provider cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface provider vlan 101
```

Step 6 Add member devices to the interface:

Example:

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 7 Add another provider cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface provider vlan 102
```

Step 8 Add the same member devices from the first interface to this new interface:

Example:

```
apic1(config-cluster-interface)# member device C1 device-interface Po1
apic1(config-member)# interface vpc VPCPolASA leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
apic1(config-cluster-interface)# member device C2 device-interface Po2
apic1(config-member)# interface vpc VPCPolASA-2 leaf 103 104
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 9 Exit out of the cluster creation mode:

Example:

```
apic1(config-cluster)# exit
```

Creating a Virtual Device Using the NX-OS-Style CLI

This example procedure creates a virtual device using the NX-OS-style CLI.

Procedure

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

```
apic1(config)# tenant t1
```

Step 3 Create a cluster:

Example:

```
apic1(config-tenant)# l4l7 cluster name ifav108-citrix type virtual vlan-domain ACIVswitch servicetype
 ADC
```

Step 4 Add a cluster device:

Example:

```
apic1(config-cluster)# cluster-device D1 vcenter ifav108-vcenter vm NSVPX-ESX
```

Step 5 Add a consumer cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface consumer
```

Step 6 Add a member device to the consumer interface:

Example:

```
apic1(config-cluster-interface)# member device D1 device-interface 1_1
apic1(config-member)# interface ethernet 1/45 leaf 102
ifav108-apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 7 Add a provider cluster interface:

Example:

```
apic1(config-cluster)# cluster-interface provider
```

Step 8 Add the same member device to the provider interface:

Example:

```
apic1(config-cluster-interface)# member device D1 device-interface 1_1
apic1(config-member)# interface ethernet 1/45 leaf 102
ifav108-apic1(config-member)# vnic "Network adapter 2"
apic1(config-member)# exit
apic1(config-cluster-interface)# exit
```

Step 9 Exit out of the cluster creation mode:

Example:

```
apic1(config-cluster)# exit
```

Example XML for Creating a Logical Device

Example XML of Creating an LDevVip Object

The following example XML creates an `LDevVip` object:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
```

```
            <vnsRsALDevToDomP tDn="uni/vmmp-VMware/dom-mininet"/>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

For Cisco ACI Virtual Edge, the following example XML creates an `LDevVip` object associated to the Cisco ACI Virtual Edge VMM domain with `ave` as the switching mode:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <vnsLDevVip name="ADCCluster1" devtype="VIRTUAL" managed="no">
            <vnsRsALDevToDomP switchingMode="AVE" tDn="uni/vmmp-VMware/dom-mininet_ave"/>
        </vnsLDevVip>
    </fvTenant>
</polUni>
```

Example XML of Creating an AbsNode Object

The following example XML creates an `AbsNode` object:

```
<fvTenant name="HA_Tenant1">
    <vnsAbsGraph name="g1">
        <vnsAbsTermNodeProv name="Input1">
            <vnsAbsTermConn name="C1">
            </vnsAbsTermConn>
        </vnsAbsTermNodeProv>

        <!-- Node1 provides a service function -->
        <vnsAbsNode name="Node1" managed="no">
            <vnsAbsFuncConn name="outside" >
            </vnsAbsFuncConn>
            <vnsAbsFuncConn name="inside" >
            </vnsAbsFuncConn>
        </vnsAbsNode>

        <vnsAbsTermNodeCon name="Output1">
            <vnsAbsTermConn name="C6">
            </vnsAbsTermConn>
        </vnsAbsTermNodeCon>

        <vnsAbsConnection name="CON2" >
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeCon-Output1/AbsTConn"/>
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-outside"/>
        </vnsAbsConnection>

        <vnsAbsConnection name="CON1" >
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsNode-Node1/AbsFConn-inside"/>
            <vnsRsAbsConnectionConns
              tDn="uni/tn-HA_Tenant1/AbsGraph-g1/AbsTermNodeProv-Input1/AbsTConn"/>
        </vnsAbsConnection>
    </vnsAbsGraph>
</fvTenant>
```

Example XML of Associating a Layer 4 to Layer 7 Service Endpoint Group with a Connector

The following example XML associates a Layer 4 to Layer 7 service endpoint group with a connector:

```
<fvTenant name="HA_Tenant1">
    <vnsLDevCtx ctrctNameOrLbl="any" descr="" dn="uni/tn-HA_Tenant1/ldevCtx-c-any-g-any-n-any"

      graphNameOrLbl="any" name="" nodeNameOrLbl="any">
        <vnsRsLDevCtxToLDev tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1"/>
        <vnsLIfCtx connNameOrLbl="inside" descr="" name="inside">
            <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG1"/>
            <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-provBD1"/>
            <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-inside"/>
        </vnsLIfCtx>
        <vnsLIfCtx connNameOrLbl="outside" descr="" name="outside">
            <vnsRsLIfCtxToSvcEPg tDn="uni/tn-HA_Tenant1/ap-sap/SvcEPg-EPG2"/>
            <vnsRsLIfCtxToBD tDn="uni/tn-HA_Tenant1/BD-consBD1"/>
            <vnsRsLIfCtxToLIf tDn="uni/tn-HA_Tenant1/lDevVip-ADCCluster1/lIf-outside"/>
        </vnsLIfCtx>
    </vnsLDevCtx>
</fvTenant>
```

Example XML of Using Static Encapsulation with a Layer 4 to Layer 7 Service Endpoint Group

The following example XML uses static encapsulation with a Layer 4 to Layer 7 services endpoint group:

```
<polUni>
    <fvTenant name="HA_Tenant1">
        <fvAp name="sap">
            <vnsSvcEPg name="EPG1" encap="vlan-3510">
            </vnsSvcEPg>
        </fvAp>
    </fvTenant>
</polUni>
```

Modifying a Device Using the GUI

After you create a device, you can modify the device.

Note To create a device or to add a device to an existing cluster, you must use the "Creating a Device" procedure.

Procedure

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***device_name***.
The Work pane displays information about the device.

Step 4 You can change some of the parameters in the **General** section.

You can add interfaces or change the path for the existing interfaces in the **Device 1** section. To add an interface, click the + button. To change the path, double-click on the path you want to change.

Step 5 After you making any changes to the parameters, click **Submit**.

Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the GUI

The following procedure enables trunking on a Layer 4 to Layer 7 virtual ASA device using the GUI.

Before you begin

- You must have configured a Layer 4 to Layer 7 virtual ASA device.

Procedure

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.
Step 2 In the Work pane, double click the tenant's name.
Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Devices** > ***device_name***.
Step 4 In the Work pane, put a check in the **Trunking Port** check box.
Step 5 Click **Submit**.

Enabling Trunking on a Layer 4 to Layer 7 Virtual ASA device Using the REST APIs

The following procedure provides an example of enabling trunking on a Layer 4 to Layer 7 virtual ASA device using the REST APIs.

Before you begin

- You must have configured a Layer 4 to Layer 7 virtual ASA device.

Procedure

Enable trunking on the Layer 4 to Layer 7 device named `InsiemeCluster`:

```
<polUni>
    <fvTenant name="tenant1">
        <vnsLDevVip name="InsiemeCluster" devtype="VIRTUAL" trunking="yes">
            ...
            ...
        </vnsLDevVip>
```

```
    </fvTenant>
</polUni>
```

Using an Imported Device with the REST APIs

The following REST API uses an imported device:

```
<polUni>
  <fvTenant dn="uni/tn-tenant1" name="tenant1">
    <vnsLDevIf ldev="uni/tn-mgmt/lDevVip-ADCCluster1"/>
    <vnsLDevCtx ctrctNameOrLbl="any" graphNameOrLbl="any" nodeNameOrLbl="any">
      <vnsRsLDevCtxToLDev tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]"/>
      <vnsLIfCtx connNameOrLbl="inside">
        <vnsRsLIfCtxToLIf
tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-inside"/>
        <fvSubnet ip="10.10.10.10/24"/>
        <vnsRsLIfCtxToBD tDn="uni/tn-tenant1/BD-tenant1BD1"/>
      </vnsLIfCtx>
      <vnsLIfCtx connNameOrLbl="outside">
        <vnsRsLIfCtxToLIf
tDn="uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-outside"/>
        <fvSubnet ip="70.70.70.70/24"/>
        <vnsRsLIfCtxToBD tDn="uni/tn-tenant1/BD-tenant1BD4"/>
      </vnsLIfCtx>
    </vnsLDevCtx>
  </fvTenant>
</polUni>
```

Importing a Device From Another Tenant Using the NX-OS-Style CLI

You can import a device from another tenant for a shared services scenario.

Procedure

Step 1 Enter the configure mode.

Example:

```
apic1# configure
```

Step 2 Enter the configure mode for a tenant.

```
tenant tenant_name
```

Example:

```
apic1(config)# tenant t1
```

Step 3 Import the device.

```
l4l7 cluster import-from tenant_name device-cluster device_name
```

Parameter	Description
import-from	Name of the tenant from where to import the device.
device-cluster	Name of the device cluster to import from the specified tenant.

Example:

```
apic1(config-tenant)# l4l7 cluster import-from common device-cluster d1
apic1(config-import-from)# end
```

Verifying the Import of a Device Using the GUI

You can use the GUI to verify that a device was imported successfully.

Procedure

Step 1 On the menu bar, choose **Tenants** > **All Tenants**.

Step 2 In the Work pane, double click the tenant's name.

Step 3 In the Navigation pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Imported Devices** > ***device_name***.

The device information appears in the **Work** pane.