Defining a Logical Device

- About Device Clusters, page 1
- About Concrete Devices, page 2
- Creating a Device Using the GUI, page 2
- Creating a Device Using the NX-OS-Style CLI, page 5
- Using an Imported Device Using the REST APIs, page 9
- Importing a Device From Another Tenant Using the NX-OS-Style CLI, page 9
- Using an Imported Device Using the Object Model CLI, page 10
- Verifying the Import of a Device Using the GUI, page 11

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 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. Assign the management IP address to the device cluster.
3. Register the device cluster with the APIC. The APIC validates the device using the device specifications from the device package.

About Managed Device Clusters

A device cluster can be configured as a managed device cluster. In managed mode, the Application Policy Infrastructure Controller (APIC) programs the devices during graph instantiation using the configuration provided to the APIC by an APIC administrator. For a managed device cluster, the APIC requires the device package for managing the devices in the device cluster.

By default, a device cluster is configured as a managed device cluster.

The following settings are needed when a device cluster is configured as managed:

- Device package
- Connectivity information for the logical device (vnsLDevViP) and devices (CDev)-management IP address, credentials, and in-band connectivity information
- Information about supported function types (go-through, go-to)
- Information about context awareness (single context or multi-context)

The APIC needs to know the topology information (LIF, CIF) for the device cluster and devices. This information is needed so that the APIC can program the appropriate ports on the leaf, and the APIC can also use this information for troubleshooting wizard purposes. The APIC also needs to know the relation to `DomP`, which is used for allocating the encapsulation.

About Unmanaged Device Clusters

A device cluster can be configured as an unmanaged device cluster. For an unmanaged device cluster, the Application Policy Infrastructure Controller (APIC) allocates only the network resources for the service graph and program on only the fabric side during graph instantiation. This might be useful if your environment already has an existing orchestrator or a dev-op tool that programs the devices in a device cluster. In some other cases, the device package for the service appliance is not available. Unmanaged mode enables the APIC to work with service devices without needing to have a device package.

The APIC needs to know the topology information (LIF, CIF) for the device cluster and devices. This information is needed so that the APIC can program the appropriate ports on the leaf, and the APIC can also use this information for troubleshooting wizard purposes. The APIC also needs to know the relation to `DomP`, which is used for allocating the encapsulation.

About Concrete Devices

A concrete device has concrete interfaces. When a concrete device is added to a logical device, 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.

Creating a Device Using the GUI

You can connect to either a physical device or a virtual machine. The fields are slightly different depending the type to which you are connecting. 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. Additionally, you can select an unknown model, which allows you to manually configure the connections.

Before You Begin

- You must have configured a tenant.

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_name*** > **L4-L7 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
Managed check box	Put a check in the box to create a managed device, or remove the check from the box to create an unmanaged device.
Name field	Enter a name for the device.
Service Type drop-down list	Choose the service type.
Device Type buttons	Choose the device type.
Physical Domain or **VMM Domain** drop-down list	Choose the physical domain or VMM domain.
Mode radio buttons	Choose the mode for the device.
Device Package drop-down list	(Only for managed devices) Choose the mode for the device.
Model drop-down list	(Only for managed devices) Choose the vendor-provided device package that you will use.

Step 6 (Only for managed devices) In the **Connectivity** section, complete the following fields:

Name	Description
APIC to Device Management Connectivity radio buttons	Choose the type of connectivity. Choose **Out-of-Band** when you are connecting to a device that is outside of the fabric or **In-Band** when you are connecting to a device through the fabric.

Step 7 (Only for managed devices) In the **Credentials** section, complete the following fields:

Name	Description
User Name field	Enter your user name.

Name	Description
Password field	Enter your password.
Confirm Password field	Enter your password again.

Step 8 In the **Device 1** section, complete the following fields:

Name	Description
Management IP Address field	(Only for managed devices) Enter the management IP address of the device to which you are connecting.
Management Port field and drop-down list	(Only for managed devices) Enter the management port or choose a value from the drop-down list.
VM drop-down list	(Only for the virtual device type) Choose a virtual machine.
Chassis drop-down list	(Only for managed devices) Choose a chassis.

Step 9 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	Choose a port, port channel, or virtual port channel to which the interface will connect.

Step 10 Click **Update**.

Step 11 (Only for an HA cluster) Complete the fields for each device.

Step 12 Complete the fields for the **Cluster** section.

Step 13 Click **Next**.
The **Device Configuration** page displays a list of possible features and parameters for the package you are using. You see a tab with the **Basic** parameters displayed and another tab **All Parameters** that displays all the available parameters with your device package. The basic parameters are included under **All Parameters**.

Step 14 In the **Features** section, choose the set of features that you want to use.
The set of parameters changes depending on the specific package you are using and the specific feature you select.

Step 15 For the parameters of the chosen features, supply the values as follows:

a) Double-click in the field you want to modify.
b) Enter the required information in the fields that appear.

c) Click **Update**.

Step 16 Click **Finish**.

Creating a Device Using the NX-OS-Style CLI

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.

Before You Begin

- You must have configured a tenant.

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.
function	(Optional) The function type. Possible values are: • go-to • go-through

Parameter	Description
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.

Parameter	Description
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
```

Using an Imported Device Using 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.

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
```

Using an Imported Device Using the Object Model CLI

The following commands use an imported device using the object model CLI:

```
# logical-device-context
cd '/aci/tenants/tenant1/l4-l7-services/device-cluster-selection-policies'
mocreate 'any' 'any' 'any'
cd 'any-any-any'
moset device-cluster
'tenants/tenant1/l4-l7-services/imported-device-clusters/[uni/tn-mgmt/lDevVip-ADCCluster1]'
moconfig commit


# vns-lifctx
cd
'/aci/tenants/tenant1/l4-l7-services/device-cluster-selection-policies/any-any-any/logical-interface-contexts'
mocreate 'inside'

cd 'inside'

moset logical-interface
'uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-inside'
moset bridge-domain 'tenants/tenant1/networking/bridge-domains/tenant1BD1'

moconfig commit


# fv-subnet
cd
'/aci/tenants/tenant1/l4-l7-services/device-cluster-selection-policies/any-any-any/logical-interface-contexts/inside/subnets'
mocreate '10.10.10.10/24'

moconfig commit


# vns-lifctx
cd
'/aci/tenants/tenant1/l4-l7-services/device-cluster-selection-policies/any-any-any/logical-interface-contexts'
mocreate 'outside'

cd 'outside'

moset logical-interface
'uni/tn-tenant1/lDevIf-[uni/tn-mgmt/lDevVip-ADCCluster1]/lDevIfLIf-outside'
moset bridge-domain 'tenants/tenant1/networking/bridge-domains/tenant1BD4'

moconfig commit


# fv-subnet
cd
'/aci/tenants/tenant1/l4-l7-services/device-cluster-selection-policies/any-any-any/logical-interface-contexts/outside/subnets'
mocreate '70.70.70.70/24'

moconfig commit


# logical-device-consumer
```

```
cd '/aci/tenants/tenant1/l4-l7-services/imported-device-clusters'
mocreate 'uni/tn-mgmt/lDevVip-ADCCluster1'
cd '[uni--tn-mgmt--lDevVip-ADCCluster1]'
moset faultcode '0'
moconfig commit
```

Verifying the Import of a Device Using the GUI

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

Step 1 On the menu bar, click the **TENANTS** tab. The **Tenant** window appears.

Step 2 On the submenu bar, click the name of the tenant for which you want to import devices.

Step 3 In the **Navigation** pane, expand the tenant's branch.

Step 4 Expand the **L4-L7 Services** branch.

Step 5 Expand the **Imported Devices** branch.

Step 6 Choose the appropriate device. The device information appears in the **Work** pane.