Monitoring a Service Graph

- Monitoring a Service Graph Instance Using the GUI, on page 1
- Monitoring Service Graph Faults Using the GUI, on page 2
- Resolving Service Graph Faults, on page 2
- Monitoring a Virtual Device Using the GUI, on page 7
- Monitoring Device Cluster and Service Graph Status Using the NX-OS-Style CLI, on page 8

Monitoring a Service Graph Instance Using the GUI

After you configure a service graph template and attach the graph to an endpoint group (EPG) and a contract, you can monitor the service graph instance. Monitoring includes viewing the state of the graph instances, functions of a graph instance, resources allocated to a function, and parameters specified for a function.

Procedure

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

Step 2 In the Work pane, double click the tenant's name for which you want to monitor its service graph.

Step 3 In the **Navigation** pane, choose **Tenant** ***tenant_name*** > **Services** > **L4-L7** > **Deployed Graph Instances**. The **Work** pane displays the following information about the active service graph instances:

Name	Description
Service Graph column	The name of the service graph template.
Contract column	The name of the contract that is shown in the service graph template.
Contained By column	The name of the network that contains the service graph template.
State column	The state of the service graph template. A state of **applied** means that the graph has been applied, and the graph policy is active within the fabric and the service device.
Description column	The description of the service graph.

Step 4 Expand the **Deployed Service Graphs** branch. The active service graph instances are listed under the branch.

Step 5 Click a service graph instance to view additional information about that instance in the **Work** pane. The default view is the topology of the graph. You can click one of the tabs in the **Work** pane to change the view for that graph.

Step 6 Expand the branch for one of the graph instances. The functions of the graph instance are listed under the instance.

Step 7 Click one of the functions to view additional information about that function in the **Work** pane. The default view is the policy of that function. You can click one of the tabs in the **Work** pane to change the view for that function. The **Work** pane displays the following information about the policy:

Name	Description
POLICY tab	The function's properties, resources allocated to the function, and the parameters of the function.
FAULTS tab	The issues that are happening on the function node.
HISTORY tab	The history of events that occurred on the function node.

Step 8 In the **Navigation** pane, click **Deployed Device**. The **Work** pane displays information about the device instances.

Monitoring Service Graph Faults Using the GUI

After you configure a service graph template and attach the graph to an endpoint group (EPG) and a contract, you can monitor a service graph template's faults.

Procedure

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

Step 2 In the Work pane, double click the tenant's name for which you want to monitor its service graph.

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

Step 4 Expand the branch for a graph instance for which you want to view its faults. The functions of the graph instance are listed under the instance.

Step 5 Click on one of the functions. By default, the **Work** pane shows the policy of that function.

Step 6 Click the **FAULTS** tab in the **Work** pane. The **Work** pane displays the faults of the function node.

Resolving Service Graph Faults

After you have observed one or more service graph template faults, resolving the issue depends on the fault. The following tables describe the faults and provide how to resolve faults.

Table 1: Connector Faults

Fault	CLI Label	Description and Resolution
missing-connection	connection associated with a connector not found	The configuration for a graph connector is invalid. The associated connection for the connector could not be found.
missing-nodeinst	NodeInst associated with a connector not found	The configuration for a graph connector is invalid. The associated NodeInst for the connector could not be found.
conn-nonrenderable	Graph connector could not be rendered.	The configuration for a graph connector is invalid. The graph could not be rendered.
invalid-bd	BD associated with a connector is not valid	The configuration for a graph connector is invalid. The associated bridge domain for the connector is not valid.
invalid-ctx	Ctx associated with a connector is not valid.	The configuration for a graph connector is invalid. The associated Ctx for the connector is not valid.
missing-peer-conn	Peer connector associated with a connector not found.	Configuration for a graph connector is invalid. The peer connector for the connection could not be found.

Table 2: AbsGraph and GraphInst Faults

Fault	CLI Label	Description and Resolution
invalid-abstract-graph-config	invalid abstract graph config	The abstract graph configuration is invalid.
missing-mandatory-param	mandatory param not found	A mandatory configuration parameter could not be resolved. Check the package for the mandatory parameter and make sure that AbsGraph has the parameter.
param-cardinality-error	invalid param cardinality	A configuration parameter does not meet cardinality requirements. Check if you specified multiple instances of a parameter without specifying cardinality=n.
epp-download-failure	epp download failure	Graph policies failed to download to the switch.

Fault	CLI Label	Description and Resolution
param-duplicate-name-failure	duplicate param name	Multiple identical copies of a parameter were found with the same name.
id-allocation-failure	id allocation failure	A unique network resource (either VLAN or VXLAN) could not be allocated.
missing-ldev	No cluster found	A cluster could not be found.
context-cardinality-violation-failure	invalid cluster context cardinality	The cluster does not support the required tenancy(multi-tenant or single tenant).
function-type-mismatch-failure	invalid function type	The function type is not supported for the selected device. Check if the AbsNode functype and resolved LDevVip function type match.
invalid-abstract-graph-config-param	invalid abstract graph config param	The abstract graph configuration parameter is invalid.
missing-mparam	No parameter definition found	A required parameter definition could not be found.
missing-abs-graph	no abs graph found	The abstract graph configuration is missing for the graph instance.
invalid-param-config	invalid param config	The parameter configuration is invalid.
invalid-param-scope	invalid parameter scope	The parameter scope is invalid. Check the vnsRsScopeToTerm parameter in the AbsGraph to see if parameter is correct.
invalid-ldev	Invalid cluster	The cluster configuration is invalid. Check the status of the resolved LDevVip and correct the fault.
missing-tenant	no tenant found	The tenant could not be found for the graph.
internal-error	internal error	An internal error occurred during graph processing.
resource-allocation-failure	resource allocation failure	A required resource could not be allocated during graph processing.
missing-abs-function	no abstract function found	The abstract function definition is missing.
param-validation-failed	param validation failure	The configuration parameter value is invalid.
missing-mconn	No connector found	A required connector could not be found.

Fault	CLI Label	Description and Resolution
cdev-missing-mgmt-ip	no mgmt ip found for cdev	The management IP address could not be found for concrete device. Check if vnsCMgmt is present for the resolved vnsCDev.
invalid-graphinst	invalid graphinst config	The graph instance is invalid.
missing-interface	no interface found	An interface could not be found.
missing-bd	no bd found	A bridge domain could not be found.
missing-terminal	Terminal node is missing a terminal	Terminal node is missing a terminal. Check the terminal node settings.
missing-namespace	no vlan/vxlan namespace found	The namespace that is needed to allocate the VLAN or VXLAN is missing. Verify that the resolved vnsLDevVip has the phyDomp parameter or the vmmDomp parameter configured that has a relation to the resolved fvnsVlanInstp.
missing-mfunc	No function found in device package	A required function could not be found in the device package. Verify that all AbsNode function types are present in the package.
missing-lif	no cluster interface found	A required cluster interface could not be found. Verify that the vnsLIf parameter in vnsLDevVip is configured correctly.
invalid-absfunc-profile	Abstract Function Profile config is invalid	The abstract function profile configuration is invalid. This fault may be due to an invalid configuration parameter that is specified in the profile.
missing-cdev	No device found	The concrete device could not be found in the cluster. Verify that a valid vnsCDev is present under the resolved vnsLDevVip.
inappropriate-devfolder	Illegal folder in configuration	No corresponding folder was found in the device package.
invalid-devctx	Device context is not legal for this folder	The device package does not allow specifying a device context for this folder.

Fault	CLI Label	Description and Resolution
insufficient-devctx	Folder must have one value for each associated CDev	The folder is concrete device specific. The folder must have at least one value for each concrete device.
cdev-missing-cif	No interface defined	A concrete device must have at least one interface defined.
cdev-missing-pathinfo	Missing path for interface	For a physical service appliance, we must know to which leaf ports the interface is connected. Verify that the vnsCifPathAtt parameter is present for all vnsCIf under the resolved vnsCDev.
missing-cif	Device interfaces does not match cluster	The device interfaces should match the interfaces configured for their cluster. Verify that the vnsCIf parameter and the vnsLIf parameter are present under the resolved vnsLDevVip.
ldevvip-missing-mgmt-ip	No Mgmt ip found for LDevVip	The management IP address could not be found for LDevVip.
lif-invalid-MIf	LIf has an invalid MifLbl	The MIfLbl contained by LIf is not present in the device package.
lif-invalid-CIf	LIf has an invalid CIf	The CIf contained by LIf is not present. Check the concrete device and CIf settings.
missing-function-node	Abstract graph missing function node	An abstract graph must have at least one function node.
graph-loop-detected	Abstract graph config has a loop	The abstract graph configuration is invalid. The configuration has a loop.
gothrough-routing-enabled-both	Both the legs of go through node has routing enabled	Both the legs of the go through node have routing enabled.
invalid-terminal-nodes	Abstract graph has invalid number of terminal nodes	An abstract graph must have at least two terminal nodes.
missing-ldev-ctx	No device context found for LDev	The device context for the device could not be found. Verify that vnsLDevCtx has values that match the contract, graph and node.
arp-flood-enabled	ARP flood is enabled on the management end point group	ARP flood must be disabled for the management endpoint group.

Fault	CLI Label	Description and Resolution
folderinst-validation-failed	FolderInst has key, that is not found in MFolder	The FolderInst's key and value should honor MFolder specifications.
paraminst-validation-failed	ParamInst has key and/or value, that are not found in MParam	ParamInst's key and value should honor MParam specifications.
invalid-mfolder	FolderInst points to an invalid MFolder	FolderInst must point to a valid MFolder.
invalid-mparam	ParamInst points to an invalid MParam	ParamInst must point to a valid MParam.
devfolder-validation-failed	DevFolder has key, that is not found in MFolder	DevFolders key and value should honor MFolder specifications.
devparam-validation-failed	DevParam has key and/or value, that are not found in MParam	DevParam's key and value should honor MParam specifications
cdev-missing-virtual-info	Virtual Object Info is missing in CDev	Virtual object information must be provided if LDevVip is of type Virtual.
invalid-rsmconnatt	Relationship to metaconnector is invalid	Correct the metaconnector DN and ensure it binds to the correct MDev hierarchy.

Monitoring a Virtual Device Using the GUI

After you configure a service graph template and attach the graph to an endpoint group (EPG) and a contract, you can monitor the virtual devices of a tenant. Monitoring the virtual devices tells you what devices are in use, which VLANs are configured for a device, the parameters passed to the devices, the statistics of the devices, and the health of the devices.

Procedure

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

Step 2 In the Work pane, double click the tenant's name for which you want to monitor its service graph.

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

Step 4 Click on one of the deployed devices. By default, the **Work** pane shows the policy of that deployed device. You can click the tabs in the **Work** pane to change the view. The tabs display the following information about the virtual device:

Tab	Description
POLICY tab	The device that is in use, the VLANs that are configured within the device, and the parameters that have been passed to the devices.

Tab	Description
OPERATIONAL tab	The statistics that are being received from the various devices.
HEALTH tab	The health of the devices.

Monitoring Device Cluster and Service Graph Status Using the NX-OS-Style CLI

The commands in this section provide examples of how to monitor device cluster and service graph status using the NX-OS-style CLI.

Showing the Operation Information of a Device Cluster

The following command shows the operational information of a device cluster:

```
show l4l7-cluster tenant tenant_name cluster device_cluster_name
```

Example:

```
apic1# show l4l7-cluster tenant HA_Tenant1 cluster Firewall
tenant-graph : HA_Tenant1-g2,HA_Tenant1-g1


Device Cluster    : Firewall
Cluster Interface : consumer1
Encap             : vlan-501
Pctag             : 32773
Devices           : FW2(int),FW1(int)
Graphs            : HA_Tenant1-g1
Contracts         : HA_Tenant1-c1


Device Cluster    : Firewall
Cluster Interface : provider1
Encap             : vlan-502
Pctag             : 32774
Devices           : FW2(ext),FW1(ext)
Graphs            : HA_Tenant1-g1
Contracts         : HA_Tenant1-c1
```

Showing the Operation Status of a Device Cluster

The following command shows the operation status of a device cluster:

```
apic1# show l4l7-graph tenant tenant_name [graph graph_name]
```

Examples:

The following example gives high-level output of the status of the `HA_Tenant1` tenant:

```
apic1# show l4l7-graph tenant HA_Tenant1
Graph           : g1
Total Instances : 1
Encaps Used     : vlan-501,vlan-502,vlan-503,vlan-504
```

```
Device Used     : uni/tn-HA_Tenant1/lDevVip-Firewall

Graph           : g2
Total Instances : 1
Encaps Used     : vlan-501,vlan-502,vlan-503,vlan-504
Device Used     : uni/tn-HA_Tenant1/lDevVip-Firewall
```

The following example gives detailed output of the status of the `g1` service graph that is associated with the `HA_Tenant1` tenant:

```
apic1# show l4l7-graph tenant HA_Tenant1 graph g1
Graph           : HA_Tenant1-g1
Graph Instances : 1

Consumer EPg  : HA_Tenant1-consEPG1
Provider EPg  : HA_Tenant1-provEPG1
Contract Name : HA_Tenant1-c1
Config status : applied

Function Node Name : Node1
 Connector   Encap       Bridge-Domain  Device Interface
 ----------  ----------  ----------     --------------------
 consumer    vlan-3001   provBD1        consumer
 provider    vlan-3335   consBD1        provider
```

Showing the Faults of a Device Cluster

The following command shows the faults of a device cluster:

```
show faults l4l7-cluster
```

Example:

```
apic1# show faults l4l7-cluster
Code            : F0772
Severity        : minor
Last Transition : 2015-09-01T01:41:13.767+00:00
Lifecycle       : soaking-clearing
Affected object : uni/tn-ts1/lDevVip-d1/lIf-ext/fault-F0772
Description     : LIf configuration ext for L4-L7 Devices d1 for tenant ts1
                  is invalid.

Code            : F1085
Severity        : cleared
Last Transition : 2015-09-01T01:39:04.696+00:00
Lifecycle       : retaining
Affected object : uni/tn-ts1/lDevVip-d1/rsmDevAtt/fault-F1085
Description     : Failed to form relation to MO uni/infra/mDev-CiscoInternal-
                  NetworkOnly-1.0 of class vnsMDev

Code            : F1690
Severity        : minor
Last Transition : 2015-09-01T01:39:04.676+00:00
Lifecycle       : soaking
Affected object : uni/tn-ts1/lDevVip-d1/vnsConfIssue-missing-
                  namespace/fault-F1690
Description     : Configuration is invalid due to no vlan/vxlan namespace
                  found
```

Showing the Faults of a Service Graph

The following command shows the faults of a service graph:

```
show faults l4l7-graph
```

Example:

```
apic1# show faults l4l7-graph
Code            : F1690
Severity        : minor
Last Transition : 2015-11-25T20:07:33.635+00:00
Lifecycle       : raised
DN              : uni/tn-HA_Tenant1/AbsGraph-WebGraph/vnsConfIssue-invalid-
                  abstract-graph-config-param/fault-F1690
Description     : Configuration is invalid due to invalid abstract graph
                  config param
```

Showing the Running Configuration of a Device Cluster

The following command shows the running configuration of a device cluster:

```
show running-config tenant tenant_name l4l7 cluster
```

Example:

```
apic1# show running-config tenant common l4l7 cluster
# Command: show running-config tenant common l4l7 cluster
# Time: Thu Nov 26 00:35:59 2015
  tenant common
    l4l7 cluster name ifav108-asa type physical vlan-domain phyDom5 service FW function
go-through
      cluster-device C1
      cluster-interface consumer_1
        member device C1 device-interface port-channel1
          interface vpc VPCPolASA leaf 103 104
          exit
        exit
      cluster-interface provider_1
        member device C1 device-interface port-channel1
          interface vpc VPCPolASA leaf 103 104
          exit
        exit
      exit
```

Showing the Running Configuration of a Service Graph

The following command shows the running configuration of a service graph:

```
show running-config tenant tenant_name l4l7 graph
```

Example:

```
apic1# show running-config tenant common l4l7 graph
# Command: show running-config tenant common l4l7 graph
# Time: Thu Nov 26 00:35:59 2015
  tenant T1
    l4l7 graph Graph-Citrix contract Contract-Citrix
     service N1 device-cluster-tenant common device-cluster ifav108-citrix mode ADC_ONE_ARM

        connector provider cluster-interface pro
          bridge-domain tenant common name BD4-Common
          exit
        connector consumer cluster-interface pro
          bridge-domain tenant common name BD4-Common
          exit
        exit
      connection C1 terminal consumer service N1 connector consumer
      connection C2 terminal provider service N1 connector provider
      exit
```