Additional Information

Revised: September 8, 2021

Troubleshooting Cisco SD-WAN Issues

This section describes problems, possible causes, recommended actions, and error messages, if applicable to the problem.

Troubleshooting Cisco SD-WAN Reachability Issues

Table 1: SD-WAN Reachability Issues

Color	Green	Red	Comment
Deployment Status	Provisioned	Provisioned-Failed: See Troubleshooting notes. For more information, see Troubleshooting Cisco SD-WAN vEdge-Cloud Deployment Deployment Errors.	Checks that VNF(s) is fully deployed and in active state.
Reachability Status	Reachable	Not Reachable: See Troubleshooting notes. For more information, see Troubleshooting Cisco SD-WAN vEdge Reachability Errors.	Checks the connectivity between the deployed vEdge and Cisco SD-WAN Control Plane.

Troubleshooting Cisco SD-WAN vEdge-Cloud Deployment Errors

After the service packs are deployed on MSX, the customer configuration templates are imported into the Cisco Network Services Orchestrator (NSO) platform for automating network orchestration. These configurations are then pushed from MSX to customer devices as part of the orchestration of device configuration. If the SD-WAN provisioning is not successful, most times, it is due to wrong parameters in the deployment data on NSO. There are multiple NSO instances if you are deploying more than one service pack. Therefore, these steps must be performed on the service pack-specific NSO node. SD-WAN uses SD-Branch's NSO, so in this case, the nso node will be nso-vbranch.

Procedure

Step 1 Log in to one of the kubernetes master nodes.

```
# grep master inventory/inventory
[kube-master]
kubernetes-master-ctsai-east-2-1 ansible_host=<master_1_ip_address> ansible_user=centos
ansible_become=true
kubernetes-master-ctsai-east-2-2 ansible_host=<master_2_ip_address> ansible_user=centos
ansible_become=true
kubernetes-master-ctsai-east-2-3 ansible_host=<master_3_ip_address> ansible_user=centos
ansible_become=true
# ssh -F ssh.cfg centos@<master_1_ip_address>
```

Step 2 Access the NSO node using this command:

```
kubectl -n vms exec -it nso-vbranch-0 -c nso-vbranch /bin/sh
```

Step 3 Change to vms user.

```
su vmsnso
```

Step 4 Run NSO CLI

```
ncs_cli -u admin
```

Step 5 Get the branch-cpe name, using the following command:

```
vmsnso@ncs> show branch-infra:branch-infra
```

Example:

```
branch-cpe axj9AUv5A06MXXSSWWSAAA {
provider admin;
type ENCS;
serial <Device serial number>;
var SD-Branch_DEVICE_TYPE {
val ENCS;
}
var contact {
val "Samuel";
}
var email {
val noreply@cisco.com;
}
var phone {
val 1112221234;
}
vnfd SD-Branch-vEdge-18.3.0 {
vdu vEdge;
```

Step 6 Check the deployment summary, using the following command. Replace the branch-cpe name with the name that was identified in step 2.

```
vmsnso@ncs> show branch-infra:branch-infra-status branch-cpe <name_from_above_command> plan component
 state | tab
```

For example:

Example:

```
vmsnso@ncs> show branch-infra:branch-infra-status branch-cpe axj9AUv5A06MXXSSWWSAAA plan component
state | tab
NAME STATE STATUS WHEN ref MESSAGE
----------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------

----------------------------------------------------------------------------------------------------

self init reached 2018-09-06T19:30:32 -
ready failed 2018-09-06T20:33:11 -
axj9AUv5A06MdGMqHpmQ5ffN init reached 2018-09-06T19:30:32 -
pnp-callhome reached 2018-09-06T19:30:32 -
ready reached 2018-09-06T19:31:28 - Ready
vEdge_SD-Branch-vEdge-18.3.0 init reached 2018-09-06T19:31:36 -
ready reached 2018-09-06T19:32:30 - Ready
vEdge_axj9AUv5A06MdGMqHpmQ5ffN init reached 2018-09-06T19:32:31 -
vm-deployed reached 2018-09-06T19:32:58 -
vm-alive reached 2018-09-06T19:42:58 -
ready failed 2018-09-06T20:33:11 - NFVIS Error - Recovery: Recovery completed with errors for VM:
[axj9AUv5A06MdGMq_vEdge-_0_828e4709-1644-4706-946a-12d7fa71c8e3]
vm-recovered failed 2018-09-06T20:33:11 -
```

The summary displays the problem, if any. In the above example, SYSTEM_IP variable is wrong, because of which ENCS was unable to configure the VNF and was unable to attach the deployed Control plane on MSX.

Troubleshooting Cisco SD-WAN vEdge Reachability Errors

If there is no connectivity between the deployed vEdge and Cisco SD-WAN Control Plane:

Procedure

Step 1 Login to the deployed vEdge and check the status of deployed vEdges.

- For a physical vEdge, directly login to the vEdge device.
- For an IOS XE device, login to IOS device then login to SD-WAN instance installed on the device.

```
ssh admin@<vEdge IP address>
a5fG2U3kulIE8EqDfHzPHKYZ# show system deployments
NAME ID STATE
----------------------------------------------------------
a5fG2U3kulIE8EqDfHzPHKYZ_vEdge.vEdge-vEdge 6 running
a5fG2U3kulIE8EqDfHzPHKYZ# vmConsole a5fG2U3kulIE8EqDfHzPHKYZ_vEdge.vEdge-vEdge
Connected to domain a5fG2U3kulIE8EqDfHzPHKYZ_vEdge.vEdge-vEdge
Escape character is ^]
viptela 18.3.0
Site001 login: admin
Password:
Welcome to Viptela CLI
admin connected from 127.0.0.1 using console on Canada_Site001
Site001#
```

Step 2 Check the status of control connection, using the following command:

```
show control connections

Site001# show control connections
PEER PEER CONTROLLER
PEER PEER PEER SITE DOMAIN PEER PRIV PEER PUB GROUP
TYPE PROT SYSTEM IP ID ID PRIVATE IP PORT PUBLIC IP PORT LOCAL COLOR PROXY STATE UPTIME ID
-------------------------------------------------------------------------------------------------------------------------------------------------------
vbond dtls 0.0.0.0 0 0 <vbond IP> 12346 5<vbond ip> 12346 default - connect 0
Site001#
Site001# show control connections
Site001#
```

If nothing shows up in the output, it shows that the vEdge is unable to establish dtls connection to vBond.

Step 3 To check why the connection has not been established, use the following command.

```
show control connections-history

Site001# show control connections-history
Legend for Errors
ACSRREJ - Challenge rejected by peer. NOVMCFG - No cfg in vmanage for device.
BDSGVERFL - Board ID Signature Verify Failure. NOZTPEN - No/Bad chassis-number entry in ZTP.
BIDNTPR - Board ID not Initialized. OPERDOWN - Interface went oper down.
BIDNTVRFD - Peer Board ID Cert not verified. ORPTMO - Server's peer timed out.
BIDSIG - Board ID signing failure. RMGSPR - Remove Global saved peer.
```

```
CERTEXPRD - Certificate Expired RXTRDWN - Received Teardown.
CRTREJSER - Challenge response rejected by peer. RDSIGFBD - Read Signature from Board ID failed.
CRTVERFL - Fail to verify Peer Certificate. SERNTPRES - Serial Number not present.
CTORGNMMIS - Certificate Org name mismatch. SSLNFAIL - Failure to create new SSL context.
DCONFAIL - DTLS connection failure. STNMODETD - Teardown extra vBond in STUN server mode.
DEVALC - Device memory Alloc failures. SYSIPCHNG - System-IP changed.
DHSTMO - DTLS HandShake Timeout. SYSPRCH - System property changed
DISCVBD - Disconnect vBond after register reply. TMRALC - Timer Object Memory Failure.
DISTLOC - TLOC Disabled. TUNALC - Tunnel Object Memory Failure.
DUPCLHELO - Recd a Dup Client Hello, Reset Gl Peer. TXCHTOBD - Failed to send challenge to BoardID.
DUPSER - Duplicate Serial Number. UNMSGBDRG - Unknown Message type or Bad Register msg.
DUPSYSIPDEL- Duplicate System IP. UNAUTHEL - Recd Hello from Unauthenticated peer.
HAFAIL - SSL Handshake failure. VBDEST - vDaemon process terminated.
IP_TOS - Socket Options failure. VECRTREV - vEdge Certification revoked.
LISFD - Listener Socket FD Error. VSCRTREV - vSmart Certificate revoked.
MGRTBLCKD - Migration blocked. Wait for local TMO. VB_TMO - Peer vBond Timed out.
MEMALCFL - Memory Allocation Failure. VM_TMO - Peer vManage Timed out.
NOACTVB - No Active vBond found to connect. VP_TMO - Peer vEdge Timed out.
NOERR - No Error. VS_TMO - Peer vSmart Timed out.
NOSLPRCRT - Unable to get peer's certificate. XTVMTRDN - Teardown extra vManage.
NEWVBNOVMNG- New vBond with no vMng connections. XTVSTRDN - Teardown extra vSmart.
NTPRVMINT - Not preferred interface to vManage. STENTRY - Delete same tloc stale entry.
EMBARGOFAIL - Embargo check failed

PEER PEER
PEER PEER PEER SITE DOMAIN PEER PRIVATE PEER PUBLIC LOCAL REMOTE REPEAT
TYPE PROTOCOL SYSTEM IP ID ID PRIVATE IP PORT PUBLIC IP PORT LOCAL COLOR STATE ERROR ERROR COUNT
DOWNTIME
------------------------------------------------------------------------------------------------------------------------------------
vbond dtls 0.0.0.0 0 0 <vbond IP> 12346 52.206.47.80 12346 default connect DCONFAIL NOERR 14
2018-09-06T16:44:56+0000
Site001#
```

As seen above, the LOCAL ERROR is mostly "DCONFAIL" which means DTLS connection failure. This happens when the vEdge is unable to reach the vBond either due to network connectivity issues or firewall is blocking the DTLS connection. For an understanding of other reachability errors, see the Cisco SD-WAN knowledge base.

Troubleshooting ENCS Reachability Issues

If the ENCS device is unreachable or unavailable, then do the following:

Procedure

Step 1 Log into the **ENCS** box. Use SSH to connect to the ENCS box.

```
ssh <username>@<management IP address>
```

Step 2 Do the following on the ENCS box:

a. Enter the configuration mode.

```
config
```

b. Revert the IP to the WAN interface of the ENCS, if the ENCS was set in the single IP mode.

```
no single-ip-mode
```

c. Remove the VPN configurations.

```
no secure-overlay
```

d. Remove all deployments.

```
no vm_lifecycle tenants tenant admin deployments deployment
```

Note If a specific VNF needs to be deleted, enter the deployment name in the above command.

e. Removes all images.

```
no vm_lifecycle images image
```

Note If a specific image needs to be deleted, enter the image name in the above command.

f. Save the changes and exit the configuration mode.

```
commit and-quit
```

g. Restarts PNP process.

```
pnp action command restart
```

Changing MSX Trace Logging Level During Runtime

Using the procedure in this section you can change any MSX trace logging level during the runtime. The following shows SD-WAN log definition in logback.xml.

```
<property name="LOG_FILE" value="logs/sdwanservice.log"/>

<!-- the rollover settings with mean a max size per log of 100Mb and 7 days -->
<property name="MAX_HISTORY" value="7"/>
<property name="MAX_FILE_SIZE" value="100MB"/>

<include resource="com/cisco/nfv/logging/nfv_base_logback.xml"/>

<!-- the specific loggers -->
<logger name="com.cisco.phiservice" level="DEBUG"/>
<logger name="com.cisco.vms.sdwanservice" level="DEBUG"/>
<logger name="com.cisco.vms.sdwanservice.integration.viptela" level="DEBUG"/>
<logger name="com.cisco.vms.svcpack.logging" level="DEBUG"/>

<logger name="org.springframework" level="INFO"/>
<logger name="org.springframework.security.oauth2" level="INFO"/>
<logger name="org.springframework.integration" level="OFF"/>
<logger name="org.springframework.oxm" level="OFF"/>
<logger name="org.springframework.http" level="ERROR"/>
```

To change the logging level during runtime:

Procedure

Step 1 Obtain the MSX client credentials.

Use the credential you use for logging in to the MSX Portal. If you do not have these credentials, contact your Service Provider Administrator.

Step 2 Obtain an access token from the MSX authorization Server. Use the following curl command to get the access token. Use the following curl command to get the access token.

```
curl -X POST 'https://<MSX_URL>/idm/v2/token?grant_type=password' -H 'Authorization: Basic
<MSX_BASIC_AUTH>' \
-H 'Content-Type: application/x-www-form-urlencoded' \
-H 'cache-control: no-cache' \
-d 'username=<MSXportal_username>&password=<MSXportal_password>'
```

Where:

- Replace <MSX_URL> by real MSX URL
- Replace <MSX_BASIC_AUTH> with real value of the Authorization of clientID and clientSecret, which is base64 of "clientID:clientSecrect". User defined their OAuth2 Authentication clientID and clientSecret in **Settings > BSS Integration > REST Configuration**

  .
- Replace <MSXportal_username> by Portal username
- Replace <MSXportal_password> by Portal password

Step 3 Check the current package logging level. Use the following curl command to check the current package log level:

```
curl -X GET https://<MSX_URL>/<service>/admin/loggers/<package>  -H 'Authorization: Bearer
<access_token>' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'
```

Where:

- Replace <MSX_URL> by real MSX URL
- Replace <service> by service endpoint (For example: sdwanservice)
- Replace <package> by real package name (For example: com.cisco.vms.sdwanservice.integration.viptela)
- Replace <access_token> from Step 2.

Step 4 Change the package logging level. Use the following curl command to update the package log level:

```
curl -X POST https://<MSX_URL>/<service>/admin/loggers/<package>  -H 'Authorization: Bearer
<access_token>' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache' \
-d '{
    "configuredLevel": "<LOG_LEVEL>"
}'
```

Where:

- Replace <MSX_URL> by real MSX URL
- Replace <service> by service endpoint (for example: sdwanservice)
- Replace <package> by real package name (for example: com.cisco.vms.sdwanservice)

Note This package name does not necessarily be defined in logback.xml, as long as this package exists in the source code.

- Replace <access_token> from Step 2.
- Replace <LOG_LEVEL> by log level you want to set.

Step 5 Verify the changes in package logger logging level. Repeat Step 3. Use the following curl command to verify the log level after the changes:

```
curl -X GET http://<MSX_URL>/<service>/admin/loggers/<package>  -H 'Authorization: Bearer
<access_token>' \
-H 'Content-Type: application/json' \
-H 'cache-control: no-cache'
# with output for logging level
{"configuredLevel":"TRACE","effectiveLevel":"TRACE"}
```

Where:

- Replace <MSX_URL> by real MSX URL
- Replace <service> by service endpoint (for example: sdwanservice)
- Replace <package> by real package name

Troubleshooting Control Plane

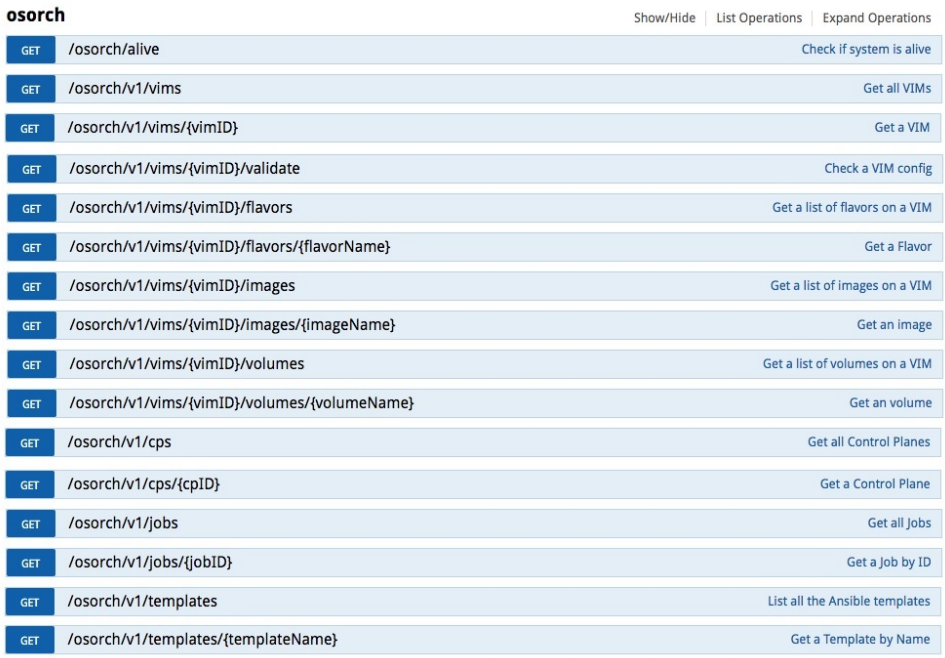
Troubleshooting Control Plane on OpenStack

If MSX is unable to reach the OpenStack control plane, then it should be due to some issues pertaining to proxy settings.

Note When both MSX and OpenStack cloud are on the corp network, proxy is not required. Ensure that the vManage IP address is added to the "no proxy" list in the "sdwanservice-rc.yml" and then restart the SD-WAN pod.

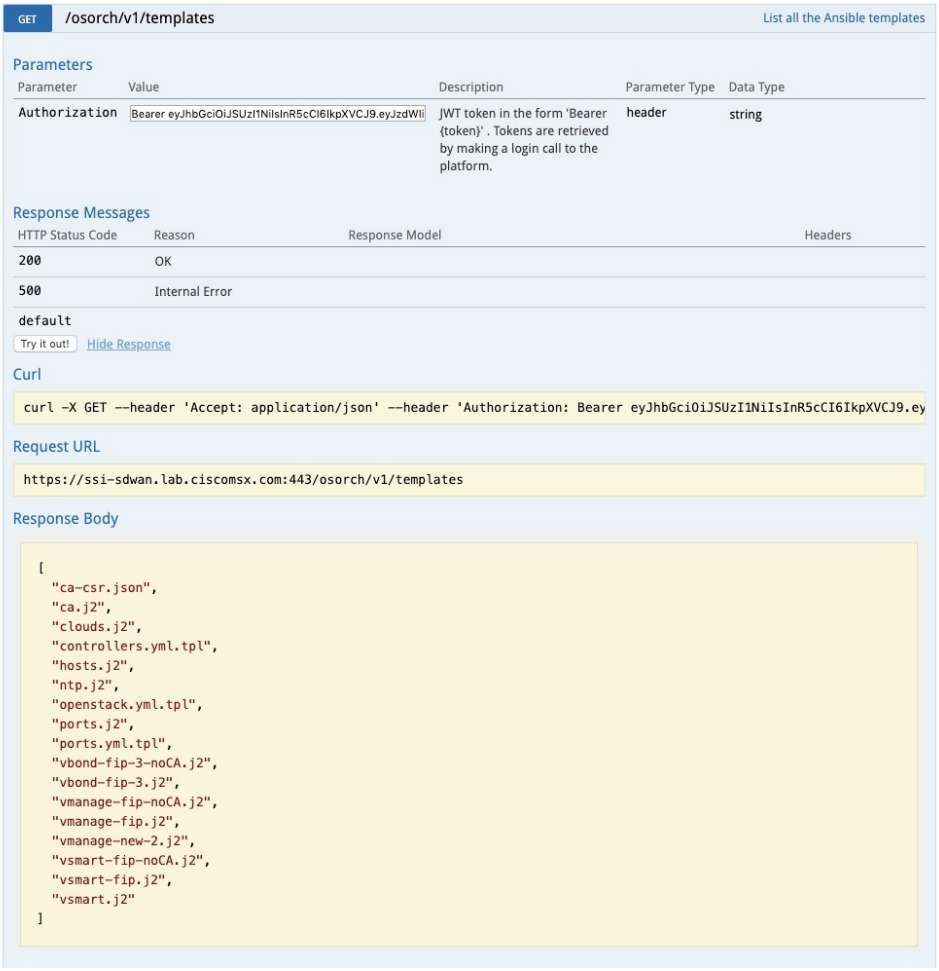
This figure shows the list of Get APIs that can be used to query the database.

Figure 1: List of Get APIs for Quering the Database



This figure shows a sample query to access the list of templates from the OS orchestrator using the curl command in this GET API page

Figure 2: Accessing the List Templates from OS Orchestrator



Troubleshooting the OS orchestrator Logs

To access the OS orchestrator logs:

Procedure

Step 1 Log in to the Kubernetes-master mode.

Step 2 Execute the given command to get the OS orchestrator pod name:

```
kubectl -n vms get po
```

Step 3 To log in to the container, execute the given command:

```
kubectl -n vms exec -it <osorch_log_name> bash
```

Step 4 To check the logs, execute the given command:

```
cd logs >
<jobID>_ansible.log
```

jobID: Specifies the job ID to access the specific job.

If there are errors during the creation of a control plane these logs can offer some guidance, it verifies the incorrect parameters and ways to resolve issues.

Change Control Plane Password or Vault Failures

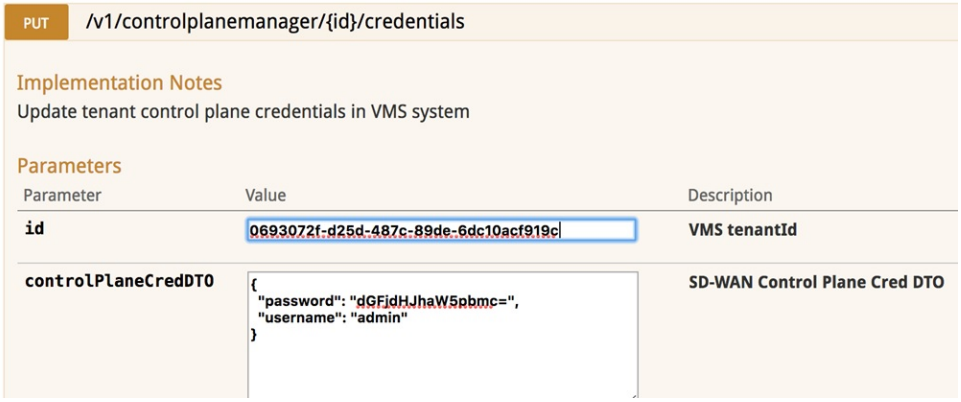
Error Message

Failed to authenticate control plane user.

Solution

Use the Swagger interface to update the credentials for the control plane manager. The password input in base64.

Figure 3: Changing the Control Plane Password

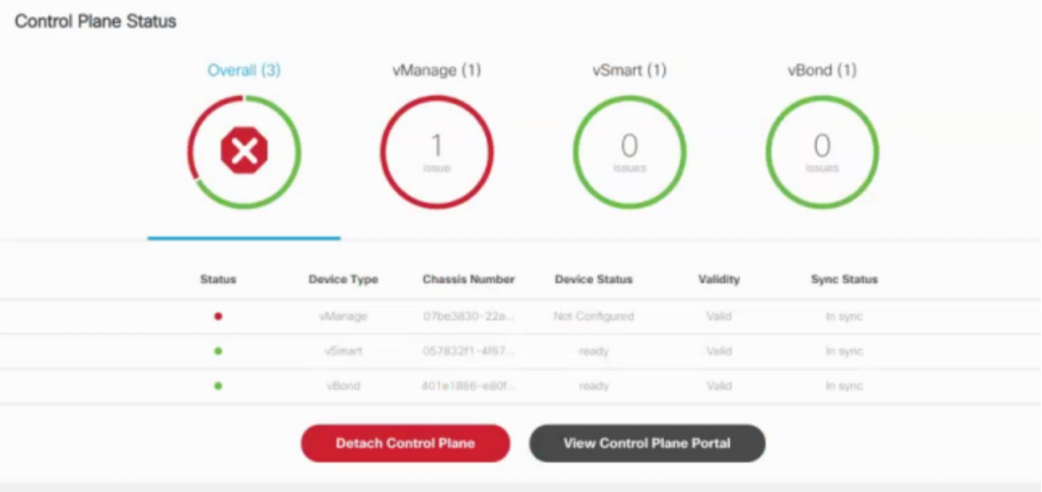


Fixing Control Plane Device Status State

Problem

After adding a new control plane, the Control Plane (vManage) remains in ‘Not Configured’ state, as shown below:

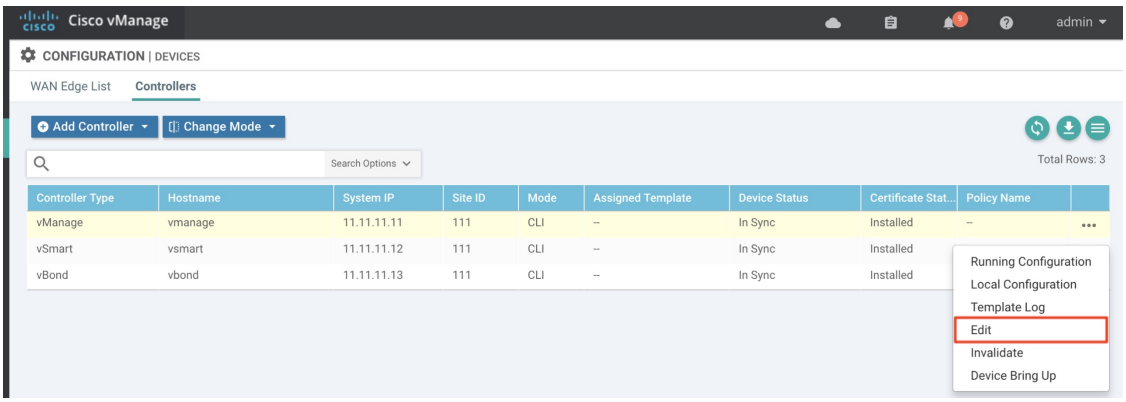
Figure 4: Control Plane Status



Reason

Incorrect way of changing the control plane password. This issue was due to changing the Control Plane password from **vManage** Console > **Configuration** > **Devices** > **Controllers**.

Figure 5: Changing the Control Plane Password from vManage

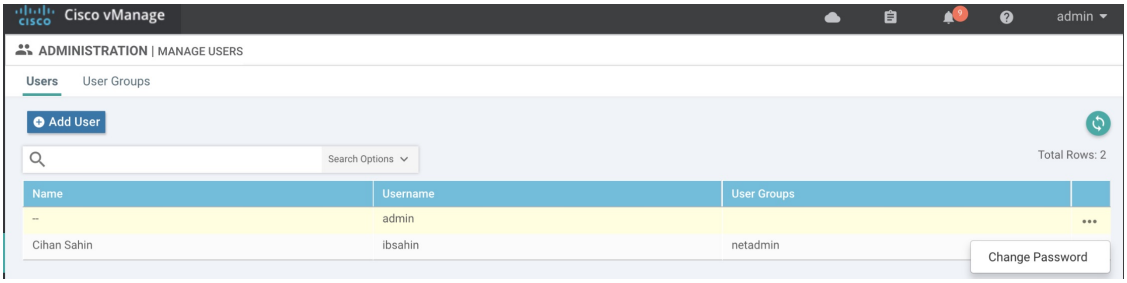


Solution

Procedure

Step 1 Change the Control Plane password from **vManage** Console > **Administration** > **Manage Users** > **Users.**

Figure 6: Changing the Control Plane Password from vManage



Step 2 Restart nms application server on vManage.

```
vmanage# request nms application-server restart
```

Note The server takes a few minutes to start.

Data Plane Troubleshooting

Check the deployment status of the vEdge device:

1. Check NSO device status.
2. Check data plane deployment i4e n the MSX Portal.

Check the reachability status (vEdge to control plane):

1. Check the vManage device state.
2. Check that the site status in SD-WAN is green.

Data Plane Deployment Status: NSO Device Status

Example:

```
vmsnso@ncs> show branch-infra:branch-infra-status branch-cpe agH89ZqVcqVObci6Pyv1jU56 plan component state
NAME                            STATE          STATUS   WHEN                 ref  MESSAGE
-------------------------------------------------------------------------------------------
self                            init           reached  2017-11-12T04:55:12  -
                                ready          reached  2017-11-12T04:57:33  -
agH89ZqVcqVObci6Pyv1jU56        init           reached  2017-11-12T04:55:12  -
                                pnp-callhome   reached  2017-11-12T04:55:12  -
                                ready          reached  2017-11-12T04:56:10  -    Ready
vEdge_SD-Branch-vEdge           init           reached  2017-11-12T04:56:11  -
                                ready          reached  2017-11-12T04:56:34  -    Ready
vEdge_agH89ZqVcqVObci6Pyv1jU56  init           reached  2017-11-12T04:56:35  -
                                vm-deployed    reached  2017-11-12T04:56:50  -
                                vm-alive       reached  2017-11-12T04:57:33  -
                                ready          reached  2017-11-12T04:57:33  -    Ready
```

Data Plane Deployment Status (MSX Portal)

To view the data plane deployment status:

Procedure

Step 1 Log in to the Cisco MSX Portal.
Step 2 In the main menu, click **Dashboard**.
Step 3 Select the tenant from the drop-down.
Step 4 Click **SD-WAN**. The **SD-WAN Service Offer** screen appears.
Step 5 Click **SD-WAN**.
Step 6 Select the SD-WAN service. The **SD-WAN** screen appears.

Data Plane Reachability Status (MSX Portal)

To view the data plane reachability status:

Procedure

Step 1 Log in to the Cisco MSX Portal.
Step 2 In the main menu, click **Dashboard**.
Step 3 Select the tenant from the drop-down.
Step 4 Click **SD-WAN**. The **SD-WAN Service Offer** screen appears.
Step 5 Click **SD-WAN**.
Step 6 Select the SD-WAN service. The **SD-WAN** screen appears.

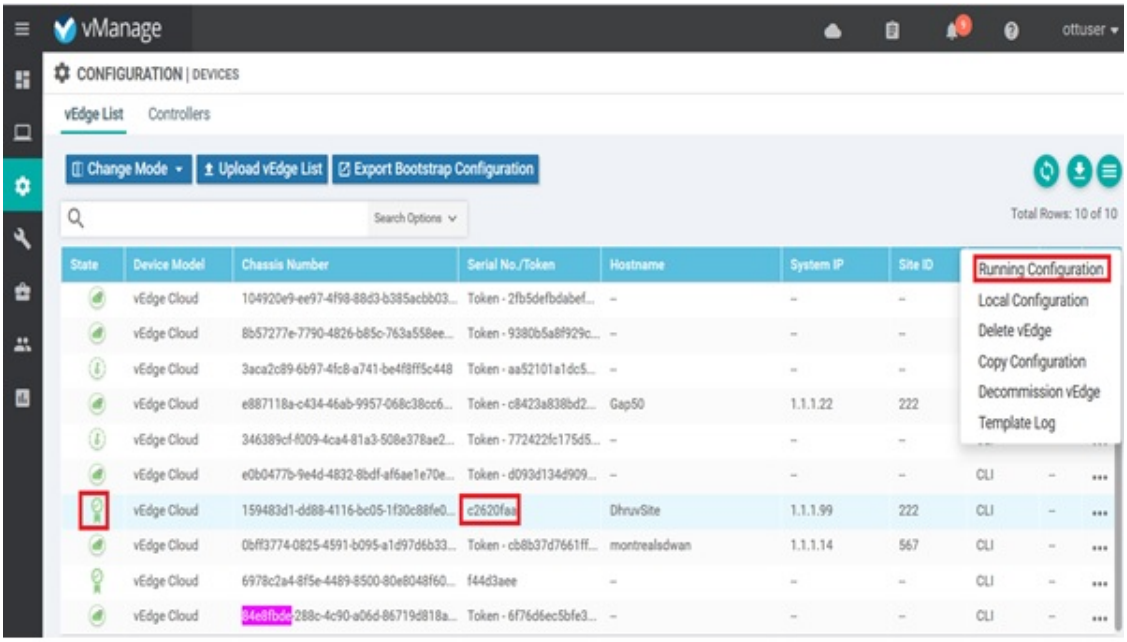
Reachability Status: vManage Device State

To view the reachability status:

Procedure

Step 1 Log in to vManage.
Step 2 In vManage, choose **Configuration > Devices**.The Configure | Devices window is displayed.

Figure 7: Reachability Status of the Devices



- The **State** column indicates if the certificate is installed.
- In the **Serial No./Token** column, you should have a serial number and not a token.
- Click **Running Configuration** to view the configuration running on the device.
- If reachability is not achieved, then verify the variables that were passed during the site deployment (especially VPN_0).

PnP Server Troubleshooting Commands

List of Devices in Contact with the PnP Server

```
admin@ncs-sm-SD-Branch> show pnp list
SERIAL        IP ADDRESS       CONFIGURED  ADDED  SYNCED  LAST CONTACT
-------------------------------------------------------------------------------
FTX1738AJME   173.36.207.85    true        true   true    2017-10-24 23:44:44
FTX1738AJMG   173.36.207.81    true        true   true    2017-10-24 23:43:50
FTX1740ALBX   173.36.207.80    true        true   true    2017-10-24 23:44:21
SSI184904LG   173.36.207.82    true        true   true    2017-10-24 23:43:56
SSI185104LT   173.36.207.84    true        true   true    2016-10-24 23:43:57
[ok][2016-10-24 23:45:49]
```

Note

- The Last Contact column displays the last date and time when the PnP server was in contact with the CPE. If the CPE has not been in recent contact with the PnP server, it may be due to connectivity or reachability issues between the PnP server and the CPE.
- Identify the active NSO instance using the following command:

```
curl -v http://consul.service.consul:8500/v1/catalog/service/nso-ha | python -m json.tool
```

CPE in Contact with the PnP Server (Without a Service)

```
admin@ncs-sm-SD-Branch> show branch-infra:branch-infra branch-cpe
%No entries found
[ok][2016-10-24 23:45:49]
```

CPE in Contact with the PnP Server (With a Service)

```
admin@ncs-sm-SD-Branch> show branch-infra:branch-infra-status branch-cpe amXqvXDO9zW2IZleho2cOBrD plan component
 state status
NAME                       STATE         STATUS
-------------------------------------------------
self                       init          reached
                           ready         reached
amXqvXDO9zW2IZleho2cOBrD   init          reached
                           pnp-callhome  reached
                           ready         reached
[ok][2017-10-25 14:20:40]
```

CPE in Contact with the PnP Server (Detailed)

```
vmsnso@ncs> show branch-infra:branch-infra-status branch-cpe amXqvXDO9zW2IZleho2cOBrD plan component

plan component self
 type      self
 state init
  status  reached
  when    2017-10-25T14:15:20
  message ""
 state ready
  status  reached
  when    2017-10-25T14:16:57
  message ""
 real-name amXqvXDO9zW2IZleho2cOBrD
plan component amXqvXDO9zW2IZleho2cOBrD
 type      branch-cpe
 state init
  status  reached
  when    2017-10-25T14:15:20
  message ""
 state pnp-callhome
  status  reached
  when    2017-10-25T14:16:22
  message ""
 state ready
  status  reached
  when    2017-10-25T14:16:57
```

```
  message Ready
 real-name amXqvXDO9zW2IZleho2cOBrD
 provider  CiscoSystems
 device    amXqvXDO9zW2IZleho2cOBrD_ENCS
[ok][2017-10-25 14:23:10]
```

View CPE Details

```
vmsnso@ncs> show pnp list

SERIAL       IP ADDRESS    CONFIGURED  ADDED  SYNCED  LAST CONTACT
-------------------------------------------------------------------------
FGL21388017  10.85.189.20  true        true   true    2017-10-25 14:24:07
FGL2138801A  10.85.189.23  false       false  false   2017-10-25 14:21:16
FGL2138801E  10.85.189.24  false       false  false   2017-10-25 14:21:40

[ok][2017-10-25 14:24:20]
vmsnso@ncs> configure

Entering configuration mode private
[ok][2017-10-25 14:24:31]

[edit]
vmsnso@ncs% show branch-infra:branch-infra branch-cpe serial FGL21388017
branch-cpe amXqvXDO9zW2IZleho2cOBrD {
    provider CiscoSystems;
    type     ENCS;
    serial   FGL21388017;
    var SD-Branch_DEVICE_TYPE {
        val ENCS;
    }
    var contact {
        val Customer;
    }
    var email {
        val abc@example.ocm;
    }
    var phone {
        val null;
    }
}
[ok][2017-10-25 14:24:34]
[edit]
vmsnso@ncs% exit
[ok][2017-10-25 14:24:53]
```

IPsec Tunnel Cannot be Established

Problem

Device fails to establish secure VPN tunnel between NFVIS and CSR Mgmt hub router.

Solution

To establish secure VPN tunnel:

Procedure

Step 1 Log in to the device and run the following command:

Example:

```
vmsnso@ncs% show pnp day0-common | display set
```

Step 2 Ensure day0-common has correct values specified for the following parameters:

Example:

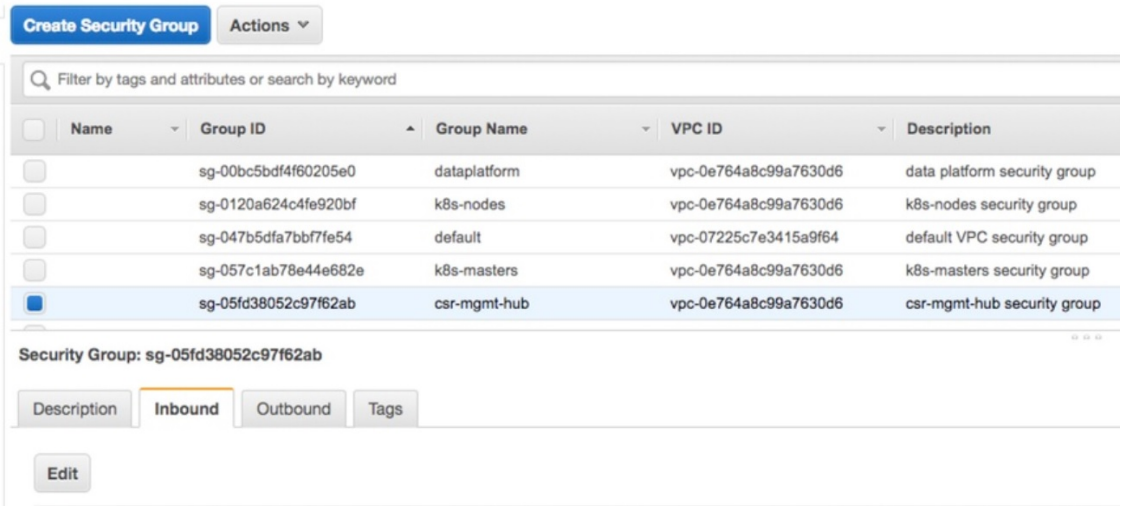
```
INT_MGMT_SUBNET_DHCP set to false
INT_MGMT_SUBNET_GW <GW IP address>
INT_MGMT_SUBNET_INVERSE_MASK <subnet mask>
INT_MGMT_SUBNET_IP <subnet value>
INT_MGMT_SUBNET_IPVERSION <IPv4 or ipv6>
INT_MGMT_SUBNET_NETMASK <subnet netmask value>
LOCAL_PRESHARED_KEY <Pre-shared key for VPN authentication on the ENCS>
MGMTHUB_OVERLAY_IP_ADDR <IP address of the interface connecting to MSX internal host on the CSR mgmt
 hub>
MGMTHUB_OVERLAY_NET <Supernet of the subnets on MSX side>
MGMT_HUB_IP <Public IP address of the CSR mgmt hub>
REMOTE_ID <Local identity configured on the CSR mgmt hub's VPN configuration>
REMOTE_PRESHARED_KEY <Pre-share key for VPN authentication on the CSR mgmt hub>
SECURE_OVERLAY_NAME
SOURCE-BRIDGE
```

Step 3 Ensure CSR Hub VPN configuration matches NFVIS's.

Step 4 Edit CSR Hub's Security group and ensure the following ports are open. The following ports are used for communication from MSX to the NFVIS via the CSR mgmt hub VPN.

- 22022 - 22024: VNF ports
- 21045: VNF port
- 830: Netconf port
- 443: Metric collection from the ENCS

Figure 8: Editing the Security Group



Step 5 Configure route to NFVIS's secure IP. The default is 10.128.0.0/16 (Assigned by SD-Branch). This can be changed using:

- SD-Branch_variables.yml file at the install time
- SD-Branch settings API

Troubleshooting Cisco Meraki SD-WAN Issues

This section describes problems, possible causes, and recommended actions that you may encounter in Cisco Meraki SD-WAN deployments.

Handling Meraki Rate Limiting Issue on MSX

Error Message

Http Error 429: This error code indicates that you have submitted more than 5 API calls in one second to the Meraki system, triggering the rate limit. Meraki Dashboard API has a rate-limiting of 5 API calls per second per organization.

Solution

Modify the rate-limiting parameters through consul. Before you modify, make sure you have API access keys for Meraki. For more information, see Managing Meraki Traffic Class Access for Tenants.

The following are the parameters that need to be modified:

- **meraki.ratelimit.max.attempts**: Use this parameters to configure the maximum attempts allowed for Meraki. Use the following command to configure this parameter. Run this curl command on the master Kubernetes node.

```
curl -s -k -X 'x-consul-token: <consul token>' PUT -H "Content-Type: application/json" -d "<Number of
retries>"
https://consul.service.consul:8500/v1/kv/userviceconfiguration/sdwanservice/meraki.ratelimit.max.attempts
 | python -mjson.tool
```

- **meraki.ratelimit.backoff**: Use this parameter to retry the Meraki request after a specified backoff time. Use the following command to configure this parameter. Run this curl command on the master Kubernetes node.

```
curl -s -k -X 'x-consul-token<consul-token>' PUT -H "Content-Type: application/json" -d "<Backoff in
milliseconds>"
https://consul.service.consul:8500/v1/kv/userviceconfiguration/sdwanservice/meraki.ratelimit.backoff |
python -mjson.tool
```

Note Replace {consul-token} with your actual consul token value from the passwords.yml file.

Checking Meraki Beat

Use the following endpoint to check if Meraki beat is up and running:

```
<IP_address>3441/admin/health

Where:

 <IP_address> is the IP address of the Meraki system.

Response:
{"description":" metrics collector","status":"UP"}description
```

If status is not **Up**,use the instructions in the deployment log to bring Meraki Beat up and running in kubernetes node. Check deployment logs for more information.

Checking Device Status

Checking Meraki Device Health

Use the following POST API to return all the query templates related to devices.

POST /api/v2/querytemplates/device

Response:


```
{
  "success": true,
  "responseObject": [
    {
      "type": "*",
      "profile": "*",
      "specificType": "*",
      "templateName": "query-ping-availability",
      "queryTemplate": "{\"query\": {\"bool\": {\"filter\": [{\"term\": {\"deviceId\": {\"value\":
\"{{deviceId}}\"}}}, {\"range\": {\"@timestamp\": {\"gte\": \"{{timestamp_gte}}\", \"lte\":
\"{{timestamp_lte}}\"}}}]}}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\",
 \"interval\": \"{{statusInterval}}\"}, \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"tot\":
 \"_count\", \"success\": \"count_success\"}, \"script\": \"def threshold = 0.5; if (params.success / params.tot
 > threshold) {return 1} else {return 0}\"}}, \"count_success\": {\"sum\": {\"script\": \"return doc['up'].value
 == true ? 1 : 0\"}}}}}, \"size\": 0}",
      "indices": [
        "heartbeat-*"
      ]
    },
    {
      "type": "*",
      "profile": "*",
      "specificType": "*",
      "templateName": "query-snmp-availability",
      "queryTemplate": "{\"query\": {\"bool\": {\"filter\": [{\"term\": {\"deviceId\": {\"value\":
\"{{deviceId}}\"}}}, {\"range\": {\"@timestamp\": {\"gte\": \"{{timestamp_gte}}\", \"lte\":
\"{{timestamp_lte}}\"}}}]}}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\",
 \"interval\": \"{{statusInterval}}\"}, \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"fail\":
 \"count_fail\", \"tot\": \"_count\"}, \"script\": \"def threshold = 0.5; if ((params.tot - params.fail) /
params.tot > threshold) {return 1} else {return 0}\"}}, \"count_fail\": {\"value_count\": {\"field\":
\"Failed\"}}}}}, \"size\": 0}",
      "indices": [
        "snmpbeat-*"
      ]
    },
    {
      "type": "CPE",
      "profile": "sdwan",
      "specificType": "MERAKI",
      "templateName": "meraki-device-status",
     "queryTemplate": "{\"sort\": [{\"@timestamp\": {\"order\": \"desc\"}}], \"query\": {\"bool\": {\"filter\":
 [{\"term\": {\"deviceId\": {\"value\": \"{{deviceId}}\"}}}, {\"range\": {\"@timestamp\": {\"gte\":
\"{{timestamp_gte}}\", \"lte\": \"{{timestamp_lte}}\"}}}, {\"bool\": {\"must\": [{\"exists\": {\"field\":
\"DeviceHealth\"}}]}}]}}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\",
 \"interval\": \"{{statusInterval}}\"}, \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"total\":
 \"_count\", \"deviceHealthSum\": \"deviceHealthSum\"}, \"gap_policy\": \"insert_zeros\", \"script\":
\"(params.deviceHealthSum / params.total >= 0.5) ? 1 : 0\"}}, \"deviceHealthSum\": {\"sum\": {\"script\":
\"return doc['DeviceHealth.status'].value == 'offline' ? 0 : doc['DeviceHealth.status'].value == 'online' ? 1
 : 'undefined'\"}}}}}, \"size\": 1}",
      "indices": [
        "merakibeat-*"
```

```
      ]
    }
  ],
  "command": "Get all device health query templates",
  "parms": {},
  "httpStatus": "OK",
  "message": "Get all device health query templates",
  "errors": [],
  "throwable": null
}
```


Problem: If meraki-device-status does not exist in the response, it means there were issues in deployment and query templates were not pushed properly.

Solution: Use the following API:

POST api/v2/querytemplates/device


```
Response:
{
  "type": "CPE",
  "profile": "sdwan",
  "specificType": "MERAKI",
  "templateName": "meraki-device-status",
  "queryTemplate": "{\"sort\": [{\"@timestamp\": {\"order\": \"desc\"}}], \"query\": {\"bool\": {\"filter\":
[{\"term\": {\"deviceId\": {\"value\": \"{{deviceId}}\"}}}, {\"range\": {\"@timestamp\": {\"gte\":
\"{{timestamp_gte}}\", \"lte\": \"{{timestamp_lte}}\"}}}, {\"bool\": {\"must\": [{\"exists\": {\"field\":
\"DeviceHealth\"}}]}}]}}, \"aggs\": {\"{{statusInterval}}\": {\"date_histogram\": {\"field\": \"@timestamp\",
 \"interval\": \"{{statusInterval}}\"}, \"aggs\": {\"status\": {\"bucket_script\": {\"buckets_path\": {\"total\":
 \"_count\", \"deviceHealthSum\": \"deviceHealthSum\"}, \"gap_policy\": \"insert_zeros\", \"script\":
\"(params.deviceHealthSum / params.total >= 0.5) ? 1 : 0\"}}, \"deviceHealthSum\": {\"sum\": {\"script\":
\"return doc['DeviceHealth.status'].value == 'offline' ? 0 : doc['DeviceHealth.status'].value == 'online' ? 1
 : 'undefined'\"}}}}}, \"size\": 1}",
  "indices": [
    "merakibeat-*"
  ]
}
```


Checking Device Status for a Specific Service ID

Use the following end point to get the statuses of the devices for a specific service ID:

GET /api/v1/status/service/{serviceId}/devices


```
Response:
{
    "statusData": [{
            "id": "<Device ID 1>",
            "locationId": null,
            "name": "127.0.0.1",
            "operationalState": "up",
            "parentId": null,
            "topLevelServiceId": "<Service ID>",
            "type": "CPE"
        },
        {
            "id": "<Device ID 2>",
            "locationId": null,
            "name": "127.0.0.1",
            "operationalState": "up",
            "parentId": null,
            "topLevelServiceId": "<Service ID>",
            "type": "CPE"
        },
```

```
        {
            "id": "<Device ID 3>",
            "locationId": null,
            "name": "127.0.0.1",
            "operationalState": "down",
            "parentId": null,
            "topLevelServiceId": "<Service ID>",
            "type": "CPE"
        }
    ]
}
```

Problem: Device ID is returned empty, which indicates issue in metric collection.

Solution: In this case, run the following query for each device on Elastic Search:

```
{
  "query": {
    "bool": {
      "filter": [
        {
          "term": {
            "deviceId": {
              "value": "{{deviceId}}"
            }
          }
        },
        {
          "range": {
            "@timestamp": {
              "gte": "{{timestamp_gte}}",
              "lte": "{{timestamp_lte}}"
            }
          }
        },
        {
          "bool": {
            "must": [
              {
                "exists": {
                  "field": "DeviceHealth"
                }
              }
            ]
          }
        }
      ]
    }
  },
  "sort": [
    {
      "@timestamp": {
        "order": "desc"
      }
    }
  ],
  "size": 1,
  "aggs": {
    "{{statusInterval}}": {
      "date_histogram": {
        "field": "@timestamp",
        "interval": "{{statusInterval}}"
      },
      "aggs": {
        "deviceHealthSum": {
```

```
          "sum": {
           "script": "return doc['DeviceHealth.status'].value == 'offline' ? 0 : doc['DeviceHealth.status'].value
 == 'online' ? 1 : 'undefined'"
          }
        },
        "status": {
          "bucket_script": {
            "buckets_path": {
              "deviceHealthSum": "deviceHealthSum",
              "total": "_count"
            },
            "gap_policy": "insert_zeros",
            "script": "(params.deviceHealthSum / params.total >= 0.5) ? 1 0"
          }
        }
      }
    }
  }
} :[..
```

In the above query, provide the values for the following:

- timestamp_gte
- timestamp_lte
- statusInterval

```
Response:
"": {
            "": 1
          }
0 Down
1 Up
```

Checking Device Connections

Use the following endpoint to get all device connections:

```
GET /manage/api/v2/devices/connections

Response:
{
  "success": true,
  "command": "getAllDeviceConnections",
  "params": {
    "serviceInstanceId": null
  },
  "message": "getAllDeviceConnections succeeded",
  "responseObject": [
    {
      "deviceInstanceId": "<Device ID1>",
      "serviceInstanceId": "<Service ID1>",
      "tenantId": null,
      "name": null,
      "profile": "sdwan",
      "type": "CPE",
      "specificType": "MERAKI",
      "category": "CPE",
      "hostName": null,
      "ipAddress": "127.0.0.1",
      "serialKey": "<Device Serial Key>",
      "createdOn": "2020-02-18T10:33:44.849412",
```

```
      "createdBy": "operatorapi181027585e8c87a9",
      "modifiedOn": "2020-02-18T10:33:44.849412",
      "modifiedBy": "operatorapi181027585e8c87a9"
    },
    {
      "deviceInstanceId": "<Device ID2>",
      "serviceInstanceId": "Service ID2",
      "tenantId": null,
      "name": "aw9DpjAZShwiRwUrNbuMZtji",
      "profile": "vbranch",
      "type": "CPE",
      "specificType": "ENCS",
      "category": "CPE",
      "hostName": "aw9DpjAZShwiRwUrNbuMZtji",
      "ipAddress": "127.0.0.1",
      "serialKey": "<Device Serial Key>",
      "createdOn": "2020-02-18T10:34:17.223363",
      "createdBy": "system",
      "modifiedOn": "2020-02-18T10:34:17.223363",
      "modifiedBy": "system"
    },
  "httpStatus": "OK"
}
```


Make sure the device connection has been created for the device with the serial number.

Problem: Device connections are not returned in the response.

Solution: Create a device connection using the following API :

POST /api/v2/devices/connections

Response:


```
{
  "": "CPE",category
  "": "CPE-<UUID>",deviceInstanceId
  "": null,hostName
  "": "127.0.0.1",ipAddress
  "": null,name
  "": " ",profilesdwan
  "": "<Serial_Key>",serialKey
  "": "<Service_instance_ID>",serviceInstanceId
  "": " MERAKI",specificType
  "": " CPE"type
}
```


The above endpoint response returns an entry for the device instance ID.

Applications Available with Cisco MSX SD-WAN

Cisco MSX allows you to set the application relevance for the applications in Cisco SD-WAN and Meraki SD-WAN managed sites. For more information, see Cisco MSX SD-WAN and Meraki Out-of-the box Applications Addendum.

Out-of-the-Box Cisco SD-WAN Device Templates Available Within MSX

Cisco MSX provides out-of-the-box device templates. The details of these templates are provided in the figures below. You can export these templates to your tenants vManage and use them as it is or modify them as per your requirements.

Note After the upgrade, the Cisco MSX SD-WAN out-of-the-box device templates assigned to your tenants will continue to work. However, if you are assigning templates to the tenants, use the new out-of-the-box templates (template names with V02 or V03 or V04). If you assign old templates, the system will show an error indicating that these are outdated OOB templates.

Figure 9: Out-of-the-Box Device Templates

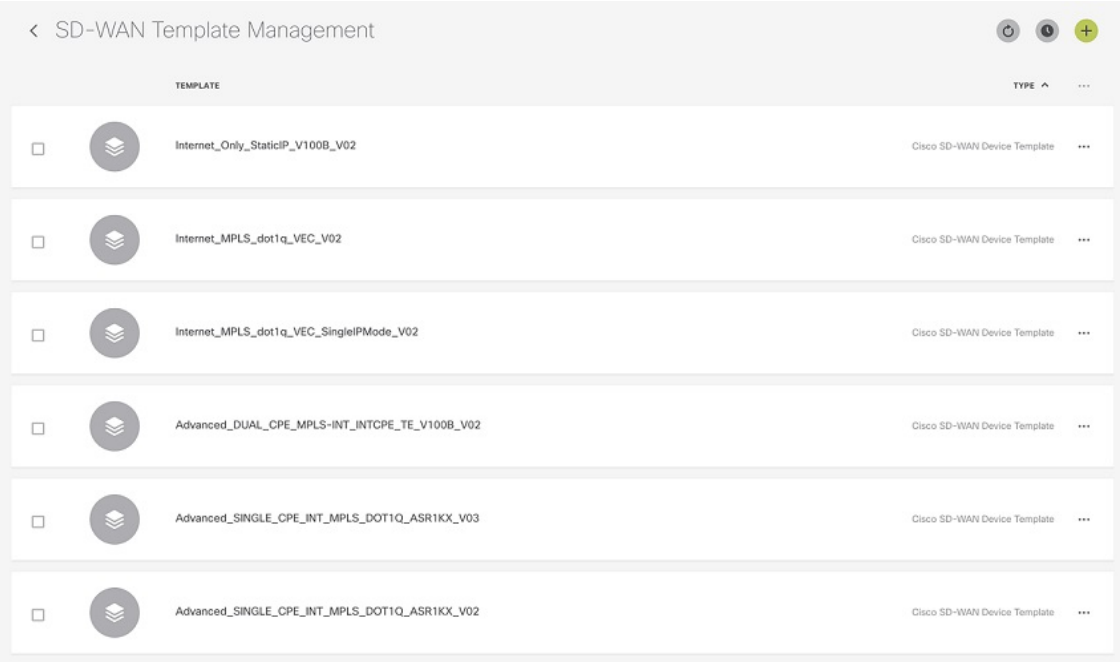


Figure 10: Advanced Dual CPE MPLS INT INTCPE

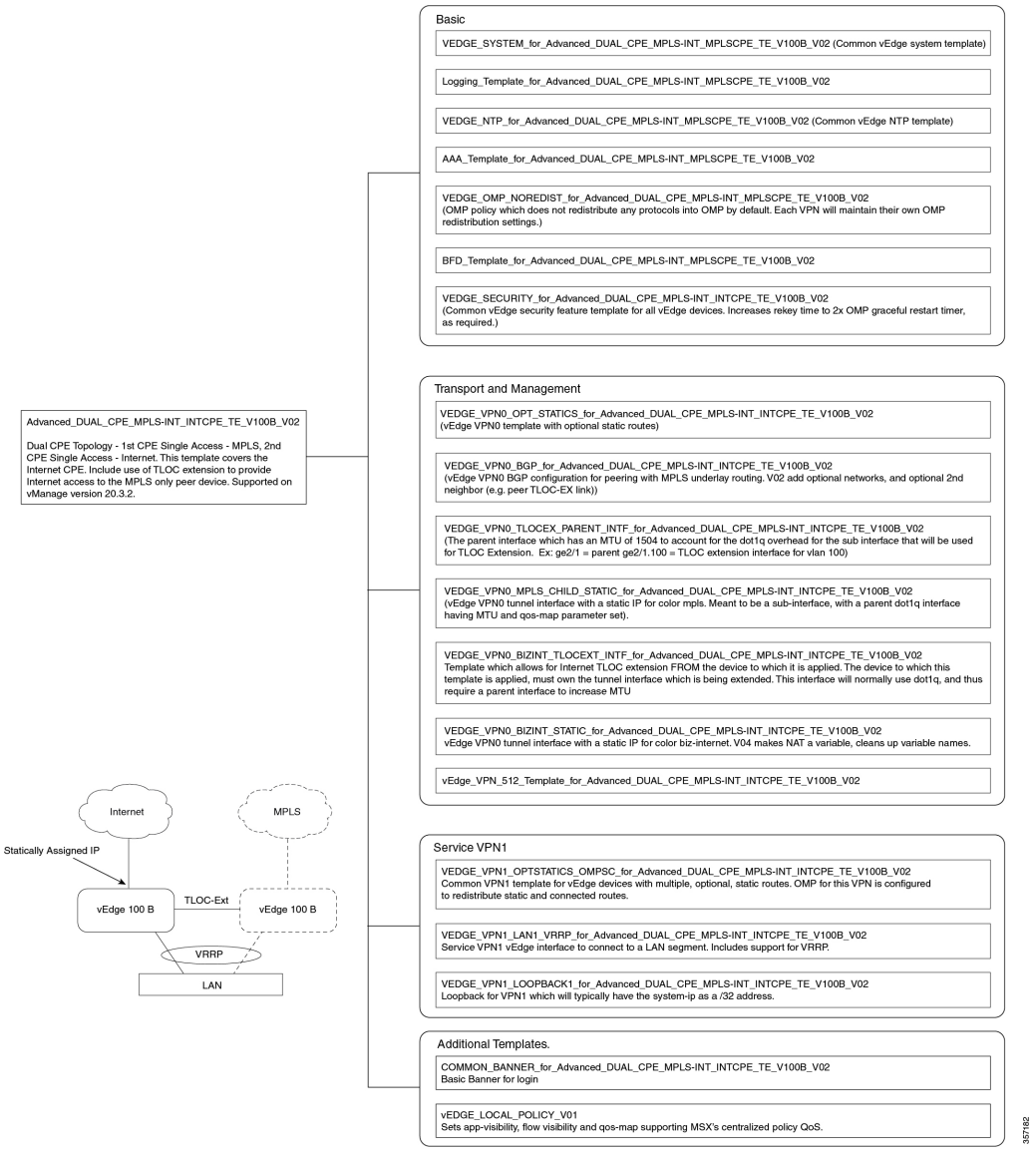


Figure 11: Internet Only StaticIP

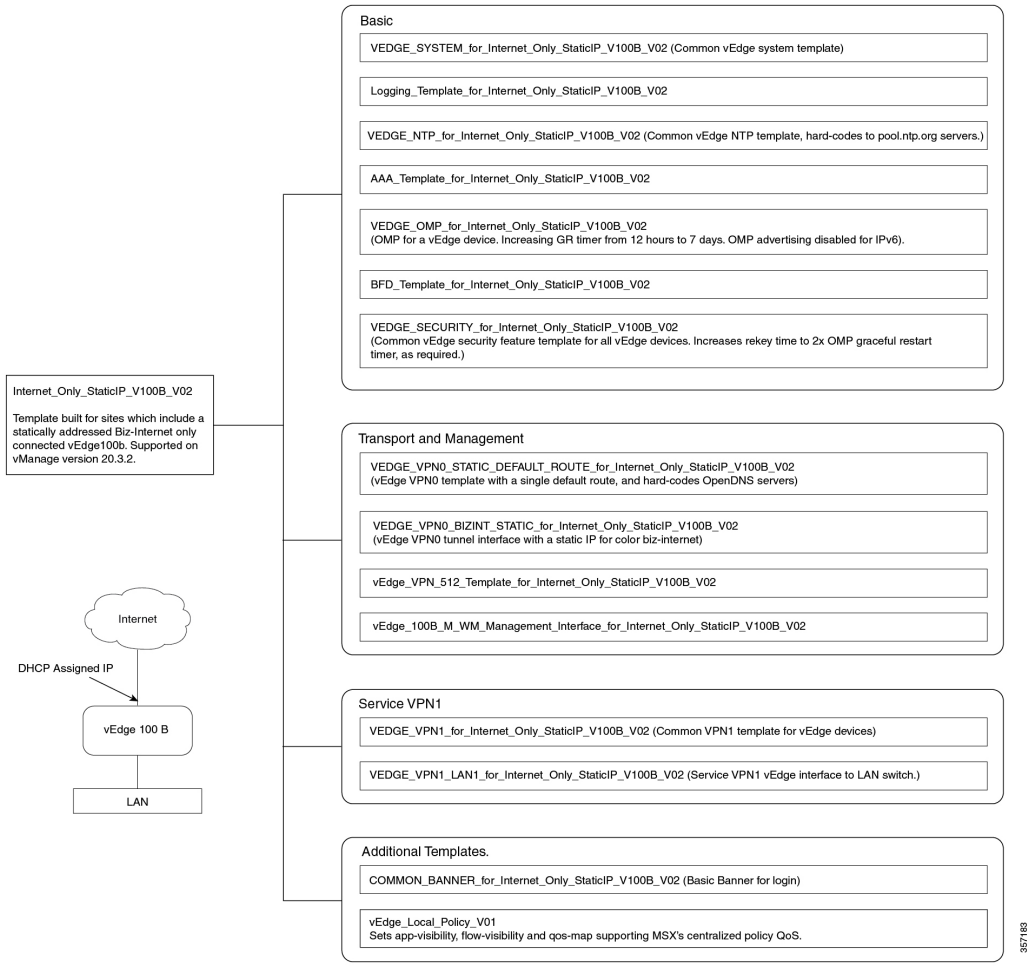


Figure 12: Internet Only DHCP

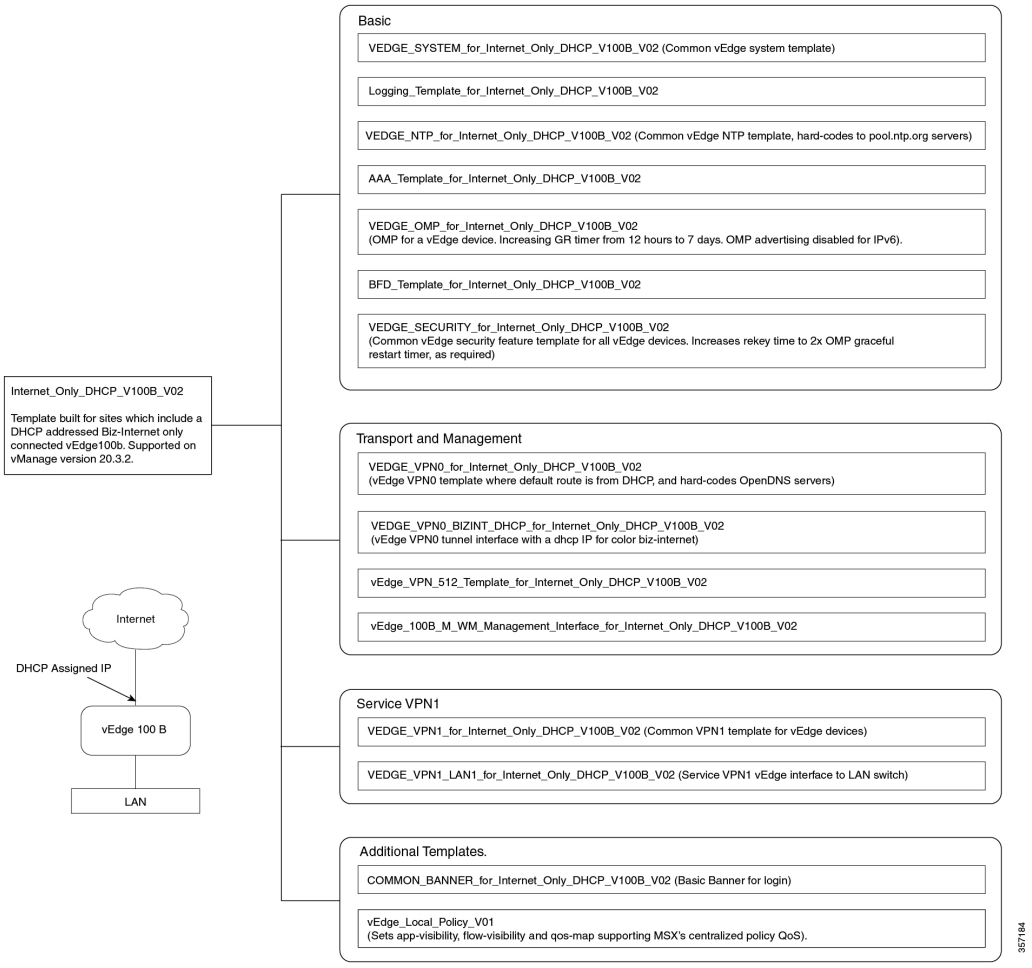


Figure 13: Internet MPLS

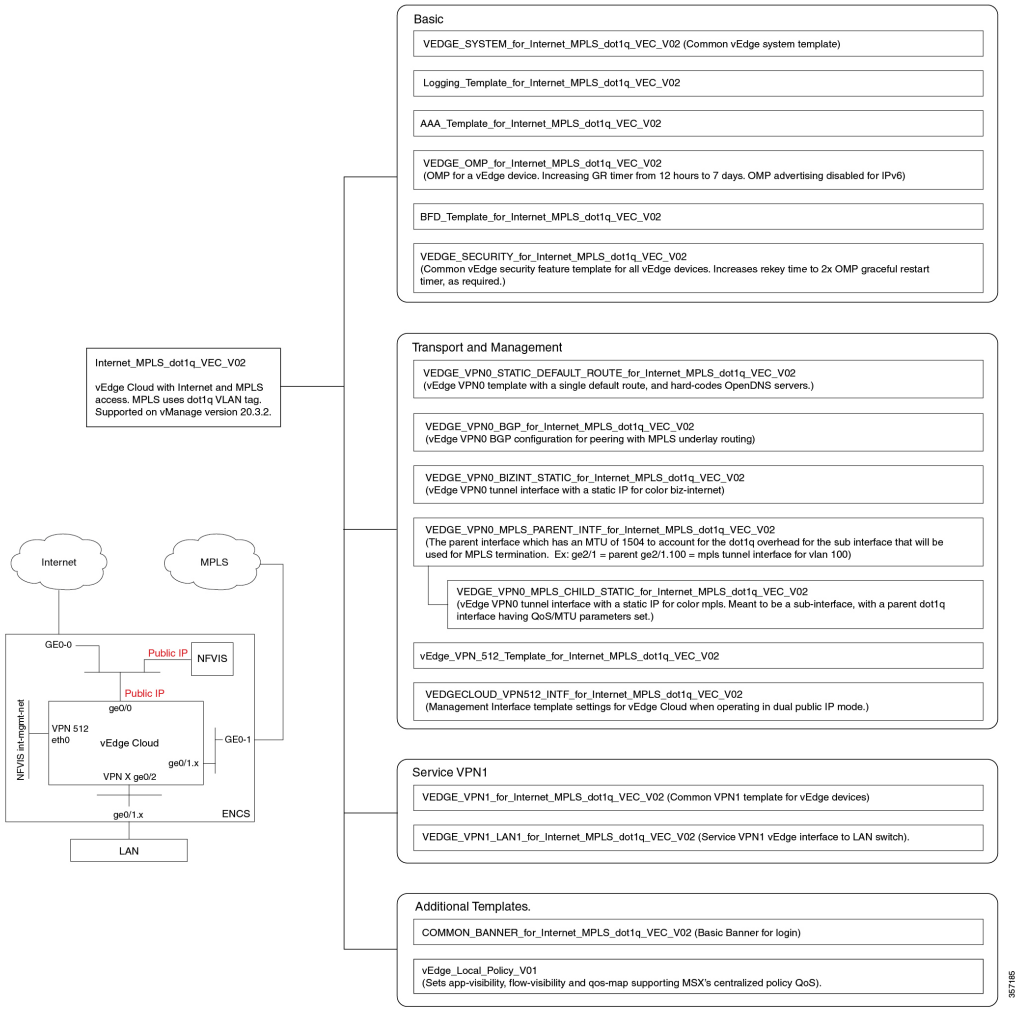


Figure 14: Internet MPLS VEC Single IP Mode

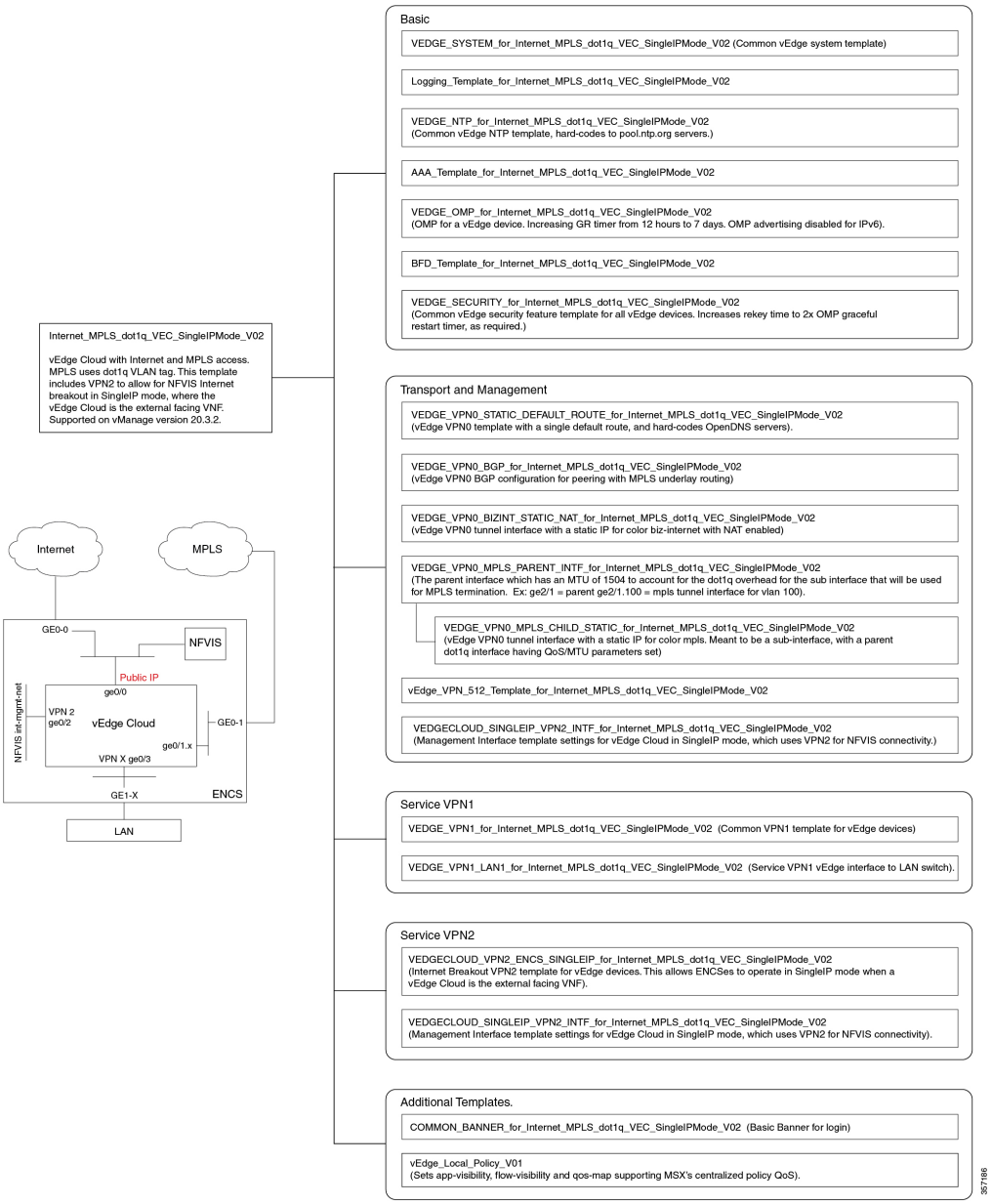


Figure 15: Advanced Single CPE INT MPLS DOT1Q ASR1KX

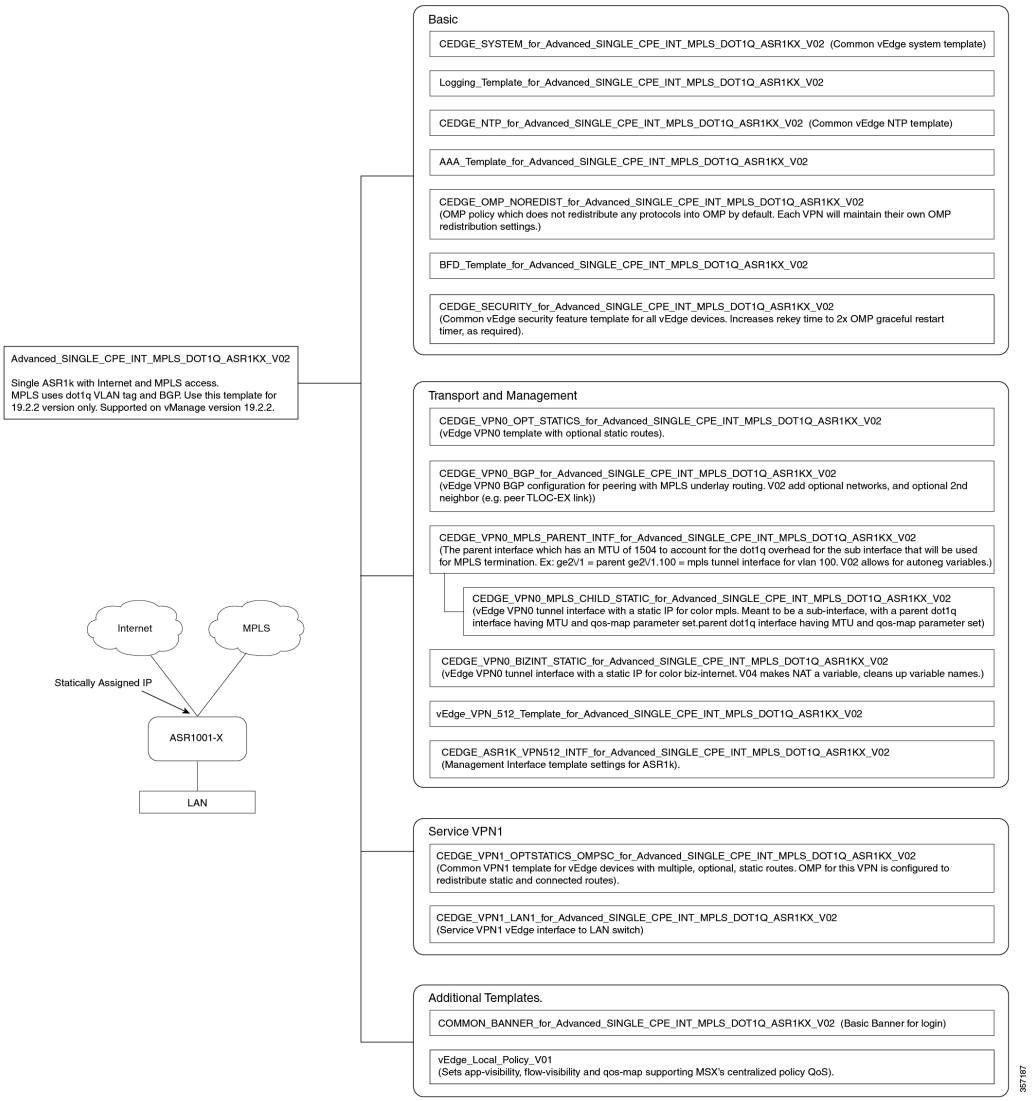


Figure 16: Actual Dual CPE MPLS-INTCPE_TE

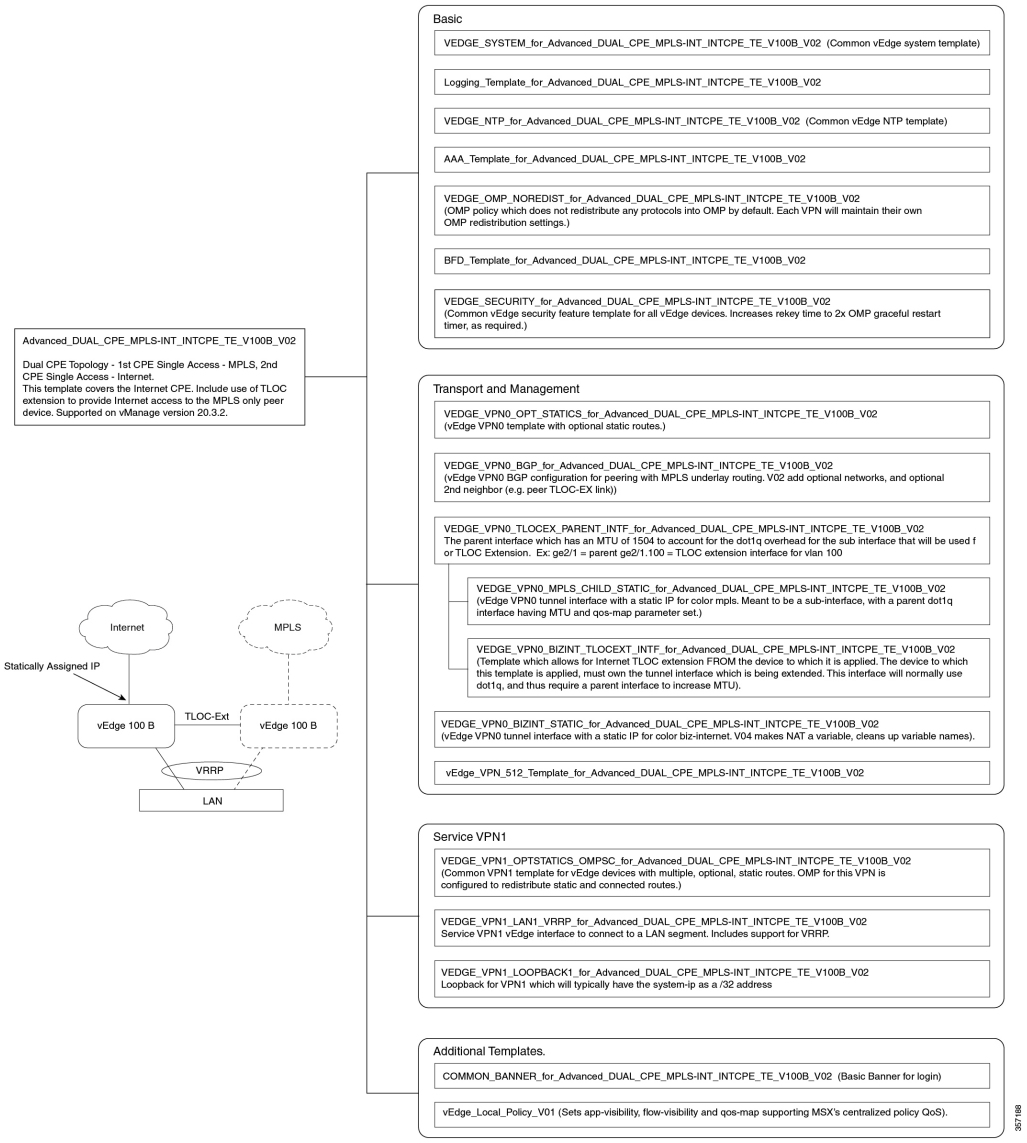


Figure 17: Advanced_SINGLE_CPE_INT_MPLS_DOT1Q_ASR1KX_V03

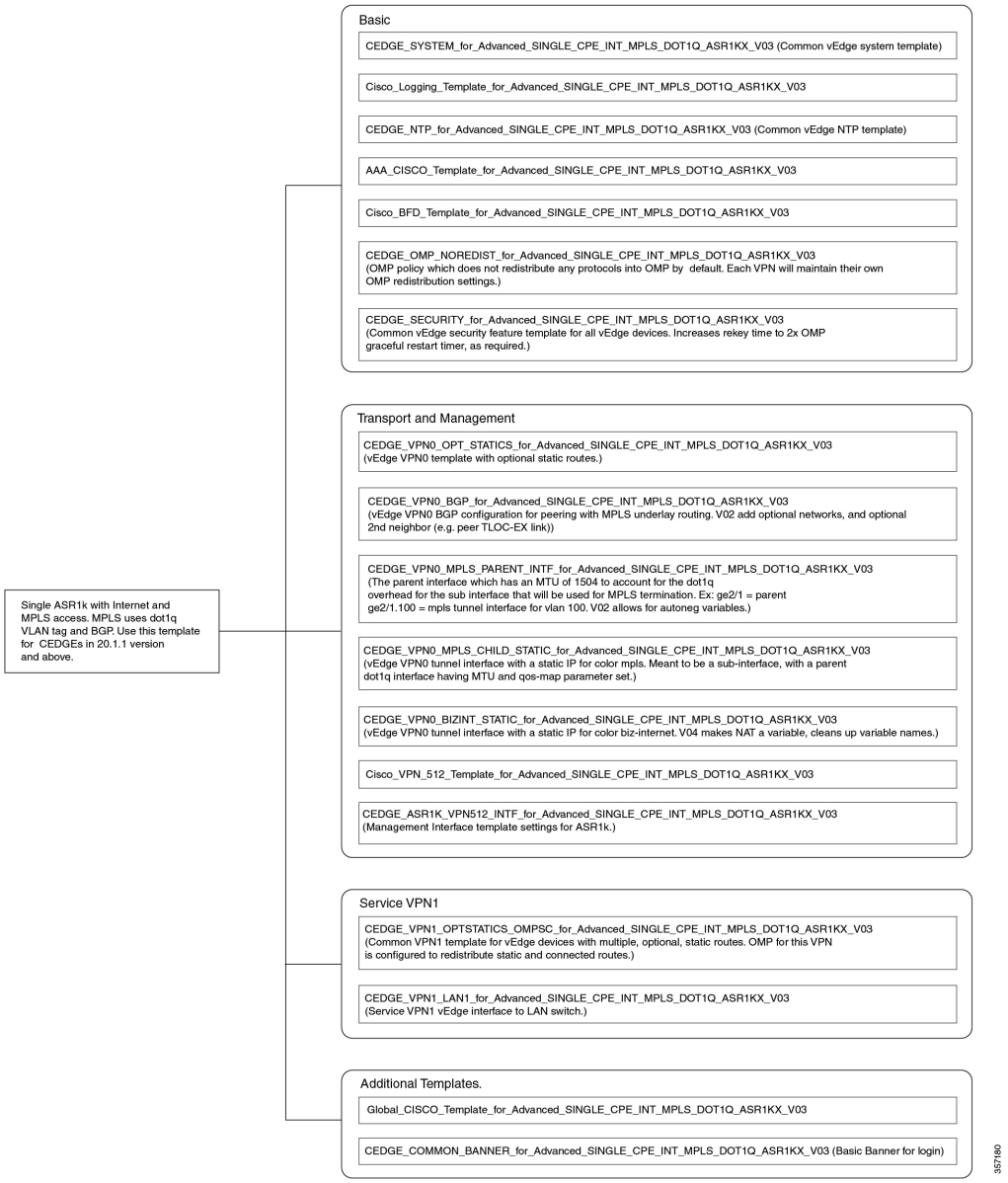
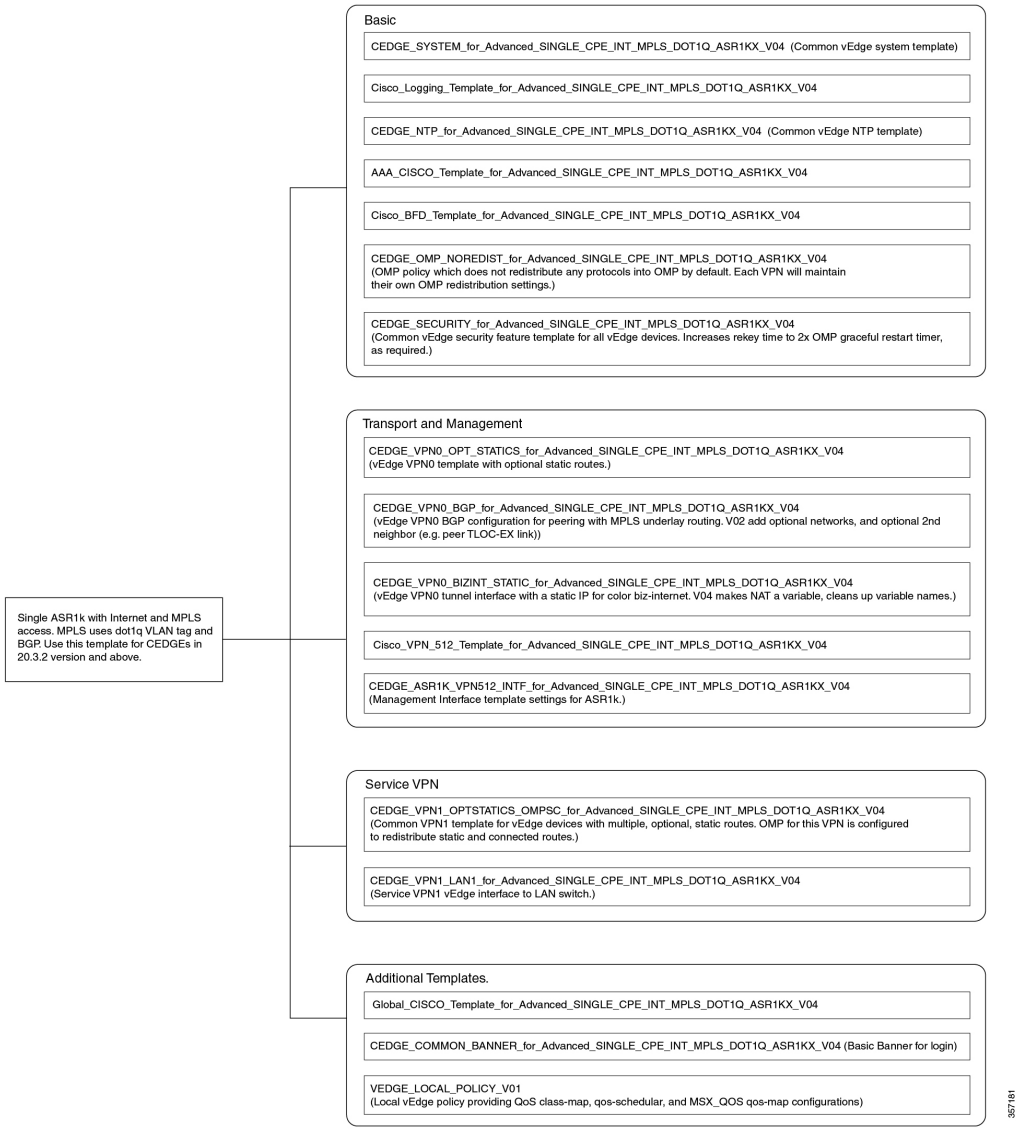


Figure 18: Advanced_SINGLE_CPE_INT_MPLS_DOT1Q_ASR1KX_V04



Sample Payloads for Creating Cisco SD-WAN Control Plane on Openstack

This section contains the sample JSON configuration files for adding VIM and control plane payloads in the provider and tenant network. The authentication certificate is a part of the control plane deployment activity in vManage and sample JSON files are given for both enterprise and symantec certificate based on the network.

Adding VIM Payload in Provider Network

This is the sample JSON file for adding VIM payload in the provider network.

Note:

- Ensure that the names used in "dtlsNetName"(VPN0) and "mgmtNetName"(VPN512) are from the provider network, that is already created in the OpenStack cloud.

- The VPN512 network should be reachable from MSX for the deployment of the control plane.
- The VPN0 network should be reachable from vEdge for the deployment of vEdge on the deployed control plane.

Provider Network.json

```
#Provider
{
"tenantID": "TestTenant",
"vim":
{
"type": "openstack",
"openstack":
{
"username": "username",
"password": "password",
"authURL": "URL",
"projectName": "admin",
"projectID": Project ID,
"domainName": "Default",
"region": "RegionOne",
"extNetName": "external",
"networkType": "provider",
"network":
{
"dtlsNetName": "Vnf-outside",
"mgmtNetName": "external"
}
}
}
}
```

Adding VIM Payload in Tenant Network

This is the sample JSON file for adding VIM payload in tenant network.

Note:

- Ensure that the names used in "dtlsNetName"(VPN0) and "mgmtNetName"(VPN512) are not repeated on the openstack cloud.
- The subnet used in "dtlsSubnet"(VPN0) and "mgmtSubnet"(VPN512) for the 'create control plane' payload should not be repeated on the openstack cloud. You can provide two subnets in this payload.
- The VPN512 network has floating IPs that should be reachable from MSX for the deployment of the control plane.
- The VPN0 network has floating IPs that should be reachable from vEdge for the deployment of vEdge on the deployed control plane.

Tenant Network.json

```
{
"tenantID": "TestTenant",
"vim": {
"type": "openstack",
"openstack": {
"username": "username",
"password": "password",
"authURL": "url",
"projectName": "admin",
```

```
"projectID": "Project ID",
"domainName": "Default",
"region": "RegionOne",
"extNetName": "external",
"networkType": "tenant",
"network": {
"dtlsNetName": "Test-Dtls",
"mgmtNetName": "Test-Mgmt"
}
}
}
}
```

Adding Control Plane Payload with Enterprise Certificate in Provider Network

This is the sample JSON file for adding control plane payload with enterprise certificate in provider network.

Provider Network with EnterpriseCA.json

```
{
"tenantID": "TestTenant",
"controlPlane": {
"vimID": "vimID",
"vmanage": {
"flavor": "viptela-vmanage-vm",
"image": "viptela-vmanage-19.1.0-genericx86-64.qcow2",
"hostname": "TestManage01",
"systemID": "system ID",
"day0": "vmanage-fip.j2",
"vpn0": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"},
"vpn512": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
}
},
"vbond": {
"flavor": "viptela-vbond-vm",
"image": "viptela-edge-19.1.0-genericx86-64.qcow2",
"hostname": "TestBond01",
"systemID": "system ID",
"day0": "vbond-fip-3.j2",
"vpn0": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
},
"vpn512": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
}
},
"vsmart": {
"flavor": "viptela-vsmart-vm",
"image": "viptela-smart-19.1.0-genericx86-64.qcow2",
"hostname": "TestSmart01",
"systemID": "system ID",
"day0": "vsmart-fip.j2",
```

```
"vpn0": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
},
"vpn512": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
}
},
"credentials": {
"username": "username",
"password": "password"
},
"org": "vmsoverlay1",
"siteID": "site ID",
"ntpServer": "ntp.esl.cisco.com",
"dnsServer": "dns serverIP address",
"createCA": true
}
}
```

Adding Control Plane Payload with Symantec Certificate in Provider Network

This is the sample JSON file for adding control plane payload with symantec certificate in provider network.

Provider Network with Symantec.json

```
{
"tenantID": "TestTenant",
"controlPlane": {
"vimID": "vim ID",
"vmanage": {
"flavor": "viptela-vmanage-vm",
"image": "viptela-vmanage-19.1.0-genericx86-64.qcow2",
"hostname": "TestManage01",
"systemID": "system ID",
"day0": "vmanage-fip-noCA.j2",
"vpn0": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
},
"vpn512": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
}
},
"vbond": {
"flavor": "viptela-vbond-vm",
"image": "viptela-edge-19.1.0-genericx86-64.qcow2",
"hostname": "TestBond01",
"systemID": "system ID",
"day0": "vbond-fip-3-noCA.j2",
"vpn0": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
},
"vpn512": {
```

```
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
}
},
"vsmart": {
"flavor": "viptela-vsmart-vm",
"image": "viptela-smart-19.1.0-genericx86-64.qcow2",
"hostname": "TestSmart01",
"systemID": "system ID",
"day0": "vsmart-fip-noCA.j2",
"vpn0": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
},
"vpn512": {
"publicIP": "IP address",
"gateway": "IP address",
"subnetMaskBits": "24"
}
},
"credentials": {
"username": "username",
"password": "password"
},
"org": "vmsoverlay1",
"siteID": "site ID",
"ntpServer": "ntp.esl.cisco.com",
"dnsServer": "IP address",
"createCA": false
}
}
```

Adding Control Plane Payload with Enterprise Certificate in Tenant Network

This is the sample JSON file for adding control plane payload with enterprise certificate in tenant network.

Tenant Network with EnterpriseCA.json

```
{
"tenantID": "TestTenant",
"controlPlane": {
"vimID": "vim ID",
"vmanage": {
"flavor": "viptela-vmanage-vm",
"image": "viptela-vmanage-19.1.0-genericx86-64.qcow2",
"hostname": "TestManage10",
"systemID": "system ID",
"day0": "vmanage-fip.j2",
"vpn0": {
"subnetMaskBits": "24"
},
"vpn512": {
"subnetMaskBits": "24"
}
},
"vbond": {
"flavor": "viptela-vbond-vm",
"image": "viptela-edge-19.1.0-genericx86-64.qcow2",
"hostname": "TestBond10",
"systemID": "50.0.1.11",
```

```
"day0": "vbond-fip-3.j2",
"vpn0": {
"subnetMaskBits": "24"
},
"vpn512": {
"subnetMaskBits": "24"
}
},
"vsmart": {
"flavor": "viptela-vsmart-vm",
"image": "viptela-smart-19.1.0-genericx86-64.qcow2",
"hostname": "TestSmart10",
"systemID": "system ID",
"day0": "vsmart-fip.j2",
"vpn0": {
"subnetMaskBits": "24"
},
"vpn512": {
"subnetMaskBits": "24"
}
},
"credentials": {
"username": "username",
"password": "password"
},
"org": "vmsoverlay1",
"siteID": "site ID",
"ntpServer": "ntp.esl.cisco.com",
"dnsServer": "IP address",
"createCA": true,
"dtlsSubnet": "IP address",
"mgmtSubnet": "IP address"
}
}
```

Adding Control Plane Payload with Symantec Certificate on Tenant Network

This is the sample JSON file for adding control plane payload with symantec certificate on tenant network.

Tenant Network with Symantec.json

```
{
"tenantID": "TestTenant",
"controlPlane": {
"vimID": "vim ID",
"vmanage": {
"flavor": "viptela-vmanage-vm",
"image": "viptela-vmanage-19.1.0-genericx86-64.qcow2",
"hostname": "TestManage10",
"systemID": "system ID",
"day0": "vmanage-fip-noCA.j2",
"vpn0": {
"subnetMaskBits": "24"
},
"vpn512": {
"subnetMaskBits": "24"
}
},
"vbond": {
"flavor": "viptela-vbond-vm",
"image": "viptela-edge-19.1.0-genericx86-64.qcow2",
"hostname": "TestBond10",
```

```
"systemID": "system ID",
"day0": "vbond-fip-3-noCA.j2",
"vpn0": {
"subnetMaskBits": "24"
},
"vpn512": {
"subnetMaskBits": "24"
}
},
"vsmart": {
"flavor": "viptela-vsmart-vm",
"image": "viptela-smart-19.1.0-genericx86-64.qcow2",
"hostname": "TestSmart10",
"systemID": "system ID",
"day0": "vsmart-fip-noCA.j2",
"vpn0": {
"subnetMaskBits": "24"
},
"vpn512": {
"subnetMaskBits": "24"
}
},
"credentials": {
"username": "username",
"password": "password"
},
"org": "vmsoverlay1",
"siteID": "site ID",
"ntpServer": "ntp.esl.cisco.com",
"dnsServer": "IP address",
"createCA": false,
"dtlsSubnet": "IP address",
"mgmtSubnet": "IP address"
}
}
```

Americas Headquarters
Cisco Systems, Inc.
San Jose, CA 95134-1706
USA

Asia Pacific Headquarters
CiscoSystems(USA)Pte.Ltd.
Singapore

Europe Headquarters
CiscoSystemsInternationalBV
Amsterdam,TheNetherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at www.cisco.com/go/offices.