Configure SD-AVC on SD-WAN

Contents

Introduction

This document describes how to configure Software Defined-Application Visibility and Control (SD-AVC) on a Software-Defined Wide Area Network (SD-WAN).

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- SD-WAN
- SD-AVC

The virtual machine of Cisco vManage must have these minimum resources:

- RAM:32 GB
- Storage:500 GB
- vCPU:16

Components Used

The information in this document is based on these software and hardware versions:

- Cisco vManage Release 20.3.x or later.
- vManage Version 20.6.3
- vBond Version 20.6.3

- vSmart Version 20.6.3
- Integrated Service Routers (ISR)4321/K9 Version 17.5.1a

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background

What is SD-AVC?

Cisco SD-AVC is a component of Cisco Application Visibility Control (AVC). AVC incorporates into the routing devices application recognition and performance monitoring capabilities traditionally available as dedicated appliances. It works as a centralized network service and operates with specific devices in the network.

For details, see SD-AVC Features and Benefits.

What is Cisco Cloud Connector?

Cisco Cloud Connector is a Cloud service provided by Cisco that improves traffic classification. It uses the latest information available about the server address used by public Internet sites and services to improve SD-AVC classification of traffic.

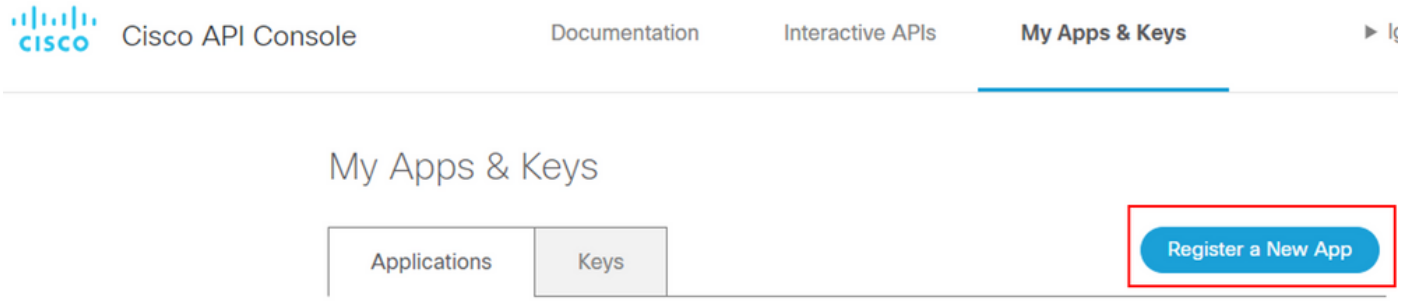
Configure

Enable Cloud Connector

1. Open the Cisco API Console and click **My Apps & Keys**.

> **Note**: The device hosted SD-AVC network requires access to Cisco SD-AVC cloud server domains: **api.cisco.com, cloudsso.cisco.com, prod.sdavc-cloud-api.com.**

2. Click **Register a New App** as shown in the image.



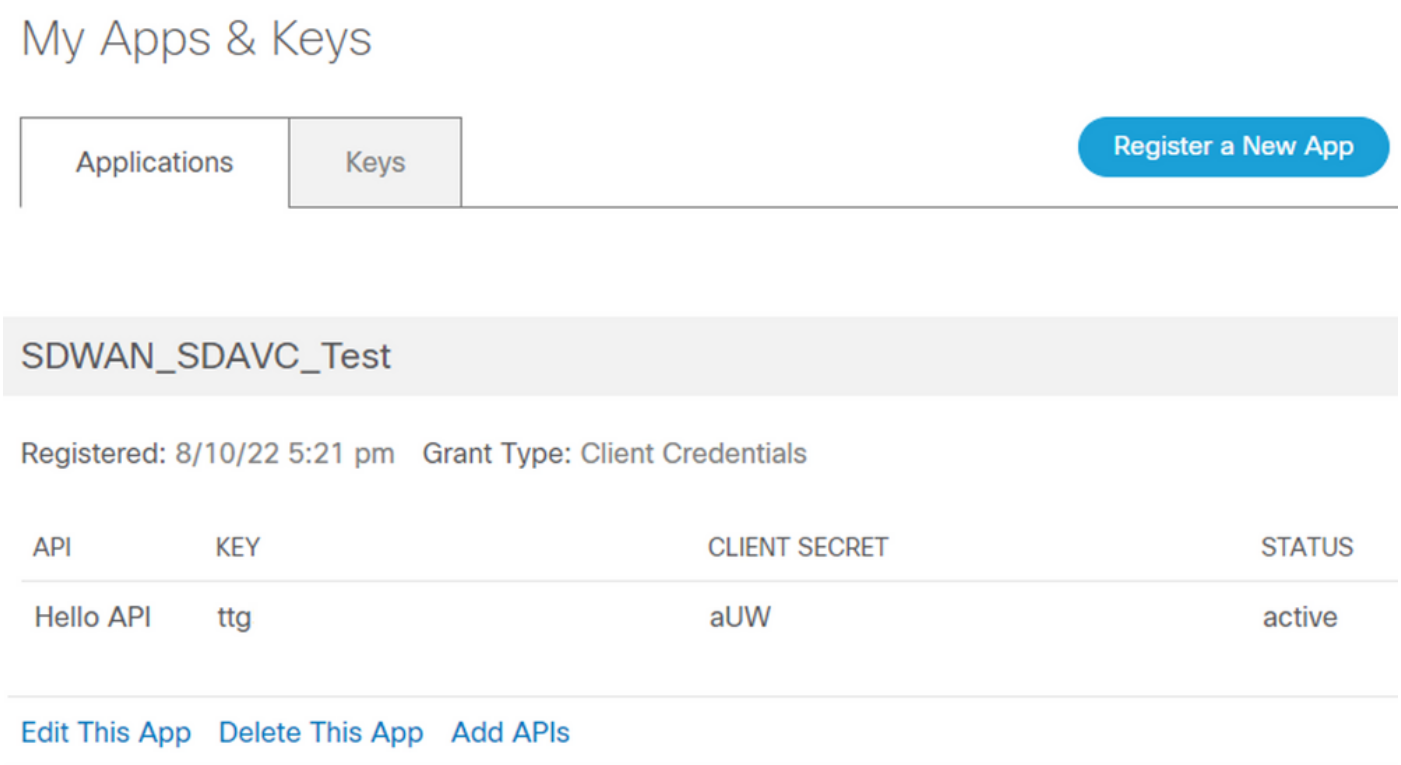
3. In the **Name of your application** field, enter a descriptive name for your application.

4. Check the **Client Credentials** check box.

5. Check the **Hello API** check box.

6. Check the check box to agree with Terms of Service.

7. Click Register. The Cisco API Console page displays the Client ID and Client Secret details. Keep this page open to complete the procedure as shown in this image.

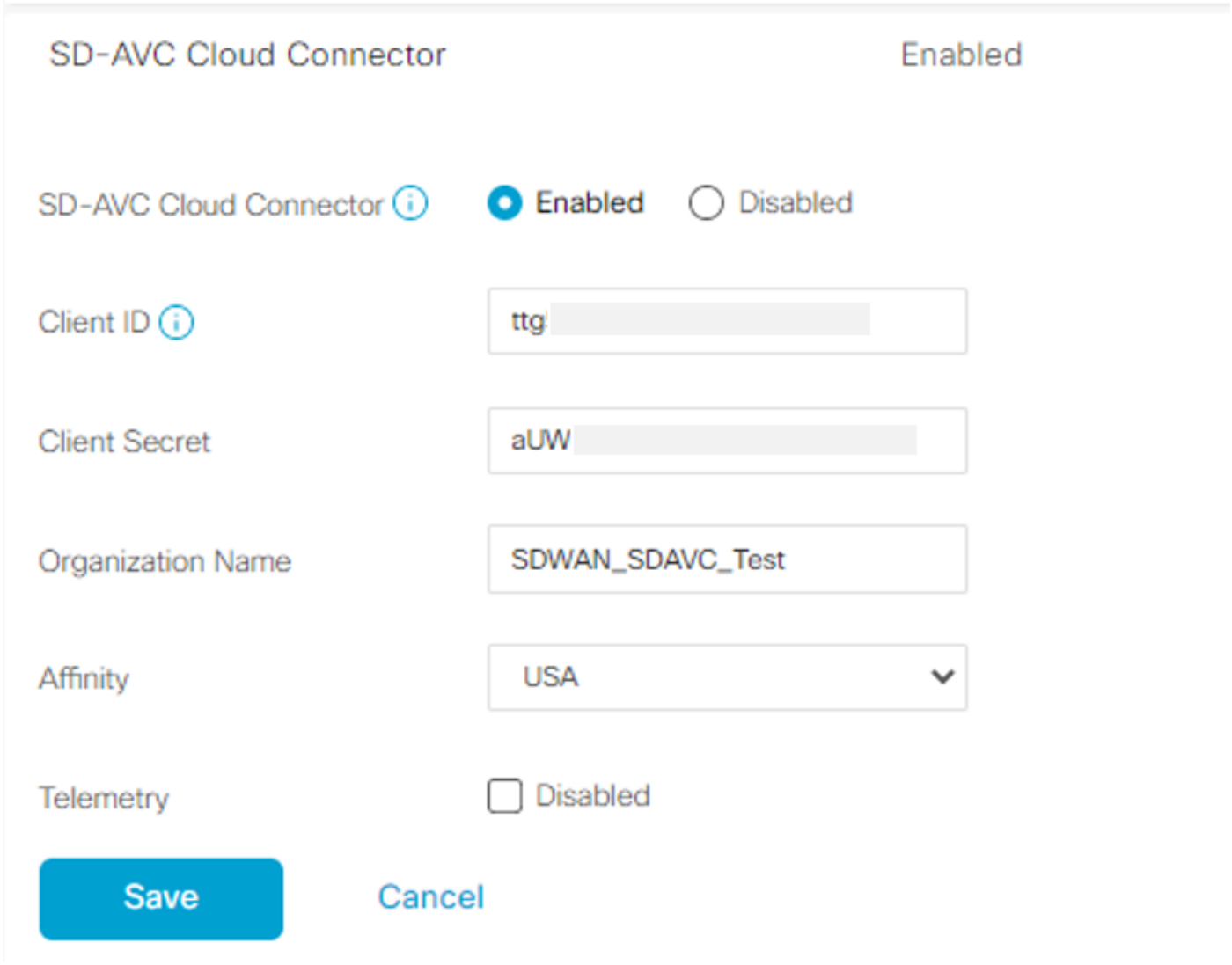


Enable SD-AVC Cloud Connector on vManage

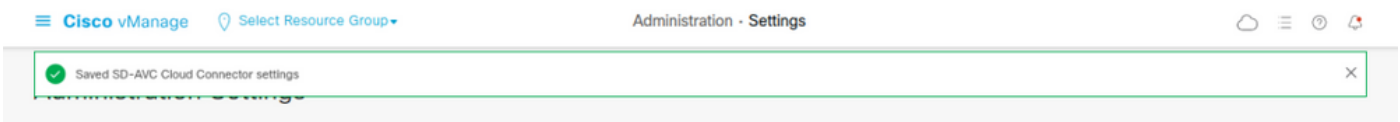
1. In the vManage GUI section, navigate to Administration > Settings > SD-AVC Cloud Connector and click **Edit**.

2.For SD-AVC Cloud Connector, click the Enabled radio button. Enter the values in these fields generated in the Enable Cloud Connector section, as shown in the image.

- Client ID
- Client Secret
- Organization Name
- Affinity
- Telemetry (optional)

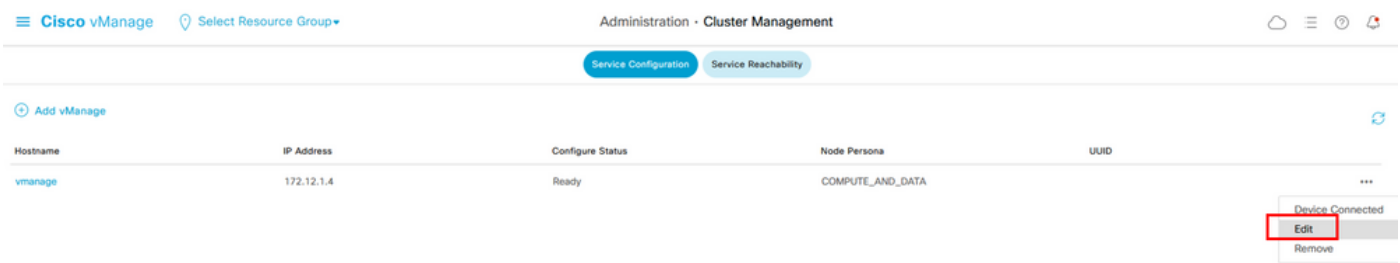


3. Click Save and verify the notification as shown in this image.



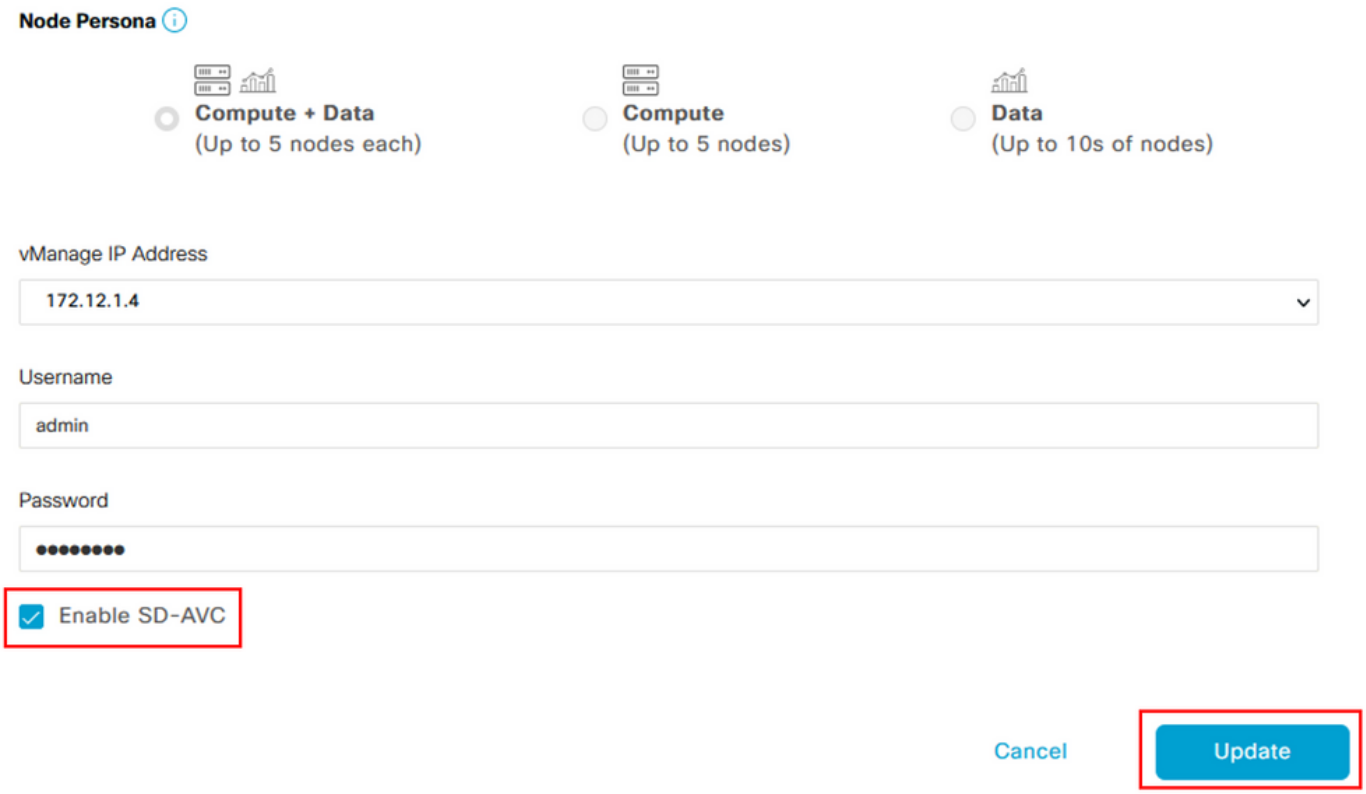
Enable SD-AVC on vManage

1. Navigate to **Administration > Cluster Management > Service Configuration**. Click **(...) More Actions** and choose **Edit.**

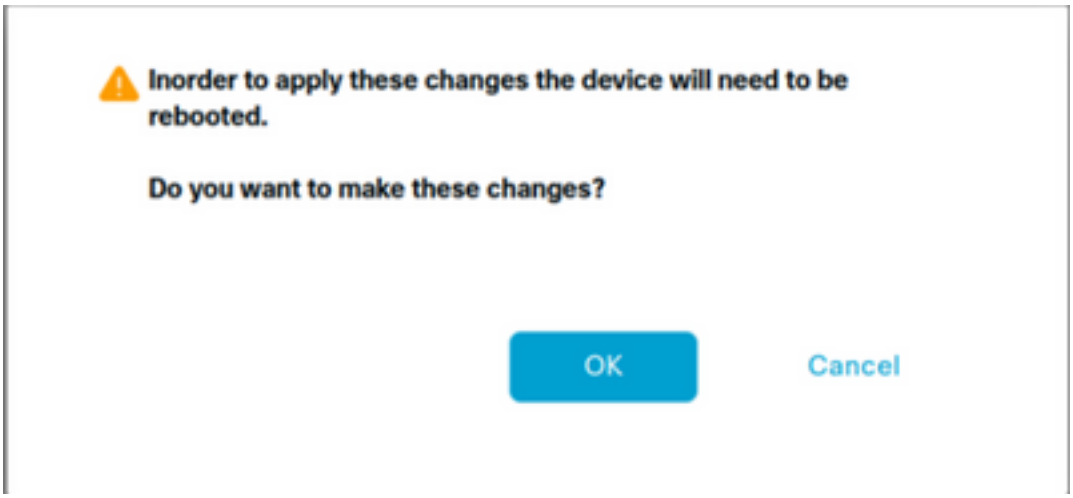


Note: Do not use a VPN 0 tunnel/transport or VPN 512 interface to enable SD-AVC. The cluster interface in vpn 0 can be used.

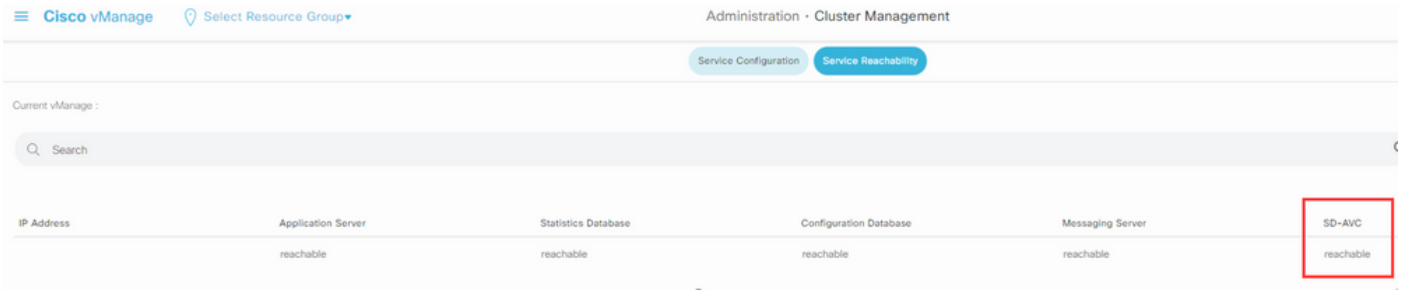
2. In the vManage IP Address section, click the IP address. Select the a non-tunnel IP address in VPN 0. Enter your credentials, check the **Enabled SD-AVC** check box, and click Update, as shown in the image.



3. Once the update has been confirmed, click OK in order to reboot the device as shown in the image.



4. After the vManage has rebooted, navigate to Administration > Cluster Management > Service Reachability. SD-AVC appears **Reachable**.

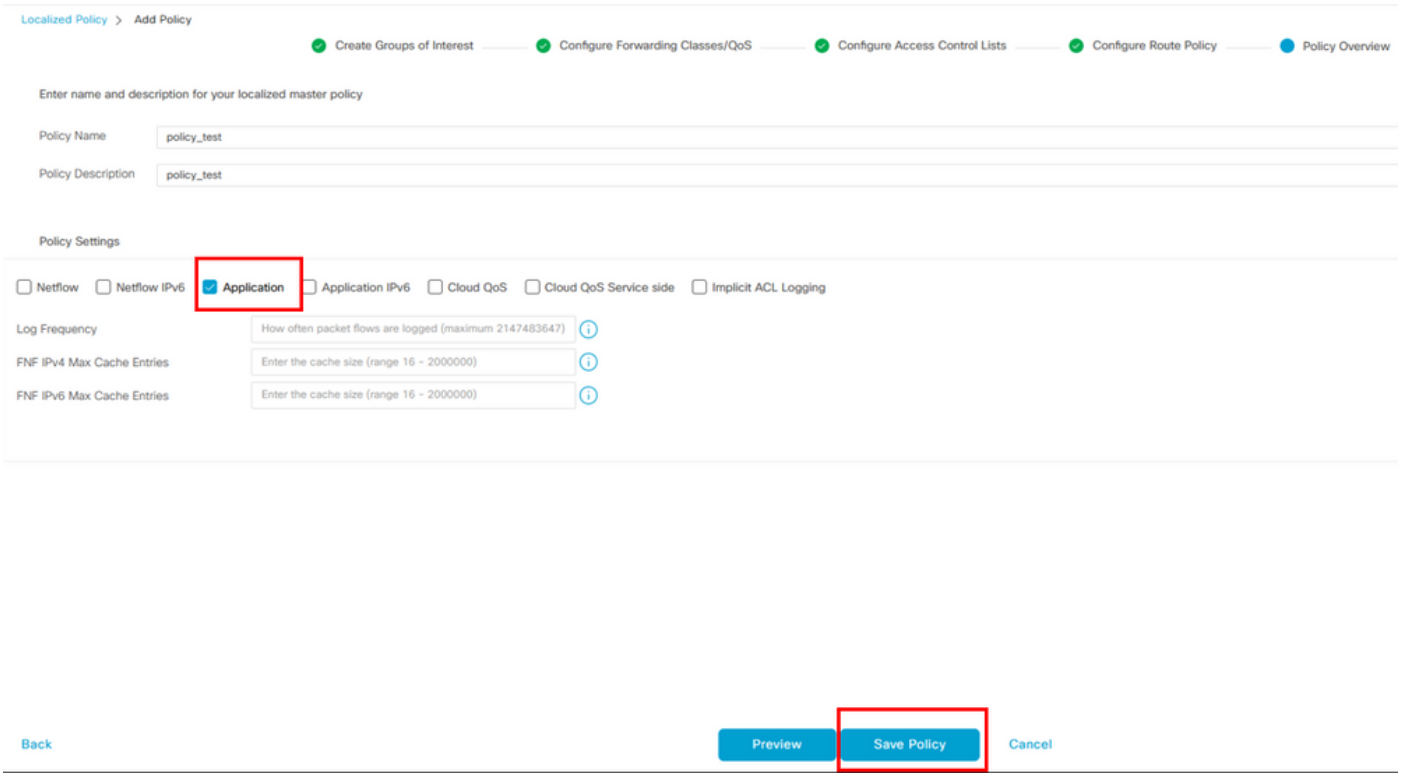


Policy Configuration

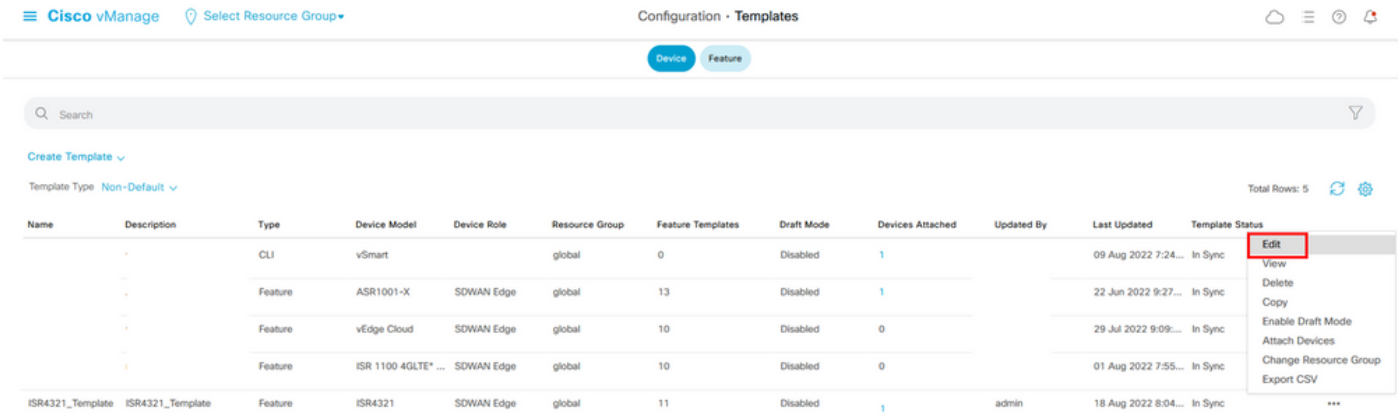
Once SD-AVC has been enabled, you need to create a localized policy and enable app visibility.

1. Navigate to the vManage GUI, and choose **Configuration > Policies > Localized Policy > Add Policy.**

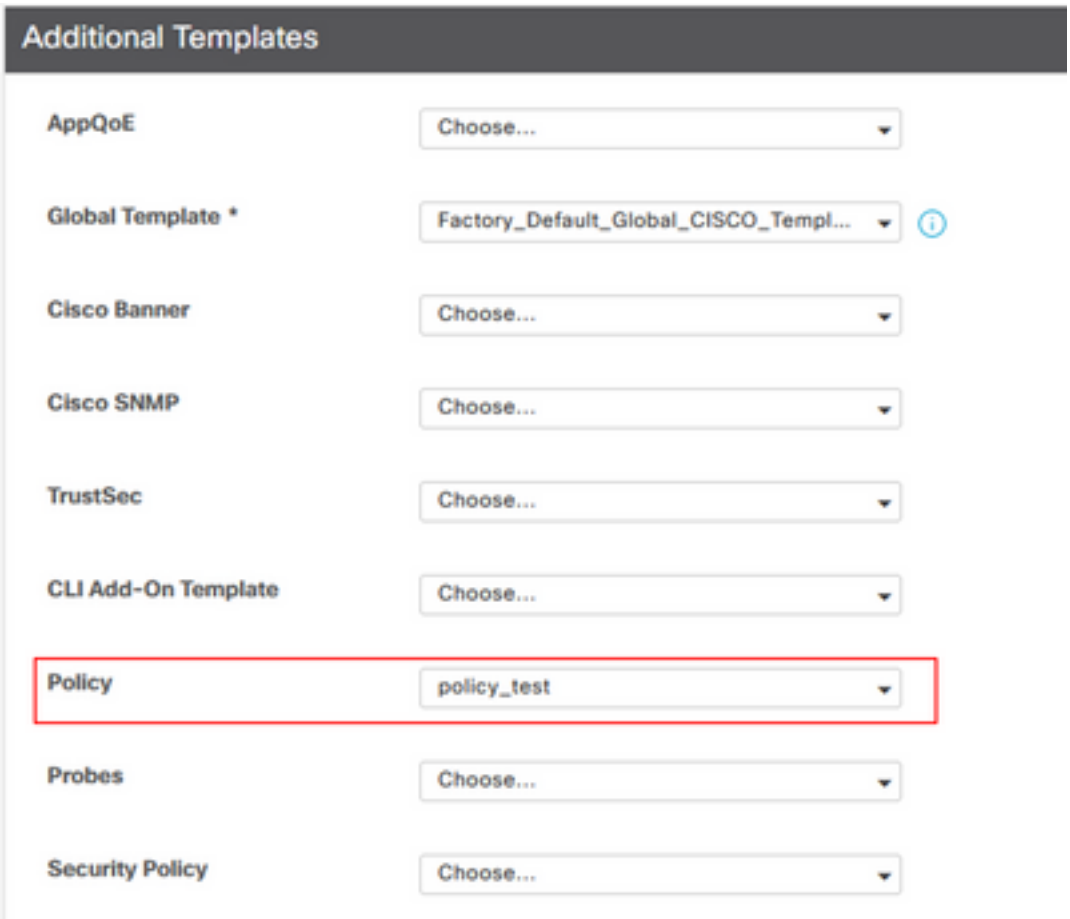
2. Navigate to **Policy Overview**,. In the Policy Settings section, check the **Application** check box and click **Save Policy.**



3. Navigate to Configuration > Templates. Identify the template name of your cEdge, click (...) More Actions and choose Edit as shown in the image.



4. Navigate to **Additional Templates**. From the **Policy** drop-down list, choose the Localized Policy created previously.



5. Save the template.

Verify

Use this section to confirm that your configuration works properly.

1. In the cEdge device, enter this command in order to verify the connectivity between the cEdge device and the SD-AVC controller.

```
ISR4321#show avc sd-service info summary
Status : CONNECTED <<<<<<<<<<<<<<<< The device is connected with SD-AVC
Device ID: ISR4321
Device segment name: <organization name>
Device address:<device ip address>
Device OS version:17.03.05
Device Type: ISR4321/K9

Active controller:
Type : Primary
IP : <system-ip>
Status: Connected
Version :4.0.0
Last connection: 21:20:28.000 UTC Thu Jul 31 2022

Active SDAVC import files
Protocol pack: Not loaded
```

```
Secondaru protocol pack PPDK_af575ccaebf99b0c4740dfc7a611d6.pack
```

2.Log in the vManage CLI and verify the container status.

```
vManage# request nms container-manager status
Container Manager is running<<<<<<<<<<<<<<<<<<
```

```
vManage# request nms-container sdavc status
b'Container: sdavc\nCreated: 7 weeks ago ago\nStatus: Up 7 weeks\n' <<<<<<<<<<<<
```

```
vManage# request nms container-manager diagnostics
NMS container manager
Checking container-manager status
Listing all images
------------------------
REPOSITORY                  TAG                 IMAGE ID            CREATED             SIZE
sdwan/cluster-oracle        1.0.1               aa5d2a4523a4        5 months ago        357MB
cloudagent-v2               fb3fc5c0841         fa24f9ef31a7        6 months ago        590MB
sdwan/host-agent            1.0.1               038ad845f080        7 months ago        152MB
sdwan/statistics-db         6.8.10              08fc31a50152        8 months ago        877MB
sdwan/coordination-server   3.6.2               5f4497812153        13 months ago       260MB
sdwan/configuration-db      4.1.7               ad351b31f7b9        13 months ago       736MB
sdwan/messaging-server      0.20.0              a46dc94d4993        13 months ago       71.2MB
sdavc                       4.1.0               721c572475f9        14 months ago       1.17GB
sdwan/support-tools         latest              0c3a995f455c        15 months ago       16.9MB
sdwan/service-proxy         1.17.0              4e3c155026d8        15 months ago       205MB
sdwan/ratelimit             master              f2f93702ef35        16 months ago       47.6MB

Listing all containers
------------------------

CONTAINER ID        IMAGE                               COMMAND
CREATED             STATUS
PORTS

NAMES
270601fc94ec        cloudagent-v2:fb3fc5c0841         "python ./main.py"       6 weeks
ago         Up 6 weeks          0.0.0.0:50051-
>50051/tcp
                                                                                 cloudagent-v2
53bba5216b24        sdwan/ratelimit:master            "/usr/local/bin/rate…"   6 weeks
ago         Up 6 weeks          6379/tcp, 0.0.0.0:8460-8462->8460-
8462/tcp
                                                           ratelimit
59bf900edf14        sdwan/service-proxy:1.17.0        "/entrypoint.sh /run…"   6 weeks
ago         Up 6
weeks

            service-proxy
62defa38c798        sdwan/messaging-server:0.20.0     "/entrypoint.sh /mes…"   6 weeks
ago         Up 6 weeks          0.0.0.0:4222->4222/tcp, 0.0.0.0:6222->6222/tcp, 0.0.0.0:8222-
>8222/tcp
                                messaging-server
3fbf32dd8d73        sdwan/coordination-server:3.6.2   "/docker-entrypoint.…"   6 weeks
ago         Up 6 weeks          0.0.0.0:2181->2181/tcp, 0.0.0.0:2888->2888/tcp, 0.0.0.0:3888-
```

```
>3888/tcp
                                coordination-server
c2e7b672774c        sdwan/configuration-db:4.1.7      "/sbin/tini -g -- /d…"   6 weeks
ago         Up 6 weeks          0.0.0.0:5000->5000/tcp, 0.0.0.0:6000->6000/tcp, 0.0.0.0:6362-
>6362/tcp, 0.0.0.0:6372->6372/tcp, 0.0.0.0:7000->7000/tcp, 0.0.0.0:7473-7474->7473-7474/tcp,
0.0.0.0:7687-7688->7687-7688/tcp   configuration-db
f42ac9b8ab37        sdwan/statistics-db:6.8.10        "/bin/tini -- /usr/l…"   6 weeks
ago         Up 17 hours         0.0.0.0:9200->9200/tcp, 0.0.0.0:9300-
>9300/tcp
                                                        statistics-db
112f3d9b578b        sdavc:4.1.0                       "/usr/local/bin/scri…"   7 weeks
ago         Up 7 weeks          0.0.0.0:10503->8080/tcp, 0.0.0.0:10502->8443/tcp, 0.0.0.0:10001-
>50000/udp
                            sdavc
06b09f3b030c        sdwan/host-agent:1.0.1            "python ./main.py --…"   7 weeks
ago         Up 7 weeks          0.0.0.0:9099-
>9099/tcp
                                                                                  host-agent
3484957576ee        sdwan/cluster-oracle:1.0.1        "/entrypoint.sh java…"   7 weeks
ago         Up 7 weeks          0.0.0.0:9090-
>9090/tcp
                                                                                  cluster-oracle
Docker info
-----------------------
Client:
 Debug Mode: false
Server:
 Containers: 10
 Running: 10
 Paused: 0
 Stopped: 0
 Images: 11
 Server Version: 19.03.12
 Storage Driver: aufs
 Root Dir: /var/lib/nms/docker/aufs
 Backing Filesystem: extfs
 Dirs: 149
 Dirperm1 Supported: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
 Volume: local
 Network: bridge host ipvlan macvlan null overlay
 Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: fd103cb716352c7e19768e4fed057f71d68902a0.m
 runc version: 425e105d5a03fabd737a126ad93d62a9eeede87f-dirty
 init version: fec3683-dirty (expected: fec3683b971d9)
 Kernel Version: 4.9.57-ltsi
 Operating System: Linux
 OSType: linux
 Architecture: x86_64
 CPUs: 16
 Total Memory: 30.46GiB
 Name: vManage
 ID: XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXX:XXXXX
 Docker Root Dir: /var/lib/nms/docker
 Debug Mode: false
 Registry: https://index.docker.io/v1/
 Labels:
 Experimental: false
```

```
 Insecure Registries:
 127.0.0.0/8
 Live Restore Enabled: false
WARNING: No cpu cfs quota support
WARNING: No cpu cfs period support
WARNING: bridge-nf-call-iptables is disabled
WARNING: bridge-nf-call-ip6tables is disabled
WARNING: the aufs storage-driver is deprecated, and will be removed in a future release.
```

Troubleshoot

This section provides information you can use to troubleshoot your configuration.

In vManage logs, verify these paths:

```
/var/log/nms/vmanage-server.log
/var/log/nms/containers/sdavc/avc/sdavc_application.log
```

Enter this command:

```
request nms container-manager {status | diagnostics}
```

In cEdge Cisco IOS® XE, enter these commands:

```
Router#show avc sd-service info connectivity
show avc sd-service info {export | import}
```