Cisco SD-AVC User Guide, Release 2.1.0

First Published: 2018-07-03

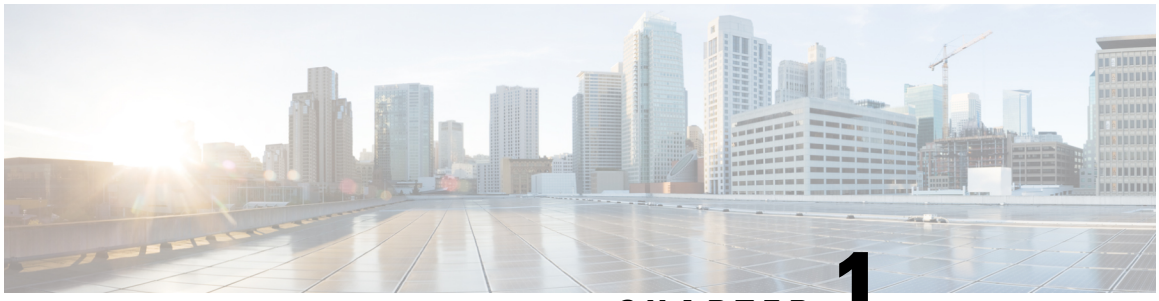
Last Modified: 2018-07-23

Americas Headquarters
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

CONTENTS

PART I

Part: Introduction



CHAPTER 1

SD-AVC Overview

SD-AVC Overview

Cisco Software-Defined AVC (SD-AVC) is a component of Cisco Application Visibility and Control (AVC). It functions as a centralized network service, operating with specific participating devices in a network.

As an SDN solution operating network-wide, Cisco SD-AVC complements solutions such as:

- Cisco Intelligent WAN (IWAN)
- Cisco EasyQoS
- Application Assurance

Features and Benefits

Feature/Benefit	Description
Network-level application recognition consistent across the network	The SD-AVC network service aggregates application data from multiple devices and sources, and provides that composite application information in return. Because SD-AVC operates at the network level, any application rule created by SD-AVC based on aggregated application data is shared and applied consistently across all participating network devices.

Feature/Benefit	Description
Improved application recognition in symmetric and asymmetric routing environments	Cisco SD-AVC further refines application recognition accuracy by helping numerous devices in a network SD-AVC aggregates application data shared by participating devices in the network, and analyzes the shared application data. It then provides this composite application information (in the form of an application rules pack) to the participating routers, improving application recognition. Because SD-AVC shares application rules across numerous network devices, devices that see only one direction of a flow can benefit from the information collected on the other direction of the same flow. See SD-AVC and Application Recognition, on page 8.
Improved first packet recognition	SD-AVC application rules are based on flow tuple (address and port) information. After a learning phase and sharing tuples among participating devices, the devices are able to identify new flows on the first packet, based on the tuple information
Protocol Pack update at the network level	SD-AVC can assist in deploying Protocol Packs to numerous routers in the network. Download the Protocol Packs to deploy, store them on the centralized SD-AVC network service, then use the SD-AVC Dashboard to select which devices in the network will receive the Protocol Packs. See Protocol Packs Page, on page 43.
SD-AVC Dashboard	Secure browser-based SD-AVC Dashboard over HTTPS for monitoring SD-AVC functionality and statistics, and for configuring Protocol Pack updates network-wide. See Using SD-AVC, on page 37.
Improved Microsoft Office 365 traffic classification	The MS-Office365 Connector component improves classification for Microsoft Office 365 traffic. The SD-AVC Dashboard displays the status of the component. See MS-Office365 Connector, on page 42.
REST API	REST API for user-defined applications. See SD-AVC REST API, on page 75.
Analysis of unclassified traffic	To improve traffic visibility, SD-AVC analyzes unclassified/unidentified traffic and provides server or socket information about unclassified traffic flows that use significant bandwidth. See Unclassified Traffic Analysis and Discovery, on page 40.

No Change to Topology

Deploying SD-AVC within an existing network does not require any changes to the network topology.

New Features and Changes, by Release

Table 1: New and Changed Features, SD-AVC Release 2.1.0

Feature	Description
REST API	The REST API enables configuring user-defined applications, providing classification of applications not covered by the standard Protocol Pack. See #unique_12.
Unclassified traffic discovery	To improve traffic visibility, SD-AVC analyzes unclassified/unidentified traffic and provides server or socket information about unclassified traffic flows that use significant bandwidth. See Unclassified Traffic Analysis and Discovery, on page 40.
Source interface configuration	On network devices operating with SD-AVC, you can specify the interface that will appear as the source address for all SD-AVC traffic between the network device and the SD-AVC network service. See Source Interface Configuration Overview, on page 93.
Ability to configure proxy DNS servers for the MS-Office365 Connector	By default, SD-AVC has two Cisco OpenDNS DNS servers configured. Improved ability to add additional DNS servers. See MS-Office365 Connector, on page 42.

Table 2: New and Changed Features, SD-AVC Release 2.0.1

Feature	Description
SD-AVC system time and displayed times	Improved display of times in the SD-AVC Dashboard. Internally, the SD-AVC network service uses standard UTC. The Dashboard displays times according to the internal SD-AVC system time, adjusted by the local time zone offset of the PC that is accessing the Dashboard. See SD-AVC System Time and Displayed Times, on page 43.
Improved ability to configure and view DNS servers for the MS-Office365 Connector	By default, SD-AVC has two Cisco OpenDNS DNS servers configured. Improved ability to add additional DNS servers. See MS-Office365 Connector, on page 42.

Table 3: New and Changed Features, SD-AVC Release 2.0.0

Feature	Description
Updated user interface	• Improved interactive display of traffic data • Improved presentation of warnings and errors affecting devices

Feature	Description
Improved control of Protocol Pack deployment	• Can update Protocol Packs for individual devices, for segments, or for all devices in the network • Ability to revert to the Protocol Pack built into the Cisco IOS release See: Protocol Packs Page, on page 43
Improved Microsoft Office 365 traffic classification	MS-Office365 Connector is a component introduced in this release that improves classification for Microsoft Office 365 traffic. The SD-AVC Dashboard displays the status of the component. This feature requires connectivity to a DNS server. By default, SD-AVC uses Cisco OpenDNS servers: 208.67.222.222 and 208.67.220.220 See: MS-Office365 Connector, on page 42
Support for more devices	Support for 4000 network devices operating with SD-AVC

Using SD-AVC in an Asymmetric Routing Scenario

The Challenge of Asymmetric Routing

One of the challenges that SD-AVC addresses well is application recognition in asymmetric routing scenarios. While it is not the only situation in which SD-AVC offers improved results, asymmetric routing demonstrates one of the advantages of aggregating application data from many sources.

Certain network configurations may produce "asymmetric routing" as an unintended effect. In asymmetric routing, the packets of a single two-way connection travel by different paths between network nodes. For example the downstream traffic from a server to a client might be routed through one path, while the upstream traffic from the client to the server might be through a different path. When this occurs, AVC operating on a hub router may see only a single direction of the traffic for that connection, posing a challenge to application recognition.

Deep Packet Inspection and Asymmetry

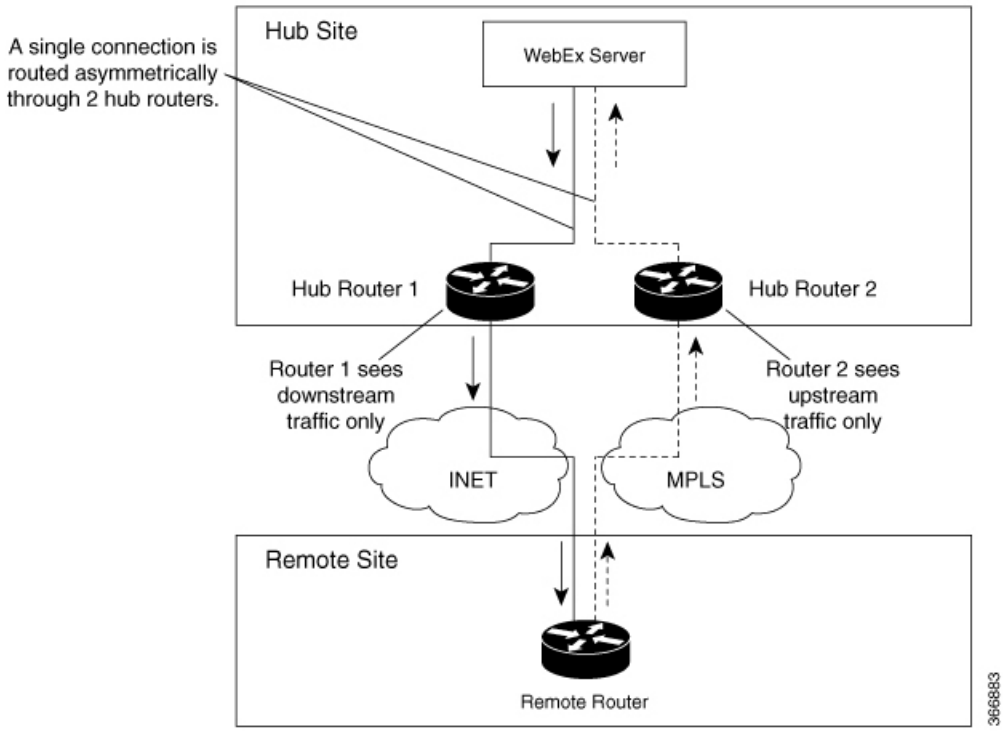
AVC deep packet inspection (DPI) operates best when it sees both directions of traffic. In symmetric routing, AVC operating on a single device that handles both directions of a flow can fully analyze metadata and other traffic attributes to help identify the application creating the flow. By contrast, an asymmetric scenario can limit the ability to recognize some types of traffic. This is especially true when AVC sees only to the downstream traffic for a particular flow.

Asymmetric routing may occur for various reasons, including from intelligent path selection by Cisco IWAN. The issue particularly affects hub routers within an enterprise network with a hub/branch topology.

Effects of Limited Application Recognition

Limiting AVC application recognition can affect classification of traffic for QoS policy, visibility, and other functionality. Consequently, a solution that overcomes the limitations caused by asymmetric routing is especially helpful for maximum network efficiency.

Figure 1: Asymmetric Routing Example

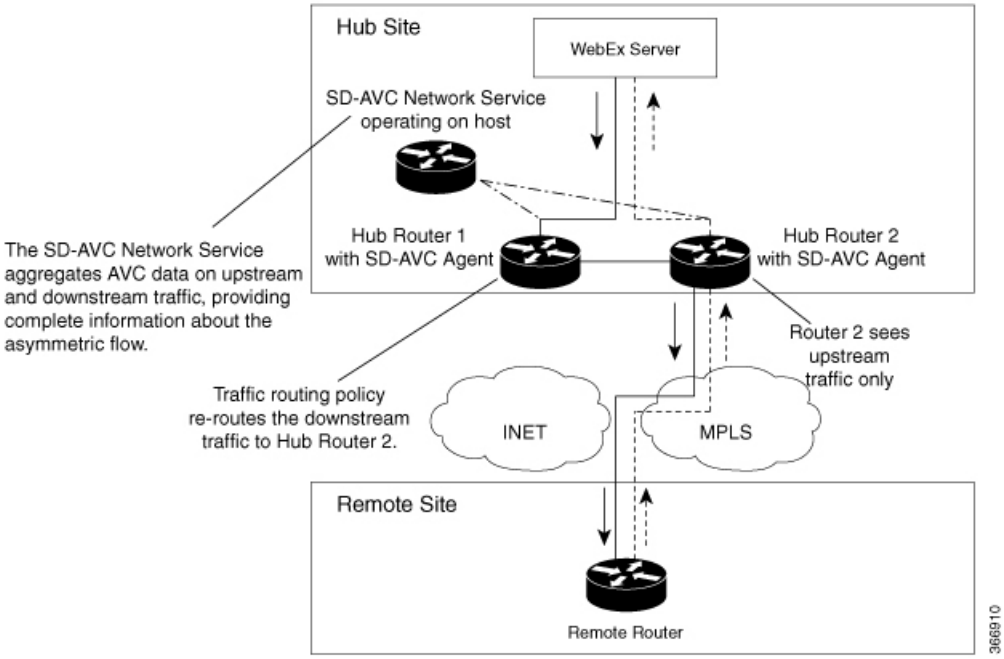


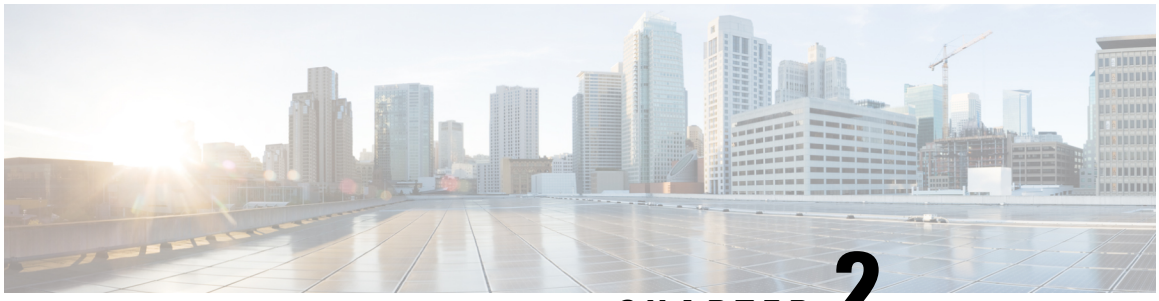
Centralized Server Aggregating Application Data

SD-AVC compiles and analyzes application data from multiple devices within the network, including devices that separately handle the downstream and upstream traffic for a single flow. Using data from multiple sources, SD-AVC synchronizes application information network-wide, overcoming the challenges of asymmetric routing. This strategy provides a major improvement to application recognition within networks, improving the effectiveness of application-based solutions.

With the improved application recognition, AVC can apply application-based policies, such as QoS, path selection, and visibility more accurately. For example, with complete information about both streams of a flow, a path selection policy can direct the downstream path through the same route as the upstream.

Figure 2: Asymmetric Routing and SD-AVC





CHAPTER 2

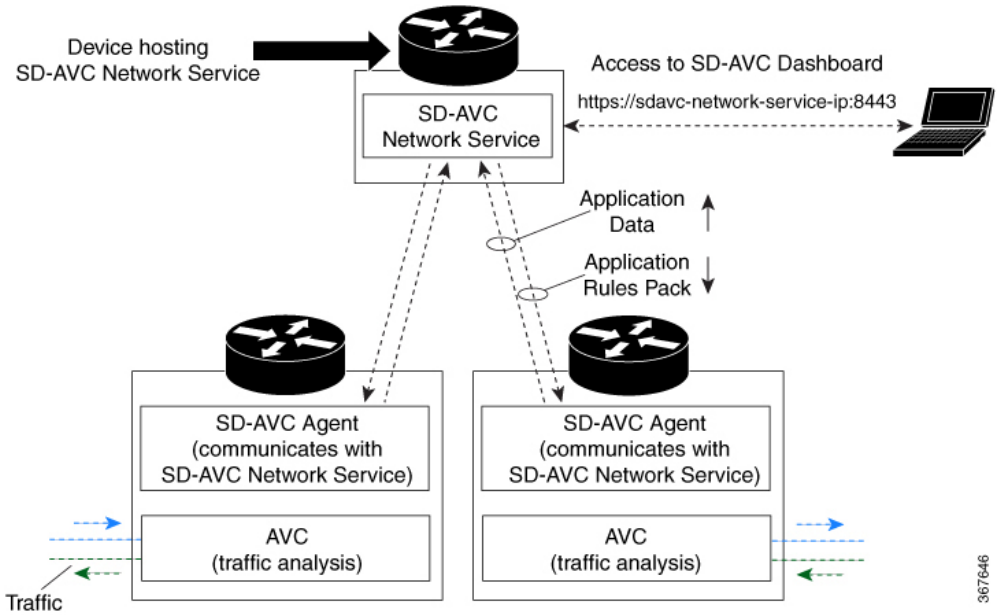
Operation

SD-AVC Architecture

SD-AVC architecture consists of two basic components:

- Centralized SD-AVC network service component operating on a host device
- SD-AVC Agent component running on each SD-AVC-enabled device in the network

Figure 3: SD-AVC Network Service and Agents



SD-AVC and Application Recognition

Cisco AVC can recognize 1400+ network applications, providing recognition of most enterprise network traffic. SD-AVC offers a network-wide approach, aggregating application information collected across the network, and centralized deployment of Protocol Pack updates.

To improve recognition of uncommon or in-house network applications, as well as for other uses, SD-AVC enables creating user-defined applications, expanding on the range of applications included in the Cisco-provided Protocol Packs. The user-defined applications are distributed to all participating devices in the network.

SD-AVC improves application recognition, and offers a solution to challenges posed by complex networks that use a variety of routing devices and routing methods. Such challenges include asymmetric routing, first packet classification, encryption, and so on.

Collecting Application Data

Devices in the network running AVC analyze traffic and generate application data. If a device is connected to SD-AVC, the SD-AVC agent operating on the device receives this application data, and processes and caches the data. Periodically, the SD-AVC agent sends the latest application data to the centralized SD-AVC network service.

As new servers are detected or as server addresses change, the agent continually discovers and validates these servers and updates the SD-AVC network service with the new information. The process of discovery and validation can take several minutes.

Server addresses usually remain constant over time, but when they do change, the SD-AVC agent detects the changes and updates the network service.

Aggregating Application Data

The SD-AVC network service aggregates application data from multiple sources, producing an application rules pack from the composite data. This is made available to network devices using SD-AVC.

Periodically, the network devices using SD-AVC request the application rules pack. Relying on devices to pull (request) the application rules pack on their own schedule improves efficiency and simplifies administration.

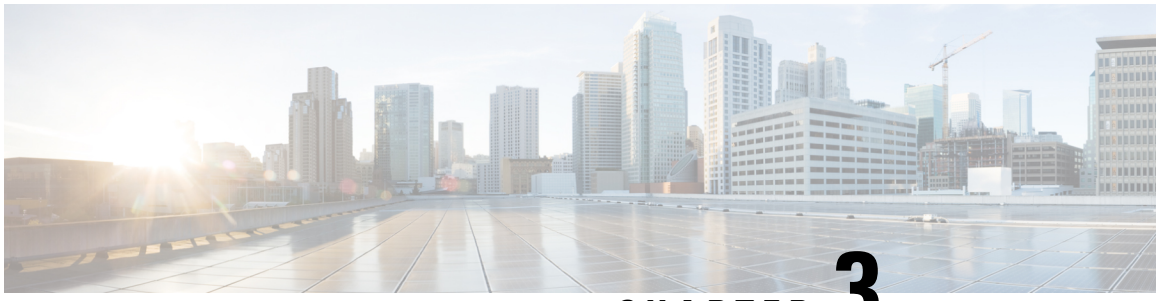
The application rules pack contains the following type of information: ID, IP address, port, network protocol, VRF name, application name, and so on.

Example:

```
ID   | IP Address | Port | Protocol | VRF-name | App-Name
=================================================================
0    | 192.0.2.1  | 5901 | TCP      | Mgt      | VNC
```

PART II

Part: Deployment



CHAPTER 3

Installation Overview

SD-AVC operates in a service/agent configuration. For details, see SD-AVC Architecture, on page 7.

- **Network Service**: The SD-AVC network service is installed as a virtualized component on a Cisco device service container, and operates on the device as a service. See: System Requirements: SD-AVC Network Service Host, on page 12
- **Agent**: Other devices in the network are enabled as agents, and communicate with the SD-AVC network service. See: Configuring Network Devices to Use SD-AVC, on page 25
- **High Availability**: SD-AVC supports a high availability (HA) configuration, using more than one SD-AVC network service. See: SD-AVC High Availability, on page 31
- **Connectivity**: Operating SD-AVC requires connectivity between the SD-AVC network service and the SD-AVC agents that operate on devices in the network. See: Configuring Connectivity, on page 13

Summary of Setup

The following table briefly describes the steps to set up SD-AVC:

Table 4: Setup

	Setup Task	Section
1	Download the open virtual appliance (OVA) file for the SD-AVC network service and install it on a host device accessible by other devices in the network.	See: Installing the SD-AVC Network Service, on page 14
2	Enable the SD-AVC agent on Cisco devices in the network, pointing them to the SD-AVC network service set up in the previous step. (In a high availability setup, include more than one SD-AVC network service instance.)	See: Configuring Network Devices, on page 25
3	Configure connectivity, or optionally, secure connectivity.	See: Configuring Connectivity, on page 13, Configuring Secure Connectivity, on page 67

- System Requirements: SD-AVC Network Service Host, on page 12
- Configuring Connectivity, on page 13

System Requirements: SD-AVC Network Service Host

The following table describes platform requirements for hosting the SD-AVC network service.

Table 5: SD-AVC Network Service Host Requirements

Host	Memory	Storage	Recommended OS (extended maintenance release trains only)	CPU
Cisco ASR1001-X Aggregation Services Routers	M-ASR1001X-16GB	NIM-SSD and SSD-SATA-400G	Cisco IOS XE Everest 16.6.1 or later	—
Cisco ASR1002-X Aggregation Services Router	M-ASR1002X-16GB	MASR1002X-HD-320G	Cisco IOS XE Everest 16.6.1 or later	—
Cisco ASR1002-HX Aggregation Services Router	M-ASR1002HX-16GB	NIM-SSD and SSD-SATA-400G	Cisco IOS XE Fuji 16.7.1 or later	—
Cisco ISR4431 Integrated Services Router	RAM: MEM-4400-4GU16G Flash: MEM-FLASH-16G	NIM-SSD and SSD-MSATA-400G	Cisco IOS XE Everest 16.6.1 or later	—
Cisco ISR4451 Integrated Services Router	RAM: MEM-4400-4GU16G Flash: MEM-FLASH-16G	NIM-SSD and SSD-MSATA-400G	Cisco IOS XE Everest 16.6.1 or later	—

Host	Memory	Storage	Recommended OS (extended maintenance release trains only)	CPU
Cisco CSR1000V Cloud Services Router	Minimum: 8 GB Recommended: 8 GB	20 GB	Cisco IOS XE Everest 16.6.1 or later	Large-scale scenario (100 or more devices): 4 cores Small-scale scenario (<100 devices): 4 cores See: Allocating VM CPUs for Cisco CSR1000V, on page 73

Configuring Connectivity

Operating SD-AVC requires connectivity between various components.

- SD-AVC network service and host
- SD-AVC network service and agents
- Connectivity to the SD-AVC Dashboard

This section describes the connectivity requirements. If secure connectivity is required, see: Configuring Secure Connectivity, on page 67

SD-AVC Network Service and Host

Connectivity is required between the SD-AVC network service, which operates as a virtualized service, and the device hosting it. The host platform requires connectivity with the service through a virtual interface called VirtualPortGroup. The virtual service communicates with the host over this virtual interface, using SSH on TCP port 22.

SD-AVC Network Service and Agents

Network devices operating with SD-AVC use an SD-AVC agent, which operates in the background on the device, to communicate with the central SD-AVC network service. Connectivity is required between each of these network devices and the SD-AVC network service (more than one network service in SD-AVC high availability configurations).

- **Ports**

  Communication between agent and service uses the following protocols and ports:

  - **UDP**: Port 50000
  - **TCP**: Ports 20, 21, 50000-60000

- **Firewalls and Access Lists**

  Ensure that communication is possible in both directions (agent to SD-AVC network service, SD-AVC network service to agent) on these ports for the relevant traffic. For example:

  - Firewall policy must enable communication in both directions.
  - If a network device has an access control list (ACL) configured, the ACL must permit communication between the SD-AVC network service and SD-AVC agents.

Connectivity to the SD-AVC Dashboard

Connecting to the SD-AVC Dashboard (see Using SD-AVC, on page 37) requires access to the device hosting the SD-AVC network service, and involves TCP traffic through port 8443. Ensure that network policy (firewall, ACL, and so on) permits this connectivity for devices requiring access to the SD-AVC Dashboard.

Using SD-AVC with Cisco IWAN

When operating SD-AVC in a Cisco IWAN environment, the SD-AVC network service may be hosted on the hub master controller (MC) or on a router dedicated for the purpose of hosting the service.

In either case, verify that the host device meets the system requirements for hosting the SD-AVC network service.

See: System Requirements: SD-AVC Network Service Host, on page 12, Installing the SD-AVC Network Service, on page 14

Installing the SD-AVC Network Service

The SD-AVC network service operates as a virtualized service on a Cisco router. It is installed as an open virtual appliance (OVA) virtual machine container, and requires a few steps of configuration on the host router. After configuration is complete, you can check service status using the browser-based SD-AVC Dashboard.

Table 6: Overview of Installation Steps

Task	Steps
System requirements	Step 1
Installation	Steps 2 to 7
Configuration, Activation	Step 8 to 12
Verification	Steps 13 to 14
Connecting to SD-AVC Dashboard	Step 15

Examples follow the steps below.

Installation Procedure

The following procedure installs the SD-AVC network service as a virtualized service on a Cisco router.

1. Verify that the intended host device meets the system requirements. See: System Requirements: SD-AVC Network Service Host, on page 12
2. Download the OVA container for the SD-AVC network service from Cisco.com, using the Download Software tool. Specify a platform that supports hosting the SD-AVC virtual service, then navigate to software downloads for the platform. Select the "SD AVC Router Virtual Service" option to display available OVA files for SD-AVC.

   Example filename: **`iosxe-sd-avc.2.1.0.ova`**
3. Copy the downloaded OVA file onto the device that will host the SD-AVC network service. Copy to one of the following locations, depending on the platform type:
   - For the CSR1000V router, use: **`bootflash`**
   - For ASR1000 Series or ISR4000 Series devices, use: **`harddisk`**

     `harddisk` refers to the SSD or HD specified in the system requirements for the platform (System Requirements: SD-AVC Network Service Host, on page 12).
4. On the device, verify that the MD5 checksum of the downloaded package matches the checksum value provided.

Note The correct MD5 checksum value apears on the Download Software page when downloading the package.

verify /md5 bootflash:*ova-filename.ova*

Example:

```
Device#verify /md5 bootflash:iosxe-sd-avc.2.1.0.ova
.....................................................................................Done!
verify /md5 (bootflash:iosxe-sd-avc.2.1.0.ova) = d8b7af1b163ccc5ad28582a3fd86c44e
```

5. Ensure that the system time is set correctly on the host device.
   - (If using an NTP server) Verify that the platform is connected to the NTP server and that the system time is correct.
   - (If setting time manually) Set the system time correctly.

Important If you change the system time after the SD-AVC service is already running, uninstall and re-install the SD-AVC service to ensure correct synchronization.

Unconfiguring or Uninstalling the SD-AVC Network Service, on page 23
Installation Overview, on page 11

6. If specific DNS servers are required, configure the server(s) on the host device.

Important Adding DNS servers after SD-AVC is active restarts the SD-AVC network service. During restart, the following are interrupted:

- Protocol Pack deployment to network devices
- Vertical debug

7. On the host device, execute the following command to extract the OVA package and install the SD-AVC network service. By default, it is installed on the same storage device where the OVA package was saved.

   service sd-avc install package *disk-with-OVA*:*OVA-filename* **media** *location-for-OVA-expansion*

   Table 7: Command Details

CLI keyword/argument	Description
disk-with-OVA	Specify one of the following, according to the platform type. The location refers to where the OVA was saved in a previous step. • CSR: **bootflash** • ASR1000 Series or ISR4000 Series: **harddisk**
OVA-filename	Downloaded OVA file.
location-for-OVA-expansion	Specify one of the following, according to the platform type: • For CSR1000V routers, use: **bootflash** • For ASR1000 Series or ISR4000 Series devices, use only: **harddisk** **Important** On ASR1000 and ISR4000 platforms, do not use **bootflash**. The CLI may allow you incorrectly to choose **bootflash**, but but this causes the step to fail. On these platforms, specify **only harddisk**.

   Examples:

   - For CSR1000V router:

     ```
     service sd-avc install package bootflash:iosxe-sd-avc.2.1.0.ova media bootflash
     ```

   - For ASR1000 Series or ISR4000 Series routers:

     ```
     service sd-avc install package harddisk:iosxe-sd-avc.2.1.0.ova media harddisk
     ```

8. Configure the SD-AVC network service.
   - Specify the router gateway interface that the virtualized service uses for external access.
   - Specify a user-selected external-facing service IP address for the SD-AVC network service. This address must be within the same subnet as the gateway interface address.

This step accomplishes the following:

- Enables routers in the network to communicate with the SD-AVC network service.
- Enables access to the browser-based SD-AVC Dashboard.

Note Use this command only in scenarios in which the gateway interface is not attached to a VRF. If the gateway interface is attached to a VRF, use the steps described in Operating the SD-AVC Network Service with Host Interface Attached to a VRF, on page 65.

service sd-avc configure gateway interface *interface* **service-ip** *service-ip-address* [**activate** | **preview**]

Table 8: Command Details

CLI keyword/argument	Description
activate	Activates the service immediately. It is not typically recommended to use this option during this configuration step. Execute the `activate` option in a separate step, as shown below.

preview	Preview the configuration without configuring or activating the service. When using this option, the configuration is not sent to the device.
	Note: If the gateway interface is attached to a VRF, see Operating the SD-AVC Network Service with Host Interface Attached to a VRF, on page 65.
	Example output:
	`! Virtual port configuration`
	`interface VirtualPortGroup31`
	`  description automatically created for sd-avc service by`
	`'service sd-avc configure' exec command`
	`  ip unnumbered gigabitEthernet1`
	`end`
	
	`! Virtual service configuration`
	`virtual-service SDAVC`
	`  description automatically created for sd-avc service by`
	`'service sd-avc configure' exec command`
	`  vnic gateway VirtualPortGroup31`
	`    guest ip address 10.56.196.101`
	`  exit`
	`end`
	
	`! Static route configuration`
	`ip route  10.56.196.101 255.255.255.255 VirtualPortGroup31`

interface	Gateway interface: The device interface that the virtualized service uses for external access. **Note**: If the interface is attached to a VRF, see Operating the SD-AVC Network Service with Host Interface Attached to a VRF, on page 65 for instructions for configuring the gateway.

CLI keyword/argument	Description
service-ip-address	External-facing IP address, must be in the same subnet as the IP of the gateway interface. **Example**: Gateway interface: 10.56.196.100 service-ip-address: 10.56.196.101

Example:

```
service sd-avc configure gateway interface gigabitEthernet1 service-ip 10.56.196.146
```

9. Activate the service.

   service sd-avc activate

   Example:

   ```
   service sd-avc activate
   ```

10. Verify that the status of the SD-AVC network service is activated.

    service sd-avc status

    If installation and activation were successful, the displayed status is:

    ```
    SDAVC service is installed, configured and activated
    ```

11. (ASR1000 Series or ISR4000 Series routers only, not CSR1000 Series) Execute the following:

    ```
    (config)#platform punt-policer service-engine 100000 100000
    ```

12. Save the new configuration.

    copy running-config startup-config

13. Ping the service IP configured in a previous step to verify that it is reachable.

14. Verify that SSH is enabled on the host device. Details vary according to different scenarios, but the following is a helpful reference:
    https://www.cisco.com/c/en/us/support/docs/security-vpn/secure-shell-ssh/4145-ssh.html

    Example (uses SSH local authentication):

    ```
    aaa new-model
    !
    aaa authentication login default local
    username cisco privilege 15 password cisco
    ip domain name cisco.com
    crypto key generate rsa
    ```

15. Wait several minutes for the service to become fully active, then use a Chrome browser to access the browser-based SD-AVC Dashboard, at the following URL, which uses the service-ip configured in an earlier step and port 8443. The SD-AVC Dashboard uses the same authentication as the platform hosting the SD-AVC network service.

    https://<service-ip>:8443

Note Accessing the SD-AVC Dashboard requires connectivity from the PC you are using to access the SD-AVC interface.

Installation Example for CSR1000V Router

The following is an example of the CLI steps used to install the SD-AVC Network Service on a Cisco CSR1000V Cloud Services Router. For this router, the first step includes “bootflash” as the location for extracting the OVA.

```
service sd-avc install package harddisk:iosxe-sd-avc.2.1.0.ova media bootflash
service sd-avc configure gateway interface gigabitEthernet1 service-ip 10.56.196.146
service sd-avc activate
service sd-avc status
copy running-config startup-config
```

Installation Example for ASR1000 Series or ISR4000 Series Routers

The following is an example of the CLI steps used to install the SD-AVC network service on a Cisco ASR1000 Series or ISR4000 Series Router. For these routers, the first step includes “harddisk” as the location for extracting the OVA.

```
service sd-avc install package harddisk:iosxe-sd-avc.2.1.0.ova media harddisk
service sd-avc configure gateway interface gigabitEthernet1 service-ip 10.56.196.146
service sd-avc activate
service sd-avc status
platform punt-policer service-engine 100000 100000
copy running-config startup-config
```

Upgrading the SD-AVC Network Service

Use the following procedure to upgrade the SD-AVC network service on the router hosting the service.

Table 9: Overview of Upgrade Steps

Task	Steps
Installation	Steps 1 to 7
Activation	Step 8
Verification	Step 9

1. Download the OVA container for the SD-AVC network service from Cisco.com, using the Download Software tool. Specify a platform that supports hosting the SD-AVC virtual service, then navigate to software downloads for the platform. Select the "SD AVC Router Virtual Service" option to display available OVA files for SD-AVC.

   Example filename: **`iosxe-sd-avc.2.1.0.ova`**

2. Copy the downloaded OVA file onto the device hosting the SD-AVC network service to be upgraded. Copy to one of the following locations, depending on the platform type:

- CSR1000V: **bootflash**
- ASR1000 Series or ISR4000 Series: **harddisk**

  harddisk refers to the SSD or HD specified in the system requirements for the platform (System Requirements: SD-AVC Network Service Host, on page 12).

3. On the device, verify the MD5 checksum of the downloaded package. The correct MD5 checksum value apears on the Download Software page when downloading the package.

   verify /md5 bootflash:*ova-filename.ova*

   Example:

   ```
   Device#verify /md5 bootflash:iosxe-sd-avc.2.1.0.ova
   ..........................................................................Done!
   verify /md5 (bootflash:iosxe-sd-avc.2.1.0.ova) = d8b7af1b163ccc5ad28582a3fd86c44e
   ```

4. Deactivate the service. This step stops the service but does not erase the database of compiled application data.

   service sd-avc deactivate

5. Verify that the service has been deactivated.

   service sd-avc status

   The following output confirms that the service has been deactivated:

   ```
   Service SDAVC is installed, configured and deactivated
   ```

6. On the host router, execute the following command to extract and install the OVA package. By default, it is installed on the same storage device where the OVA package is stored.

   service sd-avc upgrade package *disk-with-OVA*:*OVA-filename*

   Table 10: Command Details

CLI keyword/argument	Description
disk-with-OVA	Specify one of the following, according to the platform type. The location refers to where the OVA was stored in a previous step. • CSR: **bootflash** • ASR1000 Series or ISR4000 Series: **harddisk**
OVA-filename	Downloaded OVA file.

   Examples:

   - For Cisco CSR1000V router:

     ```
     service sd-avc upgrade package bootflash:iosxe-sd-avc.2.1.0.ova
     ```

   - For Cisco ASR1000 Series or ISR4000 Series routers:

     ```
     service sd-avc upgrade package harddisk:iosxe-sd-avc.2.1.0.ova
     ```

7. (Optional) During the upgrade process, view the service status.

   service sd-avc status

   During the upgrade, the following output indicates that the service is being installed:

   ```
   Service SDAVC is installing..., configured and deactivated
   ```

   The following output indicates that the upgrade is complete:

   ```
   Service SDAVC is installed, configured and deactivated
   ```

8. Activate the service.

   service sd-avc activate

   Example:

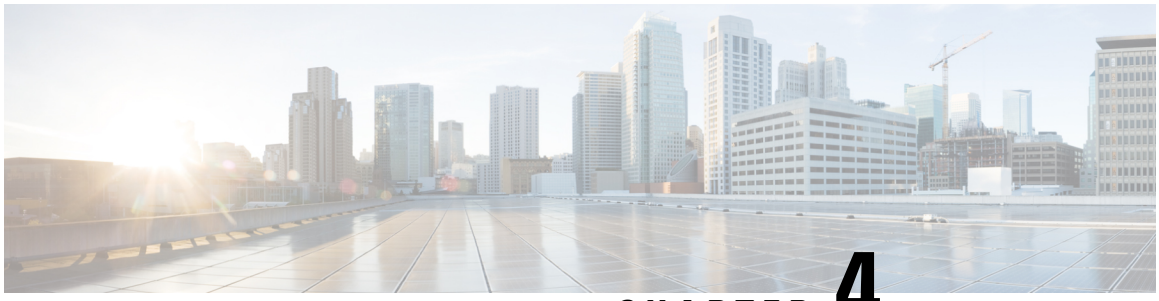
   ```
   service sd-avc activate
   ```

9. Verify that the status of the SD-AVC network service is activated.

   service sd-avc status

   If upgrade and activation were successful, the displayed status is:

   ```
   SDAVC service is installed, configured and activated
   ```



CHAPTER 4

Unconfiguring or Uninstalling the SD-AVC Network Service

Unconfiguring the SD-AVC Network Service

Use the following procedure to unconfigure the SD-AVC Network Service on the router hosting the service. Unconfiguring the service is necessary before changing the SD-AVC Network Service configuration.

1. Deactivate the service. This step stops the service but does not erase the database of compiled application data.

   service sd-avc deactivate
2. Verify that the service has been deactivated.

   service sd-avc status

   The following output confirms that the service has been deactivated:

   ```
   Service SDAVC is installed, configured and deactivated
   ```
3. Unconfigure the service.

   service sd-avc unconfigure
4. Verify that the service has been unconfigured.

   service sd-avc status

   The following output confirms that the service has been unconfigured:

   ```
   Service SDAVC is installed, not configured and deactivated
   ```

Uninstalling the SD-AVC Network Service

Use the following procedure to uninstall the SD-AVC Network Service on the router hosting the service.

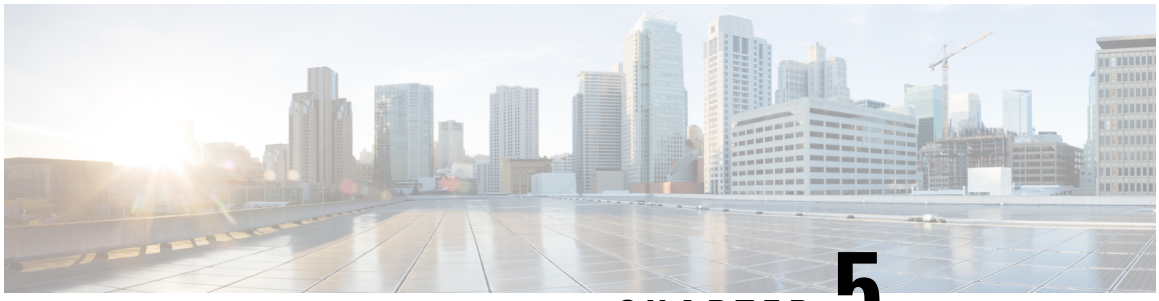
1. Deactivate and unconfigure the SD-AVC Network Service. Follow the full procedure in: Unconfiguring the SD-AVC Network Service, on page 23
2. Uninstall the service. This step deletes all information from the SD-AVC database for this SD-AVC Network Service.

   service sd-avc uninstall
3. Verify that the service has been uninstalled.

   service sd-avc status

   The following output confirms that the service has been uninstalled:

   ```
   Service SDAVC is uninstalled, not configured and deactivated
   ```



CHAPTER 5

Configuring Network Devices

Configuring Network Devices to Use SD-AVC

After the SD-AVC Network Service has been set up, use the information in this section to check the prerequisites for Cisco devices in the network to operate with the SD-AVC Network Service. Then activate and configure SD-AVC on the devices. This activates an SD-AVC agent that operates on the devices to communicate with the SD-AVC Network Service.

After configuration is complete, verify the status of each device using the SD-AVC Dashboard:

Dashboard > **Application Visibility** page > **SD-AVC Monitoring**

For High Availability SD-AVC, which employs more than one SD-AVC Network Service, see SD-AVC High Availability, on page 31.

System Requirements: Network Devices Using SD-AVC

The following table describes the supported platforms and requirements for network devices to operate with SD-AVC. When operating with SD-AVC, network devices run the SD-AVC agent, which manages communication between the devices and the SD-AVC Network Service.

Table 11: Network Device Requirements

Platform	Recommended OS (extended maintenance release trains only)
Cisco ASR1001-X Aggregation Services Router	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)

Platform	Recommended OS (extended maintenance release trains only)
Cisco ASR1002-X Aggregation Services Router	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)
Cisco ASR1001-HX Aggregation Services Router	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)
Cisco ASR1002-HX Aggregation Services Router	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)
Cisco 1100 Series Integrated Services Routers	Cisco IOS XE Fuji 16.9.1 or later
Cisco ISR4000 Series Integrated Services Routers: 4221, 4321, 4331, 4431, 4451	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)
Cisco Integrated Services Virtual Router	Cisco IOS XE Fuji 16.9.1 or later
Cisco CSR1000V Cloud Services Router	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)
Cisco Route Processor RP2, operating on Cisco ASR1004, ASR1006, or ASR1013	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)
Cisco Route Processor RP3, operating on Cisco ASR1004, ASR1006, or ASR1013	Cisco IOS XE Fuji 16.9.1 or later Cisco IOS XE Everest 16.6.4 or later (See note 1.)

Note

1. Cisco IOS XE 16.6.3 is supported, but with limited SD-AVC functionality. IOS XE 16.6.4 adds support for: Unclassified Traffic Discovery, source interface configuration, and improved scale. For questions about support for specific OS releases, please contact the SD-AVC team at:

   cs-nbar@cisco.com

Connectivity

For connectivity requirements and procedures, see Configuring Connectivity, on page 13.

Configuration Prerequisites: Network Devices Using SD-AVC

Network devices participating with SD-AVC run an SD-AVC agent (see SD-AVC Architecture, on page 7).

SD-AVC functionality depends on receiving application statistics from each participating network device. Application statistics are collected on each interface (on participating devices) on which one of the following is enabled: Cisco Performance Monitor, Easy Performance Monitor (ezPM), PfR policy, or Protocol Discovery. Each of these activates NBAR2 on the interface.

Depending on the Cisco solution in place, application statistics must be collected as follows:

- **IWAN solution**: (No additional user configuration required) Collection of application statistics is enabled by the use of Easy Performance Monitor (ezPM) and PfR policy.
- **Application Assurance solution**: (No additional user configuration required) Collection of application statistics is enabled by the use of Performance Monitor or Easy Performance Monitor (ezPM), and PfR policy.
- **EasyQoS**: (Requires user configuration) Configure Protocol Discovery on WAN-side interfaces.

Activating the SD-AVC Agent

Use the following procedure on a device in the network to activate the SD-AVC agent, enabling the device to communicate with the SD-AVC Network Service.

Note See system requirements for network devices operating with SD-AVC .

Note The term, SD-AVC Network Service, refers to the virtual service that operates on a host device and performs SD-AVC functions, such as aggregating application data. The **avc sd-service** command used in this procedure does not refer to the SD-AVC Network Service.

1. Activate SD-AVC.

   avc sd-service

   Example:

   ```
   (config)#avc sd-service
   ```

2. Configure the segment (group of devices that share the same purpose, such as routers within the same hub).

   segment cisco

   Example:

   ```
   (config-sd-service)#segment cisco
   ```

3. Enter controller mode to configure the agent to use the SD-AVC Network Service (not related to the **avc sd-service** command used in an earlier step).

   controller

   Example:

   ```
   (config-sd-service)#controller
   ```

4. Enter the service-IP used when the SD-AVC Network Service (running on a host device) was set up.

   address *service-ip*

Note For a high availability (HA) configuration, more than one SD-AVC Network Service is specified in this step. See: SD-AVC High Availability, on page 31

   Example:

   ```
   (config-sd-service-controller)#address 10.56.196.146
   ```

5. Configure VRF.

   vrf vrf_mgmt

   Example:

   ```
   (config-sd-service-controller)#vrf vrf_mgmt
   ```

   The device is now configured to operate with SD-AVC, and begins:

   - Sending collected application data to the SD-AVC Network Service
   - Receiving application rules packs periodically from the SD-AVC Network Service

6. See Scenarios that Benefit from Source Interface Configuration, on page 94 to determine whether to specify a source interface for SD-AVC traffic.
7. Using the SD-AVC Dashboard confirm that the router appears as a device in the network.

Configuration Example

The following is an example of the CLI steps used to configure the SD-AVC agent on a device.

```
(config)#avc sd-service
(config-sd-service)#segment cisco
(config-sd-service)#controller
(config-sd-service-controller)#address 10.56.196.146
(config-sd-service-controller)#vrf vrf_mgmt
```

Deactivating the SD-AVC Agent

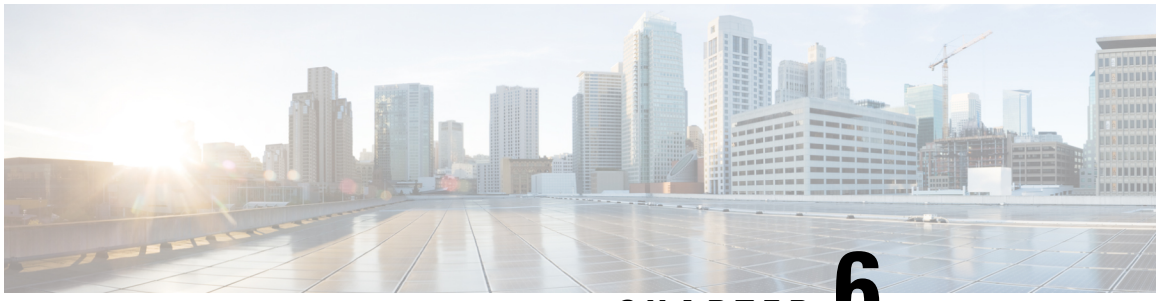
Use the following procedure on a device in the network to deactivate the SD-AVC agent and clear any SD-AVC agent configuration details that have been entered. This stops SD-AVC functionality on the device, and the device stops communicating with the SD-AVC network service.

1. Deactivate SD-AVC and remove SD-AVC agent configuration.

no avc sd-service

Example:

```
(config)#no avc sd-service
```



CHAPTER 6

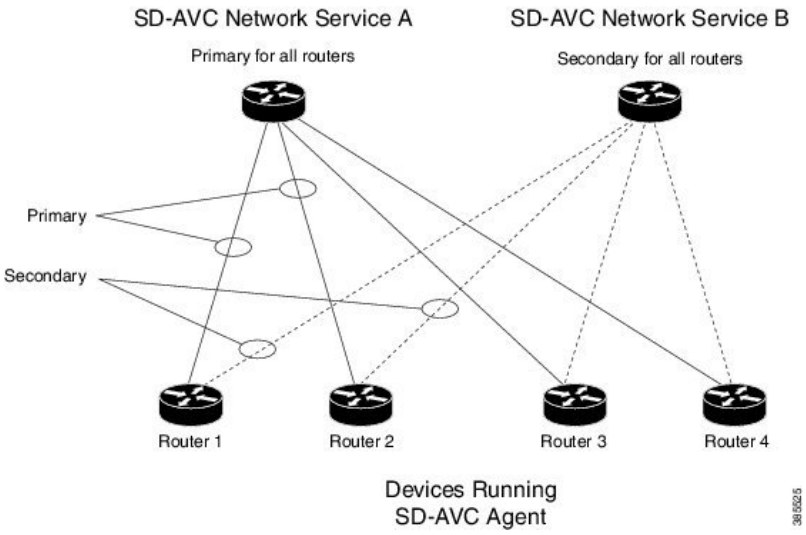
SD-AVC High Availability

SD-AVC supports a high availability (HA) configuration, using more than one SD-AVC network service. Each network device operating with SD-AVC, and consequently running the SD-AVC agent, designates a primary and secondary SD-AVC network service. If the primary SD-AVC network service becomes unavailable, the device fails over to the secondary service.

In the event of failover, the secondary SD-AVC network service receives the application data (state) maintained by the SD-AVC agents on participating network devices. This provides SD-AVC a degree of resilience, enabling the secondary network service to receive previously aggregated data and resume operation where the primary network service left off. In addition, because each SD-AVC agent maintains its state locally, classification of traffic on each device continues seamlessly during the failover from primary to secondary network service.

For all devices in the network that are operating with SD-AVC, it is recommended to use the same primary SD-AVC network service.

Figure 4: Primary and Secondary SD-AVC Network Services in High Availability Configuration



SD-AVC Network Services Collect Application Data Separately

Each SD-AVC network service collects application data from the devices that are using it as their active service. Multiple SD-AVC network services do not share application data with each other directly. So if the

primary service becomes unavailable, the agents that were using it fail over to the secondary service, and that service begins collecting application data from the agents.

- Configuring High Availability SD-AVC, on page 32
- Switchover Between Primary and Secondary SD-AVC Network Services, on page 32

Configuring High Availability SD-AVC

Setting up SD-AVC in a high availability configuration requires two steps that differ from a non-HA configuration.

1. Set up more than one SD-AVC Network Service. For information about setting up an SD-AVC Network Service, see Installation Overview, on page 11.
2. When configuring a device to use SD-AVC, specify primary and secondary SD-AVC Network Services with the **address** command. In other respects, configuring the device is identical to a non-HA configuration. For information about setting up a device, see Configuring Network Devices to Use SD-AVC, on page 25. The configuration commands are shown below.

```
avc sd-service
segment cisco
controller
address primary-network-service-ip secondary-network-service-ip
vrf vrf_mgmt
```

Example:

```
(config)#avc sd-service
(config-sd-service)#segment cisco
(config-sd-service)#controller
(config-sd-service-controller)#address 10.56.196.146 10.56.196.150
(config-sd-service-controller)#vrf vrf_mgmt
```

Switchover Between Primary and Secondary SD-AVC Network Services

If the primary SD-AVC network service for a device becomes unavailable, the device switches over to its secondary network service.

Note The primary SD-AVC network service may become unavailable either by unexpected failure, or for a planned outage, such as for an upgrade.

Appearance in Dashboard

After the switchover, the SD-AVC Dashboard for the secondary network service displays the device. To indicate that the device is in a switchover state, the **Application Visibility** page > **SD-AVC Monitoring** pane shows a yellow warning indicator. Clicking the warning indicator shows device warnings.

Functionality

After switchover, the secondary SD-AVC network service handles all operations for the device, including:

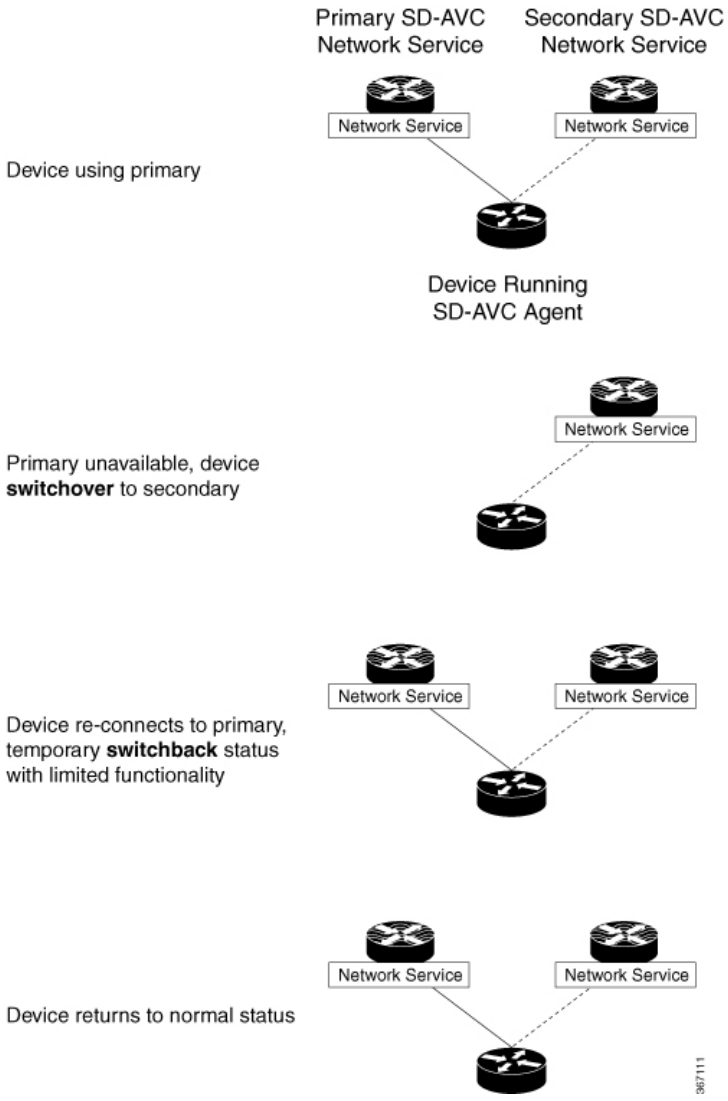
- Collecting traffic data from the device
- Displaying the traffic data
- Deploying Protocol Packs to the device if necessary

Returning to the Primary

When the primary SD-AVC network service becomes available again, the device returns to the primary network service.

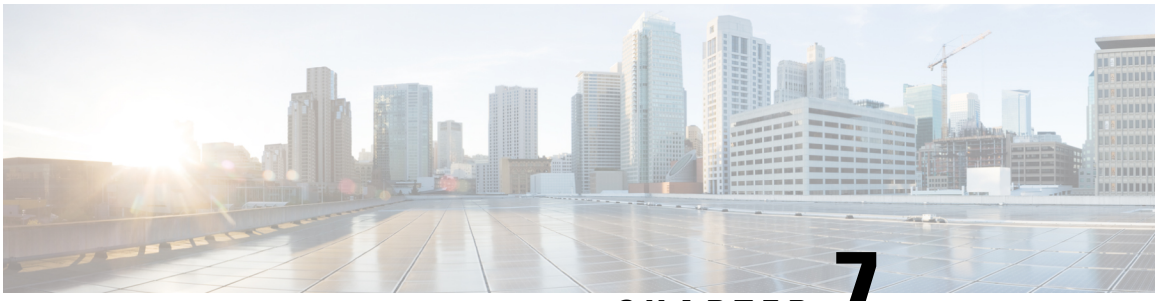
For a temporary period after re-connecting, the device status is **switchback**.

During the temporary **switchback** period, no Protocol Packs can be deployed to the device.



PART III

Part: Use



CHAPTER 7

Using SD-AVC

Using SD-AVC

Functionality	See...
Connect to the SD-AVC Dashboard	Connecting to the Dashboard, on page 37
View traffic analytics interactively, monitor devices operating with SD-AVC	Application Visibility Page, on page 38
Upload and deploy Protocol Packs	Protocol Packs Page, on page 43
View details of external sources of application classification	External Sources Page, on page 45
View system information, application rules, and debugging tools	Serviceability Page, on page 45 Application Rules Page, on page 46

Connecting to the Dashboard

Using a browser (Chrome recommended) with access to the device hosting the SD-AVC Network Service, open the SD-AVC Dashboard. The Dashboard is accessible using the service IP configured when setting up the SD-AVC Network Service, and port 8443, in the format:

https://<service-ip>:8443

Example:

```
https://10.56.196.153:8443
```

Note The SD-AVC Dashboard uses the same authentication as the platform hosting the SD-AVC Network Service. The host platform may use locally configured usernames and passwords, or it may use other methods, such as an Authentication, Authorization, and Accounting (AAA) server.

If prompted, enter the username and password used on the host platform.

Application Visibility Page

The **Application Visibility** page shows network activity handled by the devices in the network operating with SD-AVC, as well as displaying any warnings or errors for each device.

Table 12: Top of Window

Information/Control	Description
All Devices	Indicates that the application data displayed in this window includes traffic handled by all devices in the network that are operating with SD-AVC.
Filter	Filters the displayed application data to include only a single segment or a single device. (A network segment is a group of devices that share the same purpose, such as routers within the same hub.)
Time Range	Time range for application data displayed on this page.

Table 13: Summary Pane

Information/Control	Description
Classification Score	Last measured classification quality score for the device. This indicates the degree of classification quality (specificity), calculated according to traffic volume. Higher score indicates better quality.
Unclassified Traffic Discovery button ()	Displays details of unclassified traffic. See Viewing Unclassified Traffic Details, on page 41. To return, use the menu in the **Timeline** pane.
First Packet Classification	Ratio of flows classified on the first packet, to total TCP/UDP flows.
Total Usage	Total traffic volume handled in the selected time range.
SD-AVC Coverage Ratio	Ratio of flows covered by the SD-AVC application rules pack, to the total number of TCP/UDP flows.
Asymmetric Index	Last measured degree of asymmetry seen by device. This is the ratio of asymmetric flows to total flows for TCP and DNS traffic. 0 is least asymmetry, and 10 is highest asymmetry.

Information/Control	Description
Timeline	Graph of one of the following (select in dropdown menu): • Bandwidth • Classification score • First packet classification score • SD-AVC coverage ratio • Unclassified Traffic

Table 14: Applications by Usage Pane

Information/Control	Description
Table of applications	Usage and business relevance for each network application. Select one or more applications to display data for the applications in the **Timeline** pane. Use the Search field to filter the display of traffic.

Table 15: SD-AVC Monitoring Pane

Information/Control	Description
Note: When filtering to display data for a single segment or device, this pane displays information for that segment or device.	
Segment	Network segments. Click to filter display by a network segment.
Devices	Number of devices in the network. Click the magnifying glass to list devices, and for filtering options. Device warnings and alerts. Click the warning/alert for details
External Sources	Status of external sources, such as MS-Office365. Click MS-Office365 to display its configured DNS servers. See MS-Office365 Connector, on page 42.
Installed Protocol Packs	Protocol Packs installed on devices in the network.

Table 16: Business Relevance Pane

Information/Control	Description
Business Relevance Graph	Indicates portions of traffic classified as: • Business-relevant • Business-irrelevant • Default

Unclassified Traffic Analysis and Discovery

Background

The **SD-AVC Dashboard** > **Application Visibility** page shows a summary of network traffic, including a table of network applications, organized by network usage.

Traffic that has been identified and classified as belonging to a specific network application appears in the table by name.

Traffic that is not classified by Protocol Pack or external sources (example: MS-Office365) is called unclassified traffic. Unclassified traffic reduces the traffic classification score. Unclassified traffic appears as:

Label	Description
HTTP	Generic host, HTTP traffic
SSL	Generic host, SSL/HTTPS traffic
Unknown	Unknown socket

In the following example, WebEx Meeting traffic has been identified. Unclassified traffic is listed as **HTTP** and **Unknown**.

Application	Usage	Business Relevance
HTTP	0.00% (3.96 MB)	default
WebEx Meeting	6.84% (91.58 GB)	relevant
Unknown	6.35% (84.98 GB)	default

Partial Classification of Traffic

To improve traffic visibility and the classification score, SD-AVC analyzes top hosts and sockets that appear in unclassified traffic. For those using significant bandwidth, it provides a best-effort partial classification of the otherwise unclassified traffic. The process is dynamic, adapting to the network traffic of a given period.

Unclassified traffic that impacts the classification score by 1% or more meets the threshold for partial classification.

On the **SD-AVC Dashboard** > **Application Visibility** page, the partial classification appears as host or socket information in the traffic table.

Example:

- Unclassified HTTP traffic from the **am.cisco.com** domain
- Traffic on an unknown socket, with **source 128.107.107.107**, **port 50000**, using the **UDP** transport protocol

Application	Usage	Business Relevance
HTTP > am.cisco.com	7.44% (99.60 GB)	default
WebEx Meeting	6.84% (91.58 GB)	relevant
Unknown	6.35% (84.98 GB)	default
Unknown > 128.107.107.107:50000_UDP	1.94% (25.97 GB)	default

In the table, **HTTP**, **SSL**, or **Unknown** can refer to a single unclassified flow or an aggregate of numerous unclassified flows. In the example, numerous unclassified flows are represented by a single line: **Unknown**. The unclassified flows included in that line are each below the threshold for automatic partial classification, but together they total 6.35% of bandwidth.

Viewing Unclassified Traffic Details

On the **SD-AVC Dashboard** > **Application Visibility** page, in the **Summary** pane, click the **Unclassified traffic discovery** button () to display detailed information for unclassified and partially classified traffic.

- The timeline changes mode to show unclassified traffic.
- The traffic table shows all unclassified and partially classified traffic.

As with the default view, you can select items in the traffic table to display their contribution to total traffic bandwidth in the timeline.

To return to the default view, select **Bandwidth** from the dropdown menu in the **Timeline** pane.

Improves Visibility, Does Not Affect Policy

Partial classification of traffic, as described here, improves application visibility, and improves the overall classification score.

Partial classification is dynamic, adapting to current traffic, so it not applied to security (firewall) or QoS policies.

Features affected by partial classification:

- Application visibility (FNF, performance-monitor, ezPM, MACE, ...)

Features not affected by partial classification:

- MQC/QoS
- WAAS
- Performance Routing (PfR)
- NAT

Notes and Limitations

See Notes and Limitations: Classification, on page 48.

MS-Office365 Connector

MS-Office365 Connector improves classification of Microsoft Office 365 traffic. It requires connectivity between the device hosting the SD-AVC network service, and one or more DNS servers. By default, SD-AVC has two Cisco OpenDNS DNS servers configured (208.67.222.222 and 208.67.220.220).

Optionally, you can add addition DNS servers or proxy DNS servers, as described below.

Adding DNS Servers

If you need to add additional DNS servers, configure them on the platform hosting the SD-AVC network service, using the **ip name-server** command, before installing the network service.

Example (adds two DNS servers):

```
(config)#ip name-server 198.51.100.1 198.51.100.2
```

Adding a Proxy DNS Server

If you need to configure a proxy DNS server, configure it on the platform hosting the SD-AVC network service before installing the network service, as follows:

1. Before configuring a proxy DNS server, remove any standard name servers.

   no ip name-server

2. Configure the host to act as a DNS server.

   ip dns server

3. Configure the external DNS server address.

   ip dns spoofing *address*

In the following example, 198.51.100.3 is the external DNS server.

```
(config)#no ip name-server
(config)#ip dns server
(config)#ip dns spoofing 198.51.100.3
```

Viewing DNS or Proxy Servers

To view the configured DNS or proxy servers:

In the **SD-AVC Dashboard** > **Application Visibility** page > **SD-AVC Monitoring** pane, click **MS-Office365 Connector**.

A window opens, displaying a list of the default DNS servers, and any manually configured DNS and proxy servers.

Manually configured DNS servers have higher priority than the default servers. The priority of manually configured DNS servers is the order in which they were added—the first server added has the highest priority. If the highest-priority DNS server on the list is not available, SD-AVC uses the next in the list.

SD-AVC System Time and Displayed Times

SD-AVC receives the UTC time from the host platform. UTC times appear in activity logs.

The SD-AVC Dashboard displays times according to the local time zone of the PC that is accessing the Dashboard. Times appear at the bottom left of the Dashboard, in timelines of network activity, and so on.

Note If the host platform clock is set incorrectly, the times shown in logs and in the Dashboard will be incorrect.

Setting the System Time on the Host Platform

To set the system time, use:

clock set *hh*:*mm*:*ss day month year*

Example:

```
#clock set 12:13:00 27 Mar 2018
```

Setting the Time Zone on the Host Platform

Note SD-AVC receives the time from the host platform as UTC.

To set the time zone (hour offset from UTC), use the following in config mode. The timezone-name is arbitrary.

clock timezone *timezone-name offset-from-UTC*

Example:

```
(config)#clock timezone NYC -5
```

Showing the time includes the configured offset (-5 hours for New York (NYC) in the example).

Example:

```
#show clock
15:47:59.481 NYC Thu Mar 22 2018
```

To remove the time zone setting and use UTC time:

```
(config)#no clock timezone
```

Protocol Packs Page

The **SD-AVC Dashboard** > **Protocol Pack Update** page lists devices in the network, with Protocol Pack information for each.

Click **Manage & Deploy** to:

- Upload Protocol Pack files to the repository (for deploying to devices).

- Deploy Protocol Packs to devices in the network.

Understanding Protocol Pack Files

Cisco releases Protocol Packs on an ongoing basis. Each Protocol Pack release provides updates that expand and improve AVC application recognition. Typically, it is recommended to use the latest Protocol Pack compatible with the OS running on a device. The Protocol Library page indicates the latest Protocol Pack and provides compatibility information.

Protocol Packs are available using the Cisco Download Software tool. When using the tool, specify a platform and then navigate to software downloads for the platform.

Protocol Pack filename format:

pp-adv-<platform-type>-<OS>-<engine-id>-<protocol-pack-version>.pack

Platform type may be, for example, asr1k, csr1000v, or isr4000. However, a Protocol Pack may be installed on any compatible device, even if that device is not indicated by the filename.

Uploading Protocol Packs to the SD-AVC Repository

Use the SD-AVC network service to deploy Protocol Packs to participating devices in the network.

Step 1 Select a Protocol Pack to deploy (typically the latest Protocol Pack compatible with the OS running on a device). See the Protocol Library page for compatibility information.

Step 2 Download the Protocol Pack using the Cisco Download Software tool. In the filename of the downloaded Protocol Pack, note the engine ID.

Step 3 In the SD-AVC Dashboard, upload the Protocol Pack file into the Protocol Pack repository. The repository is stored on the device hosting the SD-AVC network service.

Protocol Packs page > **Manage & Deploy** button > **Protocol Pack Repository** > **Upload**

Deploying Protocol Packs to Devices

Note In SD-AVC high availability configurations, if a device switches over to its secondary SD-AVC network service, then switches back to its primary, the device has a temporary “switchback” status. During this brief period, you cannot deploy Protocol Packs to the device. See SD-AVC High Availability, on page 31.

Step 1 Open the SD-AVC Dashboard Protocol Packs page.

Protocol Packs page > **Manage & Deploy** button > **Deploy to...**

Step 2 In the **Protocol Pack Repository** pane, select a Protocol Pack or the **Builtin** option.

The **Builtin** option re-installs the original built-in Protocol Pack that was included with the OS (for example, Protocol Pack 33.0.0 for Cisco IOS-XE Fuji 16.7.1).

Step 3 In the **Deploy to...** pane, select a segment and one or more devices, then click **Continue**.

Note After selecting a Protocol Pack, only devices running an IOS version compatible with the Protocol Pack can be selected.

Step 4 Select the time to deploy the Protocol Pack(s), then click **Continue**.

Step 5 Review the deployment plan and click the **Deploy** button.

Note To return to an earlier step, click the step number.

External Sources Page

The External Sources page displays additional sources of application information used for classifying network traffic.

Source	Description
MS Office 365 Cloud	Provides domain names used by Microsoft Office 365. Click the **View Details** button for details about each domain. See MS-Office365 Connector, on page 42. **Note** Must be enabled to view details.

Serviceability Page

The Serviceability page provides system information, debugging tools, and detailed information about the application rules used to classify network traffic.

Tool	Description	
System	System information, such as disk, memory, and CPU status, and system logs.	
System	System information, such as disk, memory, and CPU status, and system logs.	
	Vertical Debug	Create rules to track specific traffic criteria, for debugging.
SD-AVC Message Capture	Collect and download SD-AVC messages (between the SD-AVC network service and one or more agents).	

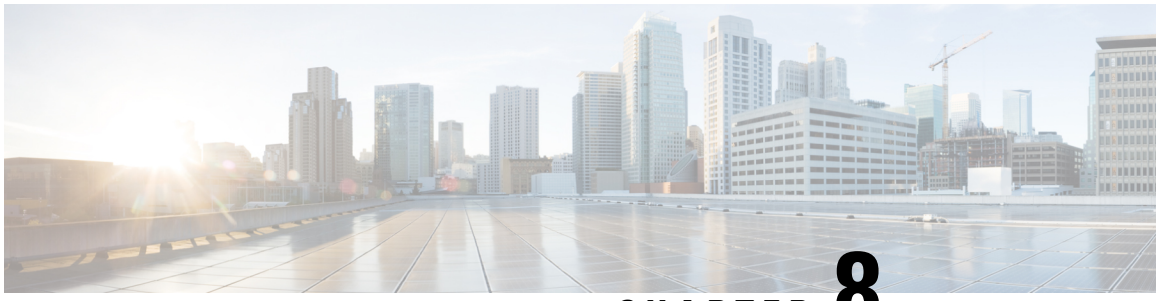
Tool	Description
Application Rules	Detailed information about the application rules used to classify network traffic. Application Rules Page, on page 46

Application Rules Page

The SD-AVC network service collects traffic classification data from network devices. The network service merges the data and sends it to devices as an application rules pack (see Operation, on page 7). This page shows the merged application rules data.

Segment: Select the network segment using the dropdown menu at the top right.

Field	Description
IP	Server IP
Port	Port
VRF	VRF name, if applicable
Application Name	Application name, defined by: • Protocol Pack protocol • User-defined protocols
Entry Type	Network cache type: • **L3** • **socket-cache**
Source	Protocol/application: • **network**: Identification of flow by Protocol Pack • **dynamic**: Identification of flow by user-defined application • **ac_hosts** or **ac_sockets**: Tracking of flow by Unclassified Traffic Discovery feature
Rating	Number of significant flow (session) hits in the network layer
Transport	Transport protocol
TTL	Time to Live: Timespan (in cycles) for tracking the socket • If there is active traffic for the socket, the TTL remains at maximum value of 384. • If there is no active traffic for the socket, the TTL value is decremented over time.



CHAPTER 8

SD-AVC Notes and Limitations

Notes and Limitations: General

Note/Limitation	Description
Maximum number of participating network devices	Maximum number of network devices participating with SD-AVC (running the SD-AVC agent): 4000

Notes and Limitations: Setup

Note/Limitation	Description
MD5 checksum of OVA download	When installing or upgrading the SD-AVC network service, download the OVA package, copy it to the device that will host the network service, then verify the MD5 checksum of the package before installing. The correct MD5 checksum value apears on the Download Software page when downloading the package.
Network Service gateway interface attached to VRF	For the SD-AVC Network Service, running on a host device, if the host interface that is used as a gateway interface is attached to a VRF, see Operating the SD-AVC Network Service with Host Interface Attached to a VRF, on page 65 for configuration details.
Running and startup configurations of participating devices	SD-AVC adds two lines to the running and startup configurations of participating devices: • To enable the MS-Office365 Connector feature, which improves classification of Microsoft Office traffic: **ip nbar protocol-pack bootflash:sdavc/sdavc_ppdk.pack force** • When SD-AVC deploys Protocol Packs to a device: **ip nbar protocol-pack harddisk:sdavc/***protocol-pack-name***.pack**

Notes and Limitations: Classification

Note/Limitation	Description
Interval before sending application data	SD-AVC requires a few minutes to learn from the network traffic before the application data is sent to the SD-AVC Network Service and compiled at the network level. See SD-AVC and Application Recognition, on page 8.
SD-AVC application rules pack less relevant for client-to-client traffic	SD-AVC provides application classification for server-based applications. The SD-AVC application rules pack is less relevant for client-to-client traffic, which is more granular and dynamic. Client-to-client traffic is classified by NBAR2 running on each network element.
Proxy or CDN	In the case of a proxy or content delivery network (CDN), multiple applications may use the same IP/port combination. The network devices themselves classify such traffic fully. However, for these applications, the SD-AVC agent operating on a device may report application data to the SD-AVC network service with a lesser degree of detail: they may be reported with less detailed classification granularity or not at all.
Reported bandwidth of Unclassified Traffic Discovery	For traffic that appears in the Unclassified Traffic view, the reported bandwidth is based on samples and may not be accurate in some cases. See Unclassified Traffic Analysis and Discovery, on page 40.
High-stress flows may not be discovered by the Unclassified Traffic Discovery feature	High-stress flows that require a large amount of system resources may be excluded from the traffic reported in the **Unclassified Traffic** view. For example, the Timeline may show a high-bandwidth of unknown/generic traffic that is not reported in the table. This is done to minimize the utilization of resources in case of high stress flows and skip the discovery mechanism. See Unclassified Traffic Analysis and Discovery, on page 40.

Notes and Limitations: High Availability

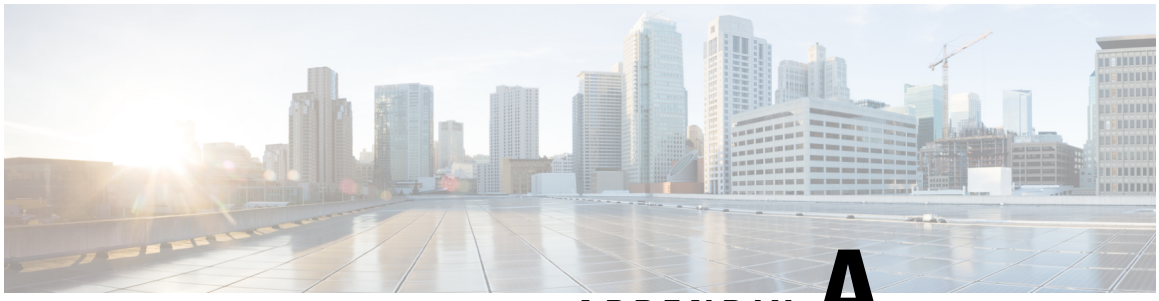
Note/Limitation	Description
Protocol Pack deployment during high availability switchover	In SD-AVC high availability configurations, if a device switches over to its secondary SD-AVC network service, then switches back to its primary, the device has a temporary “switchback” status. During this brief period, you cannot deploy Protocol Packs to the device. See SD-AVC High Availability, on page 31.

Notes and Limitations: Protocol Pack Deployment

Note/Limitation	Description
Cisco ISR4000 Series: hard disk limitation	Protocol Pack files must be loaded on the boot flash. For ISR4000 routers operating with SD-AVC, it is not recommended to install a hard disk. Doing so will cause Protocol Pack deployment by SD-AVC to fail.

Notes and Limitations: REST API

Note/Limitation	Description
Custom application source	In the initial release of the REST API, only one source is supported.
Total number of custom applications available	For each network segment: • Maximum custom applications: 1500 • Maximum L3L4 rules: 20000 • Maximum serverNames: 50000
High-availability SD-AVC configurations	High-availability SD-AVC configurations are supported. On the primary and secondary SD-AVC network services, configure the same REST API-based custom application configuration.



APPENDIX A

Troubleshooting SD-AVC

This section provides several SD-AVC troubleshooting scenarios. If this information does not provide a solution, contact Cisco TAC for assistance.

Troubleshooting Overview

The following tables describe troubleshooting for issues with:

- SD-AVC network service

  (operates on a dedicated host)

- SD-AVC agent

  (operates on each participating device in the network)

- Connectivity

  (between network service and one or more devices in the network)

Table 17: Troubleshooting: SD-AVC Network Service

Problem	How it appears	Troubleshooting
SD-AVC network service: installation failure	SD-AVC not active, **sd-avc status** shows installation failure.	**Summary** Diagnose with **sd-avc status** and then **service sd-avc trace**. Possible issues: • Not enough memory: see system requirements • Not enough disk space: see system requirements **Troubleshooting Details** Troubleshooting Commands for Network Service Issues, on page 54 System Requirements: SD-AVC Network Service Host, on page 12
SD-AVC network service: activation failure	SD-AVC not active, **sd-avc status** shows activation failure.	**Summary** Diagnose with **sd-avc status** and then **service sd-avc trace**. Possible issue: Something may be using CPU resources. Ensure that nothing is using CPU resources. **Troubleshooting Details** Troubleshooting Commands for Network Service Issues, on page 54 Activation Failure Caused by Shared CPU Resources, on page 57
SD-AVC network service: configuration failure	SD-AVC not active, **sd-avc status** shows configuration failure.	**Summary** A VRF is attached to the interface used as the management interface on the device hosting the SD-AVC network service. Remove the VRF assignment from the management interface using: **interface** *interface* **no ip vrf forwarding** **Troubleshooting Details** Configuration Failure Caused by VRF, on page 59

Table 18: Troubleshooting: SD-AVC Agent Operating on Devices in the Network

Problem	How it appears	Troubleshooting
NBAR2 is not activated on the device	On the **Dashboard** > **Application Visibility** page, the **Timeline** graph of bandwidth shows no activity.	**Summary** NBAR2 is not active: Activate NBAR2 on the device. **Troubleshooting Details** NBAR2 Not Activated on Interfaces, on page 60

Error: More than one active session	When attempting to enable the agent, an error message indicates that there is an active session already.
	Example:
	`Device(config-sd-service)# controller`
	`%% NBAR Error: There is an active session already`
	`in sd-service-controller submode`
		Summary
		Close any interfering sessions.
		Troubleshooting Details
		Active Sessions Preventing Agent Configuration, on page 60

Table 19: Troubleshooting: Connectivity between SD-AVC Network Service and Devices in the Network

Problem	How it appears	Troubleshooting
UDP	Warning in: **Dashboard** > **Application Visibility** page > **SD-AVC Monitoring** pane	**Summary** Check UDP connectivity. **Troubleshooting Details** Problem with UDP Communication with Devices, on page 61
TCP	Warning in: **Dashboard** > **Application Visibility** page > **SD-AVC Monitoring** pane	**Summary** Check TCP connectivity. **Troubleshooting Details** Problem with TCP Communication with Devices , on page 62

Problem	How it appears	Troubleshooting
FTP	Warning in: **Dashboard** > **Application Visibility** page > **SD-AVC Monitoring** pane	**Summary** 1. Check FTP connectivity: **show avc sd-service info summary** 2. Verify FTP connectivity between the SD-AVC network service and the network device. This includes checking ACL, firewalls, and so on. 3. On the device, ensure that FTP connectivity is possible from the routable interface to the SD-AVC network service. To enable FTP connections from a specific interface, use: **ip ftp source-interface** *interface-name* **Troubleshooting Details** Problem with FTP Communication with Devices, on page 62

Table 20: Troubleshooting: Protocol Packs

Problem	How it appears	Troubleshooting
Failure to load Protocol Pack on a device	When deploying Protocol Packs to one or more devices, results page shows error, such as "out of sync."	**Summary** Load the Protocol Pack manually on the device to determine whether the Protocol Pack is valid. **Troubleshooting Details** Failure to Deploy Protocol Pack to Device, on page 64

Troubleshooting SD-AVC Network Service Issues

Troubleshooting Commands for Network Service Issues

The following commands are helpful for troubleshooting SD-AVC network service issues. Execute the commands on the network service host device. The output may indicate any installation or configuration problems.

Table 21: Summary

Command	Description
service sd-avc status	Status of SD-AVC network service installation, configuration, and activation
service sd-avc trace	Memory or disk problems
show virtual-service list	Activation errors
show virtual-service global	CPU and memory usage

Command Details: service sd-avc status

Execute the command on the network service host device.

Output indicates status of SD-AVC installation, configuration, and activation.

- Installation error:

```
Service SDAVC is uninstalled, not configured and deactivated
```

- Activation error:

```
Service SDAVC is installed, configured and Activate Failed
```

Command Details: service sd-avc trace

Execute the command on the network service host device.

Output indicates memory or disk problems.

- **Memory** problem (shown in bold below):

```
service sd-avc trace
2017/11/27 02:06:42.384 [errmsg] [3071]: UUID: 0, ra: 0, TID: 0 (noise):(2):
%VMAN-2-MACH_PARSE_FAILURE: Virtual Service[SDAVC]::Parsing::XML parsing failure::Unable
 to parse VM machin
e definition::Requests 3072 MB of memory which exceeds the maximum of
1024
2017/11/27 02:06:42.383 [errmsg] [3071]: UUID: 0, ra: 0, TID: 0 (noise):(2):
%VMAN-2-MEMORY_LIMIT_WARN: Virtual service (SDAVC) defines 3072 MB of Memory
 exceeding the maximum 1024 MB.
...
```

- **Disk** problem (shown in bold below):

```
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
per-VM mac address binding from FDB
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get mac
 binding from persistent DB file
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Could not retrieve
 HA disk info for VM 'SDAVC'
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Unable to locate
fdb attributes for vm(SDAVC)
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
per-VM storage info list from FDB
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
```

```
storage pool from persistent DB file
2017/11/27 03:36:52.499 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Virtual Service
failure log[SDAVC]::Install::The installation of the virtual service failed
```

Command Details: show virtual-service list

Execute the command on the network service host device.

Output indicates activation status (**failed** in this example):

```
Virtual Service List:
Name                   Status             Package Name
------------------------------------------------------------------------------
SDAVC                  Activate Failed    avc_iosxe_221533.ova
```

Command Details: show virtual-service global

Execute the command on the network service host device.

Output indicates virtual service CPU and memory usage:

Example showing a service using 5% of CPU:

```
show virtual-service global
Maximum VCPUs per virtual service : 1
Resource virtualization limits:
Name                              Quota     Committed     Available
------------------------------------------------------------------------
system CPU (%)                       75             5            70
memory (MB)                        3072           800          2272
bootflash (MB)                    20000          6764         10672
```

Installation Failure Caused by Memory or Disk

Component(s)

Device hosting the SD-AVC network service

Background

Memory or disk allocation issues can prevent successful installation of the SD-AVC network service.

Troubleshooting

1. Use **service sd-avc status** on the network service host device to check status of installation. If installation is unsuccessful, the output shows "Service SDAVC is uninstalled."

   ```
   service sd-avc status
   Service SDAVC is uninstalled, not configured and deactivated
   ```

2. Use **service sd-avc trace** on the network service host device to indicate whether the installation problem is due to **memory** or **disk**.

   - **Memory** problem:

```
service sd-avc trace
2017/11/27 02:06:42.384 [errmsg] [3071]: UUID: 0, ra: 0, TID: 0 (noise):(2):
%VMAN-2-MACH_PARSE_FAILURE: Virtual Service[SDAVC]::Parsing::XML parsing
failure::Unable to parse VM machin
e definition::Requests 3072 MB of memory which exceeds the maximum of
 1024
2017/11/27 02:06:42.383 [errmsg] [3071]: UUID: 0, ra: 0, TID: 0 (noise):(2):
%VMAN-2-MEMORY_LIMIT_WARN: Virtual service (SDAVC) defines 3072 MB of
Memory exceeding the maximum 1024 MB.
...
```

- **Disk** problem:

```
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
per-VM mac address binding from FDB
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
mac binding from persistent DB file
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Could not
retrieve HA disk info for VM 'SDAVC'
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Unable to locate
 fdb attributes for vm(SDAVC)
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
per-VM storage info list from FDB
2017/11/27 03:36:52.500 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to get
storage pool from persistent DB file
2017/11/27 03:36:52.499 [vman] [3222]: UUID: 0, ra: 0, TID: 0 (ERR): Virtual Service
 failure log[SDAVC]::Install::The installation of the virtual service
failed
```

Solutions

Table 22: Resolving Memory or Disk Errors

Problem	Solution
Memory error	Increase the device memory to the amount specified in System Requirements: SD-AVC Network Service Host, on page 12.
Disk error	Increase the size of the harddisk or bootflash (for CSR) device according to the requirements specified in System Requirements: SD-AVC Network Service Host, on page 12.

Activation Failure Caused by Shared CPU Resources

Component(s)

Device hosting the SD-AVC network service

Background

The platform hosting the SD-AVC network service should not have other virtual services operating. Sharing CPU resources with other virtual services can prevent successful activation.

Use **service sd-avc status** on the network service host device to check status of installation. If installation has succeeded, but activation is unsuccessful, the output shows "Activate Failed."

```
service sd-avc status
Service SDAVC is installed, configured and Activate Failed
```

Troubleshooting

Use **service sd-avc trace** on the network service host device to troubleshoot. The following output shows a problem (shown in bold) with activation, due to shared CPU.

```
service sd-avc trace
2017/11/26 15:46:49.133 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Failed to find domain
 SDAVC - state query
2017/11/26 15:46:49.133 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Domain not found: No
domain with matching name 'SDAVC'
2017/11/26 15:46:49.133 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Error from libvirt:
code=42
2017/11/26 15:46:48.131 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (note): VM (SDAVC) State
Transition: next_state: LIFECYCLE_ACTIVATE_FAILED
2017/11/26 15:46:48.131 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Virtual Service failure
 log[SDAVC]::Activate::Internal error::Machine definition customization failed
2017/11/26 15:46:48.131 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Machine definition
customization failed
2017/11/26 15:46:48.131 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Customization of common
 XML parameters failed
2017/11/26 15:46:48.131 [vman] [2224]: UUID: 0, ra: 0, TID: 0 (ERR): Customize CPU tunes:
Cannot commit CPU tunes
2017/11/26 15:46:48.131 [errmsg] [2224]: UUID: 0, ra: 0, TID: 0 (noise):(2):
%VMAN-2-CPUSHARES_LIMIT: Virtual Service[SDAVC]::CPU shares limit::The virtual
 service definition exceeds the maximum number of CPU shares::Defined:
75, available: 70
```

Use **show virtual-service global** to provide details. In this example, another process is using 5% of the CPU resources (shown in bold).

```
show virtual-service global
Maximum VCPUs per virtual service : 1
Resource virtualization limits:
Name                                 Quota     Committed     Available
-----------------------------------------------------------------
system CPU (%)                          75             5            70
memory (MB)                           3072           800          2272
bootflash (MB)                       20000          6764         10672
```

Solutions

Deactivate Interface Using CPU Resources

1. Check the running configuration using **show run** on the network service host device. If an active interface is using CPU resources, deactivate the interface.

   Example

   GigabitEthernet1 is using CPU resources.

   ```
   show run | section csr_mgmt
   virtual-service csr_mgmt
   ip shared host-interface GigabitEthernet1
   ```

```
activate
```

2. Deactivate the interface.

   Example

```
conf t
virtual-service csr_mgmt
no activate
no ip shared host-interface GigabitEthernet1
```

3. Repeat the installation of the SD-AVC network service.

Configuration Failure Caused by VRF

Component(s)

Device hosting the SD-AVC network service

Background

If the host interface that is used as a gateway interface for the SD-AVC network service is attached to a VRF, the SD-AVC network service installation may be successful, but a configuration step may fail.

Troubleshooting

1. Check VRF status of the SD-AVC network service gateway interface.

   Example showing a VRF configured on the gateway interface GigabitEthernet1:

```
interface GigabitEthernet1
ip vrf forwarding Mgt
ip address 10.56.196.177 255.255.252.0

service sd-avc configure gateway interface gigabitEthernet 1 service-ip 10.56.196.180
% Error: VRF 'Mgt' is configured on gateway. This type of configuration is not
 supported.
```

Solutions

Remove the VRF assignment from the management interface. Example:

```
interface GigabitEthernet1
no ip vrf forwarding
```

Troubleshooting SD-AVC Agent Issues

NBAR2 Not Activated on Interfaces

Component(s)

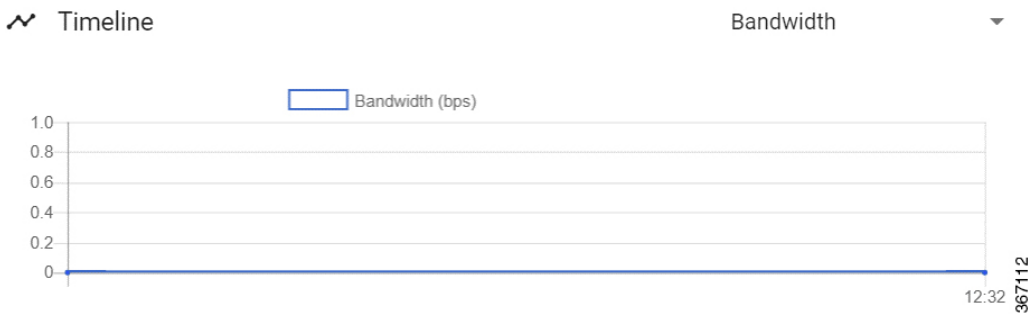
Devices in the network that are using SD-AVC

Background

The NBAR2 component must be active on any interface that processes network traffic, in order to report on traffic handled by the interface. For details, see Configuration Prerequisites: Network Devices Using SD-AVC, on page 27.

If NBAR2 is not active on an interface processing network traffic:

- The device will not report on any traffic on that interface.
- On the **Dashboard** > **Application Visibility** page, the **Timeline** graph of bandwidth will show no activity.



- The device will not receive application rules packs from the SD-AVC network service.

Troubleshooting

Verify that NBAR2 is active on interfaces that process network traffic.

Solutions

If necessary, activate NBAR2 on the interface(s).

Active Sessions Preventing Agent Configuration

Component(s)

Devices in the network that are using SD-AVC

Background

The SD-AVC agent must be enabled on any device participating with SD-AVC. This requires entering sd-service-controller submode on the device.

It is possible to connect to the device through multiple sessions. An error may occur in the following conditions, with an error message indicating the problem:

- One active session is in sd-service-controller submode.
- You attempt to open sd-service-controller submode in a new session.

Example:

```
Device(config)#avc sd-service
Device(config-sd-service)# segment sdavc
Device(config-sd-service)# controller
%% NBAR Error: There is an active session already in sd-service-controller submode
```

Solutions

Close any interfering active sessions.

1. On the device, use **show users** to display active sessions.
2. In the command output, note the line number of a session to close. Use **clear line** *line-number* to close a session.

Example:

```
Device#show users
  Line     User    Host(s)   Idle    Location
*  1       vty 0   prod      idle    00:00:00
                                     dhcp-10-11-12-13-14-15.cisco.com
   3       vty 2   prod      idle    1d04h 198.51.100.10

Device#clear line 3
[confirm]
[OK]

Device#show users
    Line   User    Host(s)   Idle    Location
*  1       vty 0   prod      idle    00:00:00
                                     dhcp-10-11-12-13-14-15.cisco.com
```

Troubleshooting SD-AVC Connectivity Issues

Problem with UDP Communication with Devices

Component(s)

SD-AVC network service

Devices in the network that use SD-AVC

Background

The SD-AVC Network Service uses UDP over port 50000 to communicate with the devices that it manages.

Troubleshooting

1. If a **Connection** warning appears in the SD-AVC Dashboard, for a specific device in the network, check connectivity on UDP port 50000. Warnings appear here:

   SD-AVC Dashboard > **Application Visibility** page > **SD-AVC Monitoring** pane

2. If no problem is found, contact Cisco TAC.

Solutions

Ensure that UDP connectivity is possible on port 50000 between the affected device and the SD-AVC network service.

Problem with TCP Communication with Devices

Component(s)

SD-AVC network service

Devices in the network that use SD-AVC

Background

The SD-AVC network service uses TCP over ports 20-21 (FTP) to communicate with the devices that it manages.

Troubleshooting

1. If an **Update** warning appears in the SD-AVC Dashboard, for a specific device in the network, check connectivity on TCP ports 20-21. Warnings appear here:

   SD-AVC Dashboard > **Application Visibility** page > **SD-AVC Monitoring** pane

2. If no problem is found, contact Cisco TAC.

Solutions

Ensure that TCP communication is possible over ports 20-21 (FTP) between the affected device and the SD-AVC network service.

Problem with FTP Communication with Devices

Component(s)

SD-AVC network service

Devices in the network that use SD-AVC

Background

The SD-AVC network service uses FTP to communicate with the devices that it manages.

A device with partial connectivity, but problems specific to FTP may show a warning in the SD-AVC Dashboard.

For FTP issues caused by connecting a device to an internal FTP server for non-SD-AVC FTP traffic, see Scenario: Internal FTP Server, on page 96.

Troubleshooting

1. If an **Update** warning appears in the SD-AVC Dashboard while the **Connection** status is green, for a specific device in the network, check the FTP connection status. Warnings appear here:

   SD-AVC Dashboard > **Application Visibility** page > **SD-AVC Monitoring** pane

2. On the device with the connectivity issue, use **show avc sd-service info summary** to check the FTP connection status. "Status: DISCONNECTED" in the output below shows an FTP connectivity problem.

```
show avc sd-service info summary

Status: DISCONNECTED

Device ID: csi-mcp-asr1k-4ru-32
Device segment name: cisco
Device address: 10.56.192.31

Active controller:
   Type  : Primary
   IP    : 64.103.125.30
   Status: Disconnected
   Last connection: Never
```

Solutions

Ensure that FTP communication is possible between the affected device and the SD-AVC network service.

1. Verify that nothing is preventing FTP network connectivity between the SD-AVC network service and the network device. This includes checking ACL, firewalls, and so on.

2. On the device with the **Update** warning, ensure that FTP connectivity is possible from the routable interface to the SD-AVC network service. To enable FTP connections from a specific interface, use:

   ip ftp source-interface *interface-name*

   Example:

```
ip ftp source-interface GigabitEthernet1
```

Troubleshooting Protocol Pack Issues

Failure to Deploy Protocol Pack to Device

Component(s)

SD-AVC network service

Cisco NBAR2 Protocol Packs

Background

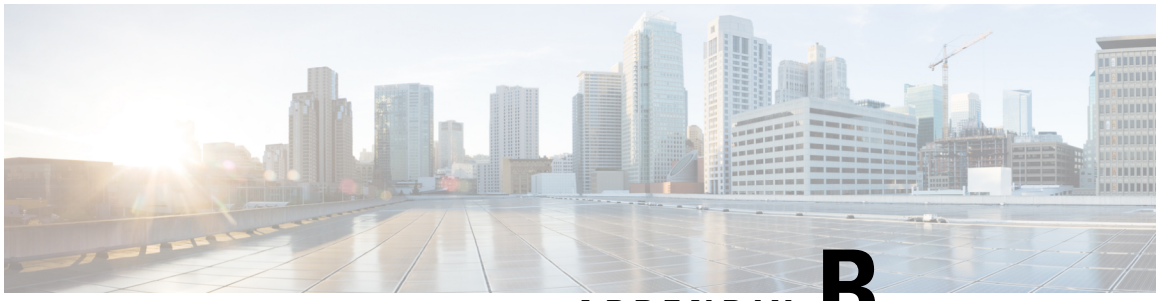
Use the SD-AVC network service to deploy Protocol Packs to one or more devices. See Deploying Protocol Packs to Devices, on page 44. When deploying Protocol Packs to one or more devices, if the deployment fails, the results page may show an error.

Troubleshooting

1. Load the Protocol Pack manually on the device indicated by the error to verfiy that the Protocol Pack is valid and can be loaded onto the device. This rules out any problems with the Protocol Pack file.

   ```
   (config)#ip nbar protocol-pack bootflash:pack_file_name.pack
   ```

2. If no problem is found, contact Cisco TAC.



APPENDIX B

Operating the SD-AVC Network Service with Host Interface Attached to a VRF

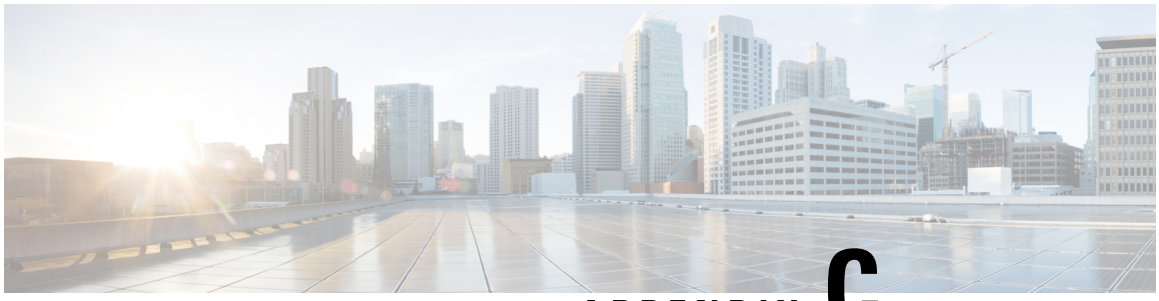
In specific use cases, it may be necessary to operate the SD-AVC Network Service on a host device on which the host interface that is used by SD-AVC as its gateway interface may be attached to a VRF. In this case, the typical installation command described in Installing the SD-AVC Network Service, on page 14 cannot be used, and manual configuration is required, using the following guidelines:

- Ensure that the virtual port group and gateway interface(s) are not on the same subnet.
- Assign the virtual port group and gateway interface(s) to a VRF.
- Ensure that the IP address of the SD-AVC network service (**guest IP** in the configuration steps below) is on the virtual port group subnet.

Example:

```
ip vrf Mgt
!
interface VirtualPortGroup31
ip vrf forwarding Mgt
ip address 10.56.197.221 255.255.255.0
!
interface GigabitEthernet1
ip vrf forwarding Mgt
ip address 10.56.196.169 255.255.255.0
!
virtual-service SDAVC
vnic gateway VirtualPortGroup31
  guest ip address 10.56.197.222
activate
!
```

Operating the SD-AVC Network Service with Host Interface Attached to a VRF



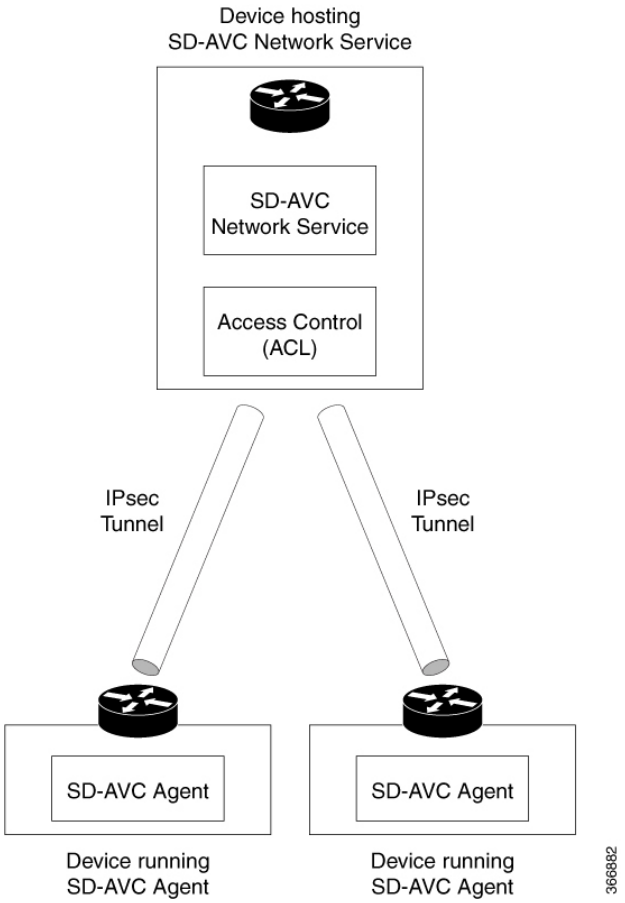
APPENDIX C

Configuring Secure Connectivity

Scenarios Requiring a Secure Connection

For network scenarios that require a secure connection between a network device running the SD-AVC agent, and the SD-AVC Network Service, you can optionally encrypt the SD-AVC communication between agent and Network Service using IPsec tunnels, and control device access using access control lists (ACL), as described in the sections that follow.

Figure 5: IPsec Tunnels between Network Devices and SD-AVC Network Service



Securing Connection between Host and SD-AVC Network Service

The SD-AVC Network Service runs as a virtual service on a Cisco device serving as a host platform. The host platform requires connectivity with the service through a virtual interface called VirtualPortGroup. The virtual service communicates with the host over this virtual interface, using SSH on TCP port 22.

Using ACL to Secure Connectivity between Host and SD-AVC Network Service

The SD-AVC network service operates as a virtualized component on a host device. To secure the connection between the host device and the SD-AVC network service, use the following:

interface VirtualPortGroup31

ip unnumbered GigabitEthernet1

ip access-group sd-avc-acl in

ip access-list extended *acl-name*

permit tcp host *SD-AVC-virtual-service-IP* **host** *host-router-IP* **eq** 22

permit tcp host *host-router-IP* **eq 22** *SD-AVC-virtual-service-IP*

Example using ACL:

```
interface VirtualPortGroup31
ip unnumbered GigabitEthernet1
ip access-group sd-avc-acl in
ip access-list extended sd-avc-acl
!! Configure SSH connection between the sd-avc-network-service to the hosted router
    permit tcp host 10.56.196.232 host 10.56.196.231 eq 22
    permit tcp host 10.56.196.231 eq 22 host 10.56.196.231
```

Securing Connection between Agents and Network Service

Network devices operating with SD-AVC communicate with a central SD-AVC Network Service. Ensure that ports, firewall policy, and so on, are configured to enable communication between the SD-AVC agents and SD-AVC Network Service(s) (see Configuring Connectivity, on page 13).

Using ACL to Secure Connection between Agent and Network Service

On the device hosting the SD-AVC Network Service, configure the UDP and TCP access control lists, as follows.

Note When using ACLs, only configured addresses will have access to the device hosting the SD-AVC Network Service.

- UDP

  The following syntax is presented for reference. For complete information about configuring ACL, see the documentation for your platform.

  permit udp [**host** *source-agent-ip* | *source-agent-network source-wildcard*] **host** *sd-avc-network-service-ip* **eq** 50000

  Example: Configuring port 50000 for UDP traffic for a range of devices (10.56.0.0 to 255).

  ```
  permit udp 10.56.0.0 0.0.255.255 host 10.56.196.232 eq 50000
  permit udp host 10.56.196.232 eq 50000 10.56.0.0 0.0.255.255
  ```

- TCP

  The following syntax is presented for reference. For complete information about configuring ACL, see the documentation for your platform.

  permit tcp [**host** *source-agent-ip* | *source-agent-network source-wildcard*] **host** *sd-avc-network-service-ip* [**eq** *port* | **range** *port-range-start port-range-end*]

  Example: Configuring required ports (20, 21, 50000-60000) for TCP traffic for a range of devices (10.56.0.0 to 255).

  ```
  permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.232 eq 20
  permit tcp host 10.56.196.232 eq 20 10.56.0.0 0.0.255.255

  permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.232 eq 21
  permit tcp host 10.56.196.232 eq 21 10.56.0.0 0.0.255.255
  ```

```
permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.232 range 50000 60000
permit tcp host 10.56.196.232 range 50000 60000 10.56.0.0 0.0.255.255
```

Using IPsec Tunnels to Secure Connection between Agent and Network Service

For network scenarios that require an encrypted connection between a network device running the SD-AVC agent, and the SD-AVC Network Service, set up IPsec tunnels to handle this communication.

For information about configuring Cisco IOS IPsec VPN connections, see Cisco IOS IPsec.

Connectivity to the SD-AVC Dashboard

Access to the SD-AVC Dashboard requires access to the device hosting the SD-AVC Network Service, and involves TCP traffic through port 8443. Ensure that network policty (firewall, ACL, and so on) permits this connectivity for devices requiring access to the SD-AVC Dashboard.

Using ACL to Secure Device Access to the SD-AVC Dashboard

On the device hosting the SD-AVC Network Service, configure the access control list as follows, to enable specific devices to connect to the SD-AVC Dashboard.

The following syntax is presented for reference. For complete information about configuring ACL, see the documentation for your platform.

ip access-list extended sd-avc-acl

permit tcp any host *sd-avc-network-service-ip* **eq 8443**

permit tcp host *source-agent-ip* **eq 8443 any**

Example: Configure PC access to SD-AVC Dashboard.

```
ip access-list extended sd-avc-acl
permit tcp any host 10.56.196.232 eq 8443
permit tcp host 10.56.196.232 eq 8443 any
```

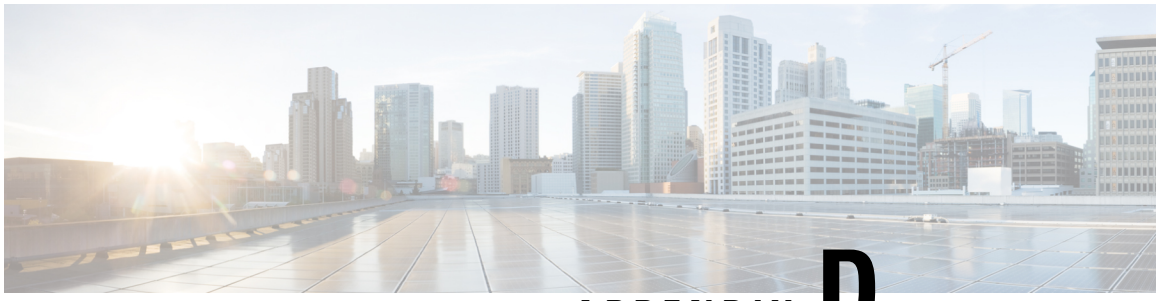
Connectivity: Complete Example

The following example configures connectivity for a newly installed SD-AVC Network Service, hosted on a platform with the address 10.56.196.232, and a range of devices in the network that are operating with SD-AVC.

- Because the SD-AVC Network Service is newly installed, the first section of the example configures connectivity between the host and the SD-AVC virtual service.
- Platform hosting the SD-AVC virtual service: 10.56.196.232
- Network devices operating with SD-AVC, connecting to the SD-AVC Network Service: Address range 10.56.0.0 0.0.255.255
- In this example, any PC may be used to connect to the SD-AVC Dashboard.

```
!! Enables extended ACL
    ip access-list extended sd-avc-acl

!! Configure SSH connection between the sd-avc-network-service to the hosted router
    permit tcp host 10.56.196.232 host 10.56.196.231 eq 22
    permit tcp host 10.56.196.231 eq 22 host 10.56.196.231

!! Configure access to the SD-AVC Dashboard
    permit tcp any host 10.56.196.232 eq 8443
    permit tcp host 10.56.196.232 eq 8443 any

!! Configure access between SD-AVC Network Service and Agents - UDP
    permit udp 10.56.0.0 0.0.255.255 host 10.56.196.232 eq 50000
    permit udp host 10.56.196.232 eq 50000 10.56.0.0 0.0.255.255

!! Configure access between SD-AVC Network Service and Agents - TCP
    permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.232 eq 20
    permit tcp host 10.56.196.232 eq 20 10.56.0.0 0.0.255.255

    permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.232 eq 21
    permit tcp host 10.56.196.232 eq 21 10.56.0.0 0.0.255.255

    permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.232 range 50000 60000
    permit tcp host 10.56.196.232 range 50000 60000 10.56.0.0 0.0.255.255

!! Configure connectivity between host and SD-AVC Network Service (virtual service)
    interface VirtualPortGroup31
    ip access-group  sd-avc-acl in
```

Configuring Secure Connectivity



APPENDIX D

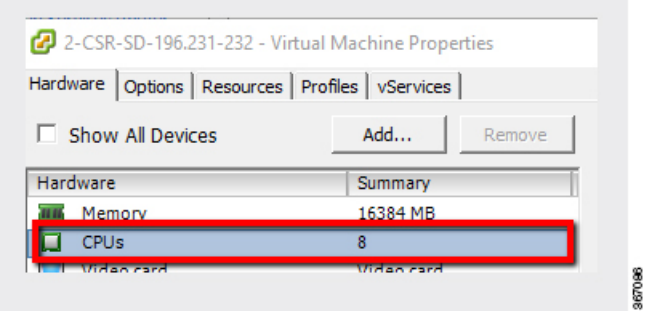
Configuring CSR1000V for SD-AVC

Allocating VM CPUs for Cisco CSR1000V

Use this task to allocate CPU resources when setting up a Cisco Cloud Services Router CSR1000V as a host for the SD-AVC network service.

Before you begin

Step 1 On the VMware ESXi hypervisor client that is hosting the Cisco CSR, edit the CSR that is hosting the SD-AVC network service. Allocate 8 CPUs to the virtual machine. (For small-scale scenarios, fewer CPUs may be necessary. See System Requirements: SD-AVC Network Service Host, on page 12.



Step 2 On the CSR device, execute the following:

```
(config)#platform resource service-plane-heavy
Please reboot to activate this template
```

Step 3 Copy the running configuration to the starting configuration.

```
copy running-config startup-config
```

Step 4 Reload the device.

```
reload
```

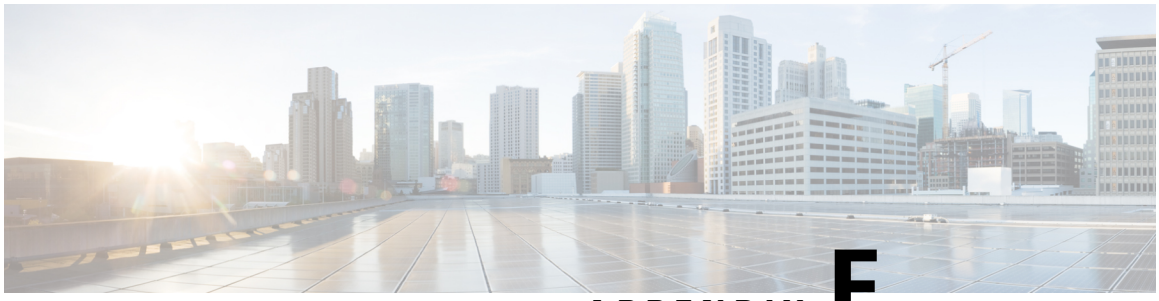
Step 5 Use **show platform software cpu alloc** to check the number of CPU cores allocated.

Check the command output for the **Control plane cpu alloc** line. The output indicates 4 CPUs (numbered 0 to 3).

```
(config)#show platform software cpu alloc
CPU alloc information:
  Control plane cpu alloc: 0-3
  Data plane cpu alloc: 4-7
  Service plane cpu alloc: 0-3
  Template used: CLI-service_plane_heavy
```

Note If the VM has only 4 cores allocated, the **Control plane cpu alloc** line in the command output shows only a single CPU (numbered 0).

```
CPU alloc information:
  Control plane cpu alloc: 0
  Data plane cpu alloc: 1-3
  Service plane cpu alloc: 0
  Template used: CLI-control_plane_heavy
```



APPENDIX E

SD-AVC REST API

REST API Overview

Using the REST API requires authentication. See Authentication from SD-AVC Network Service, on page 76.

Table 23: REST API Functionality

Custom Applications	
POST /avc-sd-service/external-api/app-rules	Defines one or more custom applications. Create User-defined Application Rules, on page 77
GET /avc-sd-service /external-api/app-rules - **GET /avc-sd-service /external-api/app-rules?sourceId=***sourceId*	Displays custom applications defined by REST API. Display User-defined Application Rules, on page 83
GET /avc-sd-service/external-api/app-rules/status - **GET /avc-sd-service /external-api/app-rules/status[?sourceId=***sourceId***]**	Displays activation status of custom applications, per device. Display User-defined Application Status, on page 84
DELETE /avc-sd-service /external-api/app-rules - **DELETE /avc-sd-service /external-api/app-rules?sourceId=***sourceId*	Deletes a set of custom applications. Delete User-defined Applications, on page 84

Authentication from SD-AVC Network Service

Using the API requires a token-based authentication from the SD-AVC network service. To acquire an authentication token:

1. Send the following HTTP request to the API:

   POST https://<SD-AVC-network-service-address>:8443/login

   Example: **POST https://192.168.0.1:8443/login**

   - In the request header, include the following key:

     Content-Type: application/x-www-form-urlencoded

   - In the request body, include the following keys, providing login credentials:

     username: *username*
     password: *password*

2. The API response body provides an authentication token. Use the token to authorize REST API calls to the SD-AVC network service.

Note The token expires after 12 hours.

Example token:

```
{"token":"Bearer eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiJhYjZkGGUxOS0zMmU3LTRlY2ItYWQ5OC
1kYmVmZTdjZjE5YzYiLCJzdWIiOiJsYWIiLCJleHAiOjE1MzAwMqk1MzJ9.EfP3wd4fZbWrOQ6Skh-I0
bbPffF4NaruB-o_OV0EQ7fwMwfmkUUNP00R58fRGKkYWR3tQu8HjoVDp37EPtDl5Q"}
```

User-defined Applications

User-defined Applications: Overview

Network devices operating with SD-AVC use Cisco NBAR2 and other tools to identify network traffic. The composite of information that NBAR2 uses to identify a network applications is called an "application" (or a "protocol" in the Protocol Packs released periodically by Cisco). User-defined applications may be specified on individual devices by CLI, or network-wide using SD-AVC.

Each application includes:

- **Signature**: Details that identify the network application
- **Attributes**: Assigned characteristics of the application, such as business-relevance, used for visibility and QoS policy. See User-defined Application Attribute Values, on page 85.

Table 24: Application Types

Application Type	Description
Protocol Pack applications	Standard applications provided by Cisco in a Protocol Pack.
User-defined applications on individual devices	Defined by CLI on individual devices, sometimes called custom protocols.
Network-wide user-defined applications	Defined by SD-AVC REST API. These appear on the **SD-AVC Dashboard** > **External Sources** page.

SD-AVC User-defined Applications

SD-AVC can provision user-defined applications at the network level, available for all participating devices in the network. In effect, this is similar to adding user-defined applications manually on each device.

Create User-defined Application Rules

API:

POST /avc-sd-service/external-api/app-rules

Description:

Defines one or more user-defined applications.

Body:

Body must include the full set of user-defined applications. Executing the API overwrites any currently defined user-defined applications for the specified source (sourceId).

```
{
    "sourceId": "string",
    "rules": [{
        "allSegments": Boolean,
        "segment": "string",
        "rules": [{
            "appName": "string",
            "serverNames": ["string"],
            "L3L4": [{
                "ipAddresses": ["string"],
                "ports": [integer(s) or range],
                "l4Protocol": "string",
                "vrf": "string"
            }],
            "attributes": {
                "category": "string",
                "sub-category": "string",
                "application-group": "string",
                "business-relevance": "string",
                "traffic-class": "string",
                "application-set": "string"
                }
        }]
    }]
}
```

Table 25: Top-level Properties

Property	Description
sourceId	(Mandatory) ID of the external source. **Note** In the initial release of the REST API, only one source is supported.
rules	(Mandatory) Contains complete list of the user-defined application rules. **Note** This property contains a sub-property also called rules.

Table 26: Sub-properties of rules

Property	Description
allSegments	(Must include either allSegments or segment.) Set to **true** to apply the user-defined applications to all segments, not only one segment. **Possible values**: true, false (default)
segment	(Must include either allSegments or segment.) List of user-defined application rules for a specific SD-AVC segment.
rules	(Mandatory) List of segment rules.

Table 27: Sub-properties of rules > rules

Property	Description
appName	(Mandatory) Name of user-defined application, reflecting name of the network application. **Note** Do not use a name that conflicts with an existing application, such as one defined in the Protocol Pack.
serverNames	(Must include at least one of serverNames, L3L4, and attributes.) List of all server names (FQDNs, SNIs, …) for the network application. **Note** Server names are case-sensitive.
L3L4	(Must include at least one of serverNames, L3L4, and attributes.) List of all IP-based rules. (See sub-properties below.)

Property	Description
attributes	(Must include at least one of serverNames, L3L4, and attributes.) Attributes to assign to the application. (See sub-properties below.)

Table 28: Sub-properties of rules > rules > L3L4

Property	Description
IpAddresses	(Mandatory) List of IPs. Can be both normal IP and subnet (using CIDR notation).
ports	Port(s) or port range. **Examples**: **"ports": [23]** **"ports": [23,24]** **"ports": [23, "25-30"]**
l4Protcol	Transport layer protocol. **Possible values**: TCP, UDP, TCP-UDP
vrf	VRF name.

Table 29: Sub-properties of rules > rules > attributes

Property	Description
application-set	(Must include at least one of serverNames, L3L4, and attributes.) Attributes to assign to the application.
application-group	(Defining a partial list of attributes is supported. If **attributes** is included, must include at least one of these properties.) See User-defined Application Attribute Values, on page 85.
category	
sub-category	
traffic-class	
business-relevance	

Response:

Response code 200 indicates success.

In case of failure, the response body provides information about the reason for failure.

Example 1: Single domain name

This example shows:

- 1 network segment: datacenter01
- 1 user-defined application: myDocs
- 1 server name
- No attributes specified

```
{
  "sourceId": "mySource",
  "rules": [{
    "segment": "datacenter01",
    "rules": [{
      "appName": "myDocs",
      "serverNames": [
        "www.myApp.com"
      ]
    }]
  }]
}
```

Example 2: Three IP addresses and ports

This example shows:

- 1 network segment: datacenter01
- 1 user-defined application: myDocs
- 3 IP addresses and 3 ports
- No attributes specified

```
{
    "sourceId": "mySource",
    "rules": [{
        "segment": "datacenter01",
        "rules": [{
            "appName": "myDocs",
            "L3L4": [{
                    "ipAddresses": ["2.2.2.2"],
                    "ports": [20]
                },
                {
                    "ipAddresses": ["3.3.3.3"],
                    "ports": [30]
                },
                {
                    "ipAddresses": ["4.4.4.4"],
                    "ports": [40]
                }
            ]
        }]
    }]
}
```

Example 3: Two user-defined applications in one network segment

This example shows:

- 1 network segment: datacenter01
- 2 user-defined applications: myDocs and myTelepresence
- No attributes specified for the myDocs user-defined application
- business-relevance attribute specified for the myTelepresence user-defined application
- IP address with subnet specified
- Individual ports and a range of ports

```
{
  "sourceId": "mySource",
  "rules": [{
    "segment": "datacenter01",
    "rules": [{
        "appName": "myDocs",
        "serverNames": [
          "www.myApp.com"
        ],
        "L3L4": [{
          "ipAddresses": ["10.1.1.0/24", "2.2.2.2"],
          "ports": [23, 34, "37 - 42"],
          "l4Protocol": "TCP",
          "vrf": "vrf1"
        }]
      },
      {
        "appName": "myTelepresence",
        "L3L4": [{
          "ipAddresses": ["2.2.2.2"],
          "ports": [35],
          "l4Protocol": "TCP"
        }],
        "attributes": {
          "business-relevance": "business-relevant"
        }
      }
    ]
  }]
}
```

Example 4: User-defined applications in two network segments

This example shows:

- 2 network segments: datacenter01, datacenter02
- 3 user-defined applications: myDocs, myTelepresence, myEnterpriseIM
- No attributes specified for: myDocs, myEnterpriseIM
- business-relevance attribute specified for myTelepresence
- IP address with subnet specified
- Individual ports and a range of ports

```
{
  "sourceId": "mySource",
  "rules": [{
```

```
      "segment": "datacenter01",
      "rules": [{
          "appName": "myDocs",
          "serverNames": [
            "www.myDocs.com"
          ],
          "L3L4": [{
            "ipAddresses": ["10.1.1.0/24", "2.2.2.2"],
            "ports": [23, 34, "37 - 42"],
            "l4Protocol": "TCP",
            "vrf": "vrf1"
          }]
        },
        {
          "appName": "myTelepresence",
          "L3L4": [{
            "ipAddresses": ["2.2.2.2"],
            "ports": [35],
            "l4Protocol": "TCP"
          }],
          "attributes": {
            "business-relevance": "business-relevant"
          }
        }
      ]
    },
    {
      "segment": "datacenter02",
      "rules": [{
        "appName": "myEnterpriseIM",
        "serverNames": [
          "www.myEnterpriseIM.com"
        ],
        "L3L4": [{
          "ipAddresses": ["2.2.2.10"],
          "ports": [23],
          "l4Protocol": "TCP"
        }]
      }]
    }
  ]
}
```

Example 5: Using allSegments and specific network segments

This example shows:

- 2 user-defined applications (myDocs, myTelepresence) for all network segments, using allSegments
- User-defined application (myEnterpriseIM) only for 1 network segment: datacenter02
- 3 user-defined applications: myDocs, myTelepresence, myEnterpriseIM
- No attributes specified for: myDocs, myEnterpriseIM
- business-relevance attribute specified for myTelepresence
- IP address with subnet specified
- Individual ports and a range of ports

```
{
  "sourceId": "mySource",
  "rules": [{
      "allSegments": true,
      "rules": [{
          "appName": "myDocs",
          "serverNames": [
            "www.myApp.com"
          ],
          "L3L4": [{
            "ipAddresses": ["10.1.1.0/24", "2.2.2.2"],
            "ports": [23, 34, "37 - 42"],
            "l4Protocol": "TCP",
            "vrf": "vrf1"
          }]
        },
        {
          "appName": "myTelepresence",
          "L3L4": [{
            "ipAddresses": ["2.2.2.2"],
            "ports": [35],
            "l4Protocol": "TCP"
          }],
          "attributes": {
            "business-relevance": "business-relevant"
          }
        }
      ]
    },
    {
      "segment": "datacenter02",
      "rules": [{
        "appName": "myEnterpriseIM",
        "serverNames": [
          "www.myEnterpriseIM.com"
        ],
        "L3L4": [{
          "ipAddresses": ["2.2.2.10"],
          "ports": [23],
          "l4Protocol": "TCP"
        }]
      }]
    }
  ]
}
```

Display User-defined Application Rules

API:

GET /avc-sd-service /external-api/app-rules

GET /avc-sd-service /external-api/app-rules?sourceId=*sourceId*

Description:

Displays the user-defined applications defined by REST API.

Response:

The response lists the user-defined applications defined for a single source or all sources. The response body uses the same JSON structure as POST.

If no *sourceId* is specified, the response lists the user-defined applications for all sources.

If *sourceId* is specified, the response lists the user-defined applications for the specified source. The *sourceId* is user-defined by POST when defining user-defined applications.

Note In the initial release of the REST API, only one source is supported.

Display User-defined Application Status

API:

GET /avc-sd-service/external-api/app-rules/status

GET /avc-sd-service /external-api/app-rules/status[?sourceId=*sourceId*]

Description:

The SD-AVC network service sends the user-defined applications defined by REST API to the devices in the network. This API displays the activation status of the applications, per device.

If *sourceId* is specified, the output is limited to that source.The *sourceId* is user-defined by POST when defining user-defined applications.

Note In the initial release of the REST API, only one source is supported.

Response:

The response lists each network device, arranged by segment. For each device:

- ID/version of application rules currently loaded on the device
- Status: SUCCESS, FAILED, IN-PROGRESS

Delete User-defined Applications

API:

DELETE /avc-sd-service /external-api/app-rules

DELETE /avc-sd-service /external-api/app-rules?sourceId=*sourceId*

Description:

Deletes a set of user-defined applications.

If no *sourceId* is specified, deletes the full set of user-defined applications.

If *sourceId* is specified, deletes the full set of user-defined applications for the specified source. The *sourceId* is user-defined by POST when defining user-defined applications.

Note In the initial release of the REST API, only one source is supported.

Response:

Response code 200 indicates success.

REST API Notes and Limitations

See Notes and Limitations: REST API, on page 49.

User-defined Application Attribute Values

When creating new user-defined applications using the SD-AVC REST API, use the following attribute values.

Note This list is provided for convenient reference, but may not be comprehensive.

SD-AVC REST API

Attribute	Possible Values
application-set	authentication-services backup-and-storage collaboration-apps consumer-browsing consumer-file-sharing consumer-gaming consumer-media consumer-misc consumer-social-networking database-apps desktop-virtualization-apps email enterprise-ipc file-sharing general-browsing general-media general-misc local-services naming-services network-control network-management remote-access saas-apps signaling software-development-tools software-updates streaming-media tunneling

Attribute	Possible Values
application-group	aol-group apple-group apple-talk-group banyan-group bittorrent-group capwap-group cisco-jabber-group cisco-phone-group corba-group dameware-group edonkey-emule-group espn-group fasttrack-group flash-group fring-group ftp-group gnutella-group google-group gtalk-group hangouts-group icq-group imap-group ipsec-group irc-group kakao-group kerberos-group ldap-group

Attribute	Possible Values
application-group (continued)	ms-cloud-group ms-crm-group ms-lync-group msn-messenger-group netbios-group nntp-group npmp-group other pop3-group prm-group qq-group skype-group smtp-group snmp-group sqlsvr-group stun-group telepresence-group tftp-group vmware-group vnc-group wap-group webex-group xns-xerox-group xunlei-group yahoo-group yahoo-messenger-group

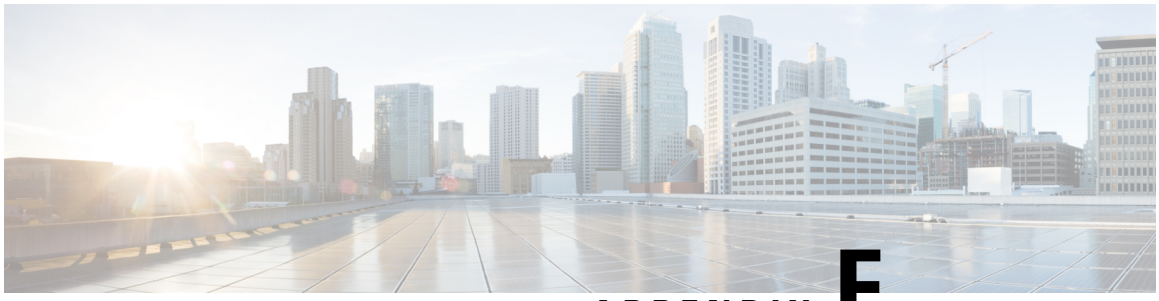
Attribute	Possible Values
category	anonymizers backup-and-storage browsing business-and-productivity-tools consumer-file-sharing consumer-internet consumer-messaging consumer-streaming database email epayement file-sharing gaming industrial-protocols instant-messaging inter-process-rpc internet-security layer3-over-ip location-based-services net-admin newsgroup other social-networking software-updates trojan voice-and-video

SD-AVC REST API

Attribute	Possible Values
sub-category	authentication-services backup-systems consumer-audio-streaming consumer-cloud-storage consumer-multimedia-messaging consumer-video-streaming consumer-web-browsing control-and-signaling desktop-virtualization enterprise-cloud-data-storage enterprise-cloud-services enterprise-data-center-storage enterprise-media-conferencing enterprise-realtime-apps enterprise-rich-media-content enterprise-sw-deployment-tools enterprise-transactional-apps enterprise-video-broadcast enterprise-voice-collaboration file-transfer naming-services network-management os-updates other p2p-file-transfer p2p-networking remote-access-terminal routing-protocol tunneling-protocols

Attribute	Possible Values
traffic-class	broadcast-video bulk-data multimedia-conferencing multimedia-streaming network-control ops-admin-mgmt real-time-interactive signaling transactional-data voip-telephony
business-relevance	business-irrelevant business-relevant default

SD-AVC REST API



APPENDIX F

Source Interface Configuration

Source Interface Configuration Overview

On network devices operating with SD-AVC, you can specify the interface to be used for communication from the device to the SD-AVC network service, using the **source-interface** command. This can be any type of interface, including virtual, such as a loopback interface.

When the network device sends packets to the SD-AVC network service, the Source IP of the packets will be the IP address of the interface specified by the **source-interface** command.

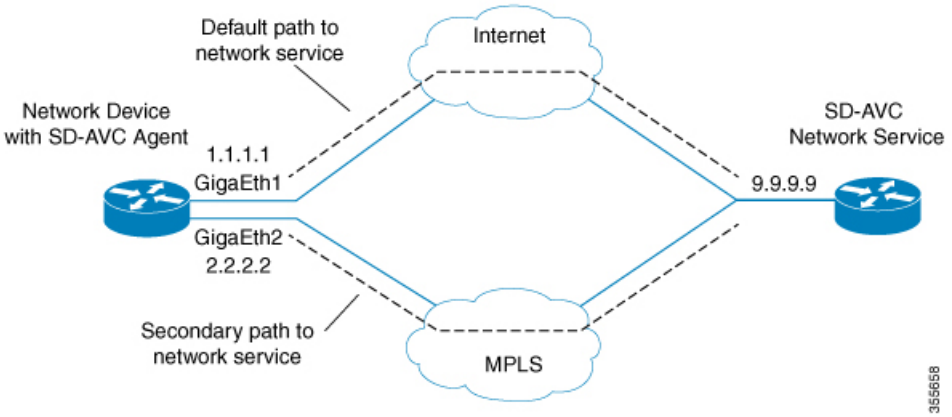
Specifying the interface for SD-AVC traffic can address several issues described in Scenarios that Benefit from Source Interface Configuration, on page 94.

Background

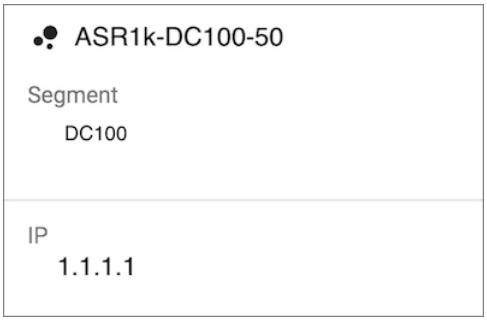
Network devices appear in the SD-AVC Dashboard, identified by an address. Typically, this is the IP of the interface on the device used for communication between the device and the SD-AVC network service. The routing table on the device determines the interface(s) used for communication with the SD-AVC network service.

In the following example, the default path for packets sent from the device to the network service will be:

Source: 1.1.1.1
Destination: 9.9.9.9



In this case, the network device appears in the SD-AVC Dashboard, identified as 1.1.1.1, as shown below.



Scenarios that Benefit from Source Interface Configuration

Specifying a source interface for SD-AVC traffic can be helpful in numerous scenarios.

- Improve visibility by providing a consistent IP address for SD-AVC traffic.
- Simplify configuring a network firewall by providing a consistent source IP address for SD-AVC traffic.
- Separate SD-AVC FTP traffic from non-SD-AVC FTP traffic.

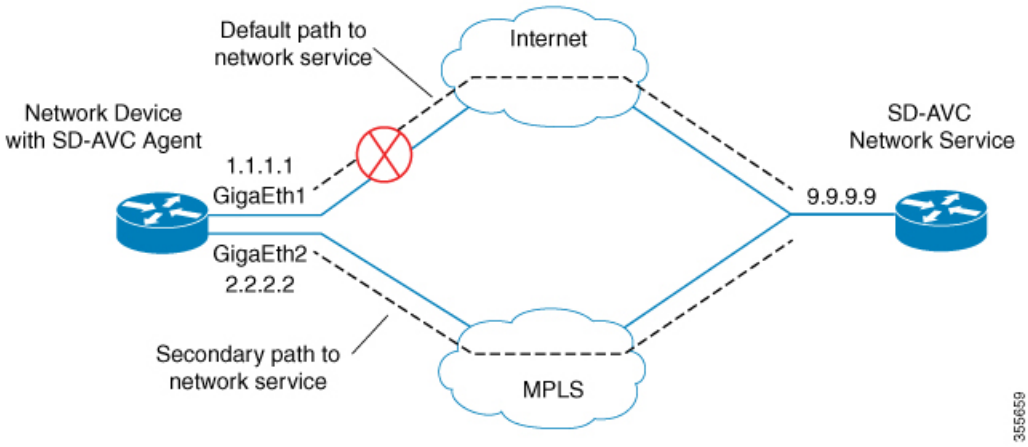
Scenario: Default Connection Down

If the default path between a network device and the SD-AVC service is not available, and traffic is routed over a different interface, the source of the packets may change. For example:

Default packet source: 1.1.1.1
Packet source when using secondary path: 2.2.2.2

In the following example, the default path is not available, and packets sent from the device to the network service will follow the secondary path (using interface 2.2.2.2) instead of the default (interface 1.1.1.1):

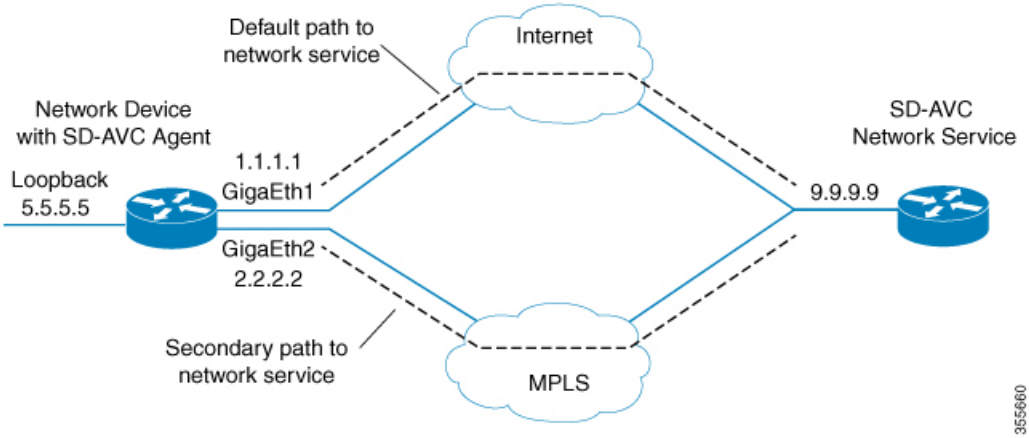
Source: 2.2.2.2
Destination: 9.9.9.9



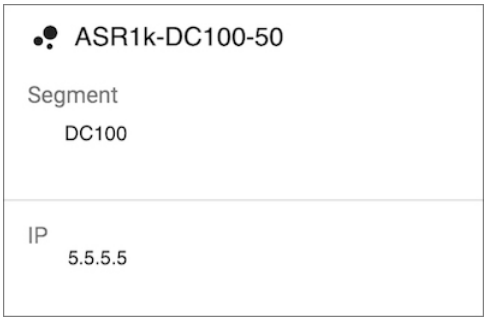
Earlier, the network device appeared in the SD-AVC Dashboard as 1.1.1.1. Now it appears as 2.2.2.2, reflecting the secondary path to the SD-AVC network service. The device hostname remains the same, but the IP has changed, as shown below. This may not be desired.

Configuring a consistent source interface ensures that the network device appears in the SD-AVC Dashboard with a consistent IP.

This can be accomplished by creating a loopback interface (5.5.5.5 in the example below) and setting it to be the source interface for all SD-AVC traffic from the device. See Specifying a Loopback as Source Interface, on page 97.



Regardless of the path used for SD-AVC traffic, the device appears consistently in the SD-AVC Dashboard as 5.5.5.5.

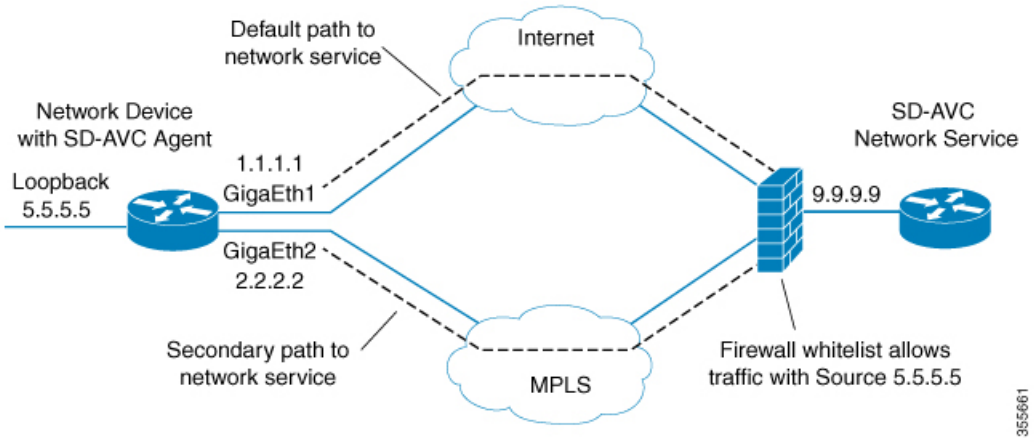


Scenario: Network Firewall Policy

In some network configurations, a firewall blocks all traffic to the SD-AVC network service, other than devices on a whitelist. This may require whitelisted devices to present themselves to the firewall with a consistent IP address. In the following illustration, traffic to the SD-AVC network service can use the 1.1.1.1 or 2.2.2.2 paths.

Configuring a consistent source interface ensures that SD-AVC traffic from the network device consistently presents itself to the firewall with the same IP. This simplifies firewall whitelist policy.

This can be accomplished by creating a loopback interface (5.5.5.5 in the example below) and setting it to be the source interface for all SD-AVC traffic from the device. See Specifying a Loopback as Source Interface, on page 97.



Scenario: Internal FTP Server

In some network configurations, a network device communicates with an FTP server through an interface that cannot reach the SD-AVC network service. This can cause conflict with the FTP communications between the SD-AVC agent on the network device and the SD-AVC network service.

To avoid conflict between different types of FTP activity, use the **source-interface** command to specify an interface that can reach the SD-AVC network service. This enables SD-AVC FTP traffic on one interface, and other FTP traffic on another interface.

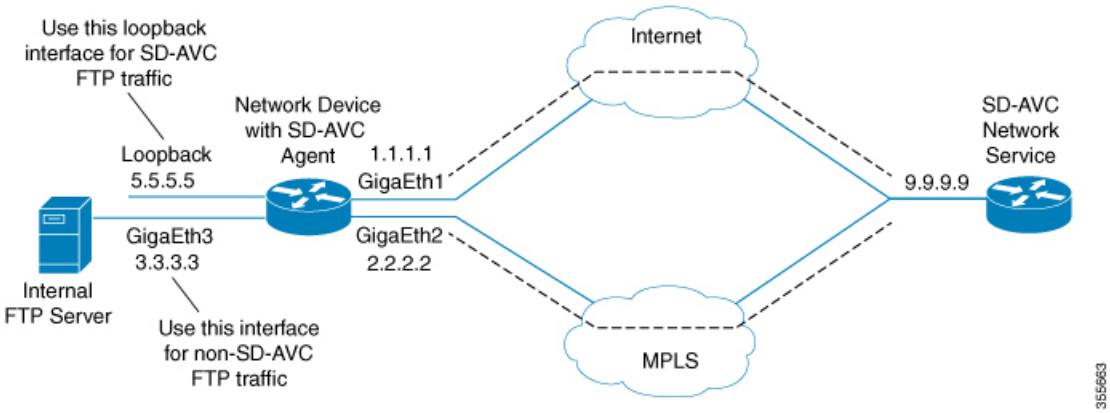
In the example below:

- Non-SD-AVC FTP traffic is on gigabitEthernet interface 3:

```
Device(config)#ip ftp source-interface gigabitEthernet 3
```

- SD-AVC FTP traffic uses the loopback interface as source (see Specifying a Loopback as Source Interface, on page 97):

```
Device(config-sd-service-controller)#source-interface loopback0
```



Configuring Source Interface for SD-AVC Communication

On network devices, use the **source-interface** command to specify the source interface, and therefore the Source IP, for SD-AVC traffic.

You can specify any physical or virtual interface on the device, but to address the scenarios described above, use a loopback interface.

Specifying a Loopback as Source Interface

To address scenarios such as those described above, create a virtual (loopback) interface and specify that SD-AVC traffic sent from the network device to the SD-AVC network service will use the virtual interface to define the Source address. The Source IP for SD-AVC packets sent from the network device will be the IP address of the specified interface.

1. On the network device, create a loopback interface (virtual), and assign it an IP address.
2. On the SD-AVC network service host, ensure access to the loopback interface on the network device.

Note This may require adding one or more routing table entries to enable access to the loopback interface. Configuring a routing table path to the loopback interface may be something like this:

ip route *device-loopback-ip* **255.255.255.255** *device-physical-interface*

Example:

```
HostDevice(config)#ip route 5.5.5.5 255.255.255.255 1.1.1.1
```

3. On the network device, use the **source-interface** command to select the loopback interface. In the example, the loopback interface is **loopback0**.

   In configuration mode:

   avc sd-service
   segment *segment*
   controller
   address *sd-avc-network-service-IP*
   source-interface *source-interface*

   Example:

```
Device(config)#avc sd-service
Device(config-sd-service)#segment sdavc
Device(config-sd-service)#controller
Device(config-sd-service-controller)#address 9.9.9.9
Device(config-sd-service-controller)#source-interface loopback0
```

In the **SD-AVC Dashboard**, the network device will be identified consistently by the specified source interface. In the example above, the source interface specified is **loopback0**, with IP 5.5.5.5.

Note The IP is updated in the Dashboard when the network device sends an update to the SD-AVC network service.