



## **Cisco SD-AVC User Guide, Release 4.0.0**

**First Published:** 2020-10-14

**Last Modified:** 2020-10-14

### **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 9**

### CHAPTER 1

#### **SD-AVC Overview 1**

- SD-AVC Overview 1
- No Change to Topology 3
- New Features and Changes, by Release 3
- Using SD-AVC in an Asymmetric Routing Scenario 8

---

### CHAPTER 2

#### **Operation 11**

- SD-AVC Architecture 11
- SD-AVC and Application Recognition 12
  - Collecting Application Data 12
  - Aggregating Application Data 13

---

### PART II

---

#### **Part: Deployment 15**

### CHAPTER 3

#### **Installation Overview 17**

- System Requirements: SD-AVC Network Service Host 18
- Configuring Connectivity 19
- Using SD-AVC with Cisco IWAN 20
- Installing the SD-AVC Network Service 20
- Upgrading the SD-AVC Network Service 26

---

### CHAPTER 4

#### **Unconfiguring or Uninstalling the SD-AVC Network Service 29**

- Unconfiguring the SD-AVC Network Service 29
- Uninstalling the SD-AVC Network Service 29

---

<b>CHAPTER 5</b>	<b>Configuring Network Devices</b>	<b>31</b>
	Configuring Network Devices to Use SD-AVC	31
	System Requirements: Network Devices Using SD-AVC	31
	Configuration Prerequisites: Network Devices Using SD-AVC	33
	Activating the SD-AVC Agent	33
	Deactivating the SD-AVC Agent	35

---

<b>CHAPTER 6</b>	<b>SD-AVC High Availability</b>	<b>37</b>
------------------	---------------------------------	-----------

---

<b>PART III</b>	<b>Part: Use</b>	<b>41</b>
-----------------	------------------	-----------

---

<b>CHAPTER 7</b>	<b>Using SD-AVC</b>	<b>43</b>
	Using SD-AVC	43
	Connecting to the Dashboard	43
	Application Visibility Page	44
	Unclassified Traffic Analysis and Discovery	46
	Background	46
	Partial Classification of Traffic	47
	Viewing Unclassified Traffic Details	47
	Improves Visibility, Does Not Affect Policy	47
	Enabling and Disabling	48
	Notes and Limitations	48
	Protocol Packs Page	48
	Understanding Protocol Pack Files	48
	Importing the Latest Protocol Packs Directly from Cisco	49
	Uploading Protocol Packs from a Local Drive to the SD-AVC Repository	49
	Deploying Protocol Packs to Devices	49
	Connectors Page	50
	Cloud Connector	50
	Enabling the Cloud Connector	52
	DNS Server Connectivity	53
	Configuring a Proxy Server	54
	Customization Page	54

Custom Applications in SD-AVC	54
Creating a Custom Application	54
Serviceability Page	56
Application Rules Page	59
SD-AVC System Time and Displayed Times	60

---

**CHAPTER 8**      **SD-AVC Notes and Limitations**    **63**

General	63
Setup	63
Classification	64
High Availability	65
Protocol Pack	65
REST API	65

---

**APPENDIX A**      **Troubleshooting SD-AVC**    **67**

Troubleshooting Overview	67
Troubleshooting SD-AVC Network Service Issues	70
Troubleshooting Commands for Network Service Issues	70
Installation Failure Caused by Memory or Disk	72
Activation Failure Caused by Shared CPU Resources	73
Configuration Failure Caused by VRF	75
Troubleshooting SD-AVC Agent Issues	76
NBAR2 Not Activated on Interfaces	76
Active Sessions Preventing Agent Configuration	76
Troubleshooting SD-AVC Connectivity Issues	77
Problem with UDP Communication with Devices	77
Problem with TCP Communication with Devices	78
Problem with FTP/HTTP Communication with Devices	79
Troubleshooting Protocol Pack Issues	80
Failure to Deploy Protocol Pack to Device	80

---

**APPENDIX B**      **Operating the SD-AVC Network Service with Host Interface Attached to a VRF**    **81**

---

**APPENDIX C**      **Configuring Secure Connectivity**    **83**

Securing Connections to the SD-AVC Network Service 83

Configuring ACL Access 85

---

**APPENDIX D**

**Configuring CSR1000V for SD-AVC 87**

Allocating VM CPUs for Cisco CSR1000V 87

---

**APPENDIX E**

**SD-AVC REST API 89**

REST API Overview 89

Authentication from SD-AVC Network Service 92

Configure Cloud Connector Credentials 93

Configure Cloud Connector Telemetry Data Location 94

System 95

System Overview 95

Display System Information 95

Display Devices 95

Delete Devices from SD-AVC 96

Display Traffic Analytics 97

Cloud Connector 98

Connect to Cloud Connector 98

Disable Cloud Connector 99

Remove Cloud Connector Credentials 99

Display Current Cloud Configuration 100

Display Cloud Data 100

Display Cloud Connector Status 101

External Sources 102

External Sources Overview 102

Enable/Disable External Sources 102

Display Status of External Sources 103

User-defined Applications 103

User-defined Applications Overview 103

Create User-defined Application Rules 104

Example 1: Single domain name 107

Example 2: Three IP addresses and ports 107

Example 3: Two user-defined applications in one network segment 108

Example 4: User-defined applications in two network segments	109
Example 5: Using allSegments and specific network segments	110
Add a User-defined Application Rule	111
Example	112
Display User-defined Application Rules	112
Display User-defined Application Status	113
Delete User-defined Applications	114
Generic Applications	115
Generic Applications Overview	115
Display Generic Application Traffic Types	115
REST API Notes and Limitations	115

---

**APPENDIX F****Source Interface Configuration 117**

Source Interface Configuration Overview	117
Background	117
Scenarios that Benefit from Source Interface Configuration	118
Scenario: Default Connection Down	118
Scenario: Network Firewall Policy	120
Scenario: Internal FTP/HTTP Server	120
Configuring Source Interface for SD-AVC Communication	121
Specifying a Loopback as Source Interface	121

---

**APPENDIX G****NBAR AWS Cloud Telemetry Matrix 123**

---

**APPENDIX H****Creating SSL Certificates to Use with SD-AVC 127**

Summary	127
Using a Certificate Signed by a Certification Authority	128
Using a Self-signed SSL Certificate Created with Keytool	128
Using a Self-signed SSL Certificate Created with OpenSSL	130

---

**APPENDIX I****Additional References 133**







## PART I

# Part: Introduction

- [SD-AVC Overview, on page 1](#)
- [Operation, on page 11](#)





# CHAPTER 1

## SD-AVC Overview

- [SD-AVC Overview, on page 1](#)
- [No Change to Topology, on page 3](#)
- [New Features and Changes, by Release, on page 3](#)
- [Using SD-AVC in an Asymmetric Routing Scenario, on page 8](#)

## 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	<p>Cisco SD-AVC further refines application recognition accuracy by helping numerous devices in a network</p> <p>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.</p> <p>See <a href="#">SD-AVC and Application Recognition, on page 12</a>.</p>
Improved first packet recognition	<p>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</p>
Protocol Pack update at the network level	<p>SD-AVC can assist in deploying Protocol Packs to numerous routers in the network. Download the Protocol Packs directly from Cisco into a repository 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.</p> <p>See <a href="#">Protocol Packs Page, on page 48</a>.</p>
SD-AVC Dashboard	<p>Secure browser-based SD-AVC Dashboard over HTTPS for monitoring SD-AVC functionality and statistics, and for configuring Protocol Pack updates network-wide.</p> <p>See <a href="#">Using SD-AVC, on page 43</a>.</p>
Cloud Connector	<p>SD-AVC connects to a cloud service provided by Cisco that improves traffic classification. Cloud Connector uses the latest information available about the server addresses used by public internet sites and services to improve SD-AVC classification of traffic.</p> <p>See <a href="#">Cloud Connector, on page 50</a>.</p>
Improved Microsoft Office 365 traffic classification	<p>The MS-Office365 Web Service component improves classification for Microsoft Office 365 traffic. The SD-AVC Dashboard displays the status of the component.</p>
REST API	<p>REST API for user-defined applications.</p> <p>See <a href="#">SD-AVC REST API, on page 89</a>.</p>
Analysis of unclassified traffic	<p>To improve traffic visibility, SD-AVC analyzes unclassified/unidentified traffic and provides server or socket information about unclassified traffic flows that use significant bandwidth.</p> <p>See <a href="#">Unclassified Traffic Analysis and Discovery, on page 46</a>.</p>

# 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 4.0.0*

Feature	Description
Store Cloud Connector telemetry data in separate locations for each segment	<p>Added the ability to specify a location to store Cloud Connector telemetry data separately for each network segment. This can be done through the REST API.</p> <p>See <a href="#">Configure Cloud Connector Telemetry Data Location, on page 94</a>.</p>
Configure custom applications in SD-AVC Dashboard	<p>Added the ability to configure user-defined custom applications using the SD-AVC Dashboard.</p> <p><b>Note</b> If you create custom applications using the SD-AVC Dashboard, do not create a new set of custom applications through the REST API using POST. Doing so overwrites the custom applications created through the SD-AVC Dashboard. You can add custom applications through the REST API using PUT.</p> <p>See <a href="#">Creating a Custom Application, on page 54</a>.</p>
Disable Behavioral Based Classification	<p>Added a control to enable/disable behavioral classification.</p> <p><b>Note</b> Using the Disable option requires that the devices in the network use Cisco IOS XE Amsterdam 17.3.x or later.</p> <p>See <a href="#">Serviceability Page, on page 56</a>.</p>
Support for Office 365 Traffic Categories	<p>Added ability to use the Microsoft Office 365 traffic category when creating traffic policy, enabling you to apply policy decisions based on Office 365 traffic category.</p> <p>See <a href="#">Office 365 Traffic Categories, on page 51</a>.</p>
Clear Traffic Classification Data for a Segment	<p>Added a <b>Clear State</b> option, enabling you to clear the collected traffic classification data for a network segment. This resets the application rules pack that the SD-AVC network service sends to devices in the network segment.</p> <p><b>Note</b> This feature requires that the devices in the network use Cisco IOS XE Amsterdam 17.3.x or later.</p> <p>See <a href="#">Application Visibility Page, on page 44</a>.</p>

Feature	Description
SD-AVC REST API: Cloud Connector status API	API added to return Cloud Connector status per segment. See <a href="#">Display Cloud Connector Status, on page 101</a> .
SD-AVC REST API: Add user-defined application to existing set	API added to add a single user-defined application to an existing set. <a href="#">Add a User-defined Application Rule, on page 111</a>
SD-AVC REST API: Management of user-defined applications by network segment	Updated the POST API for user-defined applications to enable writing a set of user-defined applications for a specific segment, without affecting other segments.  Updated the GET and DELETE APIs to enable displaying or deleting a specific user-defined application for a specific segment, or all of the user-defined applications for a specific network segment.  <a href="#">SD-AVC REST API, on page 89</a>
SD-AVC REST API: For user-defined applications, support for any subnet length for IPv4 or IPv6	Support for an unlimited subnet prefix length, and for IPv6 addresses, when configuring user-defined custom applications.  See <a href="#">Custom Applications in SD-AVC, on page 54</a> (using SD-AVC Dashboard).  See <a href="#">User-defined Applications, on page 103</a> (using SD-AVC REST API).

**Table 2: New and Changed Features, SD-AVC Release 3.2.0**

Feature	Description
SD-AVC REST API: Enhancement of the L3L4 API	Enhancement of the L3L4 API, extending the supported range of IP addresses.  See <a href="#">SD-AVC REST API, on page 89</a> .
Improved Cloud Connector reliability	Improved Cloud Connector reliability, by improved cloud server connectivity.  See <a href="#">Cloud Connector, on page 50</a> .
Proxy server configuration from SD-AVC Dashboard	Ability to configure a proxy server from the SD-AVC Dashboard.  See <a href="#">Configuring a Proxy Server, on page 54</a> .

**Table 3: New and Changed Features, SD-AVC Release 3.1.0**

Feature	Description
Cloud Connector REST APIs	REST APIs added for the Cloud Connector: connect, disable, clear credentials, display configuration, display cloud data  See <a href="#">SD-AVC REST API, on page 89</a> .

**Table 4: New and Changed Features, SD-AVC Release 3.0.0**

Feature	Description
Cloud Connector	SD-AVC connects to a cloud service provided by Cisco that improves traffic classification. Cloud Connector uses the latest information available about the server addresses used by public internet sites and services to improve SD-AVC classification of traffic.  See <a href="#">Cloud Connector</a> , on page 50.
Protocol Pack import	When Cisco releases a new Protocol Pack, SD-AVC indicates that the new Protocol Pack is available. SD-AVC now provides an option to import the Protocol Pack directly from Cisco to the local SD-AVC repository, without requiring the Software Download tool. The Protocol Pack can then be deployed to devices in the network.  See <a href="#">Protocol Packs Page</a> , on page 48.
System log server	SD-AVC keeps a system log as a local file. Beginning with this release, SD-AVC can also send system messages to an external system log server in real time.  See <a href="#">Serviceability Page</a> , on page 56.
Signed SSL certificate	By default, the browser-based SD-AVC Dashboard provides a self-signed SSL certificate that appears in a browser as untrusted. Optionally, you can register your specific domain and acquire a signed SSL certificate specifically for use with SD-AVC, and import the certificate into SD-AVC. Connecting to the SD-AVC Dashboard is then secure and trusted.  See <a href="#">Serviceability Page</a> , on page 56.
Changed TCP port range	SD-AVC uses TCP ports for communication between the central SD-AVC network service and the devices in the network running the SD-AVC agent. Port 8080 was added, changing the range from:  21 and 59990-60000  to  21, 8080, and 59990-60000

**Table 5: New and Changed Features, SD-AVC Release 2.2.1**

Feature	Description
REST API improvements	Several improvements to the SD-AVC REST API.
Optimization of device update time	SD-AVC optimizes the time interval for updating devices in the network, according to the number of devices in the network. For networks containing a relatively small number of devices, updates can occur up to 10 times faster. Updates include the latest aggregated application data, custom applications, and Protocol Pack updates.

Feature	Description
Changed TCP port range	SD-AVC uses TCP ports for communication between the central SD-AVC network service and the devices in the network running the SD-AVC agent. The range was simplified from:  21 and 59900-60000  to  21 and 59990-60000
Improved handling of proxy servers	When a network includes a proxy server, SD-AVC recognizes the proxy server IP and synchronizes the IP as a proxy, thereby preventing the SD-AVC agent from caching the IP. This prevents errors in flow classification.

**Table 6: New and Changed Features, SD-AVC Release 2.2.0**

Feature	Description
Improved scale	SD-AVC supports 1 segment with 6000 devices, or up to 12 segments with 1000 devices in each.
MS-Office365 Connector updates	The MS-Office 365 Connector (external source for SD-AVC) has been updated to incorporate the new Microsoft Office 365 web API. Recent changes that Microsoft has made to the Microsoft Office 365 web API have blocked the SD-AVC Microsoft Office 365 Connector, breaking its functionality in previous releases of SD-AVC.

**Table 7: New and Changed Features, SD-AVC Release 2.1.1**

Feature	Description
Memory and CPU allocation	Smart allocation of memory and CPU resources used for tracking sockets and L3 incoming entries.
Application rules pack distribution by network segment	For improved control, you can assign application rules pack distribution by network segment.
User-defined applications by network segment	For improved control, user-defined applications can be defined by network segment.
Debugging by device or network segment	SD-AVC <b>Dashboard</b> > <b>Serviceability</b> page > <b>Vertical Debug</b> : Can track traffic for a specific device or network segment.
Unclassified Traffic Visibility	Ability to enable or disable the Unclassified Traffic Visibility feature. See <a href="#">Serviceability Page, on page 56</a> .
User Interface improvements	Numerous improvements to usability.



**Table 8: 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 <a href="#">SD-AVC REST API</a> , on page 89.
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 <a href="#">Unclassified Traffic Analysis and Discovery</a> , on page 46.
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 <a href="#">Source Interface Configuration Overview</a> , on page 117.
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.

**Table 9: 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 <a href="#">SD-AVC System Time and Displayed Times</a> , on page 60.
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.

**Table 10: New and Changed Features, SD-AVC Release 2.0.0**

Feature	Description
Updated user interface	<ul style="list-style-type: none"> <li>• Improved interactive display of traffic data</li> <li>• Improved presentation of warnings and errors affecting devices</li> </ul>
Improved control of Protocol Pack deployment	<ul style="list-style-type: none"> <li>• Can update Protocol Packs for individual devices, for segments, or for all devices in the network</li> <li>• Ability to revert to the Protocol Pack built into the Cisco IOS release</li> </ul> <p>See <a href="#">Protocol Packs Page</a>, on page 48.</p>

Feature	Description
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
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.

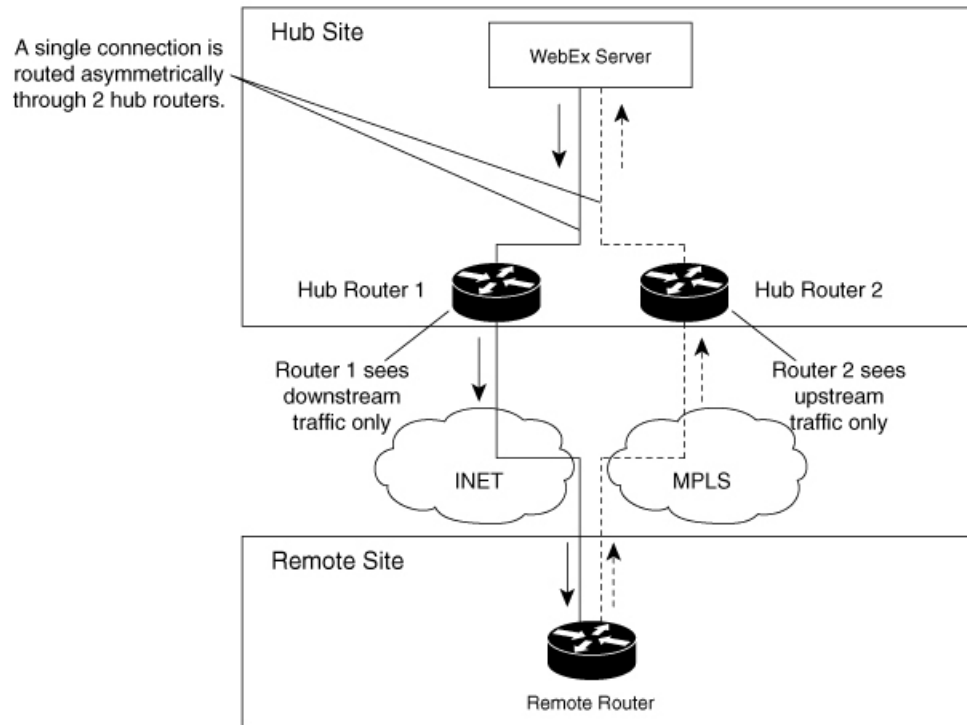
### Deep Packet Inspection and Asymmetry

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.

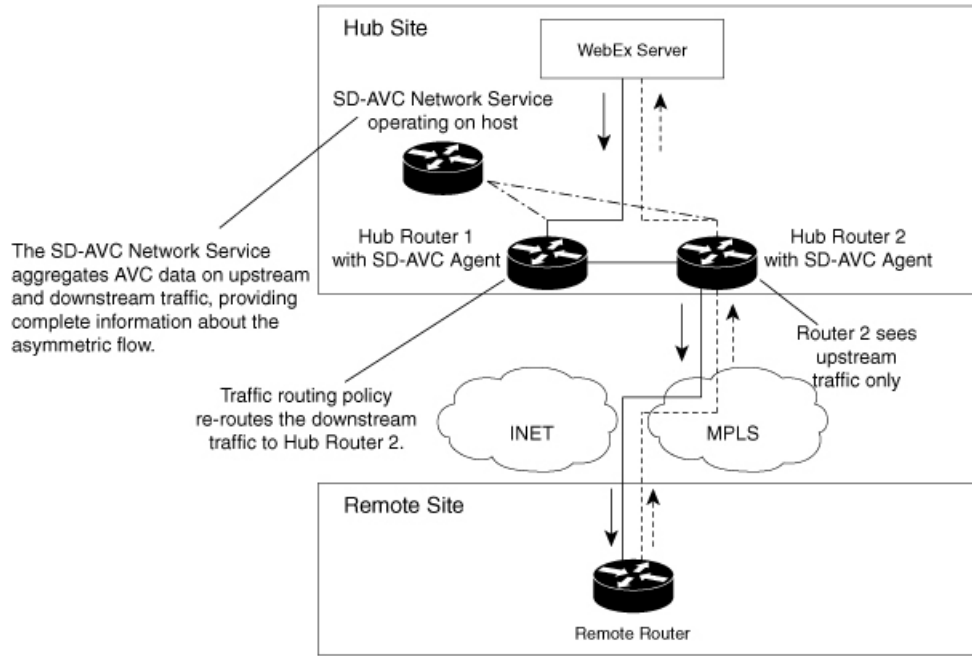
**Figure 1: Asymmetric Routing Example**

### 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



366910



## CHAPTER 2

# Operation

---

- [SD-AVC Architecture](#), on page 11
- [SD-AVC and Application Recognition](#), on page 12

## SD-AVC Architecture

SD-AVC architecture consists of two basic components:

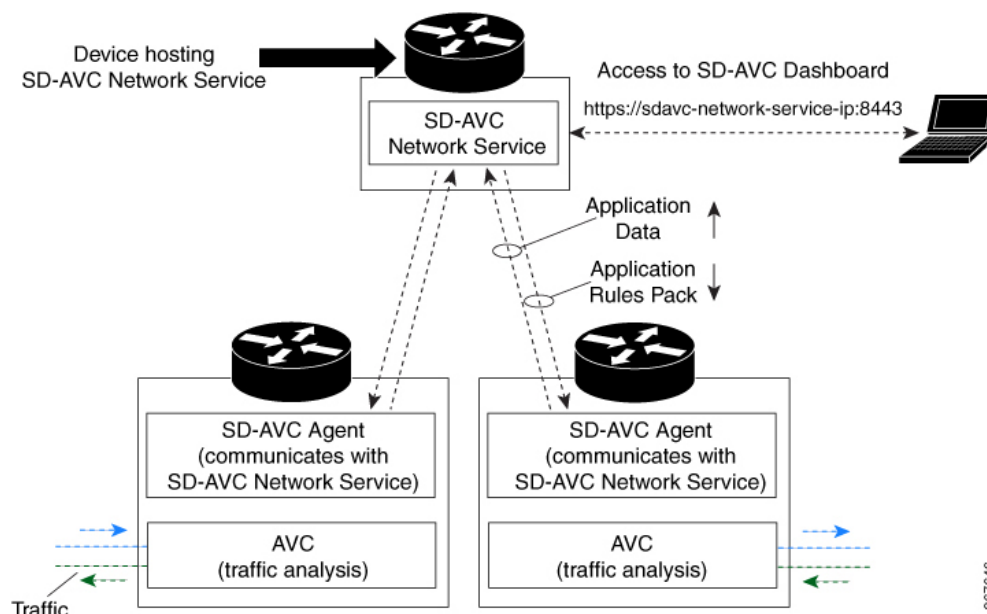
- Centralized SD-AVC network service component operating on a host device
- SD-AVC Agent component running on each SD-AVC-enabled device in the network

The SD-AVC network service communicates with SD-AVC agents in the network using:

- TCP over port 21 (FTP) for devices using Cisco IOS XE 16.11.x Gibraltar or earlier
- TCP over port 8080 (HTTP) for devices using Cisco IOS XE 16.12.1 Gibraltar or later

(See [System Requirements: Network Devices Using SD-AVC](#), on page 31.)

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

- [Installation Overview, on page 17](#)
- [Unconfiguring or Uninstalling the SD-AVC Network Service, on page 29](#)
- [Configuring Network Devices, on page 31](#)
- [SD-AVC High Availability, on page 37](#)





# CHAPTER 3

## Installation Overview

SD-AVC operates in a service/agent configuration. For details, see [SD-AVC Architecture, on page 11](#).

- **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 18](#)
- **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 31](#)
- **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 37](#)
- **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 19](#)

### Summary of Setup

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

**Table 11: 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: <a href="#">Installing the SD-AVC Network Service, on page 20</a>
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: <a href="#">Configuring Network Devices, on page 31</a>
3	Configure connectivity, or optionally, secure connectivity.	See: <a href="#">Configuring Connectivity, on page 19</a> , <a href="#">Configuring Secure Connectivity, on page 83</a>

- [System Requirements: SD-AVC Network Service Host, on page 18](#)
- [Configuring Connectivity, on page 19](#)

- [Using SD-AVC with Cisco IWAN, on page 20](#)
- [Installing the SD-AVC Network Service, on page 20](#)
- [Upgrading the SD-AVC Network Service, on page 26](#)

## System Requirements: SD-AVC Network Service Host

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

**Table 12: 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 Amsterdam 17.3.1 or later (See note 1.)	—
Cisco ASR1002-X Aggregation Services Router	M-ASR1002X-16GB	MASR1002X-HD-320G	Cisco IOS XE Amsterdam 17.3.1 or later (See note 1.)	—
Cisco ASR1002-HX Aggregation Services Router	M-ASR1002HX-16GB	NIM-SSD and SSD-SATA-400G	Cisco IOS XE Amsterdam 17.3.1 or later (See note 1.)	—
Cisco ISR4431 Integrated Services Router	RAM: MEM-4400-4GU16G Flash: MEM-FLASH-16G	NIM-SSD and SSD-MSATA-400G	Cisco IOS XE Amsterdam 17.3.1 or later (See note 1.)	—
Cisco ISR4451 Integrated Services Router	RAM: MEM-4400-4GU16G Flash: MEM-FLASH-16G	NIM-SSD and SSD-MSATA-400G	Cisco IOS XE Amsterdam 17.3.1 or later (See note 1.)	—

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 Amsterdam 17.3.1 or later (See notes 1, 2.)	Large-scale scenario (100 or more devices): 4 cores Small-scale scenario (<100 devices): 1 core See: <a href="#">Allocating VM CPUs for Cisco CSR1000V, on page 87</a>
Cisco DNA Center Traffic Telemetry Appliance (TTA)	—	—	Cisco IOS XE Amsterdam 17.3.1 or later	—



- Note**
1. Minimum supported OS: Cisco IOS XE Everest 16.6.1 or later
  2. The Cisco CSR1000V Cloud Services Router requires the following license: AX, 2.5 Gbps or higher throughput. See the [Cisco CSR1000V Data Sheet](#).

## 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 83](#)

### Connectivity between 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.

### Connectivity between 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 21, 8080, 59990-60000

#### • Firewalls and Access Lists

Ensure that communication is possible from the SD-AVC agent to the SD-AVC network service on these ports for the relevant traffic. For example:

- Firewall policy must enable communication from the SD-AVC agent to the SD-AVC network service.
- If a network device has an access control list (ACL) configured, the ACL must permit communication from the SD-AVC agent to the SD-AVC network service.

### Connectivity to the SD-AVC Dashboard

Connecting to the SD-AVC Dashboard (see [Using SD-AVC, on page 43](#)) 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 18](#), [Installing the SD-AVC Network Service, on page 20](#)

## 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 13: 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 18](#)
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 18](#)).
4. On the device, verify that the MD5 checksum of the downloaded package matches the checksum value provided.



**Note** The correct MD5 checksum value appears 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 29](#)  
[Installation Overview, on page 17](#)

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 14: Command Details**

CLI keyword/argument	Description
<i>disk-with-OVA</i>	Specify one of the following, according to the platform type. The location refers to where the OVA was saved in a previous step. <ul style="list-style-type: none"> <li>• CSR: <b>bootflash</b></li> <li>• ASR1000 Series or ISR4000 Series: <b>harddisk</b></li> </ul>
<i>OVA-filename</i>	Downloaded OVA file.
<i>location-for-OVA-expansion</i>	Specify one of the following, according to the platform type: <ul style="list-style-type: none"> <li>• For CSR1000V routers, use: <b>bootflash</b></li> <li>• For ASR1000 Series or ISR4000 Series devices, use only: <b>harddisk</b></li> </ul> <p><b>Important</b> On ASR1000 and ISR4000 platforms, do not use <b>bootflash</b>. The CLI may allow you incorrectly to choose <b>bootflash</b>, but but this causes the step to fail. On these platforms, specify <b>only harddisk</b>.</p>



**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 81.

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

**Table 15: Command Details**

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

CLI keyword/argument	Description
<b>preview</b>	<p>Preview the configuration without configuring or activating the service. When using this option, the configuration is not sent to the device.</p> <p><b>Note:</b> If the gateway interface is attached to a VRF, see <a href="#">Operating the SD-AVC Network Service with Host Interface Attached to a VRF, on page 81</a>.</p> <p><b>Example output:</b></p> <pre>! 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</pre>
<i>interface</i>	<p>Gateway interface: The device interface that the virtualized service uses for external access.</p> <p><b>Note:</b> If the interface is attached to a VRF, see <a href="#">Operating the SD-AVC Network Service with Host Interface Attached to a VRF, on page 81</a> for instructions for configuring the gateway.</p>
<i>service-ip-address</i>	<p>External-facing IP address, must be in the same subnet as the IP of the gateway interface.</p> <p><b>Example:</b></p> <pre>Gateway interface: 10.56.196.100 service-ip-address: 10.56.196.101</pre>

**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.



**Note** Upgrading clears the traffic data stored by the SD-AVC network service.

**Table 16: 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 [Software Download](#) 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 18](#)).

3. On the device, verify the MD5 checksum of the downloaded package. The correct MD5 checksum value appears 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 17: Command Details**

CLI keyword/argument	Description
<i>disk-with-OVA</i>	Specify one of the following, according to the platform type. The location refers to where the OVA was stored in a previous step. <ul style="list-style-type: none"> <li>• CSR: <b>bootflash</b></li> <li>• ASR1000 Series or ISR4000 Series: <b>harddisk</b></li> </ul>
<i>OVA-filename</i>	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, on page 29](#)
- [Uninstalling the SD-AVC Network Service, on page 29](#)

## 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 29](#)
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, on page 31](#)
- [System Requirements: Network Devices Using SD-AVC, on page 31](#)
- [Configuration Prerequisites: Network Devices Using SD-AVC, on page 33](#)
- [Activating the SD-AVC Agent, on page 33](#)
- [Deactivating the SD-AVC Agent, on page 35](#)

## 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 37](#).

## 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 18: Network Device Requirements**

Platform	Recommended OS (extended maintenance release trains only)
Cisco ASR1001-X Aggregation Services Router	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)

Platform	Recommended OS (extended maintenance release trains only)
Cisco ASR1002-X Aggregation Services Router	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco ASR1001-HX Aggregation Services Router	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco ASR1002-HX Aggregation Services Router	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco 1100 Series Integrated Services Routers	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco ISR4000 Series Integrated Services Routers: 4221, 4321, 4331, 4431, 4451	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco Integrated Services Virtual Router	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco CSR1000V Cloud Services Router	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)  (See note 2 for information about Cisco CSR1000V license requirements.)
Cisco Route Processor RP2, operating on Cisco ASR1004, ASR1006, or ASR1013	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)
Cisco Route Processor RP3, operating on Cisco ASR1004, ASR1006, or ASR1013	Cisco IOS XE Amsterdam 17.3.1 or later Cisco IOS XE Fuji 16.9.1 or later (See note 1 for details of feature support.)



- Note**
1. Cisco IOS XE 17.3.1 adds support for Microsoft Office 365 traffic categories, the Clear State function, and additional custom application functionality (IPv6 subnet, extended IPv4 subnets, port ranges).
  2. The Cisco CSR1000V Cloud Services Router requires the following license: AX, 2.5 Gbps or higher throughput. See the [Cisco CSR1000V Data Sheet](#).

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 19](#).

## Configuration Prerequisites: Network Devices Using SD-AVC

The following are prerequisites for network devices to operate with SD-AVC:

- Application statistics:

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:

- **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.
- **IWAN solution:** (No additional user configuration required) Collection of application statistics is enabled by the use of Easy Performance Monitor (ezPM) and PfR policy.

- Unique hostname:

Each network device operating with SD-AVC requires a unique hostname. The following is an example of how to configure the hostname on a device:

```
Device(config)#hostname host123
```

## 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 37](#)

---

**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 118](#) 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

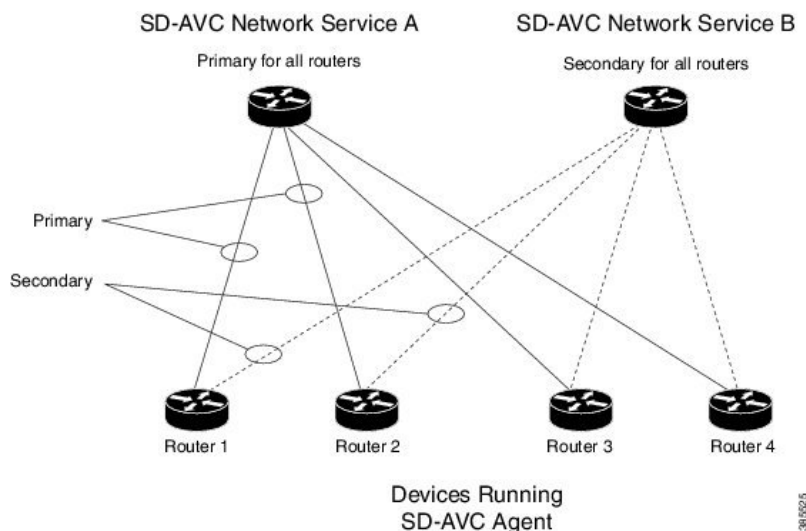
## SD-AVC High Availability

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 38](#)
- [Switchover Between Primary and Secondary SD-AVC Network Services, on page 38](#)

## 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 17](#).
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 31](#). 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:

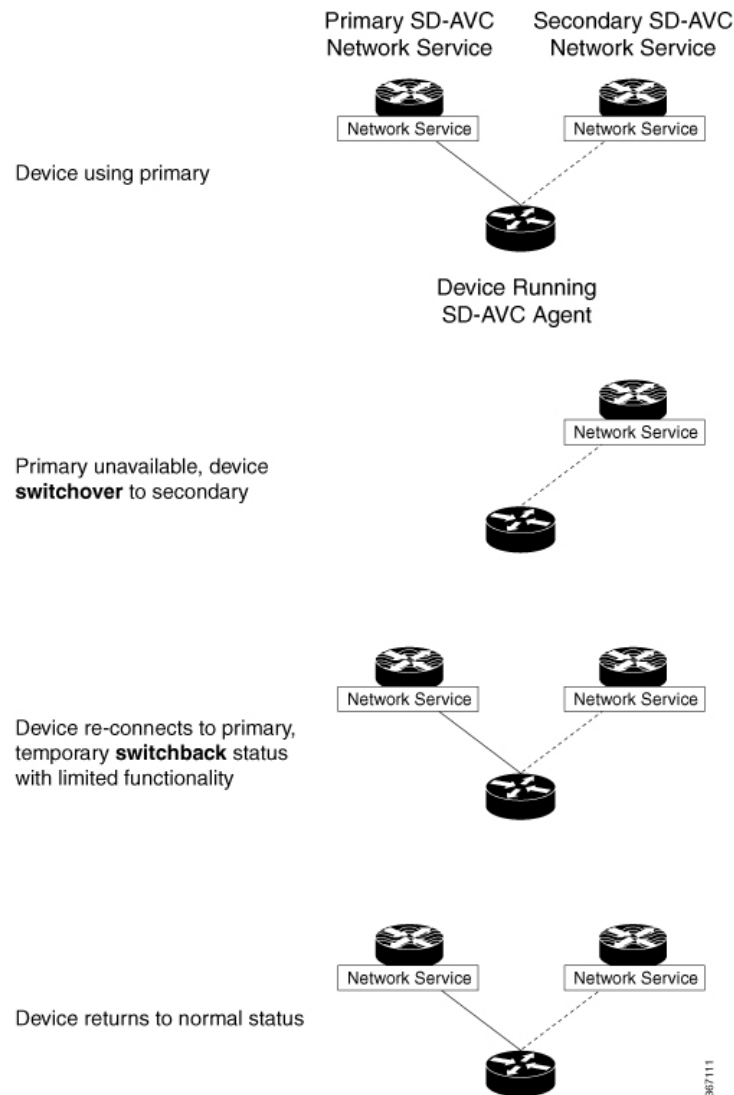
- Collecting traffic data from the device
- Displaying the traffic data
- Deploying Protocol Packs to the device if necessary

### 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**

- [Using SD-AVC, on page 43](#)
- [SD-AVC Notes and Limitations, on page 63](#)





# CHAPTER 7

## Using SD-AVC

- [Using SD-AVC, on page 43](#)
- [Connecting to the Dashboard, on page 43](#)
- [Application Visibility Page, on page 44](#)
- [Protocol Packs Page, on page 48](#)
- [Connectors Page, on page 50](#)
- [Customization Page, on page 54](#)
- [Serviceability Page, on page 56](#)
- [SD-AVC System Time and Displayed Times, on page 60](#)

## Using SD-AVC

Functionality	See...
Connect to the SD-AVC Dashboard	<a href="#">Connecting to the Dashboard, on page 43</a>
View traffic analytics interactively, monitor devices operating with SD-AVC	<a href="#">Application Visibility Page, on page 44</a>
Upload and deploy Protocol Packs	<a href="#">Protocol Packs Page, on page 48</a>
View details of external sources of application classification	<a href="#">Connectors Page, on page 50</a>
Configure custom applications	<a href="#">Custom Applications in SD-AVC, on page 54</a>
View system information, application rules, and debugging tools	<a href="#">Serviceability Page, on page 56</a> <a href="#">Application Rules Page, on page 59</a>

## 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 19: Top of Window*

Information/Control	Description
All Devices or selected segment	Indicates that the application data displayed in this window includes traffic handled by all devices in the network that are operating with SD-AVC.
Time Range	Time range for application data displayed on this page.
More Actions (...) button	<p>After selecting a specific network segment, you can use <b>More Actions &gt; Clear State</b> to clear the traffic classification data collected by the SD-AVC network service for that segment. This resets the application rules pack that the SD-AVC network service sends to devices in that network segment, and resets the application rules data stored on devices in the segment.</p> <p>The clear state process may take several minutes on the SD-AVC network service and on devices in the network segment. During this time, if a device sends packets to the SD-AVC network service, the packets are rejected.</p> <p><b>Note</b> This feature requires that the devices in the network use Cisco IOS XE Amsterdam 17.3.x or later.</p>

*Table 20: Summary Pane*

Information/Control	Description
Classification Score	<p>Last measured classification quality score for the device. This indicates the degree of classification quality (specificity), calculated according to traffic volume.</p> <p>Higher score indicates better quality.</p>


Information/Control	Description
Unclassified Traffic Discovery button 	Displays details of unclassified traffic. See <a href="#">Viewing Unclassified Traffic Details, on page 47</a> . To return, use the menu in the <b>Timeline</b> 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.
Timeline	Graph of one of the following (select in dropdown menu): <ul style="list-style-type: none"> <li>• Bandwidth</li> <li>• Classification score</li> <li>• First packet classification score</li> <li>• SD-AVC coverage ratio</li> <li>• Unclassified Traffic</li> </ul>

Table 21: 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 <b>Timeline</b> pane. Use the Search field to filter the display of traffic.

Table 22: SD-AVC Monitoring Pane

Information/Control	Description
<b>Note:</b> 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
Installed Protocol Packs	Protocol Packs installed on devices in the network.

Table 23: Business Relevance Pane

Information/Control	Description
Business Relevance Graph	<p><b>Note</b> Because business relevance depends on the network segment, this information is displayed when a single network segment or device is selected.</p> <p>Indicates portions of traffic classified as:</p> <ul style="list-style-type: none"> <li>• Business-relevant</li> <li>• Business-irrelevant</li> <li>• Default</li> </ul>

## 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

## Enabling and Disabling

To enable or disable the feature, use the control in:

**SD-AVC Dashboard > Serviceability > System**

## Notes and Limitations

See [SD-AVC Notes and Limitations, on page 63](#).

# Protocol Packs Page

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

Click **Manage & Deploy** to:

- Add Protocol Pack files to the repository, either from a local drive or by importing directly from Cisco. (Each Protocol Pack release may include one or more file versions, for compatibility with different devices in the network. Importing a Protocol Pack directly from Cisco automatically imports all file versions.) Protocol Packs in the repository are available to deploy to devices in the network.
- 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. When a new Protocol Pack is released, the SD-AVC Dashboard displays an alert on the **Application Visibility** page, in the **Devices** pane.

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.

SD-AVC can import Protocol Packs directly from Cisco. Alternatively, Protocol Packs can be downloaded using the Cisco [Software Download](#) 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.

## Importing the Latest Protocol Packs Directly from Cisco

When Cisco releases a new Protocol Pack, the SD-AVC Dashboard displays an alert on the Application Visibility page, in the Devices pane. Click the alert indicator to view details. SD-AVC can import the latest Protocol Pack release directly from Cisco, adding it to the repository. The Protocol Pack can then be deployed to devices in the network.

---

**Step 1** Protocol Packs page > **Manage & Deploy** button > **Protocol Pack Repository** > **Import from cloud**

**Step 2** If new Protocol Packs are available, they appear in a list. Select the Protocol Pack to import.

---

## Uploading Protocol Packs from a Local Drive 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 [Software Download](#) 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** > **Import from file system**

---

## 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 37](#).

---

**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.

## Connectors Page

This page displays details of the **Cloud Connector**, which manages multiple sources of application information used for classifying network traffic. The page displays any errors or warnings for the Cloud Connector, if applicable.



**Note** This page replaces the **External Sources** page that appeared in earlier SD-AVC releases.

## Cloud Connector

Beginning with the Cisco SD-AVC 3.0.0 release, SD-AVC connects to a cloud service provided by Cisco that improves traffic classification. The server addresses used by public internet sites and services change continually. Cisco Cloud Connector uses the latest information available to improve SD-AVC classification of traffic previously classified only in generic terms. For example, without Cloud Connector enabled, traffic from a web application using an unknown server might be classified simply as SSL, without any additional details. When enabled, Cloud Connector might be able to provide additional up-to-date details about this traffic.

To benefit from this service, enable the Cloud Connector in SD-AVC.

By default, SD-AVC shares telemetry with the cloud service to improve classification throughout the network.

The Cloud Connector setup enables you to specify the location for storing telemetry data for each network segment. This provides the flexibility to store telemetry data for different segments in different locations, and assists in compliance with EU General Data Protection Regulation (GDPR) regulations.

Enabling Cloud Connector in SD-AVC also enables the MS Office 365 Web Service, which was configured separately in earlier SD-AVC releases.

Source	Description
Cloud Updates	Provides application data from multiple external sources.

Source	Description
MS Office 365 Web Service	<p>Provides domain names used by Microsoft Office 365. Click the <b>View Details</b> button for details about each domain.</p> <p>See <a href="#">Office 365 Traffic Categories, on page 51</a>.</p> <p><b>Note</b> Beginning with SD-AVC 3.0.0, MS Office 365 Web Service operates through Cloud Connector. When Cloud Connector is enabled, MS Office 365 Web Service is automatically enabled also.</p> <p>In earlier releases, this component was called MS Office 365 Connector.</p>

### Telemetry Data Collection

The NBAR component of SD-AVC is configured to automatically connect and transmit telemetry data, in near real time, to Cisco. Telemetry information will be used by Cisco to improve SD-AVC functionality and facilitate development of new features that result in increased value. Telemetry information is transported securely to keep customer data private. Data collected includes general administrative information (such as SD-AVC IP address and telemetry status), cache rules (such as application name, IP addresses, and socket rating), unclassified and generic traffic (such as SSL and HTTP), analytics protocol discovery (such as number of active flows, number of TCP flows, and number of DNS flows), device information (such as deployed engine versions), and protocols information (such as application name and application attributes). Users may opt out of data collection for certain telemetry categories by turning this feature off in the “Settings” tab on the “Cloud Connector” page.

### Office 365 Traffic Categories

Cloud Connector receives information from Microsoft about domains and L3/L4 combinations (IP ranges, port, and L4 protocol) used for Office 365 traffic. Microsoft indicates the traffic category (Optimize, Allow, or Default) for each domain and L3/L4 combination. (See [Microsoft 365 Network Connectivity Principles](#).) Cisco SD-AVC identifies traffic from each of the domains and L3/L4 combinations as Office 365 traffic, and assigns it the traffic category provided by Microsoft

You can use the Office 365 traffic category when creating traffic policy, enabling you to apply policy decisions based on Office 365 traffic categorization. Recognizing Office 365 traffic by the L3/L4 combination offers the special advantage of first-packet classification, allowing traffic policy to be applied from the first packet of a flow.

**Device requirement:** To use Office 365 categories, devices must be using Cisco IOS XE Amsterdam 17.3.1 or later.

**Policy example using Office 365 traffic categories:** The following policy, configured on a device in the network, uses Office 365 traffic categories **optimize** and **allow**.

```
class-map match-any optimize
match traffic-category optimize
class-map match-all allow
match traffic-category allow
!
policy-map type ebr ebr-policy-return
parameter default flow-stickness
class optimize
    set ipv4 vrf traffic next-hop 10.0.0.254
```

```

class allow
  set ipv4 vrf traffic next-hop 10.0.0.254

interface GigabitEthernet0/0/1
service-policy type eubr input eubr-policy

```

## Enabling the Cloud Connector

### Prerequisites

- **Cloud server domain access**

The device hosting the SD-AVC network service requires access to the following Cisco SD-AVC cloud server domains:

```

api.cisco.com
cloudsso.cisco.com
prod.sdavc-cloud-api.com

```

Ensure that:

- The host device's access to these domains is not blocked by a firewall.
- If a proxy server is required, configure the server from the SD-AVC Dashboard, using **Serviceability > Proxy Settings**.

### Procedure

1. In the SD-AVC Dashboard, open the **Cloud Connector** page.
2. On the **Cloud Connector** page, click **Settings**. A pop-up displays connection information.
3. In the pop-up, click the **Cisco API Console** link. The Cisco API Console page opens in a browser.
4. On the Cisco API Console page, sign in using your Cisco credentials.
5. On the Cisco API Console page, open the **My Applications** tab. A page opens for registering a new application.




---

**Note** (The steps on the Cisco API Console page are subject to change.)

---

6. Register SD-AVC.
  - a. Name of your application:  
Use any descriptive name. Save this name for a later step.
  - b. Select the **Client Credentials** checkbox.
  - c. Select the **Hello API** checkbox.
  - d. In the Terms of Service section, select the checkbox to agree with the terms.

- e. Click **Register**. The Cisco API Console page displays the Client ID and Client Secret details. Keep this page open to complete the procedure.
7. In the SD-AVC Dashboard, complete the activation process in the open pop-up.
  - a. Enter the Client ID and Client Secret details from the Cisco API Console page.



---

**Note** These credentials expire after 90 days.

---

- b. For Organization Name, use the descriptive name that you entered on the Cisco API Console page in the “Name of your application” field.
          - c. (Optional) Click Change Data Store Location, and select a region. This determines where your telemetry data is stored. For organizations located in Europe, it is recommended to change the location to Europe, in accordance with EU General Data Protection Regulation (GDPR) regulations.
          - d. Wait for the Cisco Console API to propagate your credential information in the system. This may take a few minutes.
          - e. Click **Authenticate**. The pop-up closes.

When this process is complete, the **Cloud Connector** page shows the external sources enabled: Cloud Updates and MS Office 365 Web Service. When enabled, the box for an external source shows a **View Details** button. For either of these sources, click View Details to show details of the network traffic classification affected.



---

**Note** After enabling Cloud Connector, there may be a delay of several minutes before any details are displayed by the **View Details** button.

---

On the **Application Visibility** page, the **Cloud Connector** pane shows a green indicator when Cloud Connector is enabled.

## DNS Server Connectivity

Cloud Connector 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 additional 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
```

### Viewing DNS Servers

To view the configured DNS servers, open the **SD-AVC Dashboard > Serviceability** page > **System** pane.

## Configuring a Proxy Server

Some organizations require use of a proxy server. Use the **Proxy Settings** page to view or configure a proxy server.

1. On the **SD-AVC Dashboard > Serviceability** page, click **Proxy Settings**. The current proxy server configuration is shown.
2. To configure or update proxy server settings, enter the following:
  - Protocol: HTTP or HTTPS
  - Server IP: IP address or domain name
  - Server Port: Port number
3. If credentials are required, click **Advanced Options** and enter the credentials.
4. Click **Save**.

## Customization Page

### Custom Applications in SD-AVC

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). Custom applications, also called user-defined applications, may be specified network-wide using SD-AVC.

Each application includes a signature of details that identify the network application, such as:

- Server names
- IP addresses
- Ports
- L3 or L4 protocol

You can configure custom applications using the SD-AVC Dashboard or using the SD-AVC REST API. See [User-defined Applications, on page 103](#).

### Creating a Custom Application

The following procedure describes the typical workflow for creating a custom application in SD-AVC. It is also possible to create a custom application by these methods:



- In the SD-AVC Dashboard, on the **Application Visibility** page, select **Unclassified Traffic**, select one or more rows, and click the **Customize** button. This opens the **Custom Applications** page, with the IP address of the selected unclassified traffic entered automatically.
- Configure custom applications using the SD-AVC REST API. See [User-defined Applications, on page 103](#).



**Note** If you create custom applications using the SD-AVC Dashboard, do not create a new set of custom applications through the REST API using POST. Doing so overwrites the custom applications created through the SD-AVC Dashboard. You can add custom applications through the REST API using PUT.

1. In the SD-AVC Dashboard, on the **Customization** page, click **Applications**.
2. In the **Custom Application Rules** page, click **New Rule**.
3. In the drop-down list of network segments, select a segment.
4. In the **Application Name** field, enter the name of the source.
5. Click **New Application** and enter information as follows.

Field	Description
Server Names	Enter one or more server names.
L3/L4	Select this and enter: <ul style="list-style-type: none"> <li>• One or more IP addresses.</li> <li>• Ports or port range.</li> <li>• L3 or L4 protocol: TCP, UDP, or TCP-UDP</li> </ul>
Attributes	(optional) Specify attributes for the application: <ul style="list-style-type: none"> <li>• Category</li> <li>• Subcategory</li> <li>• Application Group</li> <li>• Business Relevance</li> <li>• Traffic Class</li> <li>• Application Set</li> </ul>

6. Click **Save & Deploy**. The new application appears in the list of applications.

# 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.
	<p><b>System Logs</b></p> <p><b>Serviceability &gt; System &gt; General Information</b></p> <p>SD-AVC keeps a system log as a local file. The log is available for download <a href="#">here</a>.</p> <p>Beginning with this release, SD-AVC can also send error messages to an external system log server in real time.</p>
	<p><b>Unclassified Traffic Visibility: Enable/Disable</b></p> <p><b>Serviceability &gt; System &gt; Settings</b></p> <p>Enables/disables the unclassified traffic analysis feature (see <a href="#">Unclassified Traffic Analysis and Discovery, on page 46</a>). When enabled, top hosts and sockets will be identified on the Application Visibility page, in the table and in the graph of traffic bandwidth.</p> <p>After enabling Unclassified Traffic Visibility, the effect is not immediate. SD-AVC gathers information about top hosts and sockets in network traffic (communicated from network devices to the SD-AVC network service by Netflow) and identifies them gradually.</p> <p>Similarly, after disabling the feature, the top hosts and sockets that have been identified may remain in the table and graph for a period of time (dependent on the time range displayed) while SD-AVC continues to analyze traffic and update the Application Visibility page.</p> <p>Default: Enabled</p>
	<p><b>Behavioral Based Classification: Enable/Disable</b></p> <p><b>Serviceability &gt; System &gt; Settings</b></p> <p>Cisco NBAR uses a method called behavioral classification to assist in classifying traffic. This method relies on heuristic techniques to determine the purpose of servers or clients.</p> <p>This method might produce undesired results in some cases – for example, when the SD-AVC network service communicates with networks through a proxy device. For these situations, disable behavioral based classification in SD-AVC.</p> <p><b>Note</b> Using the Disable option requires that the devices in the network use Cisco IOS XE Amsterdam 17.3.x or later.</p>

Tool	Description
	<p><b>SSL Certificate</b></p> <p><b>Serviceability &gt; System &gt; Settings</b></p> <p>By default, the browser-based SD-AVC Dashboard provides a self-signed SSL certificate that appears in a browser as untrusted. Optionally, you can register your specific domain and acquire a signed SSL certificate <b>specifically for use with SD-AVC</b>, and import the certificate into SD-AVC. Connecting to the SD-AVC Dashboard is then secure and trusted.</p> <p><b>Note</b> Ensure that the installed SSL certificate is valid. SD-AVC does not automatically remove an SSL certificate when it expires, so replace the certificate before it expires. An invalid certificate may prevent connection to the SD-AVC Dashboard.</p> <p>If you encounter difficulty connecting to the SD-AVC Dashboard because of an untrusted or expired certificate, connect using the IP address of the network service. You can ping the hostname to get the IP address of the network service.</p> <ol style="list-style-type: none"> <li>1. Create a certificate for the SD-AVC domain (self-signed or signed by a certification authority), and save the certificate file to a local directory.</li> <li>2. Click <b>Change</b> and upload the certificate file. <ul style="list-style-type: none"> <li>• <b>Certificate:</b> Select <b>PKCS</b> or <b>JKS</b> for the certificate format.</li> <li>• <b>Keystore Passphrase:</b> Keystore passphrase for the certificate.</li> <li>• <b>Key Alias:</b> The key alias (called friendlyName when using OpenSSL) is set when creating the certificate. It may be a default value or a specified custom name.</li> <li>• <b>Key Password:</b> Enable this option if the alias is configured with a key passphrase, and enter the passphrase.</li> </ul> </li> <li>3. Click <b>Upload &amp; Activate</b>. It may require a few minutes to activate the certificate before you can reconnect to the SD-AVC Dashboard.</li> <li>4. Log into the SD-AVC Dashboard using the hostname associated with the SSL certificate.</li> </ol>
	<p><b>Syslog Server</b></p> <p><b>Serviceability &gt; System &gt; Settings</b></p> <p>SD-AVC can send error messages to an external system log server in real time. To configure a server, enter the server address and click <b>Update</b>.</p>
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. <a href="#">Application Rules Page, on page 59</a>
Proxy Settings	View or configure proxy server settings. <a href="#">Configuring a Proxy Server, on page 54</a>

## 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 11](#)). 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: <ul style="list-style-type: none"> <li>• Protocol Pack protocol</li> <li>• User-defined protocols</li> </ul>
Entry Type	Network cache type: <ul style="list-style-type: none"> <li>• <b>L3</b></li> <li>• <b>socket-cache</b></li> </ul>
Source	Protocol/application: <ul style="list-style-type: none"> <li>• <b>network:</b> Identification of flow by Protocol Pack</li> <li>• <b>dynamic:</b> Identification of flow by user-defined application</li> <li>• <b>ac_hosts</b> or <b>ac_sockets:</b> Tracking of flow by Unclassified Traffic Discovery feature</li> </ul>
Rating	Number of significant flow (session) hits in the network layer
Transport	Transport protocol

Field	Description
TTL	<p>Time to Live: Timespan (in cycles) for tracking the socket</p> <ul style="list-style-type: none"> <li>• If there is active traffic for the socket, the TTL remains at maximum value of 384.</li> <li>• If there is no active traffic for the socket, the TTL value is decremented over time.</li> </ul>

## 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
```







## CHAPTER 8

# SD-AVC Notes and Limitations

- [General](#), on page 63
- [Setup](#), on page 63
- [Classification](#), on page 64
- [High Availability](#), on page 65
- [Protocol Pack](#), on page 65
- [REST API](#), on page 65

## General

Note/Limitation	Description
Maximum number of participating network devices	Maximum number of network devices participating with SD-AVC (running the SD-AVC agent): 6000

## 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 appears on the <a href="#">Download Software</a> 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 <a href="#">Operating the SD-AVC Network Service with Host Interface Attached to a VRF</a> , on page 81 for configuration details.

Note/Limitation	Description
Running and startup configurations of participating devices	<p>SD-AVC adds two lines to the running and startup configurations of participating devices:</p> <ul style="list-style-type: none"> <li>To enable the MS Office 365 Web Service, which improves classification of Microsoft Office traffic: <pre>ip nbar protocol-pack bootflash:sdavc/sdavc_ppdk.pack force</pre> </li> <li>When SD-AVC deploys Protocol Packs to a device: <pre>ip nbar protocol-pack harddisk:sdavc/protocol-pack-name.pack</pre> </li> </ul>

## 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 <a href="#">SD-AVC and Application Recognition, on page 12</a> .
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 <a href="#">Unclassified Traffic Analysis and Discovery, on page 46</a> .
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 <b>Unclassified Traffic</b> 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 <a href="#">Unclassified Traffic Analysis and Discovery, on page 46</a> .

## High Availability

Note/Limitation	Description
Error status and Protocol Pack deployment during high availability switchover and switchback	<p>In SD-AVC high availability configurations, if the primary SD-AVC network service becomes unavailable, network devices switch to the secondary SD-AVC network service. When the primary SD-AVC network service becomes available again, the devices switch back to primary.</p> <p>The <b>switchover</b> and <b>switchback</b> processes require approximately 30 minutes. During this time:</p> <ul style="list-style-type: none"> <li>• Service in the network continues normally without interruption.</li> <li>• The <b>SD-AVC Dashboard &gt; Application Visibility</b> page shows an error status for the devices.</li> <li>• The <b>SD-AVC Dashboard &gt; Protocol Packs</b> page shows that the devices are not active. During this brief period, SD-AVC does not deploy Protocol Packs to the devices.</li> </ul> <p>See <a href="#">SD-AVC High Availability, on page 37</a>.</p>

## Protocol Pack

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.
Protocol Pack deployment during high availability switchover and switchback	See <a href="#">High Availability, on page 65</a> .

## REST API

Note/Limitation	Description
User-defined application source	In the initial release of the REST API, only one source is supported.

Note/Limitation	Description
Total number of user-defined applications available	For each network segment: <ul style="list-style-type: none"> <li>• Maximum user-defined applications: 1100</li> <li>• Maximum L3L4 rules: 20000</li> <li>• Maximum serverNames: 50000</li> <li>• Maximum wildcards followed by period (.): 50000 (maximum serverNames) Example: *.cisco.com matches www.cisco.com, developer.cisco.com</li> <li>• Maximum prefix wildcards as part of a server name: 256 Example: *ample.com matches www.example.com</li> </ul>
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 user-defined 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, on page 67](#)
- [Troubleshooting SD-AVC Network Service Issues, on page 70](#)
- [Troubleshooting SD-AVC Agent Issues, on page 76](#)
- [Troubleshooting SD-AVC Connectivity Issues, on page 77](#)
- [Troubleshooting Protocol Pack Issues, on page 80](#)

## 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 24: Troubleshooting: SD-AVC Network Service

Problem	How it appears	Troubleshooting
SD-AVC network service: installation failure	SD-AVC not active, <b>sd-avc status</b> shows installation failure.	<p><b>Summary</b></p> <p>Diagnose with <b>sd-avc status</b> and then <b>service sd-avc trace</b>.</p> <p>Possible issues:</p> <ul style="list-style-type: none"> <li>• Not enough memory: see system requirements</li> <li>• Not enough disk space: see system requirements</li> </ul> <p><b>Troubleshooting Details</b></p> <p><a href="#">Troubleshooting Commands for Network Service Issues, on page 70</a></p> <p><a href="#">System Requirements: SD-AVC Network Service Host, on page 18</a></p>
SD-AVC network service: activation failure	SD-AVC not active, <b>sd-avc status</b> shows activation failure.	<p><b>Summary</b></p> <p>Diagnose with <b>sd-avc status</b> and then <b>service sd-avc trace</b>.</p> <p>Possible issue: Something may be using CPU resources. Ensure that nothing is using CPU resources.</p> <p><b>Troubleshooting Details</b></p> <p><a href="#">Troubleshooting Commands for Network Service Issues, on page 70</a></p> <p><a href="#">Activation Failure Caused by Shared CPU Resources, on page 73</a></p>
SD-AVC network service: configuration failure	SD-AVC not active, <b>sd-avc status</b> shows configuration failure.	<p><b>Summary</b></p> <p>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:</p> <p><b>interface <i>interface</i> no ip vrf forwarding</b></p> <p><b>Troubleshooting Details</b></p> <p><a href="#">Configuration Failure Caused by VRF, on page 75</a></p>

**Table 25: Troubleshooting: SD-AVC Agent Operating on Devices in the Network**

Problem	How it appears	Troubleshooting
NBAR2 is not activated on the device	On the <b>Dashboard &gt; Application Visibility</b> page, the <b>Timeline</b> graph of bandwidth shows no activity.	<p><b>Summary</b></p> <p>NBAR2 is not active: Activate NBAR2 on the device.</p> <p><b>Troubleshooting Details</b></p> <p><a href="#">NBAR2 Not Activated on Interfaces, on page 76</a></p>
Error: More than one active session	<p>When attempting to enable the agent, an error message indicates that there is an active session already.</p> <p><b>Example:</b></p> <pre>Device(config-sd-service)# controller %% NBAR Error: There is an active session already in sd-service-controller submode</pre>	<p><b>Summary</b></p> <p>Close any interfering sessions.</p> <p><b>Troubleshooting Details</b></p> <p><a href="#">Active Sessions Preventing Agent Configuration, on page 76</a></p>

**Table 26: Troubleshooting: Connectivity between SD-AVC Network Service and Devices in the Network**

Problem	How it appears	Troubleshooting
UDP	Warning in: <b>Dashboard &gt; Application Visibility</b> page > <b>SD-AVC Monitoring</b> pane	<p><b>Summary</b></p> <p>Check UDP connectivity.</p> <p><b>Troubleshooting Details</b></p> <p><a href="#">Problem with UDP Communication with Devices, on page 77</a></p>
TCP	Warning in: <b>Dashboard &gt; Application Visibility</b> page > <b>SD-AVC Monitoring</b> pane	<p><b>Summary</b></p> <p>Check TCP connectivity.</p> <p><b>Troubleshooting Details</b></p> <p><a href="#">Problem with TCP Communication with Devices , on page 78</a></p>

Problem	How it appears	Troubleshooting
FTP and HTTP	Warning in: <b>Dashboard &gt; Application Visibility page &gt; SD-AVC Monitoring</b> pane	<p><b>Summary</b></p> <ol style="list-style-type: none"> <li>1. Check FTP/HTTP connectivity: <b>show avc sd-service info summary</b></li> <li>2. Verify FTP/HTTP connectivity between the SD-AVC network service and the network device. This includes checking ACL, firewalls, and so on.</li> <li>3. On the device, ensure that FTP/HTTP connectivity is possible from the routable interface to the SD-AVC network service. To enable FTP/HTTP connections from a specific interface, use: <b>ip ftp source-interface</b> <i>interface-name</i> <b>ip http client source-interface</b> <i>interface-name</i></li> </ol> <p><b>Troubleshooting Details</b></p> <p><a href="#">Problem with FTP/HTTP Communication with Devices, on page 79</a></p>

Table 27: 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."	<p><b>Summary</b></p> <p>Load the Protocol Pack manually on the device to determine whether the Protocol Pack is valid.</p> <p><b>Troubleshooting Details</b></p> <p><a href="#">Failure to Deploy Protocol Pack to Device, on page 80</a></p>

# 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 28: Summary

Command	Description
<code>service sd-avc status</code>	Status of SD-AVC network service installation, configuration, and activation
<code>service sd-avc trace</code>	Memory or disk problems
<code>show virtual-service list</code>	Activation errors
<code>show virtual-service global</code>	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 29: Resolving Memory or Disk Errors*

Problem	Solution
Memory error	Increase the device memory to the amount specified in <a href="#">System Requirements: SD-AVC Network Service Host, on page 18.</a>
Disk error	Increase the size of the harddisk or bootflash (for CSR) device according to the requirements specified in <a href="#">System Requirements: SD-AVC Network Service Host, on page 18.</a>

## 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 [errormsg] [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)

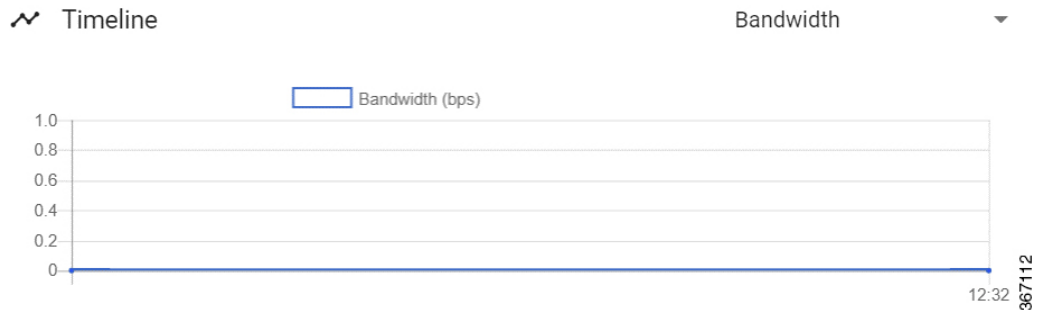
Devices in the network that are using SD-AVC

### 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 33](#).

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 communicates with SD-AVC agents in the network using:

- TCP over port 21 (FTP) for devices using Cisco IOS XE 16.11.x Gibraltar or earlier
- TCP over port 8080 (HTTP) for devices using Cisco IOS XE 16.12.1 Gibraltar or later

(See [System Requirements: Network Devices Using SD-AVC](#), on page 31.)

### Troubleshooting

1. If an FTP warning appears in the SD-AVC Dashboard, for a specific device in the network, check connectivity on TCP port 21 (FTP) or port 8080 (HTTP). 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 port 21 (FTP) and port 8080 (HTTP) between the affected device and the SD-AVC network service.



# Problem with FTP/HTTP 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/HTTP to communicate with the devices that it manages.

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

For FTP/HTTP issues caused by connecting a device to an internal FTP/HTTP server for non-SD-AVC FTP/HTTP traffic, see [Scenario: Internal FTP/HTTP Server, on page 120](#).

## Troubleshooting

1. If an FTP warning appears in the SD-AVC Dashboard while the **Connection** status is green, for a specific device in the network, check the FTP/HTTP 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/HTTP connection status. "Status: DISCONNECTED" in the output below shows an FTP/HTTP 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/HTTP communication is possible between the affected device and the SD-AVC network service.

1. Verify that nothing is preventing FTP/HTTP network connectivity between the SD-AVC network service and the network device. This includes checking ACL, firewalls, and so on.
2. To determine whether communication with the SD-AVC network service uses FTP or HTTP, execute the following command on the device. The example shows HTTP.

```
show avc sd-service info detailed | inc Transport for file copy:
```

```
Transport for file copy: http
```

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

```
ip ftp source-interface interface-name
```

Example:

```
ip ftp source-interface GigabitEthernet1  
ip http client source-interface g1
```

## 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 49](#). When deploying Protocol Packs to one or more devices, if the deployment fails, the results page may show an error.

#### Troubleshooting

1. Operating an excessive number of services on a router simultaneously can cause insufficient Quantum Forwarding Processor (QFP) memory to be available to load a new Protocol Pack. In this case, SD-AVC may display a “Low Memory” error message when you try to update a Protocol Pack.

You can use the **show platform hardware qfp active infrastructure exmem statistics** command on a router to check the status of QFP memory resources. If less than 50 MB are available, you can reload the router to free memory, and attempt to load the Protocol Pack again. If it fails a second time, you can use the **show platform hardware qfp active infrastructure exmem statistics user** command to display the individual processes using QFP memory.

2. Load the Protocol Pack manually on the device indicated by the error to verify 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
```

3. 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 20](#) 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
!
```





## APPENDIX **C**

# Configuring Secure Connectivity

- [Securing Connections to the SD-AVC Network Service, on page 83](#)
- [Configuring ACL Access, on page 85](#)

## Securing Connections to the SD-AVC Network Service

The SD-AVC Network Service, operating on a host device, communicates with:

- One or more PC-type devices running the SD-AVC Dashboard
- Network devices running the SD-AVC Agent

### Enable Connectivity

To enable connectivity, ensure that ports, firewall policy, and so on, are configured to enable communication between the SD-AVC Network Service and the other relevant devices. See [Configuring Connectivity, on page 19](#).

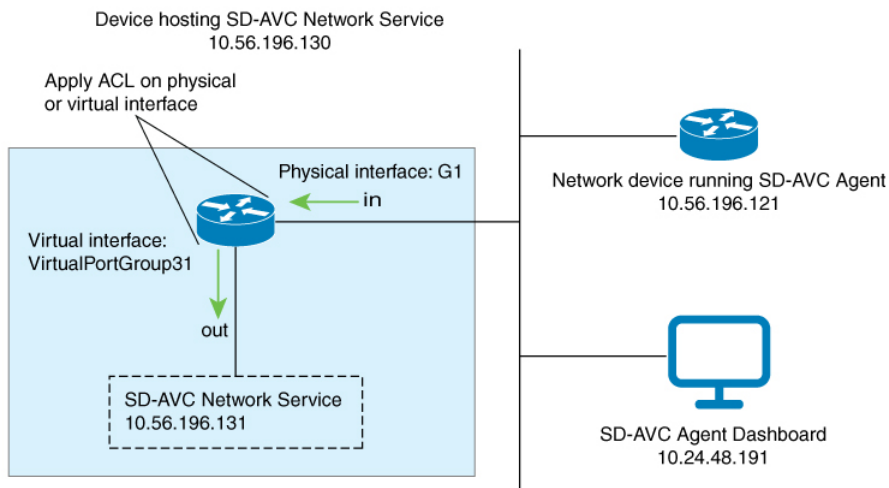
### Secure Connectivity

You can optionally use the mechanisms described below to secure the connections between the SD-AVC Network Service and other devices.

Method	Information
Access control list (ACL)	<p>Configure an ACL on the device hosting the SD-AVC Network Service to define a white list of devices authorized to communicate with the SD-AVC Network Service.</p> <p>The ACL may be applied on a physical interface of the host device, or on the virtual interface between the host device and the SD-AVC Network Service.</p> <p><b>Note</b> When using ACLs, only configured addresses will have access to the device hosting the SD-AVC Network Service.</p>

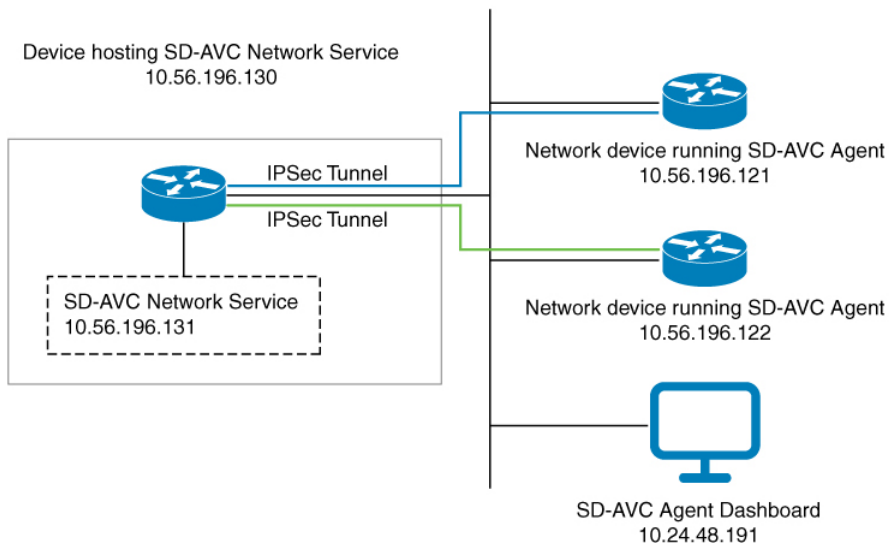
Method	Information
IPsec tunnels	<p>For network scenarios that require a secure connection between the SD-AVC Network Service and network devices running the SD-AVC agent, you can use IPsec tunnels to encrypt the SD-AVC communication.</p> <p>For information about configuring Cisco IOS IPsec VPN connections, see <a href="#">Cisco IOS IPsec</a>.</p>

Figure 5: Apply ACL to Physical Interface or Virtual Interface



355864

Figure 6: IPsec Tunnels between SD-AVC Network Service and Network Devices



355865

# Configuring ACL Access

## Prerequisites

Ports, firewall policy, and so on, have been configured to enable communication between the SD-AVC Network Service and:

- Network devices running the SD-AVC Agent
- PC-type devices that connect to the SD-AVC Network Service to display the SD-AVC Dashboard

## Configuring ACL

1. Create the ACL.

```
ip access-list extended sdavc-acl
```

2. Configure access for a PC-type device that will connect to run the SD-AVC Dashboard.

```
permit tcp host dashboard-access-device-address host sdavc-network-service-address eq 8443
```

**Example:**

```
permit tcp host 10.24.48.191 host 10.56.196.131 eq 8443
```

3. Configure access for one or more network devices running the SD-AVC Agent. For each network device, permit these ports:

```
UDP: 50000
```

```
TCP: 21, 8080, 59990-60000
```

The complete syntax options for ACL configuration, such as address wildcards, are beyond the scope of this document. For complete information about configuring ACL, see the documentation for your platform.

```
permit udp host sdavc-agent-address host sdavc-network-service-address eq 50000
```

```
permit tcp host sdavc-agent-address host sdavc-network-service-address eq 21
```

```
permit tcp host sdavc-agent-address host sdavc-network-service-address eq 8080
```

```
permit tcp host sdavc-agent-address host sdavc-network-service-address range 59990 60000
```

**Example:**

```
permit udp host 10.56.196.121 host 10.56.196.131 eq 50000
```

```
permit tcp host 10.56.196.121 host 10.56.196.131 eq 21
```

```
permit tcp host 10.56.196.121 host 10.56.196.131 eq 8080
```

```
permit tcp host 10.56.196.121 host 10.56.196.131 range 59990 60000
```

4. Apply the ACL to a physical interface of the host device or to the virtual interface between the host device and the SD-AVC Network Service. Use one of the following:

- Physical interface (note the **in** keyword):

```
interface interface
```

```
ip access-group sdavc-acl in
```

**Example:**

```
interface GigabitEthernet1
  ip access-group sdavc-acl in
```

- Virtual interface (note the **out** keyword):

```
interface virtual-interface
ip access-group sdavc-acl out
```

**Example:**

```
interface VirtualPortGroup31
  ip access-group sdavc-acl out
```

## Examples

Complete example, configuring a single device for Dashboard access and a single network device. This example uses the virtual interface option:

```
ip access-list extended sdavc-acl
  permit tcp host 10.24.48.191 host 10.56.196.131 eq 8443
  permit udp host 10.56.196.121 host 10.56.196.131 eq 50000
  permit tcp host 10.56.196.121 host 10.56.196.131 eq 21
  permit tcp host 10.56.196.121 host 10.56.196.131 range 59990 60000

interface VirtualPortGroup31
  ip access-group sdavc-acl out
```

Complete example, configuring a single device for Dashboard access, and a range of devices (10.56.0.0 to 255). This example uses the physical interface option.

```
ip access-list extended sdavc-acl
  permit tcp host 10.24.48.191 host 10.56.196.131 eq 8443
  permit udp 10.56.0.0 0.0.255.255 host 10.56.196.131 eq 50000
  permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.131 eq 21
  permit tcp 10.56.0.0 0.0.255.255 host 10.56.196.131 range 59990 60000

interface GigabitEthernet1
  ip access-group sdavc-acl in
```





## APPENDIX **D**

# Configuring CSR1000V for SD-AVC

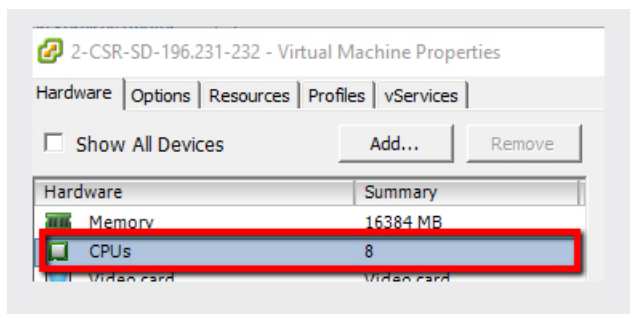
- [Allocating VM CPUs for Cisco CSR1000V, on page 87](#)

## 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 18](#).)



- 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, on page 89](#)
- [Authentication from SD-AVC Network Service, on page 92](#)
- [Configure Cloud Connector Credentials, on page 93](#)
- [Configure Cloud Connector Telemetry Data Location, on page 94](#)
- [System, on page 95](#)
- [Cloud Connector, on page 98](#)
- [External Sources, on page 102](#)
- [User-defined Applications, on page 103](#)
- [Generic Applications, on page 115](#)
- [REST API Notes and Limitations, on page 115](#)

## REST API Overview

The REST API provides numerous system functions, including:

- Displaying information about devices in the SD-AVC network
- Controlling external sources
- Displaying information about generic traffic
- Creating user-defined applications



**Note** Using the REST API requires authentication. See [Authentication from SD-AVC Network Service, on page 92](#).

**Table 30: Authentication**

|                                                                                                           |                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <b>POST</b><br><code>https://SD-AVC-network-service-address:8443/avc-sd-service/external-api/login</code> | Acquires an authentication token, enabling use of the REST API.<br><a href="#">Authentication from SD-AVC Network Service, on page 92</a> |
|-----------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------|

Table 31: System

|                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                            |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>GET</b> /avc-sd-service/external-api/system-info                                                                                                                                                                                                                                                            | Displays the SD-AVC version and system times.<br><a href="#">Display System Information, on page 95</a>                                                                                    |
| <b>GET</b> /avc-sd-service/external-api/devices                                                                                                                                                                                                                                                                | Displays devices in the SD-AVC network.<br><a href="#">Display Devices, on page 95</a>                                                                                                     |
| <b>POST</b> /avc-sd-service/external-api/remove-devices                                                                                                                                                                                                                                                        | Removes a device from the SD-AVC network.<br><a href="#">Delete Devices from SD-AVC, on page 96</a>                                                                                        |
| <b>GET</b> /avc-sd-service/external-api/visibility?period= <i>period</i><br>-<br><b>GET</b> /avc-sd-service/external-api/visibility/ <i>segmentName</i><br>?period= <i>period</i><br>-<br><b>GET</b> /avc-sd-service/external-api/visibility/ <i>segmentName</i> /<br><i>deviceName</i> ?period= <i>period</i> | Display traffic analytics (applications and bandwidth) for the complete SD-AVC network, a specific segment, or a specific device.<br><a href="#">Display Traffic Analytics, on page 97</a> |

Table 32: Cloud Connector

|                                                                                                                                                          |                                                                                                                                                                          |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>POST</b> /avc-sd-service/external-api/cloud/authorize<br><b>Note</b> Deprecated in SD-AVC 4.0.0. Use:<br>/avc-sd-service/external-api/cloud/configure | Connect to Cloud Connector.<br><a href="#">Connect to Cloud Connector, on page 98</a>                                                                                    |
| <b>POST</b> /avc-sd-service/external-api/cloud/configure                                                                                                 | Enter Cloud Connector credentials.<br><a href="#">Configure Cloud Connector Credentials, on page 93</a>                                                                  |
| <b>POST</b> /avc-sd-service/external-api/cloud/configureSegment?segment= <i>segmentName</i>                                                              | Specify the location for storing Cloud Connector telemetry data for a specific segment.<br><a href="#">Configure Cloud Connector Telemetry Data Location, on page 94</a> |
| <b>POST</b> /avc-sd-service/external-api/cloud/disable                                                                                                   | Disable Cloud Connector.<br><a href="#">Disable Cloud Connector, on page 99</a>                                                                                          |
| <b>GET</b> /avc-sd-service/external-api/cloud/status                                                                                                     | Get Cloud Connector status.<br><a href="#">Display Cloud Connector Status, on page 101</a>                                                                               |

|                                                                                          |                                                                                                                                      |
|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <b>POST</b> /avc-sd-service/external-api/cloud/removeCredentials                         | Clears the credentials that have been entered for Cloud Connector.<br><a href="#">Remove Cloud Connector Credentials, on page 99</a> |
| <b>GET</b> /avc-sd-service/external-api/cloud/getCurrentCloudConfig                      | Get current configuration for Cloud Connector.<br><a href="#">Display Current Cloud Configuration, on page 100</a>                   |
| <b>GET</b><br>/avc-sd-service/external-api/app-rules?detailed=true&sourceId=PP_Extension | Get currently collected cloud data (rules, server names, sockets, and so on).<br><a href="#">Display Cloud Data, on page 100</a>     |

**Table 33: External Sources**

|                                                                                         |                                                                                                                             |
|-----------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| <b>POST</b><br>/avc-sd-service/external-api/external-sources/ <i>externalSourceName</i> | Enables or disables receiving data from an external source.<br><a href="#">Enable/Disable External Sources, on page 102</a> |
| <b>GET</b> /avc-sd-service/external-api/external-sources                                | Displays status of external sources.<br><a href="#">Display Status of External Sources, on page 103</a>                     |

**Table 34: User-defined Applications**

|                                                         |                                                                                                                        |
|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------|
| <b>POST</b> /avc-sd-service/external-api/app-rules<br>- | Create one or more user-defined applications.<br><a href="#">Create User-defined Application Rules, on page 104</a>    |
| <b>PUT</b> /avc-sd-service/external-api/app-rules<br>-  | Add a user-defined application to an existing set.<br><a href="#">Add a User-defined Application Rule, on page 111</a> |

|                                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                         |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>GET</b> /avc-sd-service /external-api/app-rules</p> <p>-</p> <p><b>GET</b> /avc-sd-service /external-api/app-rules?sourceId=<i>sourceId</i></p> <p>-</p> <p><b>GET</b> /avc-sd-service /external-api/app-rules?segment=<i>segmentName</i></p> <p>-</p> <p><b>GET</b> /avc-sd-service /external-api/app-rules?segment=<i>segmentName</i>&amp;app=<i>applicationName</i></p>             | <p>Displays user-defined applications defined by REST API.</p> <p><a href="#">Display User-defined Application Rules, on page 112</a></p>               |
| <p><b>GET</b> /avc-sd-service/external-api/app-rules/status</p> <p>-</p> <p><b>GET</b> /avc-sd-service /external-api/app-rules/status[?sourceId=<i>sourceId</i>]</p>                                                                                                                                                                                                                         | <p>Displays activation status of user-defined applications, per device.</p> <p><a href="#">Display User-defined Application Status, on page 113</a></p> |
| <p><b>DELETE</b> /avc-sd-service /external-api/app-rules</p> <p>-</p> <p><b>DELETE</b> /avc-sd-service /external-api/app-rules?sourceId=<i>sourceId</i></p> <p>-</p> <p><b>DELETE</b> /avc-sd-service /external-api/app-rules?segment=<i>segmentName</i></p> <p>-</p> <p><b>DELETE</b> /avc-sd-service /external-api/app-rules?segment=<i>segmentName</i>&amp;app=<i>applicationName</i></p> | <p>Deletes user-defined applications.</p> <p><a href="#">Delete User-defined Applications, on page 114</a></p>                                          |

Table 35: Generic Applications

|                                                              |                                                                                                                                                               |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p><b>GET</b> /avc-sd-service/external-api/apps/generics</p> | <p>Displays the list of traffic types that contribute to "generic" traffic.</p> <p><a href="#">Display Generic Application Traffic Types, on page 115</a></p> |
|--------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------|

## Authentication from SD-AVC Network Service

Using the REST 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/avc-sd-service/external-api/login>

**Example:**

```
POST https://192.168.0.1:8443/avc-sd-service/external-api/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": "Bearer eyJhbGciOiJIUzUxMiJ9.eyJqdGkiOiJhYjZkGGUxOS0zMmU3LTRlY2ItYWQ5OC1kYmVmZTdjaE5YzYiLCJzdWIiOiJsYWIiLCJleHAiOiJlMzAwMgk1MzJ9.EfP3wd4fZbWrOQ6Skh-I0bbPffF4NaruB-o_OV0EQ7fwMwfmkUUNP00R58fRGKkYWR3tQu8HjovDp37EPtD15Q" }
```

3. Use this token in the "Authorization" request header field of each HTTP request.

## Configure Cloud Connector Credentials

### API

POST /avc-sd-service/external-api/cloud/configure

### Description

Configures the credentials for Cloud Connector. Use this when specifying the location for storing Cloud Connector telemetry data for one or more network segments.

After using this API to configure the credentials, you can specify the location for storing Cloud Connector telemetry data individually for each segment. See [Configure Cloud Connector Telemetry Data Location, on page 94](#).



**Note** This API is an alternative to the following API. Do not use them together.

POST /avc-sd-service/external-api/cloud/authorize

### Body

```
{
  "clientID": "clientId",
  "clientSecret": "secret",
  "orgName": "organizationName",
}
```

Table 36: Properties

| Property                | Description                            |
|-------------------------|----------------------------------------|
| <i>clientId</i>         | Client ID.                             |
| <i>secret</i>           | Secret for authentication.             |
| <i>organizationName</i> | Organization name. May include spaces. |

## Configure Cloud Connector Telemetry Data Location

### API

POST /avc-sd-service/external-api/cloud/configureSegment?segment=*segmentName*

### Description

Enables or disables transmitting and receiving Cloud Connector telemetry data, and specifies the location for storing the telemetry data. When using this API, specify the Cloud Connector credentials using the following API:

POST /avc-sd-service/external-api/cloud/configure

See [Configure Cloud Connector Credentials, on page 93](#).

### Body

```
{
  "txConfig": {
    "isEnabled": [true | false],
    "location": "locationId"
  },
  "rxConfig": {
    "isEnabled": [true | false]
  }
}
```

Table 37: Properties

| Property                    | Description                                                                                                                                                                                                        |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| segment= <i>segmentName</i> | Specifies a network segment.                                                                                                                                                                                       |
| txConfig                    | isEnabled: Enable or disable transmitting Cloud Connector telemetry data for the specified segment.<br><br>location: Specify the location for storing Cloud Connector telemetry data. See <i>locationId</i> below. |
| rxConfig                    | isEnabled: Enable or disable receiving Cloud Connector telemetry data for the specified segment.                                                                                                                   |
| <i>locationId</i>           | Location for storing Cloud Connector telemetry data for the specified segment.<br><br>Values: ASIA, CANADA, EU, US                                                                                                 |



### Example

In the following example, transmitting and receiving telemetry data are enabled, and the location is specified as "US".

```
{
  "txConfig": {
    "isEnabled": true,
    "location": "US"
  },
  "rxConfig": {
    "isEnabled": true
  }
}
```

## System

### System Overview

The REST API can display information about the SD-AVC system, and change the configuration.

### Display System Information

#### API

GET /avc-sd-service/external-api/system-info

#### Description

Displays:

- Current time: Time in UNIX format.
- System uptime: SD-AVC uptime in milliseconds.
- SD-AVC version
- Office 365 Connector status (enabled/disabled, errors, warnings)

#### Example Response

```
{
  "systemTime": "2019-06-26T12:19:02Z",
  "systemUpTimeSec": "13490106",
  "version": "4.0.0",
  "o365Connector": {
    "isEnabled": true,
    "error": [],
    "warning": []
  },
  "cloudStage": "test"
}
```

### Display Devices

#### API

GET /avc-sd-service/external-api/devices

### Description

Displays the devices in the SD-AVC network, organized by segment, in JSON format. The response includes errors and warnings, and additional information per device.

### Response

The output shows errors and warnings for:

- total network
- each segment
- each device

### Example Response

The example represents a network with one segment (datacenter-01) and one device (asr-device-100) within that segment.

```
{
  "total":{
    "connection":{
      "error":[],
      "warn":[]
    },
  },
  "segments":[
    {
      "name":"datacenter-01",
      "connection":{
        "error":[],
        "warn":[]
      },
      "devices":[
        {
          "name":"asr-device-100",
          "ip":"192.168.1.0",
          "connection":{
            "error":[],
            "warn":[]
          }
        }
      ]
    }
  ]
}
```

## Delete Devices from SD-AVC

### API

POST /avc-sd-service/external-api/remove-devices

### Description

Removes a device from the SD-AVC network. Specify the device and segment in the body.

### Body

```
{
  "devices": [
    {
      "name": "device-name-1",
      "ip": "address-1"
    },
    {
      "name": "device-name-2",
      "ip": "address-2"
    }
  ],
  "segment": "segment-name"
}
```

**Example Body**

```
{
  "devices": [
    {
      "name": "dev1",
      "ip": "10.10.10.10"
    },
    {
      "name": "dev2",
      "ip": "10.10.10.11"
    }
  ],
  "segment": "dnac"
}
```

**Example Response**

```
{"success":true,"message":"2 devices from segment dnac were deleted successfully"}
```

## Display Traffic Analytics

**API**

GET /avc-sd-service/external-api/visibility?period=*period*

-

GET /avc-sd-service/external-api/visibility/*segmentName*?period=*period*

-

GET /avc-sd-service/external-api/visibility/*segmentName*/[*deviceName* | *deviceAddress*]?period=*Period*

**Description**

Displays traffic analytics (applications and bandwidth) for the complete SD-AVC network, a specific segment, or a specific device. Optionally, specify a period for the analytics. The response includes:

- Application name and bandwidth (bytes) used by the application
- Total bandwidth (bytes) used

Table 38: Properties

| Property             | Description                                                                                                                                                                                                  |
|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>segmentName</i>   | (Optional)<br>Specifies a segment. Response includes only analytics from this segment.                                                                                                                       |
| <i>deviceName</i>    | (Optional)<br>Specifies a device by name. Response includes only analytics from this device.                                                                                                                 |
| <i>deviceAddress</i> | (Optional)<br>Specifies a device by IP address. Response includes only analytics from this device.                                                                                                           |
| <i>period</i>        | Use <code>?period=<i>period</i></code> to specify the period to include in the analytics.<br>Possible values for <i>period</i> : 120, 720, 1440, 2880 minutes (These correspond to 2, 12, 24, and 48 hours.) |

**Example:**

In this example, the period is set to 24 hours (1440 minutes).

```
GET /avc-sd-service/external-api/visibility/datacenter01/device-300?period=1440
```

```
{
  "apps": [{
    "name": "vmwarevsphere",
    "bandwidth": 226331127989634
  }, {
    "name": "telepresencecontrol",
    "bandwidth": 146787859067274
  }, {
    "name": "unknown",
    "bandwidth": 132586088501412
  }],
  "totalBandwidth": 505705075558320
}
```

## Cloud Connector

### Connect to Cloud Connector

**API:**

```
POST /avc-sd-service/external-api/cloud/authorize
```

**Description:**

Connect to the Cloud Connector, using credentials. See [Cloud Connector, on page 50](#).

**Note**

- Deprecated in SD-AVC 4.0.0. Use: POST /avc-sd-service/external-api/cloud/configure
- If continuing to use this deprecated API, do not use it together with the POST /avc-sd-service/external-api/cloud/configure API.

**Example:**

```
cisco_client_id=YOUR_CLIENT_ID&cisco_client_secret=YOUR_CLIENT_SECRET&cloud_organization_name=
ORAGANIZATION_NAME&cloud_data_affinity=usa&telemetry_enabled=1
```

**Example Responses:**

```
{"success": "AUTH_SUCCESS"}
```

or

```
{"error": "INVALID_CREDENTIALS"}
```

## Disable Cloud Connector

**API:**

POST /avc-sd-service/external-api/cloud/disable

**Description:**

Disables the Cloud Connector. See [Cloud Connector, on page 50](#).

**Note**

Use this API when you have configured Cloud Connector credentials with the "authorize" API:

POST /avc-sd-service/external-api/cloud/authorize

If you have configured Cloud Connector credentials using the "configure" API...

POST /avc-sd-service/external-api/cloud/configure

...then do not use this API to disable Cloud Connector. Instead, use the following "configureSegment" API, specifying "false" for the transmit and receive options:

POST /avc-sd-service/external-api/cloud/configureSegment?segment=name

See [Configure Cloud Connector Credentials, on page 93](#).

**Example Response:**

```
{ "success": true }
```

## Remove Cloud Connector Credentials

**API:**

POST /avc-sd-service/external-api/cloud/removeCredentials

**Description:**

Clears the credentials that have been entered for Cloud Connector.

**Example Response:**

```
{ "success": true }
```

## Display Current Cloud Configuration

**API:**

GET /avc-sd-service/external-api/cloud/getCurrentCloudConfig

**Description:**

Displays the current configuration for Cloud Connector.

**Example Response:**

```
{ "cisco_client_id": {"key": "cisco_client_id", "currentValue": "MY_CLIENT_ID", "defaultValue": ""},
  "cloud_credentials_renew_threshold": {"key": "cloud_credentials_renew_threshold", "currentValue": "90"},
  "cloud_credentials_renew_time": {"key": "cloud_credentials_renew_time", "currentValue": "1561547912794", "defaultValue": "0"},
  "cloud_data_affinity": {"key": "cloud_data_affinity", "currentValue": "usa"},
  "cloud_enabled": {"key": "cloud_enabled", "currentValue": "1", "defaultValue": "0"},
  "cloud_organization_name": {"key": "cloud_organization_name", "currentValue": "CSCO", "defaultValue": ""},
  "telemetry_enabled": {"key": "telemetry_enabled", "currentValue": "1"}}
```

## Display Cloud Data

**API:**

GET /avc-sd-service/external-api/app-rules?sourceId=PP\_Extension

**Description:**

Displays the currently collected cloud data (rules, server names, sockets, and so on).

**Example Response:**

```
[{
  "sourceId": "PP_Extension",
  "rules": [{
    "allSegments": true,
    "rules": [{
      "appName": "slack",
      "serverNames": ["slack-redir.net",
        "www.slack-redir.net"],
      "L3L4": [{
        "ipAddresses": ["34.204.245.22"],
        "ports": [443],
        "l4Protocol": "TCP"]
    }
  ]
},
  {
    "appName": "facebook",
    "L3L4": [{
      "ipAddresses": [
        "31.13.24.0/21",
        "31.13.64.0/19"]
    }
  ]
}
]
```

## Display Cloud Connector Status

### API:

GET `https://SD-AVC-network-service-address:8443/avc-sd-service/external-api/cloud/status`

GET

`https://SD-AVC-network-service-address:8443/avc-sd-service/external-api/cloud/status?segment=segmentName`

### Description:

Display status of the Cloud Connector for the specified segment.

### Body:

```
{
  "cloudConnector": {
    "isEnabled": true,
    "error": []
  }
}
```

- **isEnabled**: Cloud status for the segment. Values: true, false
- **error**: Detected errors.
  - **CONNECTIVITY**: SD-AVC cannot reach the cloud server. Connectivity problems may include DNS issues, and so on.
  - **CREDENTIALS**: Credentials for connecting to the cloud server are invalid. For example, the client secret may have expired.
  - **INTERNAL\_CLOUD**: SD-AVC cannot retrieve the cloud rules. The output provides a reason for the error. The stated reason may be helpful when troubleshooting with Cisco technical assistance.

**Table 39: Properties**

| Property                                | Description                                                                                                                                                                                                                                                                                                                             |
|-----------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>segment=<i>segmentName</i></code> | (Optional)<br>Specifies a segment. Response includes only analytics from this segment.<br>When no segment is specified, the output includes all segments.<br><b>Note</b> If you have configured SD-AVC to store telemetry data for different segments in different locations, then you must include this parameter when using this API. |

### Example Responses:

Cloud Connector enabled, no errors:

```
{
  "cloudConnector": {
    "isEnabled": true,
    "error": []
  }
}
```

Cloud Connector enabled, INTERNAL\_CLOUD error, with reason:

```

{
  "cloudConnector": {
    "isEnabled": true,
    "error": [
      {
        id: "INTERNAL_CLOUD",
        reason: ["MINOR_PP"]
      }
    ]
  }
}

```

## External Sources

### External Sources Overview

External sources provide additional application information that SD-AVC uses for classifying network traffic. They are managed by Cloud Connector. To use external sources, ensure that Cloud Connector is enabled. See [Cloud Connector, on page 50](#).

### Enable/Disable External Sources

#### API

POST /avc-sd-service/external-api/external-sources/*externalSourceName*

#### Description

Enables or disables receiving data from an external source.

**Table 40: Properties**

| Property                  | Description                                                                                                                                                                                                                                         |
|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>externalSourceName</i> | (Mandatory)<br>Name of the external source.<br><b>Note</b> In this release, the only external source to specify is "ms-office-365". To use this external source, Cloud Connector must be enabled. See <a href="#">Cloud Connector, on page 50</a> . |
| start<br>stop             | In the JSON body of the command, enables or disables the external source.                                                                                                                                                                           |

#### Examples

POST /avc-sd-service/external-api/external-sources/ms-office-365

```

{
  "action": "start"
}

```

POST /avc-sd-service/external-api/external-sources/ms-office-365

```

{

```



```
    "action": "stop"
  }
```

## Display Status of External Sources

### API

GET /avc-sd-service/external-api/external-sources

### Description

Displays external sources and their status: true = enabled, false = disabled.

### Example

GET /avc-sd-service/external-api/external-sources

### Example Response

In this example, the MS Office 365 Web Service, an external source, is enabled.

```
{
  "sources": [{
    "ms-office-365": true
  }]
}
```

## 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.

### 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.

### Terminology of Applications and Protocols

The protocols provided in a Protocol Pack and the user-defined applications configured in SD-AVC or in other ways function similarly, but the terminology varies. The table below describes the terminology.

Table 41: 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.<br>These appear on the <b>SD-AVC Dashboard &gt; Connectors</b> page.                            |
| Custom applications defined in Cisco SD-WAN     | Defined by Cisco SD-WAN, operating together with Cisco SD-AVC.<br>These are equivalent to SD-AVC user-defined applications. |

## Create User-defined Application Rules

### API

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

POST /avc-sd-service/external-api/app-rules?segment=*segmentName*

### Description

Defines one or more user-defined applications.

Table 42: Properties

| Property                    | Description                                                                  |
|-----------------------------|------------------------------------------------------------------------------|
| segment= <i>segmentName</i> | Defines the full set of user-defined applications for the specified segment. |

### 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 43: Top-level Properties**

| Property | Description                                                                                                                                                                                                                                                                                                |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sourceId | <p>(Mandatory)</p> <p>ID of the external source.</p> <p><b>Note</b> In the initial release of the REST API, only one source is supported.</p> <p><b>Note</b> If you have configured custom applications using the SD-AVC Dashboard, then you must use the following sourceID: "sdavc_ui_custom_source"</p> |
| rules    | <p>(Mandatory)</p> <p>Contains complete list of the user-defined application rules.</p> <p><b>Note</b> This property contains a sub-property also called rules.</p>                                                                                                                                        |

**Table 44: Sub-properties of rules**

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

**Table 45: Sub-properties of rules > rules**

| Property | Description                                                                                                                                                                                                                         |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| appName  | <p>(Mandatory)</p> <p>Name of user-defined application, reflecting name of the network application.</p> <p><b>Note</b> Do not use a name that conflicts with an existing application, such as one defined in the Protocol Pack.</p> |

| Property    | Description                                                                                                                                                                                   |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| serverNames | (Must include at least one of serverNames, L3L4, and attributes.)<br>List of all server names (FQDNs, SNIs, ...) for the network application.<br><b>Note</b> Server names are case-sensitive. |
| L3L4        | (Must include at least one of serverNames, L3L4, and attributes.)<br>List of all IP-based rules.<br>(See sub-properties below.)                                                               |
| attributes  | (Must include at least one of serverNames, L3L4, and attributes.)<br>Attributes to assign to the application.<br>(See sub-properties below.)                                                  |

Table 46: Sub-properties of rules &gt; rules &gt; L3L4

| Property    | Description                                                                                                                                                                          |
|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| IpAddresses | (Mandatory)<br>List of IPs. Can be both normal IP and subnet (using CIDR notation).                                                                                                  |
| ports       | Port(s) or port range.<br>If this property is defined, you must also include <b>l4protocol</b> .<br><b>Examples:</b><br>"ports": [23]<br>"ports": [23, 24]<br>"ports": [23, "25-30"] |
| l4Protocol  | Transport layer protocol.<br>If this property is defined, you must also include <b>ports</b> .<br><b>Possible values:</b> TCP, UDP, TCP-UDP                                          |
| vrf         | VRF name.                                                                                                                                                                            |

Table 47: Sub-properties of rules &gt; rules &gt; attributes

| Property        | Description                                                                                                   |
|-----------------|---------------------------------------------------------------------------------------------------------------|
| application-set | (Must include at least one of serverNames, L3L4, and attributes.)<br>Attributes to assign to the application. |

| Property           | Description                                                                                                                            |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| application-group  | (Defining a partial list of attributes is supported. If <b>attributes</b> is included, must include at least one of these properties.) |
| 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",
```

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

```

    "rules": [{
      "appName": "myDocs",
      "L3L4": [{
        "ipAddresses": ["2.2.2.2"],
        "ports": [20],
        "l4Protocol": "TCP"
      },
      {
        "ipAddresses": ["3.3.3.3"],
        "ports": [30],
        "l4Protocol": "TCP"
      },
      {
        "ipAddresses": ["4.4.4.4"],
        "ports": [40],
        "l4Protocol": "TCP"
      }
    ]
  }]
}

```

**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"
      }
    ]
  }
}

```

## Add a User-defined Application Rule

### API

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

PUT /avc-sd-service/external-api/app-rules?segment=*segmentName*

### Description

Add a user-defined application to an existing set of applications in a particular segment.

### Body

The body must include a single user-defined application. See [Create User-defined Application Rules, on page 104](#) for descriptions of the properties to use for user-defined applications. Executing the API overwrites any currently defined user-defined applications for the specified source (sourceId).

If you use the name (appName field) as an existing user-defined application, this API overwrites the existing application.



**Note** If you have configured custom applications using the SD-AVC Dashboard, then you must use the following sourceID: "sdavc\_ui\_custom\_source"

**Table 48: Properties**

| Property                    | Description                                                                                                                                                                                                                                                                                                                  |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| segment= <i>segmentName</i> | <p>In the current release, this parameter is optional. If used, it must match the segment specified in the API body.</p> <p>Example:</p> <p>PUT /avc-sd-service/external-api/app-rules?segment=segment1</p> <p><b>Note</b> The segment must be specified in the API body, even if it is added as a parameter in the URI.</p> |

```

{
  "sourceId": string,
  "rules": [{
    "allSegments": boolean,
    "segment": string,
    "rules": [{
      "appName": string,
      "serverNames": [string],
      "L3L4": [{

```

## Example

```

        "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
    }
}]]
}

```

**Response**

Response code 200 indicates success.

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

**Example**

This example shows:

- 1 network segment: datacenter01
- 1 user-defined application: segment1\_myDocs
- 1 server name
- No attributes specified

```

{
  "sourceId": "mySource",
  "rules": [{
    "segment": "segment1",
    "rules": [{
      "appName": "segment1_myDocs",
      "serverNames": [
        "www.myApp.com"
      ]
    }]
  }]
}

```

**Display User-defined Application Rules****API**

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

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

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

GET /avc-sd-service /external-api/app-rules?segment=*segmentName*&app=*applicationName*

**Description**

Displays the user-defined applications.

**Table 49: Properties**

| Property                         | Description                                                                                                                                                                                                                                                                                                                                                                                                   |
|----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>sourceId=sourceId</code>   | <p>If no <i>sourceId</i> is specified, the response lists the user-defined applications for all sources.</p> <p>If <i>sourceId</i> is specified, the response lists the user-defined applications for the specified source. The <i>sourceId</i> is user-defined by POST when defining user-defined applications.</p> <p><b>Note</b> In the initial release of the REST API, only one source is supported.</p> |
| <code>segment=segmentName</code> | Lists the full set of user-defined applications in the specified segment.                                                                                                                                                                                                                                                                                                                                     |
| <code>app=applicationName</code> | <p>(Must also specify the segment)</p> <p>Lists the specified application.</p> <p>Example:</p> <pre>GET /avc-sd-service /external-api/app-rules?segment=datacenter01&amp;app=segment1_myDocs</pre>                                                                                                                                                                                                            |

**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.

## 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*

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

DELETE /avc-sd-service /external-api/app-rules?segment=*segmentName*&app=*applicationName*

### Description

Deletes a set of user-defined applications.

If no source, segment, or application are specified, the API deletes the full set of user-defined applications.

**Table 50: Properties**

| Property                    | Description                                                                                                                                                                                                                                                                                                                                |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| sourceId= <i>sourceId</i>   | <p><b>DEPRECATED in SD-AVC 4.0.0: This option is not recommended.</b></p> <p>Deletes the full set of user-defined applications for the specified source. The <i>sourceId</i> is user-defined by POST when defining user-defined applications.</p> <p><b>Note</b> In the initial release of the REST API, only one source is supported.</p> |
| segment= <i>segmentName</i> | Deletes the full set of user-defined applications in the specified segment.                                                                                                                                                                                                                                                                |
| app= <i>applicationName</i> | <p>(Must also specify the segment)</p> <p>Deletes the specified application.</p> <p>Example:</p> <pre>DELETE /avc-sd-service /external-api/app-rules?segment=datacenter01&amp;app=segment1_myDocs</pre>                                                                                                                                    |

### Response

Response code 200 indicates success.

# Generic Applications

## Generic Applications Overview

"Generic" network traffic is not attributed to a specific network application. This portion of network traffic reduces the classification index, which is shown in the SD-AVC Dashboard.

## Display Generic Application Traffic Types

### API

GET /avc-sd-service/external-api/apps/generics

### Description

Displays the list of traffic types that contribute to generic traffic. The response is preconfigured - it does not depend on current traffic.

### Response

```
["statistical-conf-audio","rtp-audio","spdy","statistical-p2p","rtp-video","http","statistical-conf-video","quic","statistical-download","ssl","unknown","rtp"]
```

## REST API Notes and Limitations

See [SD-AVC Notes and Limitations](#), on page 63.





## APPENDIX F

# Source Interface Configuration

---

- [Source Interface Configuration Overview, on page 117](#)
- [Background, on page 117](#)
- [Scenarios that Benefit from Source Interface Configuration, on page 118](#)

## 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.

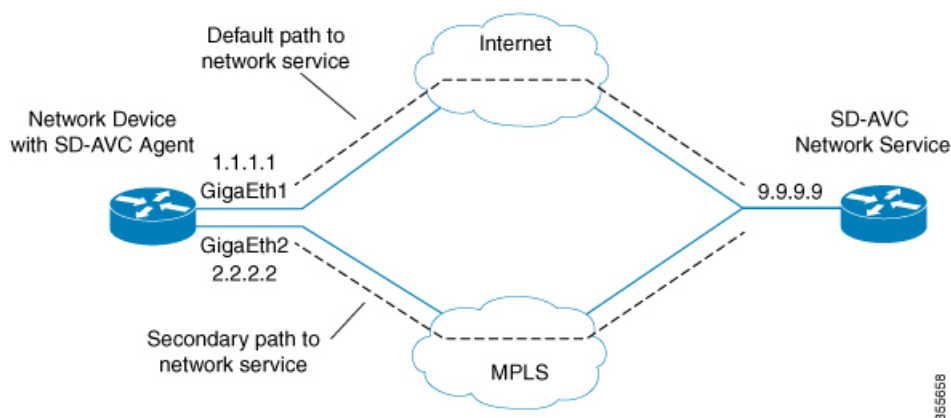
Specifying the interface for SD-AVC traffic can address several issues described in [Scenarios that Benefit from Source Interface Configuration, on page 118](#).

## 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.

|                                                |
|------------------------------------------------|
| <p>ASR1k-DC100-50</p> <p>Segment<br/>DC100</p> |
| <p>IP<br/>1.1.1.1</p>                          |

## 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

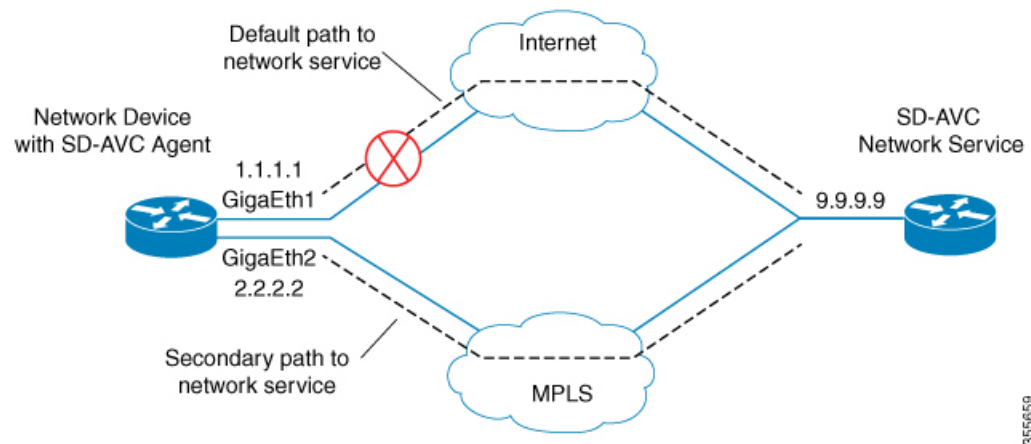
Packet source when using secondary path: 2.2.2.2

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



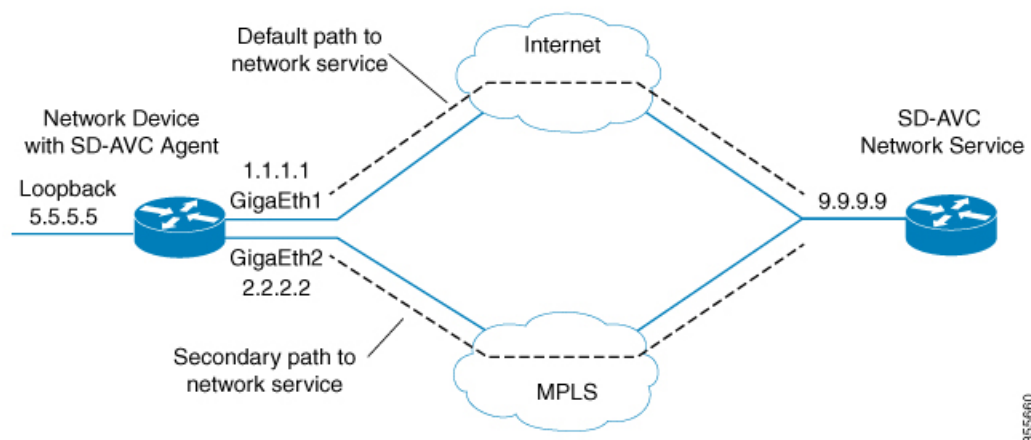


Earlier, the network device appeared in the SD-AVC Dashboard as 1.1.1.1. Now it appears as 2.2.2.2, reflecting the secondary path to the SD-AVC network service. The device hostname remains the same, but the IP has changed, as shown below. This may not be desired.

|                  |
|------------------|
| ● ASR1k-DC100-50 |
| Segment<br>DC100 |
| IP<br>2.2.2.2    |

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 121.



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

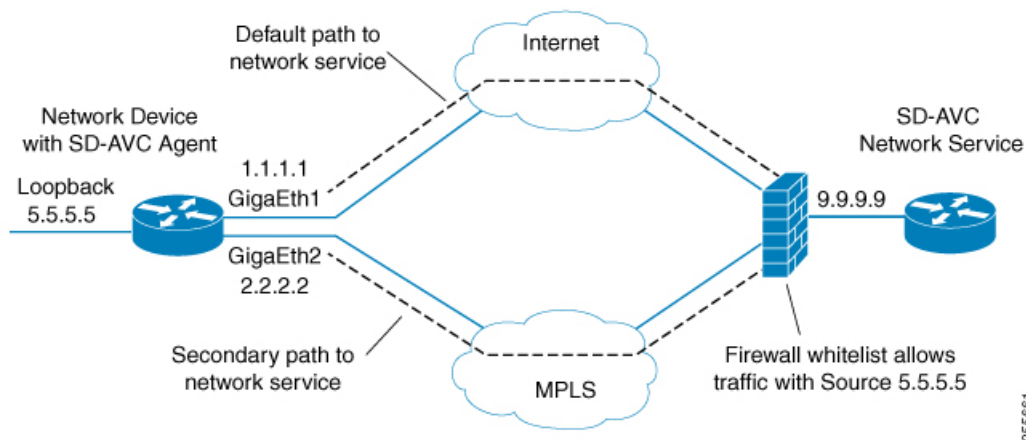
|                  |         |
|------------------|---------|
| ● ASR1k-DC100-50 |         |
| Segment          | DC100   |
| IP               | 5.5.5.5 |

## 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 121.



## Scenario: Internal FTP/HTTP Server

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

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

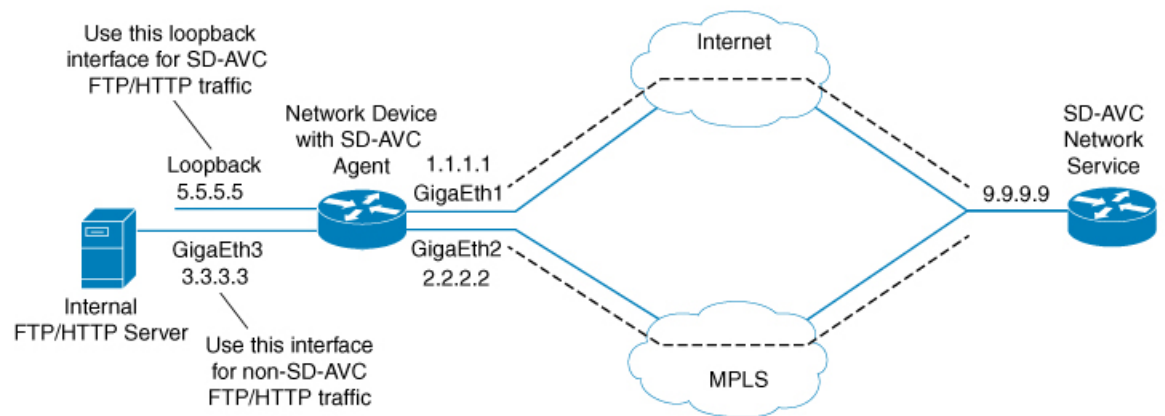
In the example below:

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

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

- SD-AVC FTP/HTTP traffic uses the loopback interface as source (see [Specifying a Loopback as Source Interface, on page 121](#)):

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



35-66851

## 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
```

- 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.

|                  |         |
|------------------|---------|
| ● ASR1k-DC100-50 |         |
| Segment          | DC100   |
| IP               | 5.5.5.5 |



## APPENDIX **G**

# NBAR AWS Cloud Telemetry Matrix

By default, the Cisco Cloud Connector telemetry collection is **on**.

| Data Category          | Specific Data Collected | Purpose for Collection/Benefits from Collection                                 | Mandatory Collection – Y or N? (no ability to opt in/out) |
|------------------------|-------------------------|---------------------------------------------------------------------------------|-----------------------------------------------------------|
| General Administrative | SDAVC id                | Detect Network Agent sending the telemetry                                      | Y (If you enable cloud this will always be sent)          |
| General Administrative | SDAVC IP                | Detect Network Agent sending the telemetry                                      | Y (If you enable cloud this will always be sent)          |
| General Administrative | Segment                 | Detect logical segment the data belong to                                       | Y (If you enable cloud this will always be sent)          |
| General Administrative | Telemetry Status        | Detect telemetry status (enabled/disabled)                                      | Y (If you enable cloud this will always be sent)          |
| CACHE RULES            | Application name        | Report application info to enhance the application recognition service          | N (You can opt out)                                       |
| CACHE RULES            | Ip address              | Report internet IP used to enhance the application recognition service          | N (You can opt out)                                       |
| CACHE RULES            | port                    | Report internet port used to enhance the application recognition service        | N (You can opt out)                                       |
| CACHE RULES            | L4 protocol             | Report internet L4 protocol used to enhance the application recognition service | N (You can opt out)                                       |
| CACHE RULES            | vrf                     | Report internet vrf to enhance the application recognition service              | N (You can opt out)                                       |
| CACHE RULES            | Socket rating           | Enhance application recognition using telemetry                                 | N (You can opt out)                                       |

| <b>Data Category</b>               | <b>Specific Data Collected</b>      | <b>Purpose for Collection/Benefits from Collection</b>         | <b>Mandatory Collection – Y or N? (no ability to opt in/out)</b> |
|------------------------------------|-------------------------------------|----------------------------------------------------------------|------------------------------------------------------------------|
| ANALYTICS<br>UV INFO               | Ip address                          | Detect unclassified traffic to resolve                         | N (You can opt out)                                              |
| ANALYTICS<br>UV INFO               | port                                | Detect unclassified traffic to resolve                         | N (You can opt out)                                              |
| ANALYTICS<br>UV INFO               | L4 protocol                         | Detect unclassified traffic to resolve                         | N (You can opt out)                                              |
| ANALYTICS<br>UV INFO               | bandwidth                           | Detect unclassified traffic to resolve                         | N (You can opt out)                                              |
| ANALYTICS<br>UV INFO               | classification                      | Detect unclassified traffic to resolve                         | N (You can opt out)                                              |
| ANALYTICS<br>UV INFO               | domain                              | Detect unclassified traffic to resolve                         | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Application name                    | Top application bandwidth usage – detect trends and apps usage | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | bandwidth                           | Top application bandwidth usage – detect trends and apps usage | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of active flows                 | Understand the scale of network                                | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of fif flows                    | Understand the scale of network                                | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of SDAVC early classified flows | Get feedback on SDAVC impact on network                        | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of TCP flows                    | Get insight regarding TCP percent in network                   | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of TCP asymmetric flows         | Get insight on asymmetric traffic                              | N (You can opt out)                                              |
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of DNS flows                    | Get insight regarding DNS percent in network                   | N (You can opt out)                                              |

| <b>Data Category</b>               | <b>Specific Data Collected</b> | <b>Purpose for Collection/Benefits from Collection</b> | <b>Mandatory Collection – Y or N? (no ability to opt in/out)</b> |
|------------------------------------|--------------------------------|--------------------------------------------------------|------------------------------------------------------------------|
| ANALYTICS<br>PROTOCOL<br>DISCOVERY | Num of DNS asymmetric flows    | Get insight on asymmetric traffic                      | N (You can opt out)                                              |
| DEVICE INFO                        | Num of devices                 | Understand the scale of network                        | N (You can opt out)                                              |
| DEVICE INFO                        | Num of active devices          | Find if there is dysconnectivity of devices            | N (You can opt out)                                              |
| DEVICE INFO                        | Installed PP versions          | Be able to suggest new version base on PP status       | N (You can opt out)                                              |
| DEVICE INFO                        | Deployed engine versions       | Be able to support features per device engine version  | N (You can opt out)                                              |
| PROTOCOLS<br>INFO                  | Application name               | See static application configuration on network        | N (You can opt out)                                              |
| PROTOCOLS<br>INFO                  | Application attributes         | See static application configuration on network        | N (You can opt out)                                              |
| PROTOCOLS<br>INFO                  | Application id                 | See static application configuration on network        | N (You can opt out)                                              |
| PROTOCOLS<br>INFO                  | Is custom application          | See static application configuration on network        | N (You can opt out)                                              |
| PROTOCOLS<br>INFO                  | Is generic application         | See static application configuration on network        | N (You can opt out)                                              |







## APPENDIX **H**

# Creating SSL Certificates to Use with SD-AVC

- [Summary, on page 127](#)
- [Using a Certificate Signed by a Certification Authority, on page 128](#)
- [Using a Self-signed SSL Certificate Created with Keytool, on page 128](#)
- [Using a Self-signed SSL Certificate Created with OpenSSL, on page 130](#)

## Summary

### Create certificate to be signed by certification authority

|   | Task                                                           | Where to find...                                                                           |
|---|----------------------------------------------------------------|--------------------------------------------------------------------------------------------|
| 1 | Create certificate keys.                                       | See <a href="#">Using a Certificate Signed by a Certification Authority, on page 128</a> . |
| 2 | Generate a certificate signing request (CSR).                  |                                                                                            |
| 3 | Send the CSR file to be signed by the certification authority. |                                                                                            |
| 4 | Install the signed certificate in the SD-AVC Dashboard.        | See "Serviceability Page" in <a href="#">Using SD-AVC, on page 43</a> .                    |

### Create self-signed certificate

|   | Task                                                    | Where to find...                                                                                                                                                                           |
|---|---------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | Create self-signed certificate keys.                    | See <a href="#">Using a Self-signed SSL Certificate Created with Keytool, on page 128</a> .<br>See <a href="#">Using a Self-signed SSL Certificate Created with OpenSSL, on page 130</a> . |
| 2 | Install the signed certificate in the SD-AVC Dashboard. | See "Serviceability Page" in <a href="#">Using SD-AVC, on page 43</a> .                                                                                                                    |

## Using a Certificate Signed by a Certification Authority

You can use the **keytool** or **OpenSSL** command line utilities to create a certificate to be signed by a certification authority, and used with Cisco SD-AVC.

### Using Keytool

1. Create certificate keys.

#### Example:

```
keytool -genkey -alias sdavc_alias -keyalg RSA -sigalg SHA1withRSA -keysize 2048 -keystore mykeystore.keystore
```

2. Generate a certificate signing request (CSR).

#### Example:

```
keytool -certreq -alias sdavc_alias -keyalg RSA -sigalg SHA1withRSA -file mycsrfile.csr -keystore mykeystore.keystore
```

The command produces a CSR file called **mycsrfile.csr**.

3. Send the CSR file to be signed by the certification authority.
4. Install the signed certificate in the SD-AVC Dashboard. See "Serviceability Page" in [Using SD-AVC, on page 43](#).

### Using OpenSSL

1. Create certificate keys.

#### Example:

```
openssl genrsa -des3 -out server.key 2048
```

2. Generate a certificate signing request (CSR).

#### Example:

```
openssl req -new -key server.key -sha256 -out server.csr
```

3. Send the CSR file to be signed by the certification authority.
4. Install the signed certificate in the SD-AVC Dashboard. See "Serviceability Page" in [Using SD-AVC, on page 43](#).

## Using a Self-signed SSL Certificate Created with Keytool

You can use the **keytool** command line utility to create a self-signed certificate, and use the certificate with Cisco SD-AVC.

This utility creates certificates in [Java KeyStore](#) (JKS) format.

The example shows how to create a self-signed certificate and how to display the details of the certificate. Details such as alias are required when configuring SD-AVC to use the certificate.



**Note** Keytool is not a Cisco product. The brief guidelines provided here are for convenience. Complete information is available online.

### Creating and Installing the SSL Certificate

This example shows the command, followed by interactive input. It creates a certificate with:

- **Alias:** abc\_ssl
- **Passphrase:** 123456

#### 1. Create certificate keys.

```
keytool -genkey -keyalg RSA -alias abc_ssl -keystore my_keystore.jks -storepass 123456
-validity 360 -keysize 2048
What is your first and last name?
[Unknown]: hostname.cisco.com
What is the name of your organizational unit?
[Unknown]: dev
What is the name of your organization?
[Unknown]: cisco
What is the name of your City or Locality?
[Unknown]: san-jose
What is the name of your State or Province?
[Unknown]: ca
What is the two-letter country code for this unit?
[Unknown]: us
Is CN=hostname.cisco.com, OU=dev, O=cisco, L=san-jose, ST=ca, C=us correct? (type "yes"
or "no")
[no]: yes

Enter key password for <abc_ssl>:
(RETURN if same as keystore password):
```

#### 2. Install the signed certificate in the SD-AVC Dashboard. See "Serviceability Page" in [Using SD-AVC, on page 43](#).

### Viewing the Certificate Details

View the certificate details. Note that the output includes the alias name (which may be a default value, or a specified custom alias name, as in this example), and keystore type (jks in this example).

```
1. keytool -list -v -keystore my_keystore.jks
Enter keystore password:

Keystore type: jks
Keystore provider: IBMJCE

Your keystore contains 1 entry

Alias name: abc_ssl
Creation date: Apr 30, 2019
Entry type: keyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=hostname.cisco.com, OU=dev, O=cisco, L=san-jose, ST=ca, C=us
Issuer: CN=hostname.cisco.com, OU=dev, O=cisco, L=san-jose, ST=ca, C=us
Serial number: 5cc899de
```

```

Valid from: 4/30/19 9:54 PM until: 4/24/20 9:54 PM
Certificate fingerprints:
    MD5:  38:B7:B4:28:43:48:11:88:C5:B1:E0:47:79:26:CD:A7
    SHA1: 7C:60:01:35:26:67:40:64:65:D0:E2:B5:2B:30:1F:7D:5E:16:44:C3
    SHA256:
42:82:63:BF:CF:87:95:B7:5A:FA:38:12:45:F9:88:D5:FD:00:68:A8:96:28:63:32:0C:D4:E5:A0:86:68:25:53

    Signature algorithm name: SHA256withRSA
    Version: 3

```

## Using a Self-signed SSL Certificate Created with OpenSSL

You can use the **OpenSSL** command line utility to create a self-signed certificate, and use the certificate with Cisco SD-AVC.

This utility creates certificates in numerous formats.

The example shows how to create a certificate and how to display the details of the certificate. Details such as alias/friendlyName, are required when configuring SD-AVC to use the certificate.



**Note** OpenSSL is not a Cisco product. The brief guidelines provided here are for convenience. Complete information is available online.

### Creating and Installing the SSL Certificate

This example shows the command, followed by interactive input. It creates and exports a certificate with:

- **Alias/friendlyName:** abc\_ssl
- **Output filename:** my\_cakey.pem

#### 1. Create certificate keys.

```

openssl req -newkey rsa:2048 -x509 -keyout my_cakey.pem -out my_cacert.pem -days 3650
Generating a 2048 bit RSA private key
.....+++
...+++
writing new private key to 'my_cakey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:us
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:city
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:hostname.cisco.com
Email Address []:anyEmail@cisco.com

```

```
openssl pkcs12 -export -in my_cacert.pem -inkey my_cakey.pem -out my_identity.p12 -name
"abc_ssl"
Enter pass phrase for my_cakey.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

2. Convert the format.

```
openssl pkcs12 -export -in my_cacert.pem -inkey my_cakey.pem -out my_identity.p12 -name
"abc_ssl"
Enter pass phrase for my_cakey.pem:
Enter Export Password:
Verifying - Enter Export Password:
```

3. Install the signed certificate in the SD-AVC Dashboard. See "Serviceability Page" in [Using SD-AVC, on page 43](#).

### Viewing the Certificate Details

View the certificate details. Note that this command provides the alias/friendlyName, which may be a default value, or a specified custom alias name, as in this example.

```
1. openssl pkcs12 -info -in my_identity.p12
Enter Import Password:
MAC Iteration 2048
MAC verified OK
PKCS7 Encrypted data: pbeWithSHA1And40BitRC2-CBC, Iteration 2048
Certificate bag
Bag Attributes
    localKeyID: 2E 12 BE F7 56 D3 1D C0 39 9A 52 29 AD 18 3A 95 05 AA A5 86
    friendlyName: abc_ssl
```





## APPENDIX

# Additional References

| Topic                               | Document                                                       |
|-------------------------------------|----------------------------------------------------------------|
| SD-AVC release notes, release 4.0.0 | <a href="#">Cisco SD-AVC Release Notes, Release 4.0.0</a>      |
| Cisco AVC product page              | <a href="#">Cisco Application Visibility and Control (AVC)</a> |

