Virtual Services Container

Virtual Services Container

Prerequisites for a Virtual Services Container

- You must have a Cisco device installed with an operating system release that supports virtual services and has the needed system infrastructure required for specific applications like Cisco Plug-in for OpenFlow.

> **Note** A compatibility matrix is delivered with each Cisco application. Refer to this matrix for information about which operating system release supports the features and infrastructure necessary for a particular application such as Cisco Plug-in for OpenFlow.

- You must download an open virtual application (OVA) package that is compatible with the device operating system, and downloaded from an FTP server connected to the device.

- You must have enough memory for installation and deployment of application. Refer to the application configuration guide for specific recommendations.

Information About Virtual Services Container

Virtual Services Containers and Applications

A virtual services container is a virtualized environment on a device. It is also referred to as a virtual machine (VM), virtual service, or container.

You can install an application within a virtual services container. The application runs in the virtual services container of the operating system of a device. The application is delivered as an open virtual application

(OVA), which is a tar file with a .ova extension. The OVA package is installed and enabled on a device through the device CLI.

Cisco Plug-in for OpenFlow is an example of an application that can be deployed within a virtual services container.

Some of the files that can be found in an OVA file are the following:

- Virtual machine definition file, in libvirt XML format, with Cisco extensions.

- Manifest file, listing the contents of a distribution. It contains the hash information for each file in the OVA package.

- Certificate file containing the signature of a manifest file. This file is used in validating the integrity of an OVA package.

- Version file, used to check compatibility with the virtualization infrastructure.

How to Configure a Virtual Services Container

This section includes the following required and optional tasks:

Installing and Activating an Application in a Virtual Services Container

This task copies an open virtual application (OVA) package from an FTP file location, installs the application in a virtual services container, provisions the application, and activates it.

SUMMARY STEPS

1. **enable**
2. **copy** *from***://***source-directory-url destination-directory-url*
3. **virtual-service install name** *virtual-services-name* **package** *file*
4. **configure terminal**
5. **virtual-service** *virtual-services-name*
6. **activate**
7. **end**
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**enable** **Example:** Device> enable	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**copy** *from***://***source-directory-url destination-directory-url* **Example:** Device# copy tftp://myserver.com/downloads/ofa-1.0.0-n3000-SPA-k9.ova bootflash:/ofa-1.0.0-n3000-SPA-k9.ova	Downloads the new OVA package to the device for upgrade. Possible values are: • sftp: • tftp: • ftp: • http: • bootflash:
Step 3	**virtual-service install name** *virtual-services-name* **package** *file* **Example:** Device# virtual-service install name openflow_agent package bootflash:/ofa-1.0.0-n3000-SPA-k9.ova	Installs an OVA package from the specified location onto a device. Ensure that the ova file is located in the root directory of the storage device • The *virtual-services-name* defined here should be used in all occurrences of this argument in this document.
Step 4	**configure terminal** **Example:** Device# configure terminal	Enters global configuration mode.
Step 5	**virtual-service** *virtual-services-name* **Example:** Device(config)# virtual-service openflow_agent	Configures a virtual services container and enters virtual services configuration mode. • Use the *virtual-services-name* defined during installation of the application. • Ensure that installation is complete before proceeding to the next step using the **show virtual-service list** command.
Step 6	**activate** **Example:** Device(config-virt-serv)# activate	Activates the installed virtual services container.
Step 7	**end** **Example:** Device(config-virt-serv)# end	Exits virtual services configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

What to Do Next

You can now begin using your application.

Deactivating and Uninstalling an Application from a Virtual Services Container

(Optional) Perform this task to uninstall and deactivate an application from within a virtual services container.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **virtual-service** *virtual-services-name*
4. **no activate**
5. **no virtual-service** *virtual-services-name*
6. **end**
7. **virtual-service uninstall name** *virtual-services-name*
8. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**virtual-service** *virtual-services-name* **Example:** `Device(config)# virtual-service openflow_agent`	Enters virtual services configuration mode to configure a specified application. • Use the *virtual-services-name* defined during installation of the application.

	Command or Action	Purpose
Step 4	**no activate** **Example:** `Device(config-virt-serv)# no activate`	Disables the application.

Step 5	**no virtual-service** *virtual-services-name*
	
	Example:
	`Device(config)# no virtual-service`
	`openflow_agent`
		Unprovisions the application.
		
		• Use the *virtual-services-name* defined during installation of the application.
		
		• This command is optional for all devices running Cisco IOS-XE.

Step 6	**end** **Example:** `Device(config-virt-serv)# end`	Exits virtual services configuration mode and enters privileged EXEC mode.

Step 7	**virtual-service uninstall name** *virtual-services-name*
	
	Example:
	`Device# virtual-service uninstall name`
	`openflow_agent`
		Uninstalls the application.
		
		• Use the *virtual-services-name* defined during installation of the application.
		
		• Run this command only after receiving a successful deactivation response from the device.

Step 8	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Upgrading an Application in a Virtual Services Container

(Optional) Perform this task to upgrade a virtual services container application.

Note An application upgrade may require an upgrade of the device operating system. Check the compatibility matrix of the respective application software release before upgrading it.

SUMMARY STEPS

1. **enable**
2. **copy** *from***:**//*source-directory-url destination-directory-url*
3. **configure terminal**
4. **virtual-service** *virtual-services-name*
5. **no activate**
6. **end**
7. **virtual-service upgrade name** *virtual-services-name* **package** *file*
8. **configure terminal**
9. **virtual-service** *virtual-services-name*
10. **activate**
11. **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

Step 2	**copy** *from***:**//*source-directory-url destination-directory-url*
	
	Example:
	`Device# copy`
	`tftp://myserver.com/downloads/ofa-1.0.0-n3000-SPA-k9.ova`
	`bootflash:/ofa-1.0.0-n3000-SPA-k9.ova`
		Downloads the new OVA package to the device for upgrade. Possible values are:
		
		• sftp:
		
		• tftp:
		
		• ftp:
		
		• http:
		
		• bootflash:

Step 3	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 4	**virtual-service** *virtual-services-name* **Example:** `Device(config)# virtual-service openflow_agent`	Enters virtual services configuration mode for configuring a specified application. • Use the *virtual-services-name* defined during installation of the application.

	Command or Action	Purpose
Step 5	**no activate** **Example:** `Device(config-virt-serv)# no activate`	Disables the application.
Step 6	**end** **Example:** `Device(config-virt-serv)# end`	Exits virtual services configuration mode and enters privileged EXEC mode.
Step 7	**virtual-service upgrade name** *virtual-services-name* **package** *file* **Example:** `Device# virtual-service upgrade name openflow_agent package bootflash:/ofa-1.0.0-n3000-SPA-k9.ova`	Upgrades the application using the specified OVA file. • Use the *virtual-services-name* defined during installation of the application. • Run this command only after receiving a successful deactivation message from the device.
Step 8	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 9	**virtual-service** *virtual-services-name* **Example:** `Device(config)# virtual-service openflow_agent`	Enters virtual services configuration mode for configuration of the specified application. • Use the *virtual-services-name* defined during installation of the application.
Step 10	**activate** **Example:** `Device(config-virt-serv)# activate`	Activates the application.
Step 11	**copy running-config startup-config** **Example:** `Device# copy running-config startup-config`	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

What to Do Next

You can now begin using your application.

Collecting General Troubleshooting Information

Information collected using the commands listed below can be sent to Cisco Technical Support for troubleshooting purposes.

SUMMARY STEPS

1. **show system sysmgr service name vman**
2. **virtual-service move name** *virtual-services-name* **[core | log] to** *destination-url*
3. **show mgmt-infra trace settings vman_trace**
4. **set trace control vman_trace buffer-size** *buffer-size*
5. **set trace control vman_trace clear [location active]**
6. **set trace vman_trace level {debug | default | err | info | warning} [location active]**

DETAILED STEPS

	Command or Action	Purpose

Step 1	**show system sysmgr service name vman**
	
	Example:
	`Device# ` **`show system sysmgr service name vman`**
	
	`Service "vman" ("vman", 209):`
	`        UUID = 0x49B, PID = 3283, SAP = 808`
	`        State: SRV_STATE_HANDSHAKED (entered at time Tue`
	`Mar  5 01:11:41 2013).`
	`        Restart count: 1`
	`        Time of last restart: Tue Mar  5 01:11:41 2013.`
	`        The service never crashed since the last reboot.`
	`        Tag = N/A`
	`        Plugin ID: 0`
		This command shows the health of the virtualization manager (VMAN) process.

Step 2	**virtual-service move name** *virtual-services-name* **[core	log] to** *destination-url* **Example:** `Device# ` **`virtual-service move name openflow_agent core to`** **`bootflash:/`**	Moves application log or core files to a specified destination location. This command can be used when the application running in the container has an issue (but the container is running as expected).

Step 3	**show mgmt-infra trace settings vman_trace**
	
	Example:
	`Device# ` **`show mgmt-infra trace settings vman_trace`**
	
	`One shot Trace Settings:`
	
	`Buffer Name: vman_trace`
	`Default Size: 262144`
	`Current Size: 262144`
	`Traces Dropped due to internal error: Yes`
	`Total Entries Written: 2513`
	`One shot mode: No`
	`One shot and full: No`
	`Disabled: False`
		This command displays trace settings of a trace buffer.

	Command or Action	**Purpose**
Step 4	**set trace control vman_trace buffer-size** *buffer-size*	This command sets the trace buffer size.
Step 5	**set trace control vman_trace clear [location active]**	This command clears the trace buffer.
Step 6	**set trace vman_trace level {debug	default	err	info	warning} [location active]**	This command sets the trace level.

Verifying Virtual Services Container Applications

SUMMARY STEPS

1. **show virtual-service [global]**
2. **show virtual-service detail** [**name** *virtual-services-name*]
3. **show virtual-service list**
4. **show virtual-service storage pool list**
5. **show virtual-service storage volume list**
6. **show virtual-service version name** *virtual-services-name* **installed**
7. **show virtual-service tech-support**
8. **show virtual-service redundancy state**
9. **show virtual-service utilization name** *virtual-services-name*
10. **show virtual-service utilization statistics CPU**

DETAILED STEPS

Step 1 **show virtual-service [global]**
This command displays available memory, disk space, and CPU allocated for applications.

Example:

Step 2 **show virtual-service detail** [**name** *virtual-services-name*]
This command displays a list of resources committed to a specified application, including attached devices.

Example:

Step 3 **show virtual-service list**
This command displays an overview of resources utilized by the applications.

Example:

Step 4 **show virtual-service storage pool list**
This command displays an overview of storage locations (pools) used for virtual service containers.

Example:
```
Device# show virtual-service storage pool list

Virtual-Service storage pool list

Name                  Pool Type   Path
----------------------------------------------------------------------
virt_strg_pool_bf_vdc_1  directory   /bootflash/virt_strg_pool_bf_vdc_1
```

Step 5 **show virtual-service storage volume list**
This command displays an overview of storage volume information for virtual service containers.

Example:
```
Device# show virtual-service storage volume list

Virtual-Service storage volume list

Name                  Capacity   In Use   Virtual-Service
----------------------------------------------------------------------------------
_rootfs.ofa           90 MB      Yes      ofa
```

Step 6 **show virtual-service version name** *virtual-services-name* **installed**
This command displays the version of an installed application.

Example:
```
Device# show virtual-service version name openflow_agent installed

Virtual service openflow_agent installed version:
 Name : CiscoPluginForOpenFlow
 Version : 1.1.0_fc1
```

Step 7 **show virtual-service tech-support**
Displays all relevant container-based information.

Step 8 **show virtual-service redundancy state**

Example:
```
Device# show virtual-service redundancy state

Device# show virtual-service redundancy state
Virtual Service Redundancy State:

Switch No.      Role      Configure sync status    OVA sync status
-------------------------------------------------------------------------
3               Active    N/A                      N/A
```
Displays state of virtual-services.

Step 9 **show virtual-service utilization name** *virtual-services-name*

Example:
```
cat4k-openflow1#sh virtual-service utilization name openflow_agent
Virtual-Service Utilization:

CPU Utilization:
  CPU Time:  0 % (30 second average)
  CPU State: R : Running
```

```
        Memory Utilization:
          Memory Allocation: 262144 Kb
          Memory Used:        19148 Kb


        Storage Utilization:
          Name: _rootfs, Alias: _rootfs
            RD Bytes:    0                 WR Bytes:    0
            RD Requests: 0                 WR Requests: 0
            Errors:      0
            Capacity(1K blocks):  89243    Used(1K blocks): 66976
            Available(1K blocks): 17659    Usage: 80 %
          Name: cisco, Alias: cisco
            RD Bytes:    0                 WR Bytes:    0
            RD Requests: 0                 WR Requests: 0
            Errors:      0
            Capacity(1K blocks):  861512   Used(1K blocks): 218216
            Available(1K blocks): 643296   Usage: 26 %
          Name: /mnt/ofa, Alias: /mnt/ofa
            RD Bytes:    0                 WR Bytes:    0
            RD Requests: 0                 WR Requests: 0
            Errors:      0
            Capacity(1K blocks):  4955     Used(1K blocks): 35
            Available(1K blocks): 4664     Usage: 1  %
          Name: /cisco/core, Alias: /cisco/core
            RD Bytes:    0                 WR Bytes:    0
            RD Requests: 0                 WR Requests: 0
            Errors:      0
            Capacity(1K blocks):  138119   Used(1K blocks): 91053
            Available(1K blocks): 39935    Usage: 70 %
          Name: /tmp1, Alias: /tmp1
            RD Bytes:    0                 WR Bytes:    0
            RD Requests: 0                 WR Requests: 0
            Errors:      0
            Capacity(1K blocks):  861512   Used(1K blocks): 218216
            Available(1K blocks): 643296   Usage: 26 %
          Name: /cisco123, Alias: /cisco123
            RD Bytes:    0                 WR Bytes:    0
            RD Requests: 0                 WR Requests: 0
            Errors:      0
            Capacity(1K blocks):  856308   Used(1K blocks): 19200
            Available(1K blocks): 837108   Usage: 3  %
```
Displays virtual-services utilization information.

Step 10   **show virtual-service utilization statistics CPU**
Displays virtual service CPU utilization statistics.

Troubleshooting Virtual Services Containers

Troubleshooting Installation of Applications in a Virtual Services Container

Problem Installation of an application in a virtual services container is not successful.

Possible Cause Installation of the application may still be ongoing.

Solution Check the status of the installation using the **show virtual-service list** command. The following is sample output when the application has an Installed status.

```
Device# show virtual-service list
```

```
Virtual Service List:
Name                    Status              Package Name
----------------------------------------------------------------
multiova                Activated           multiova-working.ova
WAAS                    Installed           ISR4451X-WAAS-5.2.0-b...
```

Possible Cause An application with the same name has already been installed.

Solution Ensure that an application of the same name has not been installed using the **show virtual-service list** command. You can verify this by referencing the Name field.

Possible Cause The target media has not been installed. Target media for various devices are given below:

- **Possible Cause** Cisco Nexus 3000 Series device—bootflash

- **Possible Cause** Cisco 4500 Series device—bootflash

- **Possible Cause** Cisco 3850 and 3650 device—flash

Solution Ensure that the target media is installed using the **show version** command.

```
Device# show version

Cisco Nexus Operating System (NX-OS) Software
TAC support: http://www.cisco.com/tac
Documents: http://www.cisco.com/en/US/products/ps9372/tsd_products_support_serie
s_home.html
Copyright (c) 2002-2013, Cisco Systems, Inc. All rights reserved.
The copyrights to certain works contained herein are owned by
other third parties and are used and distributed under license.
Some parts of this software are covered under the GNU Public
License. A copy of the license is available at
http://www.gnu.org/licenses/gpl.html.

Software
  BIOS:      version 1.2.0
  loader:    version N/A
  kickstart: version 6.0(2)U1(1)
  system:    version 6.0(2)U1(1)
  Power Sequencer Firmware:
             Module 1: version v4.4
  BIOS compile time:       08/25/2011
  kickstart image file is: bootflash:///n3000-uk9-kickstart.6.0.2.U1.0.78.bin
  kickstart compile time:  5/7/2013 12:00:00 [05/07/2013 19:45:30]
  system image file is:    bootflash:///n3000-uk9.6.0.2.U1.0.78.bin
  system compile time:     5/7/2013 12:00:00 [05/07/2013 20:54:48]

Hardware
  cisco Nexus 3048 Chassis ("48x1GE + 4x10G Supervisor")
  Intel(R) Celeron(R) CPU       P450 with 3980876 kB of memory.
  Processor Board ID FOC16434LJ2

  Device name: n3k-202-194-2
  bootflash:    2007040 kB

Kernel uptime is 0 day(s), 19 hour(s), 5 minute(s), 45 second(s)

Last reset at 132996 usecs after  Wed May  8 18:27:54 2013

  Reason: Reset Requested by CLI command reload
  System version: 6.0(2)U1(1)
  Service:

plugin
  Core Plugin, Ethernet Plugin
```

Possible Cause There is insufficient space to install an application.

Solution Ensure that sufficient space exists using the **dir** command.

```
Device# dir bootflash:

         407    May 08 21:35:52 2013   admin.rc.cli
        1332    Feb 28 16:51:27 2013   bxmnt-n3k
        3348    May 08 16:21:57 2013   config-sumana-08-may-13
     2826744    Feb 13 15:00:49 2013   dd2
     2826744    Jan 30 15:26:15 2013   dplug
    10273827    Apr 10 03:09:52 2013   gdb
      123496    Apr 10 03:12:46 2013   libexpat.so.0
        2016    Feb 28 15:18:33 2013   linux-mount-setup-n3k
     2826744    Jan 29 19:51:24 2013   lltor-dplug_md.bin
       49152    Nov 29 00:52:45 2012   lost+found/
        1903    Jan 11 16:08:49 2013   mts.log
    31884800    Apr 01 18:40:52 2013   n3000-uk9-kickstart.6.0.2.U1.0.36.bin
    31864320    Apr 08 15:53:00 2013   n3000-uk9-kickstart.6.0.2.U1.0.44.bin
    32757760    May 08 16:37:08 2013   n3000-uk9-kickstart.6.0.2.U1.0.78.bin
   232540777    Apr 04 18:24:30 2013   n3000-uk9.6.0.2.U1.0.40.bin
   232535711    Apr 08 15:51:49 2013   n3000-uk9.6.0.2.U1.0.44.bin
   232632475    May 08 16:36:35 2013   n3000-uk9.6.0.2.U1.0.78.bin
    53555200    May 08 15:37:44 2013   n3k_ofa.ova
    55101440    Feb 28 20:27:39 2013   n3k_ofa.ova-gdb
    52613120    Apr 04 18:26:55 2013   n3k_ofa.ova.port-channel2
    58675200    Feb 01 14:47:44 2013   n3k_ofa.ova1
    58675200    Feb 01 20:40:47 2013   n3k_ofa.ova31-6
     2201210    Feb 27 20:30:02 2013   of_agent
    56729600    May 08 16:41:33 2013   ofa-0.1.0_46-n3000-SSA-k9.ova
        4096    Jan 29 17:52:15 2013   onep/
        8552    Apr 04 18:10:50 2013   saveApril3
        7536    Feb 28 19:08:06 2013   saveConfigFeb28
        4096    Jan 29 00:48:00 2010   vdc_2/
        4096    Jan 29 00:48:00 2010   vdc_3/
        4096    Jan 29 00:48:00 2010   vdc_4/
        4096    May 08 18:56:52 2013   virt_strg_pool_bf_vdc_1/
        4096    Apr 09 20:24:06 2013   virtual-instance/
           0    May 08 16:51:44 2013   virtual-instance-upgrade.conf
          63    May 08 16:51:44 2013   virtual-instance.conf

Usage for bootflash://sup-local
 1558257664 bytes used
   90365952 bytes free
 1648623616 bytes total
```

Possible Cause Disk quota for container is insufficient.

Solution Ensure that disk quota available for virtual services is sufficient using the **show virtual-services global** command.

```
Device# show virtual-service global

Virtual Service Global State and Virtualization Limits:

Infrastructure version : 1.5
Total virtual services installed : 1
Total virtual services activated : 1

Machine types supported   : LXC
Machine types disabled    : KVM

Maximum VCPUs per virtual service : 1
Resource virtualization limits:
Name                            Quota      Committed      Available
--------------------------------------------------------------
system CPU (%)                      6              1              5
memory (MB)                       256            256              0
bootflash (MB)                    256            164             92
```

Possible Cause An invalid OVA package has been used for installation (Invalid package/Parsing error/Invalid machine specification error).

Solution Ensure that the OVA package copied to the device matches in size with the OVA package on the FTP server. Refer to the compatibility matrix for details or Contact Cisco Technical Support to ensure that the OVA file provided is compatible with the device operating system and not corrupted.

Possible Cause The virtual services container does not install properly due to unknown reasons.

Solution Uninstall the virtual services container. If the problem persists, collect general troubleshooting information and contact Cisco Technical Support. For more information, see Collecting General Troubleshooting Information, on page 8.

Troubleshooting Activation of Applications in a Virtual Services Container

Problem Activation of an application in a virtual services container is not successful.

Possible Cause Activation of the application may still be ongoing.

Solution Check the status of activation using the **show virtual-service list** command. The following is sample output when the application has an Activated status.

```
Device# show virtual-service list

Virtual Service List:
Name                     Status            Package Name
-----------------------------------------------------------------
WAAS                     Activated         ISR4451X-WAAS-5.2.0-b...
```

Possible Cause The virtual services container does not have sufficient resources for activation of the application.

Solution Check if the device has sufficient resources for virtualization, including memory, disk space, and CPU utilization. You can view the resource requirement for virtualization using the **show virtual-service** command.

```
Device# show virtual-service

Virtual Service Global State and Virtualization Limits:

Infrastructure version : 1.5
Total virtual services installed : 1
Total virtual services activated : 1

Machine types supported   : LXC
Machine types disabled    : KVM

Maximum VCPUs per virtual service : 1
Resource virtualization limits:
Name                         Quota     Committed     Available
-----------------------------------------------------------------
system CPU (%)                   6             1             5
memory (MB)                    256           256             0
bootflash (MB)                 256           164            92
```

Possible Cause The application does not activate properly due to unknown reasons.

Solution Deactivate and uninstall the application. If the problem persists, collect general troubleshooting information and contact Cisco Technical Support. For more information, see .

Troubleshooting Uninstallation of Applications in a Virtual Services Container

Problem Uninstallation of an application from the virtual services container is not successful.

Possible Cause The application being uninstalled has not deactivated completely.

Solution Check the activation status of an application using the **show virtual-service list** command. The following is sample output when the application is in the Deactivated status and can be uninstalled.

```
Device# show virtual-service list


Virtual Service List:
Name                    Status          Package Name
------------------------------------------------------------------
WAAS                    Deactivated     ISR4451X-WAAS-5.2.0-b...
```

Possible Cause The application does not uninstall gracefully due to unknown reasons.

Solution As a last resort, delete the `virtual-instance.conf`, using the **delete** command and then reload the device.

```
Device# delete bootflash:virtual-instance.conf
Device#  reload
```

Solution If the problem persists, collect general troubleshooting information and contact Cisco Technical Support. For more information, see .

Troubleshooting Deactivation of Applications in a Virtual Services Container

Problem Deactivation of an application is not successful.

Possible Cause The application being deactivated is not activated.

Solution Check the status of activation of the application using the **show virtual-service list** command. The following is sample output from a **show virtual-service list** when the application is in the Activated state and can be deactivated.

```
Device# show virtual-service list

Virtual Service List:
Name                    Status          Package Name
------------------------------------------------------------------
oneFW                   Activated       iosxe-cx-9.0.2-hudson...
```

Possible Cause Deactivation takes a long time (5 minutes).

Solution Check if application directories are in use. Ensure that there are no shells open in the application file system directories on the device.

Possible Cause The application does not deactivate gracefully due to unknown reasons.

Solution As a last resort, uninstall the application (if you haven't done so yet) and delete the `virtual-instance.conf` configuration file, using the **delete** command and reload the device. This step deletes all applications installed in the virtual services container.

```
Device#  delete bootflash:virtual-instance.conf
Device#  reload
```

Solution If the problem persists, generate general troubleshooting information and contact Cisco Technical support. For more information, see Collecting General Troubleshooting Information, on page 8.

Configuration Examples for a Virtual Services Container

Example: Cisco Plug-in for OpenFlow Virtual Services Container Installation Configuration

```
Device# enable
Device# copy scp://myserver.com/downloads/ofa-1.0.0-n3000-SPA-k9.ova
bootflash:/ofa-1.0.0-n3000-SPA-k9.ova
Device# virtual-service install name openflow_agent package
bootflash:ofa-1.0.0-n3000-SPA-k9.ova
Device# configure terminal
Device(config)# virtual-service openflow_agent
Device(config-virt-serv)# activate
Device(config-virt-serv)# end
Device# copy running-config startup-config
```

Example: Verifying Cisco Plug-in for OpenFlow Virtual Services Container Installation Configuration

```
Device# show virtual-service list
Virtual Service List:

Name                    Status            Package Name
-------------------------------------------------------------------
openflow_agent          Installed         ofa-1.0.0-n3000-SPA-k9.ova
```

Additional References for the Virtual Services Container

Related Documents

Related Topic	Document Title
Cisco commands	

Technical Assistance

Description	Link
The Cisco Support and Documentation website provides online resources to download documentation and tools. Use these resources to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password.	http://www.cisco.com/cisco/web/support/index.html

Feature Information for Virtual Services Container

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Table 1: Feature Information for the Virtual Services Container

Feature Name	Releases	Feature Information
Virtual Services Container		Cisco Plug-in for OpenFlow runs in an operating system-level virtual services container on a device. Cisco Plug-in for OpenFlow is delivered in an open virtual application (OVA). The OVA package is installed and enabled on the device through the CLI.

Glossary

application

Application installed within and hosted from a virtual ervices container on a device.

container

This is another name for virtual service container.

guest

Application instance running within a container.

host

Operating system installed on a device.

KVM

Kernel Virtual Machine. This is a virtualization infrastructure for the Linux kernel.

LxC

Linux Container. Operating system virtualization technology that shares the host kernel with the guest, but provides namespace extensions to the kernel.

logical Switch

An Cisco Plug-in for OpenFlow switch configured on a device and controlled by an external controller using flows defined on the controller.

OVA

This is an open virtual application. Software package used to install an application and related metafiles within a container. This is a tar file with a .ova extension.

physical Switch

A physical device on which Cisco Plug-in for OpenFlow application is installed and deployed.

virtual machine

This is another name for virtual service container.

virtual service

This is another name for virtual service container.

virtual services container

This is a virtualized environment on a device on which an application can be hosted. A virtualized environment on a Cisco device is called a Cisco virtual-services container.

VMAN

This is the virtualization manager. A process that manages virtual service containers and runs as a host process.