Cisco Nexus Data Broker Embedded Deployment Guide, Release 3.2.2

First Published: 2017-03-17

Last Modified: 2017-04-05

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

CONTENTS

CHAPTER 1

Cisco Nexus Data Broker Embedded Overview

This chapter contains the following sections:

About Cisco Nexus Data Broker Embedded

Visibility into application traffic has traditionally been important for infrastructure operations to maintain security, troubleshooting, and compliance mechanisms, and to perform resource planning. With the technological advances and growth in cloud-based applications, it has become imperative to gain increased visibility into the network traffic. Traditional approaches to gain visibility into network traffic are expensive and rigid, making it difficult to do in large-scale deployments.

Cisco Nexus Data Broker Embedded with Cisco Nexus Switches provides a software-defined, programmable solution to aggregate copies of network traffic using Switched Port Analyzer (SPAN) or network Test Access Points (TAP) for monitoring and visibility. As opposed to traditional network taps and monitoring solutions, this packet-brokering approach offers a simple, scalable and cost-effective solution that is well suited for customers who need to monitor higher-volume and business-critical traffic for efficient use of security, compliance, and application performance monitoring tools.

The Cisco Nexus Data Broker Embedded option provides the flexibility for you to run the Cisco Nexus Data Broker software directly on a Cisco Nexus 3000, 3100, 3200, 3500, or 9000 Series switch in a single-switch deployment. This is suitable for smaller, co-located facilities where customers need only a single Cisco Nexus 3000, 3100, 3200, 3500, or 9000 Series switch for TAP/SPAN aggregation, because it eliminates the requirement to have a separate virtual machine for the Cisco Nexus Data Broker application.

If Cisco Nexus Data Broker does not work after reloading of the device in Embedded mode, you have to run the **python activator script** script from the **ndb** directory in the GitHub repository. Before launching the script, ensure that the Cisco Nexus Data Broker is in activated mode. If it is in the deactivated mode, the script does not execute. You can use **show virtual-service list CLI** command to display the status of Cisco Nexus Data Broker. Use **python bootflash:<python activator script> -v ndb** command to execute the script.

The activator script is different for the various Cisco NXOS versions:

- NDBActivator2.0_A6_A8_Plus.py: For Cisco NXOS versions A6 and A8.
- NDBActivator2.0_I3_I4.py: For Cisco NXOS versions I3 and I4.
- NDBActivator2.0_I5_Plus.py: For Cisco NXOS version I5.

When the Python script is run, it creates a file in the virtual machine that is known as the interfaces file. It contains the details of the interfaces and the management IP address and it updates the **launcher.sh** file. The **embndb** folder is created by the Python script.

The Cisco Nexus Data Broker Embedded solution supports the following:

- Support for the OpenFlow mode or the NX-API mode of operation.

Note The OpenFlow mode and the NX-API mode are supported on both Cisco Nexus 3000 Series and Cisco Nexus 9000 Series switches. Cisco Nexus 9500 supports only NX-API mode of deployment. Cisco Nexus 3500 supports only Openflow mode of deployment.

You can enable only one mode, either OpenFlow or NX-API mode, at a time.

In order to start or stop the Cisco Nexus Data Broker application in embedded mode, you should activate or de-activate the **ofa** file. Do not use **./runxnc.sh** as it is not the right way to start the application.

Note Starting with Cisco Nexus 3000 Release 7.x, the NX-API configuration is supported on the following Cisco Nexus 3100 Series switches:

- Cisco Nexus 3172 switches
- Cisco Nexus 3132 switches
- Cisco Nexus 3164 switches
- Cisco Nexus 31128 switches
- Cisco Nexus 3232 switches
- Cisco Nexus 3264 switches

Note SPAN session that includes production switch and APIC configurations are not supported in Embedded Nexus Data Broker.

- Support for Layer-7 filtering for the HTTP traffic using the HTTP methods.
- Support for VLAN and MPLS tag stripping.
- The ability to aggregate traffic from multiple TAP or SPAN ports connected to a single switch.
- Support for Q-in-Q to tag input source TAP and SPAN ports.
- Symmetric hashing or symmetric load balancing.
- Rules for matching monitoring traffic based on Layer 1 through Layer 4 information.

- The ability to replicate and forward traffic to multiple monitoring tools.
- Timestamp tagging using the Precision Time Protocol.
- Packet truncation beyond a specified number of bytes to discard payload.
- Security features, such as role-based access control (RBAC), and integration with an external Active Directory using RADIUS or TACACS for authentication and authorization.
- End-to-end path visibility and both port and flow level statistics for troubleshooting.
- Robust Representational State Transfer (REST) API and web-based GUI for all functions.

Supported Web Browsers

The following web browsers are supported for Cisco Nexus Data Broker Embedded:

- Firefox 18.x and later versions
- Chrome 24.x and later versions

Note JavaScript 1.5 or a later version must be enabled in your browser.

Prerequisites for Cisco Nexus Series Switches

Cisco Nexus Data Broker is supported on Cisco Nexus 3000, 3100, 3200, 3500, and 9000 series switches. Before you deploy the software, you must do the following:

- Ensure that you have administrative rights to log in to the switch.
- Embedded installation for NDB version I5 is currently supported on Cisco Nexus 9000 Switches.
- Verify that the management interface of the switch (mgmt0) has an IP address configured by running the switch# **show running-config interface mgmt0** command.
- Add the VLAN range in the database that is to be used in Cisco Nexus Data Broker for tap aggregation and inline monitoring redirection to support VLAN filtering. For example, the syntax is **vlan <range of VLAN IDs>**. For example, the VLAN range is <1-3967>.

For running the OpenFlow and NX-API mode on the Cisco Nexus Series switches, see the following pre-requisites.

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 3000 Series switches	Enter the **# hardware profile openflow** command at the prompt.	With Cisco Nexus 3000 Series switches, only Openflow mode is supported.

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 3164Q switches	The OpenFlow mode is not supported on the Nexus 3164Q switches.	Enter the following commands at the prompt: • **# hardware profile tcam region qos 0** • **# hardware profile tcam region racl 0** • **# hardware profile tcam region vacl 0** • **# hardware profile tcam region ifacl 1024 double-wide** • **# hardware access-list tcam region mac-ifacl 512**
Cisco Nexus 3172 Series switches	Enter the **# hardware profile openflow** command at the prompt.	Use the **hardware profile mode tap-aggregation** [**l2drop**] CLI command to enable tap aggregation and to reserve entries in the interface table that are needed for VLAN tagging. The l2drop option drops non-IP traffic ingress on tap interfaces.
Cisco Nexus 3200 Series switches	Enter the **hardware access-list tcam region openflow 256** command at the prompt.	Enter the following commands at the prompt: • **# hardware access-list tcam region e-racl 0** • **# hardware access-list tcam region span 0** • **# hardware access-list tcam region redirect 0** • **# hardware access-list tcam region vpc-convergence 0** • **# hardware access-list tcam region racl-lite 256** • **# hardware access-list tcam region l3qos-intra-lite 0** • **# hardware access-list tcam region ifacl 256 double-wide** • **# hardware access-list tcam region mac-ifacl 512**

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 9300 Series switches	Enter the **hardware access-list tcam region openflow 512 double-wide** command at the prompt to configure the MAC filters. For other scenarios, enter the **hardware access-list tcam region openflow 512** command.	Enter the following commands at the prompt: • **# hardware access-list tcam region qos 0** • **# hardware access-list tcam region vacl 0** • **# hardware access-list tcam region racl 0** • **# hardware access-list tcam region redirect 0** • **# hardware access-list tcam region vpc-convergence 0** • **#hardware access-list tcam region ifacl 1024 double-wide** • **# hardware access-list tcam region mac-ifacl 512**
Cisco Nexus 9200 and 9300-EX switches	The OpenFlow mode is not supported on the 9200 and 9300-EX switches.	Enter the following commands at the prompt: • **#hardware access-list tcam region ing-l2-span-filter 0**(For Cisco Nexus 93108 series switch only) • **#hardware access-list tcam region ing-l3-span-filter 0**(For Cisco Nexus 93108 series switch only) • **# hardware access-list tcam region ing-racl 256** • **# hardware access-list tcam region ing-l3-vlan-qos 256** • **# hardware access-list tcam region egr-racl 256** • **# hardware access-list tcam region ing-ifacl 1024**

Cisco Nexus Data Broker Software Release Filename Matrix

See the Cisco Nexus Data Broker software release filename matrix for more information on the software images:

Mode of Deployment	NXOS Image	Mode	File Name
Embedded	7.0(3)I5(1)	NXAPI	`ndb1000-sw-app-emb-i5-k9-3.2.2.zip`
Embedded	7.0(3)I4 ,7.0(3)I3	NXAPI	`ndb1000-sw-app-emb-nxapi-3.2.2-k9.zip`
Embedded	7.0(3)I4, 7.0(3)I3	Openflow	`ndb1000-sw-app-emb-3.2.2-ofa_mmemb-2.1.4-r2-nxos-SPA-k9.zip`
Embedded	6.0(2)U6(3), 6.0(2)A8(1)	Openflow	`ndb1000-sw-app-emb-3.2.2-ofa_mmemb-1.1.5-r3-n3000-SPA-k9.zip`
Centralized	Upto 7.0(3)I5(1)	NXAPI, Openflow	`ndb1000-sw-app-k9-3.2.2.zip`

CHAPTER 2

Deploying Cisco Nexus Data Broker Embedded for OpenFlow

This chapter contains the following sections:

Obtaining the Cisco Nexus Data Broker Embedded Software for OpenFlow

Attention Starting with Cisco NXOS Release I5, Openflow is not supported for Cisco NDB.

Step 1 In a web browser, navigate to Cisco.com.

Step 2 Under **Support**, click **All Downloads**.

Step 3 In the center pane, click **Cloud and Systems Management**.

Step 4 If prompted, enter your Cisco.com username and password to log in.

Step 5 In the right pane, click **Network Controllers and Applications**, and then click **Cisco Nexus Data Broker.**

Step 6 Download and unzip the **Cisco Nexus Data Broker Release 3.2.2** application bundle zip file. For more information regarding the NDB zip file name, see Cisco Nexus Data Broker Software Release Filename Matrix.
The application bundle zip file contains the following:

- The Cisco Nexus Data Broker Software Application package, for example, **ndb1000-sw-app-emb-k9-3.2.2.ova**

- The Cisco Plug-in for OpenFlow package, for example, **ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova**

What to Do Next

Install the software on a Cisco Nexus 3000, 3100, 3200, 3500, or 9000 Series switch.

Upgrading to Release 3.2.2

This process involves using the GUI to download the configuration, perform the upgrade, and then upload the configuration.

Before You Begin

Step 1 Navigate to the **System** tab under **Administration**.
The **System Administration** window is displayed.

Step 2 Click **Download Configuration**.
It downloads the configuration in a zip file format. The name of the zip file is **configuration_startup.zip**.

Step 3 Download the configuration in Cisco NDB 3.1 or Cisco NDB 3.2.

Step 4 Deactivate Cisco NDB and uninstall Cisco NDB using the following steps:

Step 5 **configure terminal**

Example:
```
device# configure terminal
```

Step 6 **virtual-service virtual-services-name**

Example:

```
device(config)# virtual-service <virtual-services-name>
```

Step 7 **no activate**

Example:
```
device(config-virt-serv)# no activate
```

Step 8 **no virtual-service <virtual-services-name>**

Example:
```
device(config)# no virtual-service <virtual-services-name>
```

Step 9 **end**

Example:
```
device(config-virt-serv)# end
```

Step 10 **virtual-service uninstall name virtual-services-name**

Example:

```
# virtual-service uninstall name <virtual-services-name>
```

Step 11 **copy running-config startup-config**

Example:

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

Step 12 Install and activate Cisco NDB 3.2.2 using the following steps:

Step 13 **virtual-service install name <virtual-services-name> package bootflash: ndb1000-sw-app-emb-k9-3.2.2.ova**

Step 14 **show virtual-service list**

Use the show command to check the status of the virtual service installation. After the status of the virtual service becomes listed as **Installed**, run the following commands to activate the service.

Step 15 **configure terminal**

Step 16 device(config)# **virtual-service <virtual-services-name>**

Step 17 device(config)# **activate**

Step 18 device(config)# **end**

Step 19 device(config)# **copy running-config startup-config**

Step 20 Run the **<python activator script>** script using the **python bootflash:<python activator script> -v <ndb virtual service name>** command.

Note The NDB activator script is different for the different Cisco NXOS versions:

- NDBActivator2.0_A6_A8_Plus.py: For Cisco NXOS versions A6 and A8.
- NDBActivator2.0_I3_I4.py: For Cisco NXOS versions I3 and I4.

Note For NXOS devices with A6/A8 version, run the activator script in root user. Copy the activator script in the bootflash of the device and complete the following steps:

```
N3K-130# run bash
bash-3.2$ sudo su
bash-3.2# cd /bootflash/
bash-3.2# python NDBActivator2.0_A6_A8_Plus.py -v ndb
2017-02-27 09:08:05,923 - __main__ - INFO - Successfully created /embndb/interface file with management
 interface details
bash-3.2#
```

Example:

```
device#  configure terminal
device(config)#  virtual-service <virtual-services-name>
device(config)#  no activate
device(config)#  show virtual-service list  (Wait until deactivated complete)
device(config)#  activate
device(config)#  show virtual-service list (Wait until activated complete)
device(config)#  end
device(config)#  copy running-config startup-config
```

Step 21 Upload Cisco NDB configuration that is downloaded in step 1 in the Cisco NDB user interface (UI).

Installing and Activating the Cisco Nexus Data Broker Embedded Software for OpenFlow

Before You Begin

Note You cannot install a new version of the Cisco Nexus Data Broker Embedded if you already have an existing Cisco Monitor Manager Embedded application installed and active.

Before you begin installing a new version of the Cisco Nexus Data Broker Embedded, you must:

- Deactivate your current Cisco Monitor Manager Embedded OVA file.
- Uninstall the Cisco Monitor Manager Embedded OVA file.

Important Ensure that you have at least 1 GB of available space in the bootflash. For example, the **ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova** and **ndb1000-sw-app-emb-k9-3.2.2.ova** file require a total of 850 MB of space in the bootflash for the decompression and installation processes. For more information regarding the NDB zip file name, see Cisco Nexus Data Broker Software Release Filename Matrix.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova bootflash: vrf management** OR switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **download_dir ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova bootflash: vrf management**	Copies the Cisco Plug-in for OpenFlow package from the directory where you downloaded it to the switch.
Step 2	switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **ndb1000-sw-app-emb-k9-3.2.2.ova bootflash:vrf management**	Copies the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch.
Step 3	switch# **show virtual-service list**	Monitors the status of the copy processes.
Step 4	switch# **virtual-service install name ofa_ndbemb package bootflash:ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova** OR switch# **virtual-service install name ofa_ndbemb package bootflash:ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova**	Installs the Cisco Plug-in for OpenFlow package on the switch.
Step 5	switch# **virtual-service install name ndb_emb package bootflash:ndb1000-sw-app-emb-k9-3.2.2.ova**	Installs the Cisco Nexus Data Broker Embedded package on the switch.
Step 6	switch# **show virtual-service list**	Monitors the status of the installations.

	Command or Action	Purpose
		Note Do not continue until both OVA files have been successfully installed.
Step 7	switch# **configure terminal**	Enters global configuration mode on the switch.
Step 8	switch (config)# **virtual-service ofa_ndbemb**	Starts the virtual service for the Cisco Plug-in for OpenFlow package and enters virtual service configuration mode on the switch.
Step 9	switch(config-virt-serv)# **activate**	Activates the Cisco Plug-in for OpenFlow package.
Step 10	switch(config-virt-serv)# **exit**	Returns to global configuration mode.
Step 11	switch(config)# **virtual-service ndb_emb**	Starts the virtual service for the Cisco Nexus Data Broker Embedded package and enters virtual service configuration mode on the switch.
Step 12	switch(config-virt-serv)# **activate**	Activates the Cisco Nexus Data Broker Embedded package.
Step 13	switch(config-virt-serv)# **exit**	Exits virtual service configuration mode on the switch.
Step 14	switch(config)# **show virtual-service list**	Monitors the status of the package activations.
Step 15	Run the **NDB python activator script** script from the **ndb** directory in the GitHub repository at https://github.com/datacenter/nexus9000/blob/master/nexusdatabroker/ using the **python bootflash:<python NDB activator script> -v ndb** command.	Creates `/embndb/interface` file with management interface details: • If the Cisco NDB version is 2.x.x, the following error message is displayed, "Not supported version, please upgrade to the newer version" • If the Cisco NDB version is 3.0.0 or 3.1.0, the */xnclite/launcher.sh* file is updated. • If the Cisco NDB version is 3.2.0, */xnclite/launcher.sh* is not updated. . **Note** The NDB activator script is different for the different Cisco NXOS versions: • NDBActivator2.0_A6_A8.py: For Cisco NXOS versions A6 and A8. • NDBActivator2.0_I3_I4.py: For Cisco NXOS versions I3 and I4.

Step 16	Deactivate the NDB virtual service and activate it.
	Example:
	`device#  configure terminal`
	`device(config)#  virtual-service <virtual-services-name>`
	`device(config)#  no activate`
	`device(config)#  show virtual-service list  (Wait`
		Update the configuration changes.

	Command or Action	Purpose

	`until deactivated complete)`
	`device(config)#  activate`
	`device(config)#  show virtual-service list (Wait`
	`until activated complete)`
	`device(config)#  end`
	`device(config)#  copy running-config`
	`startup-config`	

Configuring the Cisco Plug-in for OpenFlow

The Cisco Plug-in for OpenFlow needs to be connected to the Cisco Nexus Data Broker locally running on the Cisco Nexus 3000, 3100, 3200, 3500, or 9000 Series switch.

Note The steps in this procedure continue the steps that were completed in the previous section.

Before You Begin

Install and activate the Cisco Nexus Data Broker package and the Cisco Plug-in for OpenFlow package.

Enter the following pre-requisite command **hardware profile openflow** for the Cisco Nexus 3000 and 3100 Series switches. Enter the following pre-requisite command **hardware profile forwarding-mode openflow-hybrid** for the Cisco Nexus 3500 Series switches.

Step 1 Enter the configuration mode on the switch.
configure terminal

Step 2 Enter the Cisco Plug-in for OpenFlow configuration mode on the switch.
switch(config)# **openflow**

Step 3 Choose the switch to which you want to connect.
switch(config-ofa)# **switch** *switch_num*

Caution Set the *switch_num* to **1**. This is the default value. Only expert users should set the *switch_num* number to any value other than 1.

Step 4 Choose the pipeline to which you want to connect.
switch(config-ofa-switch)# **pipeline** *pipeline_num*

Caution Set the *pipeline_num* to **201** for Cisco Nexus 3000, 3100, 3200, and 9300 Series switches. This is the default value. Only expert users should set the *pipeline_num* number to any value other than 201.

Set the *pipeline_num* to **203** for Cisco Nexus 3500 Series switches This is the default value. Only expert users should set the *pipeline_num* number to any value other than 203.

Step 5 Configure the controller address using vrf management.
switch(config-ofa-switch)# **controller ipv4** *management_interface_address* **port** *port_num* **vrf management security none**

Note

- The controller ipv4 address should match the management interface (mgmt0) address.
- By default, the Cisco Plug-in for OpenFlow listens on port 6653.

Step 6 Assign ports to the Cisco Plug-in for OpenFlow.
switch(config-ofa-switch)# **of-port interface** *ethernet_port_num*

Example:
switch(config-ofa-switch)# **of-port interface** ethernet1/10

Step 7 Exit from the current configuration command mode and return to EXEC mode.
switch(config-ofa-switch)# **end**

Step 8 Verify that the Cisco Plug-in for OpenFlow is connected to the Cisco Nexus Data Broker.
switch# **show openflow switch** *switch_num* **controllers**

See the Cisco Plug-in for OpenFlow Configuration Guide 1.3

Logging in to the Cisco Nexus Data Broker GUI

The default HTTPS web link for the Cisco Nexus Data Broker GUI is `https://`*`Nexus_Switch_Management_IP`*`:8443/monitor`

Note You must manually specify the https:// protocol in your web browser. The controller must also be configured for HTTPS.

Step 1 In your web browser, enter the Cisco Nexus Data Broker web link, for example, *https://Nexus_Switch_Management_IP:8443/monitor*.

Step 2 On the launch page, do the following:

a) Enter your username and password.
The default username and password is admin/admin.

b) Click **Log In**.

What to Do Next

See the *Cisco Nexus Data Broker Configuration Guide* for the procedures that you need to configure Cisco Nexus Data Broker.

CHAPTER 3

Deploying Cisco Nexus Data Broker Embedded for NX-API

This chapter contains the following sections:

Obtaining the Cisco Nexus Data Broker Embedded Software for NX-API

Step 1 In a web browser, navigate to Cisco.com.

Step 2 In the center pane, click **Cloud and Systems Management**.

Step 3 If prompted, enter your Cisco.com username and password to log in.

Step 4 In the right pane, click **Network Controllers and Applications**, and then click **Cisco Nexus Data Broker.**

Step 5 Download and unzip the **Cisco Nexus Data Broker Release 3.2.2** application bundle zip file. For more information regarding the NDB zip file name, see Cisco Nexus Data Broker Software Release Filename Matrix.
The application bundle zip file contains the following:

- The Cisco Nexus Data Broker Software Application package, for example, **ndb1000-sw-app-emb-k9-3.2.2.ova**

The activator script is different for the various Cisco NXOS versions:

- NDBActivator2.0_I3_I4.py: For Cisco NXOS versions I3 and I4.
- NDBActivator2.0_I5_Plus.py: For Cisco NXOS version I5.

What to Do Next

Install the software on a Cisco Nexus 3000, 3100, 3200, 3500, or 9000 Series switch.

Installing and Activating the Cisco Nexus Data Broker Embedded Software for NX-API Mode for NXOS Versions upto I4

Before You Begin

Note You cannot install a new version of the Cisco Nexus Data Broker Embedded if you already have an existing Cisco Monitor Manager Embedded application installed and active.

Before you begin installing a new version of the Cisco Nexus Data Broker Embedded, you must:

- Deactivate your current Cisco Monitor Manager Embedded OVA file.
- Uninstall the Cisco Monitor Manager Embedded OVA file.

Important Ensure that you have at least 1 GB of available space in the bootflash. The **ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova** and **ndb1000-sw-app-emb-k9-3.2.2.ova** file require a total of 850 MB of space in the bootflash for the decompression and installation processes. For more information regarding the NDB zip file name, see Cisco Nexus Data Broker Software Release Filename Matrix.

SUMMARY STEPS

1. switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **ndb1000-sw-app-emb-k9-3.2.2.ova bootflash:vrf management**
2. switch# **show virtual-service list**
3. switch# **virtual-service install name ndb_emb package bootflash:ndb1000-sw-app-emb-k9-3.2.2.ova**
4. switch# **show virtual-service list**
5. switch# **configure terminal**
6. switch(config)# **virtual-service ndb_emb**
7. switch(config-virt-serv)# **activate**
8. switch(config-virt-serv)# **exit**
9. switch(config)# **show virtual-service list**
10. Run the **NDB python activator script** script from the **ndb** directory in the GitHub repository at https://github.com/datacenter/nexus9000/blob/master/nexusdatabroker/ using the **python bootflash:<python NDB activator script> -v ndb** command.
11. Deactivate the NDB virtual service and activate it.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **ndb1000-sw-app-emb-k9-3.2.2.ova bootflash:vrf management**	Copies the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch.
Step 2	switch# **show virtual-service list**	Monitors the status of the copy processes.
Step 3	switch# **virtual-service install name ndb_emb package bootflash:ndb1000-sw-app-emb-k9-3.2.2.ova**	Installs the Cisco Nexus Data Broker Embedded package on the switch.
Step 4	switch# **show virtual-service list**	Monitors the status of the installations. **Note** Do not continue until both OVA files have been successfully installed.
Step 5	switch# **configure terminal**	Enters global configuration mode on the switch.
Step 6	switch(config)# **virtual-service ndb_emb**	Starts the virtual service for the Cisco Nexus Data Broker Embedded package and enters virtual service configuration mode on the switch.
Step 7	switch(config-virt-serv)# **activate**	Activates the Cisco Nexus Data Broker Embedded package.
Step 8	switch(config-virt-serv)# **exit**	Exits virtual service configuration mode on the switch.
Step 9	switch(config)# **show virtual-service list**	Monitors the status of the package activations.

	Command or Action	Purpose
Step 10	Run the **NDB python activator script** script from the **ndb** directory in the GitHub repository at https://github.com/datacenter/nexus9000/blob/master/nexusdatabroker/ using the **python bootflash:<python NDB activator script> -v ndb** command.	Will create `/embndb/interface` file with management interface details: • if version is 2.x.x, the following error message is displayed, "Not supported version, please upgrade to the newer version" • if version is 3.0.0 or 3.1.0, the */xnclite/launcher.sh* file is updated. • If version is 3.2.0, */xnclite/launcher.sh* is not updated. . **Note** The NDB activator script is different for the different Cisco NXOS versions: • NDBActivator2.0_I3_I4.py: For Cisco NXOS versions I3 and I4.

Step 11	Deactivate the NDB virtual service and activate it.
	Example:
	`device#  configure terminal`
	`device(config)#  virtual-service`
	`<virtual-services-name>`
	`device(config)#  no activate`
	`device(config)#  show virtual-service list`
	`(Wait until deactivated complete)`
	`device(config)#  activate`
	`device(config)#  show virtual-service list (Wait`
	` until activated complete)`
	`device(config)#  end`
	`device(config)#  copy running-config`
	`startup-config`
		Update the configuration changes.

Installing and Activating the Cisco Nexus Data Broker Embedded Software for NX-API Mode for NXOS I5

You can now install NDB directly on a device in embedded mode on NXOS I5 release. To install Cisco Nexus Data Broker Embedded software on NXOS I5 release, use the NDB activator script, NDBActivator2.0_I5_Plus.py. The activator script performs the following functions:

- Resizes the Guestshell resources.
- Unzips and places the XNC folder into the Guestshell home directory.
- Configures the NXAPI to listen to network management namespace.
- Configures the Guestshell to management VRF.
- Configures the system unit file.

- Starts the NXAPI application.

Before You Begin

Note By default, you cannot install a new version of the Cisco Nexus Data Broker Embedded if you already have an existing Cisco Nexus Data Broker Embedded application installed and active. You can use force attribute to forcefully run the activator script even if it is already activated. For example:

```
Syntax: python <file path>NDBActivator2.0_I5_Plus.py –v guestshell+ <zip file path>
--force

Example:  python bootflash:NDBActivator2.0_I5_Plus.py –v guestshell+
/bootflash/ndb1000-sw-app-emb-i5-nxapi-k9-3.2.2.zip --force
```

Before you begin installing a new version of the Cisco Nexus Data Broker Embedded, you must:

- To uninstall NDB application, destroy the guestshell using the command, guestshell destroy.
- Download ndb1000-sw-app-emb-i5-k9-3.2.2.zip and extract.
- Copy NDBActivator2.0_I5_Plus.py and ndb1000-sw-app-emb-i5-nxapi-k9-3.2.2.zip to device.

Important Ensure that you have sufficient space available in the bootflash. The **ndb1000-sw-app-emb-i5-nxapi-k9-3.2.2.zip** file require a total of ~600 MB of space in the bootflash (/volatile folder) for the decompression processes. The script runs only on NXOS platform, version I5, with memory greater than 8GB.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **NDBActivator2.0_I5_Plus.py bootflash:vrf management**	Copies the NDBActivator2.0_I5_Plus.py from the directory where you downloaded it to the switch.
Step 2	switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **ndb1000-sw-app-emb-i5-nxapi-k9-3.2.2.zip bootflash:vrf management**	Copies the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch.
Step 3	switch# **show virtual-service list**	Monitors the status of the copy processes.
Step 4	switch# **python bootflash:NDBActivator2.0_I5_Plus.py -v guestshell+ /bootflash/ndb1000-sw-app-emb-i5-nxapi-k9-3.2.2.zip**	Installs the Cisco Nexus Data Broker Embedded package on the switch.
Step 5	switch# **show virtual-service list**	Monitors the status of the installations. **Note** To start the NDB application, use the **guestshell enable** command. If the NDB application is initiated through the Python script, guestshell is enabled automatically.

	Command or Action	Purpose
		Note To stop the NDB application, use the **guestshell disable** command. If you have used the **guestshell disable** command to stop NDB application, then to enable it again, you need to use **guestshell enable** command. **Note** Do not continue until installation completes successfully. NDB application starts after it is installed successfully.

Adding a Device

You need to manually add a device to the NDB application to monitor it.

SUMMARY STEPS

1. Log in to NDB user interface.
2. Click **Administration** Tab and then click **DEVICE CONNECTIONS** Tab.
3. To add a new device, click **Add Devices**, **Add Device** dialog box appears.
4. In the **Add Device Dialog** box, enter the following details:
5. Click **Add Device** in the **Add Device** dialog box to add the device with the provided credentials.

DETAILED STEPS

Step 1 Log in to NDB user interface.

Step 2 Click **Administration** Tab and then click **DEVICE CONNECTIONS** Tab.

Step 3 To add a new device, click **Add Devices**, **Add Device** dialog box appears.

Step 4 In the **Add Device Dialog** box, enter the following details:

- **Address**: IP address of the new device.
- **User Name**: User name for accessing the device.
- **Password**: Password to validate the user.
- **Connection Type**: Type of connection the new device will use, select *NXAPI*.
- **Port Number**: Port number through which the device will communicate.

Step 5 Click **Add Device** in the **Add Device** dialog box to add the device with the provided credentials.

Upgrading to Release 3.2.2 for Cisco NXOS Releases Upto I4

This process involves using the GUI to download the configuration, perform the upgrade, and then upload the configuration.

Step 1 Navigate to the **System** tab under **Administration**.
The **System Administration** window is displayed.

Step 2 Click **Download Configuration**.
It downloads the configuration in a zip file format. The name of the zip file is **configuration_startup.zip**.

Step 3 Download the configuration in Cisco NDB 3.1 or Cisco NDB 3.2.

Step 4 Deactivate Cisco NDB and uninstall Cisco NDB using the following steps:

Step 5 **configure terminal**

Example:

```
device# configure terminal
```

Step 6 **virtual-service virtual-services-name**

Example:

```
device(config)# virtual-service <virtual-services-name>
```

Step 7 **no activate**

Example:

```
device(config-virt-serv)# no activate
```

Step 8 **no virtual-service <virtual-services-name>**

Example:

```
device(config)# no virtual-service <virtual-services-name>
```

Step 9 **end**

Example:

```
device(config-virt-serv)# end
```

Step 10 **virtual-service uninstall name virtual-services-name**

Example:

```
# virtual-service uninstall name <virtual-services-name>
```

Step 11 **copy running-config startup-config**

Example:

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

Step 12 Install and activate Cisco NDB 3.2.2 using the following steps:

Step 13 **virtual-service install name <virtual-services-name> package bootflash: ndb1000-sw-app-emb-k9-3.2.2.ova**

Step 14 **show virtual-service list**
Use the show command to check the status of the virtual service installation. After the status of the virtual service becomes listed as **Installed**, run the following commands to activate the service.

Step 15 **configure terminal**

Step 16 device(config)# **virtual-service <virtual-services-name>**

Step 17 device(config)# **activate**

Step 18 device(config)# **end**

Step 19 device(config)# **copy running-config startup-config**

Step 20 Run the **<python NDB activator script>** script using the **python bootflash:<python activator script> -v <ndb virtual service name>** command.

Note The NDB activator script is different for the different Cisco NXOS versions:

- NDBActivator2.0_A6_A8.py: For Cisco NXOS versions A6 and A8.
- NDBActivator2.0_I3_I4.py: For Cisco NXOS versions I3 and I4.
- NDBActivator2.0_I5.py: For Cisco NXOS version I5.

Example:

```
device#  configure terminal
device(config)#  virtual-service <virtual-services-name>
device(config)#  no activate
device(config)#  show virtual-service list  (Wait until deactivated complete)
device(config)#  activate
device(config)#  show virtual-service list (Wait until activated complete)
device(config)#  end
device(config)#  copy running-config startup-config
```

Step 21 Upload Cisco NDB 3.2 configuration that you downloaded in step 1 in the Cisco NDB user interface (UI).

Upgrading to Release 3.2.2 for Cisco NXOS Release I5

This process involves using the GUI to download the configuration, perform the upgrade, and then upload the configuration.

Step 1 Navigate to the **System** tab under **Administration**.
The **System Administration** window is displayed.

Step 2 Download the ndb1000-sw-app-emb-i5-nxapi-k9-3.2.2.zip file into the device.

Step 3 Click **Download Configuration**.
It downloads the configuration in a zip file format. The name of the zip file is **configuration_startup.zip**.

Step 4 Download the configuration in Cisco NDB 3.1 or Cisco NDB 3.2.

Step 5 Deactivate Cisco NDB and upgrade Cisco NDB using the following steps:

Step 6 **configure terminal**

Example:

```
device# configure terminal
```

Step 7 **virtual-service virtual-services-name**

Example:

```
device(config)# virtual-service <virtual-services-name>
```

Step 8 **no activate**

Example:

```
device(config-virt-serv)# no activate
```

Step 9 **no virtual-service <virtual-services-name>**

Example:

```
device(config)# no virtual-service <virtual-services-name>
```

Step 10 **end**

Example:

```
device(config-virt-serv)# end
```

Step 11 **virtual-service uninstall name virtual-services-name**

Example:

```
# virtual-service uninstall name <virtual-services-name>
```

Step 12 **copy running-config startup-config**

Example:

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

Step 13 Upgrade the NXOS version to release I5.1.

Step 14 Download the *ndb1000-sw-app-emb-i5-k9-3.2.2.zip* file into the standalone device.

Step 15 Run the NDBActivator2.0_I5_Plus.py script in the device console.

Example:

```
python bootflash:NDBActivator2.0_I5_Plus.py -v guestshell+
/bootflash/ndb1000-sw-app-emb-i5-k9-3.2.2.zip
```

NDB application starts after the installation completes successfully.

Step 16 Log in to the NDB application using the credentials.
You need to manually add a device in NDB.

Step 17 To add a new device, click **Administration** Tab and then click **DEVICE CONNECTIONS** Tab.

Step 18 To add a new device, click **Add Devices**, **Add Device** dialog box appears.

Step 19 In the **Add Device Dialog** box, enter the following details:

- **Address**: IP address of the new device.
- **User Name**: User name for accessing the device.
- **Password**: Password to validate the user.
- **Connection Type**: Type of connection the new device will use, select *NXAPI*.
- **Port Number**: Port number through which the device will communicate.

Step 20 Click **Add Device** in the **Add Device** dialog box to add the device with the provided credentials.

Step 21 device(config)# **copy running-config startup-config**

Step 22 Upload Cisco NDB 3.2 configuration that you downloaded in step 1 in the Cisco NDB user interface (UI).

Logging in to the Cisco Nexus Data Broker GUI

The default HTTPS web link for the Cisco Nexus Data Broker GUI is `https://`*`Nexus_Switch_Management_IP`*`:8443/monitor`

Note You must manually specify the https:// protocol in your web browser. The controller must also be configured for HTTPS.

Step 1 In your web browser, enter the Cisco Nexus Data Broker web link, for example, *https://Nexus_Switch_Management_IP:8443/monitor*.

Step 2 On the launch page, do the following:

a) Enter your username and password.
The default username and password is admin/admin.

b) Click **Log In**.

What to Do Next

See the *Cisco Nexus Data Broker Configuration Guide* for the procedures that you need to configure Cisco Nexus Data Broker.

CHAPTER 4

Installing or Upgrading the Cisco Nexus Data Broker Software in Centralized Mode

This chapter contains the following sections:

Installing Cisco Nexus Data Broker in Centralized Mode

Installing or Upgrading the Cisco Nexus Data Broker Software in Centralized Mode

Important Direct upgrade path to Cisco Nexus Data Broker Release 3.2.2 is available from Cisco Nexus Data Broker release 3.0 or above. If you are running a previous release, upgrade to Release 3.0 first before upgrading to Release 3.2.2.

- To complete a new installation of Cisco Nexus Data Broker, see the *Installing the Cisco Nexus Data Broker Software* section.

Installing the Cisco Nexus Data Broker Software in Centralized Mode

Step 1 In a web browser, navigate to **www.cisco.com**.

Step 2 Under **Support**, click **All Downloads**.

Step 3 In the center pane, click **Cloud and Systems Management**.

Step 4 If prompted, enter your Cisco.com **username** and **password** to log in.

Step 5 In the right pane, click **Network Controllers and Applications**, and then click **Cisco Nexus Data Broker.**

The file information for Release 3.2.2 is displayed: Cisco Nexus Data Broker Software Application: ndb1000-sw-app-k9-3.2.2.zip

Step 6 Download the Cisco Nexus Data Broker application bundle.

Step 7 Create a directory in your Linux machine where you plan to install Cisco Nexus Data Broker.
For example, in your Home directory, create `CiscoNDB`.

Step 8 Copy the Cisco Nexus Data Broker zip file into the directory that you created.

Step 9 Unzip the Cisco Nexus Data Broker zip file.
The Cisco Nexus Data Broker software is installed in a directory called `xnc`. The directory contains the following:

- `runxnc.sh` file—The file that you use to launch Cisco Nexus Data Broker.
- `version.properties` file—The Cisco Nexus Data Broker build version.
- `captures` directory—The directory that contains output dump files from analytics run in Cisco Nexus Data Broker.

  Note The `captures` directory is created after you execute the Cisco Nexus Data Broker analytics tool.

- `configuration` directory—The directory that contains the Cisco Nexus Data Broker initialization files.

  This directory also contains the `startup` subdirectory where configurations are saved.

- `bin` directory—The directory that contains the following script:
  - `xnc` file—This script contains the Cisco Nexus Data Broker common CLI.
- `etc` directory—The directory that contains profile information.
- `lib` directory—The directory that contains the Cisco Nexus Data Broker Java libraries.
- `logs` directory—The directory that contains the Cisco Nexus Data Broker logs.

  Note The `logs` directory is created after the Cisco Nexus Data Broker application is started.

- `plugins` directory—The directory that contains the OSGi plugins.
- `work` directory—The webserver working directory.

  Note The `work` directory is created after the Cisco Nexus Data Broker application is started.

Upgrading the Application Software in Centralized Mode

Use the **upgrade** command to upgrade to Cisco Nexus Data Broker Release 3.2.1. When you are upgrading from Release 2.2.0 and/or Release 2.2.1, you first need to upgrade to Release 3.0.0 or Release 3.1.0 or Release 3.2.0, or Release 3.2.1 and only then you can upgrade to Cisco Nexus Data Broker Release 3.2.2. This upgrade is an in-place upgrade, which means that the product bits are replaced. A backup archive is created to restore your original installation, if necessary.

Note Once you upgrade to Cisco Nexus Data Broker Release 3.2.2, you cannot use the downgrade option to rollback to 3.2.1, 3.2.0 or 3.1.0 or 3.0.0. You have to use the configuration archive that is created during the upgrade process to rollback the software.

Note When you upgrade the software, the hostname should not be changed during the upgrade process. While upgrading to Cisco Nexus Data Broker Release 3.2.2, user should not allowed to change the hostname. If the hostname is changed during the upgrade, the upgrade process is not done successfully.

When you execute the **upgrade** command, the installation and the configuration are upgraded. However, any changes you made to the shell scripts or configuration files, for example, `runxnc.sh` and `config.ini`, are overwritten. After you complete the upgrade process, you must manually reapply your changes to those files.

Before You Begin

- Stop all controller instances that use the Cisco Nexus Data Broker installation. This will avoid conflicts with the file system, which is updated during the upgrade.
- If you are using high availability clustering, stop all application instances in the cluster to ensure that there are no inconsistencies.
- Back up your `config.ini` and `runxnc.sh` files.

Important You should manually backup your `config.ini` and `runxnc.sh` files before upgrading, because the backup process does not back them up for you. If you do not backup your files before upgrading, any changes you made will be lost.

Note When you run `runxnc.sh` script after upgrading from Cisco Nexus Data Broker, make sure that you upgrade your current Java version and you have set JAVA_HOME to point to the correct JAVA version. If the current Java version used is lower than 1.8.0_45, the Java process does not start and it does not get the Web access.

Note When you run `runxnc.sh` script, there is a thread in the script that monitors the log and the Cisco Nexus Data Broker JAVA process to monitor the health of the Cisco Nexus Data Broker. The default value for this option is 30 Seconds.

Step 1 In a web browser, navigate to Cisco.com.

Step 2 Under **Support**, click **All Downloads**.

Step 3 In the center pane, click **Cloud and Systems Management**.

Step 4 In the right pane, click **Network Controllers and Applications**, and then click **Cisco Nexus Data Broker.**

Step 5 Download the Cisco Nexus Data Broker Release 3.2.1 application bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.2.1.zip

Step 6 Create a temporary directory in your Linux machine where you plan to upgrade to Cisco Nexus Data Broker.
For example, in your `Home` directory, create `CiscoNDB_Upgrade`.

Step 7 Unzip the Cisco Nexus Data Broker Release 3.2.1 zip file into the temporary directory that you created.

Step 8 Navigate to the `xnc` directory that was created when you installed the Cisco Nexus Data Broker release earlier.

Step 9 Backup your Cisco Nexus Data Broker release installation using your standard backup procedures.

Step 10 Stop running all Cisco Nexus Data Broker release processes.

Step 11 Navigate to the `xnc/bin` directory in the temporary directory that you created for the Cisco Nexus Data Broker Release 3.2.2 upgrade software.

Step 12 Upgrade the application by entering the **./xnc upgrade --perform --target-home** {*xnc_directory_to_be_upgraded*} **[--verbose] [--backupfile** {*xnc_backup_location_and_zip_filename*}**]** command.
You can use one of the following options:

Option	Description
--perform --target-home {*xnc_directory_to_be_upgraded*}	Upgrades the Cisco XNC Monitor Manager installation to Cisco Nexus Data Broker.
--perform --target-home {*xnc_directory_to_be_upgraded*} **--backupfile** {*xnc_backup_location_and_zip_filename*}	Upgrades the Cisco XNC Monitor Manager installation to Cisco Nexus Data Broker and creates a backup .zip file in the directory path that you set. **Note** You must provide the name of the backup file and the .zip extension.
--rollback --target-home {*xnc_directory_to_be_upgraded*}	Rolls back to the previous Cisco XNC Monitor Manager installation.
--rollback --target-home {*xnc_directory_to_be_upgraded*} **--backupfile** {*xnc_backup_location_and_zip_filename*}	Rolls back to the previous Cisco XNC Monitor Manager installation using the backup file in the absolute path that you set.
--verbose	Displays detailed information to the console. This option can be used with any other option and is disabled by default.

Option	Description
--validate --target-home *{xnc_directory_to_be_upgraded}*	Validates the installation.
./xnc help upgrade	Displays the options for the **upgrade** command.

Step 13 Navigate to the `xnc` directory where you originally installed Cisco XNC Monitor Manager.

Step 14 Start the application processes that you previously stopped.

Note Press Ctrl–F5, or press the Cmd, Shift, and R keys simultaneously when you access Cisco Nexus Data Broker through a web UI following an upgrade.

Step 15 If you have any upgrade-related issues, perform the following tasks:

a) Stop all application processes.

b) Navigate to the temporary directory that you created in Step 6.

c) Enter the **./xnc upgrade --rollback --target-home** *{xnc_directory_to_be_downgraded}* **--backupfile** *{xnc_backup_location_and_zip_filename}* **[--verbose]** command.

d) Restart the application processes.

Note Press Ctrl–F5, or press the Cmd, Shift, and R keys simultaneously when you access Cisco XNC Monitor Manager through a web UI following a rollback.

Starting the Application

Note When you are running xnc for the first time, the URL that you need to connect to and the port that it is listening on are displayed on the screen. For example, when you run the ./runxnc.sh script, the following message is displayed on the screen: Web GUI can be accessed using below URL: *[https://<IP_address>: 8443]*.

You can use one of the following options:

Option	Description
no option	
-jmx	
-jmxport *port_number*	Enables JMX remote access on the specified JVM port.
-debug	
-debugsuspend	
-debugport *port_number*	Enables debugging on the specified JVM port.
-start	**Note**

Option	Description
-start *port_number*	**Note**
-stop	
-status	
-console	
-help	Displays the options for the **./runxnc.sh** command.
-tls	To enable TLS, start the controller by entering the **./runxnc.sh -tls -tlskeystore** *keystore_file_location* **-tlstruststore** *truststore_file_location* command.
-osgiPasswordSync	To set the OSGi web console password same as the XNC password if the XNC password is changed. **Note** This step is optional. If the application is started without this option, the OSGi console can be accessed through the default credentials.

Note Use runxnc.sh script to start Cisco Nexus Data Broker. You have to set a path variable named JAVA_HOME. It sets the path variables that are used for startup and launches the OSGi framework with the specified options. If a user attempts to start the Cisco Nexus Data Broker application with Java version lower than 1.7, an error message is displayed and the application aborts. To resolve the issue, upgrade your current Java version and restart Cisco Nexus Data Broker. If the current Java Version used is lower than 1.8.0_45, a warning message is issued before the start that Upgrade to 1.8.0_45 or above is recommended.

Verifying That The Application is Running

Step 1 Navigate to the `xnc` directory that was created when you installed the software.

Step 2 Verify that the application is running by entering the **./runxnc.sh -status** command.
The controller outputs the following, which indicates that the controller is running the Java process with PID 21680:

```
Controller with PID:21680 -- Running!
```

What to Do Next

Connect the switches to the controller. For more information, see the configuration guide for your switches.