Cisco Nexus Data Broker Deployment Guide, Release 3.9.x

First Published: 2020-04-04

Last Modified: 2023-03-16

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

Trademarks

Trademarks

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS REFERENCED IN THIS DOCUMENTATION ARE SUBJECT TO CHANGE WITHOUT NOTICE. EXCEPT AS MAY OTHERWISE BE AGREED BY CISCO IN WRITING, ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS DOCUMENTATION ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED.

The Cisco End User License Agreement and any supplemental license terms govern your use of any Cisco software, including this product documentation, and are located at: http://www.cisco.com/go/softwareterms.Cisco product warranty information is available at http://www.cisco.com/go/warranty. US Federal Communications Commission Notices are found here http://www.cisco.com/c/en/us/products/us-fcc-notice.html.

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.

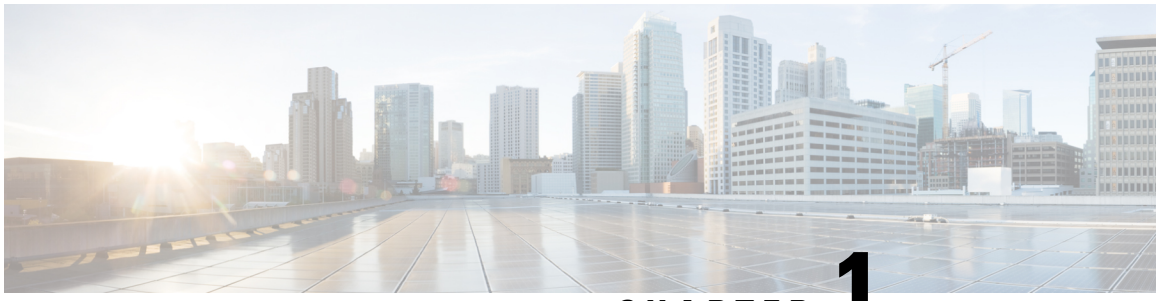
Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability for delay in the delivery or failure to deliver any products or feature roadmap items that may be set forth in this document.

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.

The documentation set for this product strives to use bias-free language. For the purposes of this documentation set, bias-free is defined as language that does not imply discrimination based on age, disability, gender, racial identity, ethnic identity, sexual orientation, socioeconomic status, and intersectionality. Exceptions may be present in the documentation due to language that is hardcoded in the user interfaces of the product software, language used based on RFP documentation, or language that is used by a referenced third-party product.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: www.cisco.com go trademarks. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

CONTENTS



CHAPTER 1

Cisco Nexus Data Broker Overview

This chapter contains the following sections:

About Cisco Nexus Data Broker

Visibility into application traffic has traditionally been important for infrastructure operations to maintain security, troubleshooting, and compliance and 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 for managers of large-scale deployments.

Cisco Nexus Data Broker 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 Point (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.

With the flexibility to use a variety of Cisco Nexus Switches and the ability to interconnect them to form a scalable topology provides the ability to aggregate traffic from multiple input TAP or SPAN ports, and replicate and forward traffic to multiple monitoring tools which may be connected across different switches. Combining the use of Cisco plugin for OpenFlow and the Cisco NX-API agent to communicate to the switches, Cisco Nexus Data Broker provides advance features for traffic management.

Cisco Nexus Data Broker provides management support for multiple disjointed Cisco Nexus Data Broker networks. You can manage multiple Cisco Nexus Data Broker topologies that may be disjointed using the same application instance. For example, if you have 5 data centers and want to deploy an independent Cisco Nexus Data Broker solution for each data center, you can manage all 5 independent deployments using a single application instance by creating a logical partition (network slice) for each monitoring network.

Starting with Cisco NDB release 3.6, when a new switch is discovered on NDB, the following connections are installed on the ISL interfaces:

- Default-Deny-ISL connection with Default-Deny-All, Default-Deny-MPLS, and Default-Deny-ARP filters. This connection is supported on all the types of switches in NXAPI mode.
- Default-Deny-ISL-ICMP connection with Default-Deny-ICMP and Default-Deny-ICMP-All filters. This connection is supported on 9200, 9300EX, 9300FX, 9500EX, and 9500FX switches in NXAPI mode.

All the ACLs related to the default filters are installed on the ISL interfaces of the new switch. By default, this feature is enabled for all the new ISL interfaces.

Starting with Cisco Nexus Data Broker, Release 3.8:

- Add newly supported feature list.

Note You can configure a maximum of 30 unique Port ACLs (PACLs) for the Cisco Nexus 9300 FX Platform.

Note Each PACL takes one label. If the same PACL is configured on multiple interfaces, the same label is shared. If each PACL has unique entries, the PACL labels are not shared, and the label limit is 30.

Note You can manage this feature using the mm.addDefaultISLDenyRules attribute in config.ini file. By default, themm.addDefaultISLDenyRules attribute is not be present in config.in file. To disable this feature, you need to add the mm.addDefaultISLDenyRules attribute to config.ini file ans set it to false and restart the device. For example:

```
mm.addDefaultISLDenyRules = false
```

Note A Cisco Nexus Data Broker instance can support either the OpenFlow or NX-API device configuration mode, it does not support both device types.

Note Starting with Cisco NDB release 3.6, Global ACLs are automatically added to all the interfaces on a device. By default, Global ACLs are enabled for a device. To manage Global ACLs, you need to add the configure.global.acls parameter in the config.ini file. Set the configure.global.acls parameter to *false* and restart the device to disable Global ACLs on the device.

Note Starting with Cisco NDB release 3.6, constistency check option is now available for NX-API based devices along with the OpenFlow based devices.

Note Starting with Cisco NDB Release 3.4, you can configure the timeout interval for NDB GUI. By default, a user is logged out if the session is inactive for more than 10 minutes. You can configure the inactive timeout interval by modifying the timeout interval attribute in the *xnc/configuration/web.xml* file. You need to restart the NDB to apply the new interval.

Note Starting with Cisco NDB Release 3.6.2, you can now configure the inactivity timeout interval in NDB GUI instead of updating the *xnc/configuration/web.xml* file. By default, a user is logged out if the session is inactive for more than 10 minutes. You need to re-log in to the NDB to apply the new interval. For more information, see *Configuring Inactivity Timeout* section. .

Note Starting with Cisco Nexus Data Broker, Release 3.3:

- Advanced filtering based on TCP AND UDP flags is supported to filter the traffic.
- IPv6, QinQ, and UDF are supported for NX-OS I6 release platform.
- You can define a User Defined Filter (UDF) and use it while creating a filter for traffic management.
- Edit Priority field for the connections is configurable. By default, edit is enabled for the Cisco NDB administrator role.

Note Starting with Cisco NDB release 3.2.2, IPv6 addressing is supported in centralized mode. You can configure NDB to use either IPv6 addressing or both IPv4 and IPv6 addressing. Set `ipv6.strict` attribute in config.ini file to *true* to make NDB accessible only through IPv6 address. If you set the `ipv6.strict` attribute to *false*, you can access NDB through IPv4 or IPv6 address.

Note Starting with Cisco Nexus Data Broker Release 3.1, the user strings for Cisco Nexus Data Broker can contain alphanumeric characters including the following special characters: period (.), underscore (_), or hyphen (-) . These are the only special characters that are allowed in the user strings.

Note The hostname string for Cisco Nexus Data Broker can contain between 1 and 256 alphanumeric characters including the following special characters: period (.), underscore (_), or hyphen (-) . These are the only special characters that are allowed in the user strings.

Note Nexus 3548 does not support Block-Tx feature.

Cisco Nexus Data Broker provides 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, 9200, and 9300-EX switches support 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.

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

When using OpenFlow mode, NX-API is available for auxiliary configurations only, for example, Enabling Q-in-Q on the SPAN and TAP ports.

Cisco Nexus 9300-EX Series switches support only Cisco NX-OS Release 7.0(3)I5(1) and later releases.

The configuration that is supported in the AUX mode is:

- Pull and push of interface description
- Q-in-Q configuration
- Redirection
- Port Channel load balancing
- MPLS Stripping

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

- The features that are supported with the Cisco Nexus 9500 Series switches are:
  - The NX-API feature is supported. (OpenFlow is not supported.)
  - The MPLS strip feature is supported.
  - The label age CLI feature is not supported.

- Support for Layer-7 filtering for the HTTP traffic using the HTTP methods.
- Support for VLAN filtering.
- Support for MPLS tag strippng.
- A scalable topology for TAP and SPAN port aggregation.
- Support for Q-in-Q to tag input source TAP and SPAN ports.
- 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.
- Time stamping using Precision Time Protocol (PTP).
- Packet truncation beyond a specified number of bytes to discard payload.
- Reaction to changes in the TAP/SPAN aggregation network states.
- Security features, such as role-based access control (RBAC), and integration with an external Active Directory using RADIUS, TACACS, or LDAP for authentication, authorization, and accounting (AAA) functions.
- End-to-end path visibility, including both port and flow level statistics for troubleshooting.
- Robust Representational State Transfer (REST) API and a web-based GUI for performing all functions
- Support for Cisco plugin for Open Flow, version 1.0
- Cisco Nexus Data Broker adds NX-API plugin to support Cisco Nexus 9000 Series switches as TAP/SPAN aggregation. The NX-API supports JSON-RPC, XML, and JSON. Cisco Nexus Data Broker interacts with Cisco Nexus 9000 Series using the NX-API in JSON message formats.
- Beginning with Cisco Nexus Data Broker, Release 3.1, Cisco Nexus Data Broker is certified with Cisco Nexus 9200 Series and Cisco Nexus 9300-EX Series switches.

  The following features are supported on the Cisco Nexus 9300-EX, -FX, -FX2 Series switches:

  - Symmetric Load Balancing
  - Q-in-Q
  - Switch Port Configuration
  - MPLS Stripping
  - BlockTx
  - Truncate

- Beginning with Cisco Nexus Data Broker, Release 3.1, Cisco Nexus Data Broker is shipped with a certificate for the HTTPS connection between the Cisco Nexus Data Broker and a browser. Now with this feature, you can change to a different certificate than the shipped certificate.

  The script **generateWebUIcertificate.sh** is available in the **xnc/configuration** folder. If you execute this script, it moves the shipped certificate to **old_keystore** and the new certificate is generated in **keystore**. On the next Cisco Nexus Data Broker restart, this new certificate is used.

With Cisco Nexus Data Broker, you can:

- Classify Switched Port Analyzer (SPAN) and Test Access Point (TAP) ports.
- Integrate with Cisco ACI through Cisco APIC to configure SPAN destinations and SPAN sessions.
- Add monitoring devices to capture traffic.
- Filter which traffic should be monitored.
- Redirect packets from a single or multiple SPAN or TAP ports to multiple monitoring devices through delivery ports.
- Restrict which users can view and modify the monitoring system.
- If Cisco Nexus 9000 Series switch is using 7.0(3)I4(1) or later version in NX-API mode and if a flow is installed using a VLAN filer, then the device goes through an IP access list and it does not match on the Layer 2 packet.
- Configure these additional features, depending upon the type of switch:
  - Enable MPLS Tag stripping.
  - Set VLAN ID on Cisco Nexus 3000 Series switches.
  - Symmetric load balancing on Cisco Nexus 3100 Series switches and Cisco Nexus 9000 Series switches.
  - Q-in-Q on Cisco Nexus 3000 Series switches, 3100 Series switches, and Cisco Nexus 9000 Series switches.
  - Timestamp tagging and packet truncation on Cisco Nexus 3500 Series switches.
  - You can now configure the **watchdog_timer** configuration parameter in the **config.ini** file. If the value of the parameter is set to 0, the watchdog timer functionality is not available. The value of 30 seconds is a minimum value of the parameter and if the value of the parameter is set to a value more the 30 seconds, the watchdog timer monitors the JAVA process for the configured time interval.

Supported Web Browsers

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

- Firefox 45.x and later versions
- Chrome 45.x and later versions
- Internet Explorer 11 and later versions
- Microsoft Edge 42 or 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.
- Verify that the management interface of the switch (mgmt0) has an IP address configured using the **show running-config interface mgmt0** command.
- Ensure that the switch is in Multiple Spanning Tree (MST) mode. You can use **spanning-tree mode mst** command to enable MST mode on a switch.
- 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 VLAN range is <1-3967>.
- Ensure that the spanning tree protocol is disabled for all the VLANs. You can use the **no spanning-tree vlan 1-3967** to disable spanning tree on all the VLANs.
- For the first NDB deployment with NXOS version 9.2(1), ensure that the **feature nxapi** and **nxapi http port 80** commands are configured on the NDB switch. If you upgrading NDB switch from NXOS version I7(x) to 9.2(1), the **feature nxapi** and **nxapi http port 80** configurations are not required.

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

Note The hardware command that is a pre-requisite for the IPv6 feature is **hardware access-list tcam region ipv6-ifacl 512 double-wide**.

Note The TCAM configurations are based on the type of filters required. You may configure multiple TCAM entries from a specific region based on the network requirement. For example, *ing-ifacl* is the TCAM region to cater MAC, IPv4, IPv6 filters in case of N93180YC-E. You may configure multiple TCAM from this region to fit more filtering ACL TCAM entries.

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 3000 Series switches	Enter the **# hardware profile openflow** command at the prompt.	

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 3164Q, 3132Q switches	Enter the **# hardware profile openflow** command at the prompt. **Note** 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** • **#feature nxapi** • **#feature lldp**
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.

Device Models	OpenFlow Mode	NX-API Mode
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** • **# hardware access-list tcam region ipv6-ifacl 256** • **#feature nxapi** • **#feature lldp**
Cisco Nexus 3500 series switches	Enter either of the following commands at the prompt to configure OpenFlow TCAM: • **# hardware profile forwarding-mode openflow-hybrid** • **#hardware profile forwarding-mode openflow-only**	

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 IPv4 and IPv6, enter the **hardware access-list tcam region openflow 512** command. **Note** IPv6 and IPv4 dual stack is not supported in I6 and I7.	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** • **# hardware access-list tcam region ipv6-ifacl 512** • **#feature nxapi** • **#feature lldp**

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 9200, 9300-EX, 9336C-FX2, and 93240YC-FX2 switches	The OpenFlow mode is not supported on the 9200, 9300-EX, 9336C-FX2, and 93240YC-FX2 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 0** • **hardware access-list tcam region ing-l3-vlan-qos 0** • **# hardware access-list tcam region egr-racl 0** • **# hardware access-list tcam region ing-ifacl 1024** • **#feature nxapi** • **#feature lldp**
Cisco Nexus 9500-EX and 9500-FX Series switches	The OpenFlow mode is not supported on the Cisco Nexus 9500-EX and 9500-FX Series switches.	Enter the following commands at the prompt: • **# hardware access-list tcam region ing-racl 0** • **# hardware access-list tcam region ing-l3-vlan-qos 0** • **# hardware access-list tcam region egr-racl 0** • **# hardware access-list tcam region ing-ifacl 1024** • **#feature nxapi** • **#hardware acl tap-agg** • **#feature lldp**

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)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	NXAPI	`ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip`
Embedded	7.0(3)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	OpenFlow	`ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip`
Embedded	7.0(3)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	NXAPI	`ndb1000-sw-app-emb-nxapi-3.9.0-k9.zip`

Mode of Deployment	NXOS Image	Mode	File Name
Embedded	7.0(3)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	Openflow	ndb1000-sw-app-emb-3.9.0-ofa_mmemb-2.1.4-r2-nxos-SPA-k9.zip
Embedded	7.0(3)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	Openflow	ndb1000-sw-app-emb-3.9.0-ofa_mmemb-1.1.5-r3-n3000-SPA-k9.zip
Centralized	7.0(3)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	NXAPI	ndb1000-sw-app-k9-3.9.0.zip
Centralized	7.0(3)I4(1) to 7.0(3)I4(9), 7.0(3)I6(1), 7.0(3)I7(2) to 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4)	OpenFlow	ndb1000-sw-app-k9-3.9.0.zip

Nexus Data Broker Hardware and Software Interoperability Matrix

See Cisco Nexus Data Broker Release Notes, Release 3.9 for the latest matrix.

Python Activator Scripts for NX-OS Images

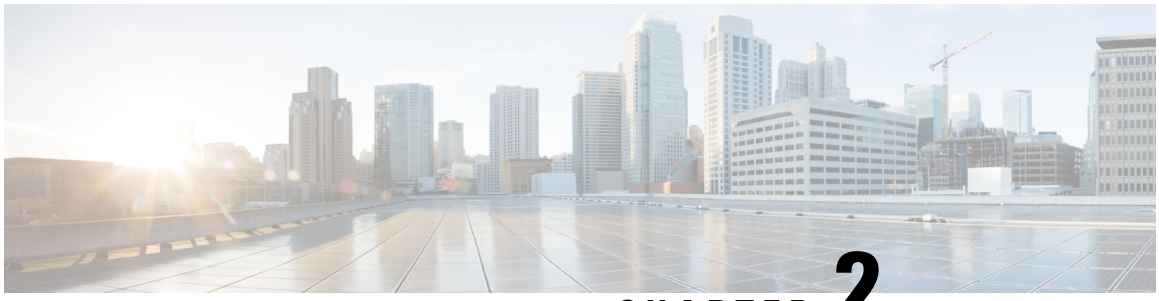
The following table lists the Python Activator scripts and corresponding NX-OS Image names:

Note The activator scripts are availble for download at: https://github.com/datacenter/nexus-data-broker.

Note Check the Guestshell version using the **show guestshell** command. If the Guestshell version is 2.2 or earlier, either upgrade the Guestshell or destroy and re-run the script to start NDB embedded.

Table 1: Python Activator Scripts for NX-OS Images

Python activator script file name	NX-OS Image
NDBActivator2.0_A6_A8_Plus.py	Cisco NXOS versions A6 and A8
NDBActivator2.0_I3_I4.py	Cisco NXOS versions I3 and I4
NDBActivator3.0_I5_Plus.py	Cisco NXOS version I5 and above.



CHAPTER 2

Deploying Cisco Nexus Data Broker Embedded for OpenFlow

This chapter contains details of procedures for deploying Cisco Nexus Data Broker on Cisco Nexus series switches.

Generating TLS certificate between NDB server and NDB switch for Embedded mode of deployment is not supported. For details about TLS, see the Managing TLS Certificate, KeyStore and TrustStore Files chapter in the *Cisco Nexus Data Broker Configuration Guide.*

The sections in this chapter are:

Obtaining the Cisco Nexus Data Broker Embedded Software for OpenFlow

To obtain the Cisco Nexus Data Broker Embedded Software for OpenFlow, complete the following steps:

Attention Starting with Cisco NXOS Release 7.0(3)I5(1), Openflow is supported in Embedded deployment.

Note This procedure is applicable for the following Cisco Nexus 3000, Nexus 9000 platforms:

- Cisco Nexus 9000 Series switches (excluding EX, FX, and 9500 models) running NXOS version I4 or lower.
- Cisco Nexus 3000 and 3100 Series switches running NXOS version I4 or lower.
- Cisco Nexus 3548 switches running version 6.0(2)A6 or 6.0(2)A8.

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 3.9** application bundle zip file. For more information regarding the NDB zip file name, see Cisco Nexus Data Broker Software Release Filename Matrix.

Step 7 Download the activator script. The Python activator script needed to activate the NDB is available at: https://github.com/datacenter/nexus-data-broker. For more information regarding the Python activator script file name, see Cisco Nexus Data Broker Software Release Filename Matrix.

Step 8 Unzip the application bundle to extract the two files:

- ndb1000-sw-app-emb-k9-3.9.0.ova – NDB application OVA
- ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova (N9K switches) OR ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova (N3K and N3548 switches) – Openflow Plugin OVA

Step 9 Copy the OpenFlow Plugin OVA to bootflash of the switch.

Example:

copy ftp://user@10.10.10.1/download_dir/ndb1000-sw-app-emb-k9-3.9.0.ova bootflash: vrf management

Step 10 Copy the NDB application OVA to bootflash of the switch.

Example:

copy ftp://user@10.10.10.1/download_dir/ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova bootflash: vrf management

Example:

copy ftp://user@10.10.10.1/download_dir/ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova bootflash: vrf management

Step 11 Copy the Python activator script to switch.

Example:

copy ftp://user@10.10.10.1/download_dir/NDBActivator2.0_A6_A8_Plus.py bootflash:NDBActivator.py vrf management

Example:

copy ftp://user@10.10.10.1/download_dir/NDBActivator2.0_I3_I4.py bootflash:NDBActivator.py vrf management

What to do next

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

Enable Auxiliary mode for Openflow

This task has details about enabling AUX mode in openflow for embedded and centralized modes.

Step 1 Navigate to **Administration** > **Devices** > **Device Connections**.

Step 2 Click **Add Device**.

Provide the device, connection and port details.

Step 3 Check the **Set Auxiliary Node** check box.

Note If the device has already been discovered by Openflow, the **Set Auxiliary Node** checkbox is enabled by default.

Step 4 Click **Add Device** to enable the selected device in auxiliary mode.

Configure these commands in the switch for NX-AUX support:

```
feature nxapi
nxapi http port 80
nxapi use-vrf management
```

The **nxapi use-vrf management** command is supported from NX-OS Release 7.x onwards.

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

To install and activate Cisco Nexus Data Broker (NDB) for OpenFlow, you need two OVA files for deployment, one file for OpenFlow plugin and other for NDB application.

OpenFlow agent in the switch cannot be installed as a virtual service. The OpenFlow support is made natively available on the switch. You can configure OpenFlow on a switch using the **feature openflow** command. For detailed information about OpenFlow switch configuration, see *Configuring the Cisco Plug-in for OpenFlow*.

Note Cisco Nexus 93xx EX/FX platforms do not support OpenFlow as a virtual service or as a native OpenFlow feature.

Note Download Cisco NDB Application bundle and the activator scripts as mentioned in section Obtaining the Cisco Nexus Data Broker Embedded Software for OpenFlow.

Note This procedure is applicable for the following Cisco Nexus 3000 and Nexus 9000 platforms:

- Cisco Nexus 9000 Series switches (excluding EX, FX, and 9500 models) running NXOS version I4 or lower.
- Cisco Nexus 3000 and 3100 Series switches running NXOS version I4 or lower.
- Cisco Nexus 3548 switches running version 6.0(2)A6 or 6.0(2)A8.

Note To add a device to NDB in embedded mode, you need to use the device IP address. Currently, NDB embedded mode does not support hostname for adding a device.

Note Starting with Cisco NXOS Release 7.0(3)I5(1), OpenFlow agent in the switch is not installed as a virtual service. OpenFlow agent is built into the switch OS and you can configure OpenFlow on a switch by using the **feature openflow** command.

```
switch#
conf t
feature openflow
```

Note OpenFlow switch configurations are mandatory on the switch. For detailed information about OpenFlow switch configuration, see Configuring the Cisco Plug-in for OpenFlow.

Note Cisco Nexus 93xx series EX and Nexus 93xx series FX platforms do not support OpenFlow as a virtual service or as a native OpenFlow feature.

Before you begin

Configure the pre-requisite profile, **hardware profile openflow** for Cisco Nexus 3000, 3100, 3200, and Nexus 9000 Series switches.

```
switch# configure terminal
switch (config)# hardware profile openflow
switch (config)# spanning-tree mode mst
switch (config)# vlan 1-3967
switch (config)# no spanning-tree vlan 1-3967
switch (config)# int <int> - <int>  Enter the interface ranges for use in openflow
switch (config-if-range) swithcport
switch (config-if-range) switchport mode trunk
switch (config-if-range) exit
switch (config)# copy running-config startup-config
switch (config)# reload
```

For Cisco Nexus 3500 Series switches, configure the pre-requisite profile **hardware profile forwarding-mode openflow-hybrid** for the Cisco Nexus 3500 Series switches.

```
device-3548# configure terminal
device-3548(config)# hardware profile forwarding-mode openflow-hybrid
device-3548(config)# spanning-tree mode mst
device-3548(config)# vlan 1-3967
device-3548(config)# no spanning-tree vlan 1-3967
device-3548(config)# int <int> - <int>  Enter the interface ranges for use in openflow
device-3548(config-if-range) swithcport
device-3548(config-if-range) switchport mode trunk
device-3548(config-if-range) exit
device-3548(config)# copy running-config startup-config
device-3548(config)# reload
```

Ensure that the OpenFlow switch configurations are complete, for detailed information about OpenFlow switch configuration, see Configuring the Cisco Plug-in for OpenFlow.

Step 1 Verify that NDB application and OpenFlow plugin is not installed and activated. If already installed, please use the upgrade procedure.

Example:

```
switch# show virtual-service list
```

Step 2 Install OpenFlow plugin OVA, with service name **ofa**.

Example:

```
switch# virtual-service install name ofa bootflash:ofa_mmemb-2.1.4-r2-nxos-SPA-k9.ova
#    (for N9K switches)
```

Example:

```
switch# virtual-service install name ofa package bootflash:ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova
# (for N3K and N3548 switches)
```

Step 3 Install Cisco NDB Application OVA, with service name **ndb**.

Example:

```
switch# virtual-service install name ndb package bootflash:ndb1000-sw-app-emb-k9-3.9.0.ova
```

Step 4 Monitor the installation status.

Example:

```
switch# show virtual-service list
```

Note Do not continue until both OVA files have been successfully installed.

Step 5 Enter the global configuration mode on the switch.

Example:

```
switch# configure terminal
```

Step 6 Enter virtual service configuration mode for OpenFlow plugin **ofa** on the switch

Example:

```
switch (config)# virtual-service ofa
```

Step 7 Activate the Cisco Plug-in for OpenFlow OVA.

Example:

```
switch(config-virt-serv)# activate
```

Step 8 Return to global configuration mode.

Example:

```
switch(config-virt-serv)# exit
```

Step 9 Enter virtual service configuration mode for NDB Application **ndb** on the switch.

Example:

```
switch(config)# virtual-service ndb
```

Step 10 Activate the Cisco Nexus Data Broker Embedded application OVA.

Example:

```
switch(config-virt-serv)# activate
```

Step 11 Exit virtual service configuration mode on the switch.

Example:

```
switch(config-virt-serv)# exit
```

Step 12 Monitor the service status.

Example:

```
switch# show virtual-service list
```

Example:

```
N9k(config-virt-serv)# show virtual-service list
Virtual Service List:
Name Status Package Name
------------------------------------------------------------------------
ndb Activated ndb1000-sw-app-emb-k9-3.9.0.ova
ofa Activated ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova
```

Step 13 Run the NDB activator script to setup management interface file and to activate the NDB.

Note On a Cisco Nexus 3548 switch, a user with username/password set to admin/admin and with *admin* privileges can only activate a NDB application.

Example for installing NDB on Cisco Nexus 3548 switch.

Example:

```
device-3548# run bash
bash-3.9$ sudo su
Password: #(Enter the password for the "admin" user when prompted).
bash-3.9# cd /bootflash/
bash-3.9# python NDBActivator.py -v ndb
2017-05zx-27 09:08:05,923 - __main__ - INFO - Successfully created /embndb/interface file with
management interface details
bash-3.9#
```

Example for installing NDB on Cisco Nexus 3000, Nexus 3100 and Nexus 9000 switches.

Example:

```
switch# python bootflash:NDBActivator.py -v ndb
```

Step 14 Deactivate the NDB virtual service and activate it for the activator configuration changes to apply.

Example:

```
device# configure terminal
device(config)# virtual-service ndb
device(config)# no activate
device(config)# show virtual-service list (repeat until status shows deactivated)
device(config)# activate
device(config)# show virtual-service list (repeat until status shows activated)
device(config)# end
device(config)# copy running-config startup-config
```

Installing and Activating the Cisco Nexus Data Broker Embedded Software for OpenFlow Mode for NXOS Versions in 7.0(3)I7(2) or Later

Cisco NDB is not installed as OVA, it is installed in the Guestshell. You can now install NDB directly on a device in embedded mode on NXOS 7.0(3)I7(2) or later release on the Guestshell. To install Cisco Nexus Data Broker Embedded software on NXOS 7.0(3)I7(2) or later release, use the NDB activator script, NDBActivator2.0_I5_Plus.py. After you install NDB, you need to enable it. To enable NDB, use the **guestshell enable** command. To disable NDB, use **guestshell disable** command.

The activator script performs the following functions:

- Resizes the Guestshell resources.
- Unzips and places the XNC folder into the Guestshell home directory.
- Configures the Guestshell to management VRF.

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 forcefully run the activator script even if it is already activated using **python bootflash** command with **--force** attribute. For example:

```
Syntax: python <file path>NDBActivato3.0_I5_Plus.py -v guestshell+ <zip file path> --force

Example:  python bootflash:NDBActivator3.0_I5_Plus.py -v guestshell+
/bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.zip -force
```

Note If you are using Cisco NXOS version I4.7 in embedded mode, you need to uninstall the NDB OVA and OpenFlow agent OVA and then upgrade the device to NXOS version I7.2.

Note Before you begin installing a new version of the Cisco Nexus Data Broker in Embedded mode, you must download and copy NDBActivator3.0_I5_Plus.py and ndb1000-sw-app-emb-i6-plus-k9-3.9.zip files to device.

Note If an instance of NDB exists in the guestshell and you need to install a new NDB instance, you need to destroy the existing Guestshell and re-install the Guestshell and the NDB. To uninstall NDB application, destroy the Guestshell using the command, **guestshell destroy**.

```
N9K-switch# guestshell destroy
```

Important Ensure that you have sufficient space available in the bootflash. The **ndb1000-sw-app-emb-i6-plus-k9-3.9.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 7.0(3)I5(1) or later, with memory greater than 8GB.

Step 1 switch# **copy** ftp://10.10.10.1 **NDBActivator3.0_I5_Plus.py bootflash: vrf management**

Copies the NDBActivator3.0_I5_Plus.py from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Step 2 switch# **copy** ftp://10.10.10.1 **ndb1000-sw-app-emb-i6-plus-k9-3.9.zip bootflash: vrf management**

Copies the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Step 3 switch# **show virtual-service list**

Monitors the status of the copy processes.

Step 4 switch# **guestshell enable**

Enables the guestshell.

Step 5 switch# **python bootflash:NDBActivator3.0_I5_Plus.py -v guestshell+ /bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.zip**

Installs the Cisco Nexus Data Broker Embedded package on the switch.

Step 6 switch# **show virtual-service list**

Monitors the status of the installations.

Note Do not continue until both the OVA files are installed succesfully.

Step 7 switch# **show processes cpu sort | grep java**

Example:

switch# show processes cpu sort | grep java

19587 3 6 551 0.00% java

switch#

Verify whether NDB installed and initiated successfully. Stop the NDB using **guestshell disable** command and then enable it using the **guestshell enable** command.

Step 8 switch# **guestshell disable**

Disables the NDB.

Step 9 switch# **guestshell enable**

Enables the NDB.

Step 10 Log in to the NDB application using the credentials.

Step 11 Configure the OpenFlow switch, see Configuring the Cisco Plug-in for OpenFlow.

Installing, Activating and upgrading the Cisco Nexus Data Broker for OpenFlow Mode for NXOS Version 7.0(3)I7(2) or Later on Low Memory Devices

Complete the following steps to upgrade Nexus 3K Switches (Nexus 3548P and 3548X) with low memory (less than 4 Gb). This procedure is applicable when you are upgrading devices with NXOS version 6.0(2)A6(x) and 6.0(2)A8(x) to NXOS version 7.0(3)I7(X) or 9.2(x).

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 forcefully run the activator script even if it is already activated using **python bootflash** command with **--force** attribute. For example:

```
Syntax: python <file path>NDBActivato3.0_I5_Plus.py -v guestshell+ <zip file path> --force

Example:  python bootflash:NDBActivator3.0_I5_Plus.py -v guestshell+
/bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip -force
```

Note If you are using Cisco NXOS version I4.7 in embedded mode, you need to uninstall the NDB OVA and OpenFlow agent OVA and then upgrade the device to NXOS version I7.2.

Step 1 Log into the current NDB instance.

Step 2 Navigate to **ADMNINISTRATION** > **Devices** > **NODES LEARNED** tab.

Step 3 Note the Node ID (also known as Data Path ID (DPID)) value for the switch listed on the **NODES LEARNED** tab.

Step 4 Click **Save** to save the NDB configuration for NDB release 3.1 and earlier.

Note Ensure that you save the NDB configuration because configuration in NDB releases 3.1 and earlier are not saved automatically.

Step 5 Navigate to **ADMINSTRATION** > **System**.

Step 6 Click **Download Configuration** to download the current configuration on the NDB.

Step 7 Remove the OVA files (ndb and ofa virtual services) from the switch. For example:

Example:

```
N3K-130(config)# conf t
N3K-130(config)# virtual-service ndb
N3K-130(config-virt-serv)# no activate
N3K-130(config-virt-serv)# sh virtual-service list
Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ndb                     Deactivating       ndb1000-sw-app-emb-k9-3.9.0.ova
ofa                     Activated          ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova

N3K-130(config-virt-serv)# sh virtual-service list

Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ndb                     Deactivated        ndb1000-sw-app-emb-k9-3.9.0.ova
ofa                     Activated          ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova

N3K-130(config-virt-serv)# no virtual-service ndb
N3K-130(config)# ex
N3K-130# virtual-service uninstall name ndb
N3K-130# sh virtual-service list

Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ofa                     Activated          ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova

N3K-130# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
N3K-130(config)# virtual-service ofa
N3K-130(config-virt-serv)# no activate
N3K-130(config-virt-serv)# sh virtual-service list

Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ofa                     Deactivating       ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova

N3K-130(config-virt-serv)# sh virtual-service list

Virtual Service List:

Name                    Status             Package Name
------------------------------------------------------------------------
ofa                     Deactivating       ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova

N3K-130(config-virt-serv)# sh virtual-service list

Virtual Service List:
```

```
Name                    Status             Package Name
------------------------------------------------------------------------
ofa                     Deactivated        ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova

N3K-130(config-virt-serv)# no virtual-service ofa
N3K-130(config)# ex
N3K-130# virtual-service uninstall name ofa
N3K-130#
N3K-130# sh virtual-service list

Virtual Service List:
```

Step 8 Copy the latest NXOS image to your local server and use the compact option to generate a compact NXOS image while copying the image to the switch bootflash. For example:

Important Starting with Cisco NXOS Release 7.0(3)I5(2), you can compact the image while copying the image to the switch bootflash or USB drive using compact option in the **copy** command. Download and keep the non-compact image in any server. The compact option in the **copy** command overrides the bootflash space limitation as the image is compacted at the time of transferring the image to the switch bootflash or USB drive.

```
Syntax: switch# copy scp :source image name bootflash :image name compact vrf vrf name

copy scp://vm-ubuntu-1604@10.16.206.136//home/vm-ubuntu-1604/praba/nxos_img/nxos.7.0.3.I7.5A.bin
bootflash: compact vrf management
```

Note The compact option with **copy** command is allowed only with SCP protocol.

Step 9 Upgrade the switch to the downloaded compact image. Refer to Cisco NX-OS Upgrade guide.

Step 10 Download the guestshell.ova from NX-OS software download page. If the guestshell.ova is not available on the download page, you need to extract it from the downloaded NXOS image. Refer to Extracting Guestshell section for more information.

Step 11 Install the guestshell on the low memory switch using the guestshell enable package command.

```
N3K-130# guestshell enable package bootflash:guestshell.ova
N3K-130#
N3K-130# show virtual-service list
Virtual Service List:
Name                    Status             Package Name
------------------------------------------------------------------------
guestshell+             Activated          guestshell.ova
```

Step 12 Install and activate the NDB for OpenFlow mode . For detailed information, see Installing and Activating the Cisco Nexus Data Broker Embedded Software for OpenFlow Mode for NXOS Versions in 7.0(3)I7(2) or Later for more information.

Step 13 Configure the Cisco Plugin. For more information, see the Configuring the Cisco Plug-in for OpenFlow section.

Step 14 Use the show openflow switch command to check the current DPID in the switch after upgrading to Cisco NX-OS Release I7(5A)

```
N3K-130# show openflow switch 1 | grep DPID
                DPID: 0x0001a44c116a7620
```

Note If DPID is different from the DPID in Step 1, you need to configure DPID described in the next step.

Step 15 Enable the Openflow feature and add controller configuration along with DPID value to the switch.

```
N3K-130(config)# feature openflow
       N3K-130(config)# openflow
```

```
      N3K-130(config-ofa)# switch 1 pipeline 203
N3K-130(config-ofa-switch)# controller ipv4 10.16.206.130 port 6653 vrf management security none
N3K-130(config-ofa-switch)# datapath-id ?
 <0-18446744073709551615>  64-bit hex value [0x1-0xffffffffffffffff]
              N3K-130(config-ofa-switch)# datapath-id 0x001a44c116a7620
N3K-130(config-ofa-switch)# of-port interface ethernet 1/1-48
```

Note DataPath ID should be the same as mentioned in the device currently, after the NXOS upgrade. Any change in the DataPath ID will result in the upgrade failure.

Extracting Guestshell

Complete these steps to extract the guestshell from the downloaded NX-OS image.

Before you begin

Ensure that guestshell extraction is performed on a switch with more than 4 Gb disk space and 1 Gb of memory. Load the regular non-compact I7(5) image the switch.

Step 1 Log into a switch with non-compact image installed.

Step 2 Use the **feature bash-shell** command to enable the bash-shell feature.

```
N3548X(config)# feature bash-shell
```

Step 3 Use the **run bash** command to enter the bash-shell mode.

```
N3548X# run bash
bash-4.2$
```

Step 4 Use the **sudo su** command to switch to super user.

```
bash-4.2$ sudo su
```

Step 5 Set the following environment variables where the guest shell ova is to be stored.

```
root@N3548X#TARGET="isanboot/bin/guestshell.ova"
```

Step 6 Set the system image environment variable from where the guest shell needs to be extracted.

```
root@N3548X#SYSIMG="/bootflash/nxos.7.0.3.I7.5_full.bin"
```

Step 7 Use the **isanboot** command to extract the guest shell ova from the specified NXOS image.

Example:

```
root@N3548X#/isanboot/bin/x_nbi_seg $SYSIMG stdout 3 | /bin/zcat | /bin/cpio -ivd $TARGET
```

Step 8 Exit the super user mode using the **exit** command.

```
root@N3548X#exit
logout
```

Step 9 Exit the bash-shell mode using the **exit** command.

```
bash-4.2$ exit
```

Step 10 Verify the extracted files using the **dir boot flash** command.

```
N3548X# dir bootflash:///isanboot/bin
   50309120    Jun 25 21:45:57 2001  guestshell.ova  <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
```

Step 11 Use the **run bash** command to enter the bash-shell mode.

```
N3548X# run bash
bash-4.2$
```

Step 12 Use the **copy** command to copy the guestshell.ova to the switch boot flash.

```
bash-4.2$ cp guestshell.ova /bootflash/
bash-4.2$
```

Step 13 Change the current directory to boot flash directory.

```
bash-4.2$ cd /bootflash/
```

Step 14 Exit the bash-shell mode using the **exit** command..

```
bash-4.2$ exit
```

Note After extracting the guestshell.ova file, you can delete both isanboot and guestshell.ova. You can also disable bash-shell feature on this switch

Upgrading to Release 3.8

This process involves using the GUI to download the configuration, perform the upgrade to Release 3.8 (or later), and then upload the configuration.

Note For detailed information about upgrading Cisco Nexus OS from I4(6) to I6(1), see *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 7.x*.

Note This procedure is applicable for the following Cisco Nexus 3000, Nexus 9000 platforms:

- Cisco Nexus 9000 Series switches (excluding EX, FX, and 9500 models) running NXOS version I4 or lower.
- Cisco Nexus 3000 and 3100 Series switches running NXOS version I4 or lower.
- Cisco Nexus 3548 switches running version 6.0(2)A6 or 6.0(2)A8.

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**.

OR

Navigate to the **System** tab under **Administration** > **Backup/Restore** tab. Click **Backup and Backup Locally** to download the configuration in zip file format.

Step 3 Download the configuration in Cisco NDB version 3.9.

Step 4 Deactivate and uninstall Cisco NDB. For example:

Example:

```
configure terminal
virtual-service ndb
no activate
no virtual-service ndb
end
virtual-service uninstall name ndb
copy running-config startup-config
```

Step 5 Install and activate Cisco NDB 3.9 using the following steps:

a) **virtual-service install name <virtual-services-name> package bootflash: ndb1000-sw-app-emb-k9-3.9.0.ova**

b) **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.

c) **configure terminal**

d) device-3548(config)# **virtual-service <virtual-services-name>**

e) device-3548(config)# **activate**

f) device-3548(config)# **end**

g) device-3548(config)# **copy running-config startup-config**

Note Verify that the OpenFlow plugin is present in the **show virtual-service** list. If it is present, continue with the next step. If it is not listed, follow Step 2 in Installing and Activating the Cisco Nexus Data Broker Embedded Software for OpenFlow.

Step 6 Run the **<python activator script>** in the bash shell **python bootflash:<python activator script> -v <ndbvirtual service name>** command. The script deactivates and activates the NDB OVA automatically. If the NDB OVA is not activated automatically, you need to activate is manually.

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:

Note Starting with Cisco NXOS, Release 7.0(3)I5(1), only NXAPI embedded is supported, OpenFlow embedded is not supported.

Note Cisco Nexus 3548 switches, running NXOS A6/A8 version, need **root** user privileges to run the activator script. Ensure that activator script is downloaded and copied to bootflash with the name NDBActivator.py as mentioned in the previous section.

```
device-3548# run bash
bash-3.9$ sudo su
Password:    #(Enter the password for the "admin" user).
bash-3.9# cd /bootflash/
bash-3.9# python NDBActivator.py -v ndb
2017-05zx-27 09:08:05,923 - __main__ - INFO - Successfully created /embndb/interface file with
```

```
management interface details
bash-3.9#
```

Example:

```
device# configure terminal
device-3548(config)# virtual-service <virtual-services-name>
device-3548(config)# no activate
device-3548(config)# show virtual-service list (Wait until deactivated complete)
device-3548(config)# activate
device-3548(config)# show virtual-service list (Wait until activated complete)
device-3548(config)# end
device-3548(config)# copy running-config startup-config
```

Step 7 Upload Cisco NDB configuration that you downloaded earlier in the Cisco NDB User Interface (UI). Navigate to **Administration** > **System** > **BACKUP/RESTORE** tab, click **Restore locally** and follow the prompts to upload the configuration zip file.

Upgrading to Release 3.8 for Cisco NXOS Releases 7.0(3)I7(2) or Later

This process involves downloading, upgrading (to Release 3.8 and later), and finally uploading the configuration using the GUI.

Note For detailed information about upgrading Cisco Nexus OS from I4(6) to I6(1), see *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 7.x*.

Note While upgrading NDB embedded, after configuration upload from NDB, wait for at least 60 seconds before restarting the guestshell.

Step 1 For embedded NDB release 3.3 or earlier, navigate to the **System** tab under **Administration** and click **System Administration** to open the **System Administration** window. Click **Download Configuration**. For embedded NDB release 3.6 or later, navigate to the **System** tab under **Administration** -> **Backup/Restore** tab. Click **Backup and Backup locally**.

It downloads the configuration in a zip file format. The name of the zip file is **configuration_startup.zip**

Note For Cisco NXOS I4 or earlier versions, uninstall both the NDB OVA and Embedded OVA and then upgrade the NXOS version to I7.2.

Note For Cisco NXOS I5 and later versions, disable and destroy the guestshell and then upgrade the NXOS version to I7.2.

Step 2 Copy the NDBActivator3.0_I5_Plus.py from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Example:

switch# copy scp://10.10.10.1 NDBActivator3.0_I5_Plus.py bootflash:vrf management

Step 3 Copy the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Example:

```
switch# copy scp://10.10.10.1 ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip bootflash:vrf management
```

Step 4 Monitor the status of the copy processes.

Example:

switch# show virtual-service list

Step 5 Enable the guestshell.

Example:

switch# guestshell enable

Step 6 Install the Cisco Nexus Data Broker Embedded package on the switch.

Example:

switch# python bootflash:NDBActivator3.0_I5_Plus.py -v guestshell+ /bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip

Step 7 Monitor the status of the installations.

Example:

switch# show virtual-service list

Note Do not continue until installation completes successfully. NDB application starts after it is installed successfully. Auto discovery for embedded Nexus devices is not available, you need to manually add a device using the Web GUI.

Step 8 Verify whether NDB installed and initiated successfully.

Example:

```
switch# show processes cpu sort | grep java
 19587 3 6 551 0.00% java
```

Step 9 Log in to the NDB application using the credentials.

Step 10 Upload Cisco NDB 3.x configuration that you downloaded in step 1 in the Cisco NDB user interface (UI). After uploading you need to disable and enable the guestshell to apply the uploaded configuration.

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 This procedure is applicable if you are installing the Cisco NDB with Openflow for the first time.

Note The flow can have only up to 16 output actions.

Before you begin

Configure the pre-requisite profile, **hardware profile openflow** for the Cisco Nexus 3000, 3100, 3200, and hardware access-list tcam region openflow <tcam> double-wide (optional) for the Nexus 9000 Series switches. For Cisco Nexus 3500 Series switches, configure the pre-requisite profile **hardware profile forwarding-mode openflow-hybrid** for the Cisco Nexus 3500 Series switches. Use the **copy running-config startup-config** to save the configuration and reload the switch.

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.

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 Switch This is the default value. Only expert users should set the *pipeline_num* number to any value other than 203.

Starting with NXOS version I6(1), you must configure the switch and pipeline in a single command.

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:

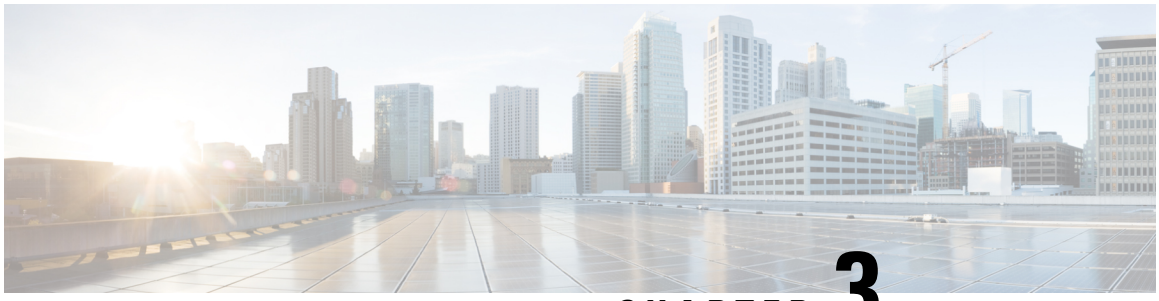
a) Enter your username and password.

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 details of procedures for deploying NDB for NX-API.

Before you proceed with the upgrade/ install procedures discussed in this chapter, compare the **md5sum** between the NDB CCO image and image file copied to linux. Use the following command to check (linux):

```
cisco@NDB-virtual-machine:~/3.9/$ md5sum ndb1000-sw-app-k9-3.9.0.zip
Displayed output: c2d273dce4abbba03c06aeee8774b901 ndb1000-sw-app-k9-3.9.0.zip
```

Generating TLS certificate between NDB server and NDB switch for Embedded mode of deployment is not supported. For details about TLS, see the *Managing TLS Certificate, KeyStore and TrustStore Files* chapter in the *Cisco Nexus Data Broker Configuration Guide.*

This chapter contains the following topics:

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 3.9** 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-k9-3.9.0.ova**

Step 6 Download activator script. The Python activator script needed to activate the NDB is available at: https://github.com/datacenter/nexus-data-broker. For more information regarding the Python activator script file name, see Python Activator Script Filename Matrix table in Cisco Nexus Data Broker Software Release Filename Matrix, on page 12.

What to do next

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

Installing and Activating the Cisco Nexus Data Broker Embedded Software for NX-API Mode for NXOS Versions in 7.0(3)I4(x)

Before you begin

Note For a TACACS user to start NDB in embedded mode, the user should be logged in to the switch with network administrator privileges.

Note To add a device to NDB in embedded mode, use the device IP address. Currently, NDB embedded mode does not support hostname for adding a device.

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.

Step 1 switch# **copy** [*scp:* | *ftp:* | *http:*] *//download_dir* **ndb1000-sw-app-emb-k9-3.9.0.ova bootflash:vrf management**

Copies the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

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-k9-3.9.0.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 the OVA file is successfully installed.

Step 5 switch# **configure terminal**

Enters global configuration mode on the switch.

Step 6 switch(config)# **virtual-service Name_Of_OFA_NDB_EMB_File**

Example:

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.

Step 10 **switch# python bootflash: Name_Of_python_NDB_Activator_Script -v ndb_emb** .

Example:

switch# python bootflash: NDBActivator3.0_I5_Plus.py -v ndb_emb

Creates `/embndb/interface` file with management interface details using the NDB Python activator script:

The script is available in the **ndb** directory in the GitHub repository at https://github.com/datacenter/nexus-data-broker.

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

- NDBActivator2.0_I3_I4.py: For Cisco NXOS versions 7.0(3)I3(x) and 7.0(3)I4(x).

Step 11 Deactivate the NDB virtual service and activate it.

Update the configuration changes.

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
```

Installing and Activating the Cisco Nexus Data Broker Embedded Software for NX-API Mode for NXOS 7.0(3)I6(1) or Later

Cisco NDB is not installed as OVA, it is installed in the Guestshell. The Guestshell is installed and activated in the NXOS 7.0(3)I6(1) and later releases. You can now install NDB directly on a device in embedded mode on NXOS 7.0(3)I6(1) or later release. To install Cisco Nexus Data Broker Embedded software on NXOS 7.0(3)I6(1) or later release, use the NDB activator script, NDBActivator3.0_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 Guestshell to management VRF.

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 forcefully run the activator script even if it is already activated using **python bootflash** command with **--force** attribute. For example:

```
Syntax: python <file path>NDBActivator3.0_I5_Plus.py –v guestshell+ <zip file path> --force

Example:  python bootflash:NDBActivator3.0_I5_Plus.py –v guestshell+
/bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip --force
```

Note To uninstall NDB application, destroy the Guestshell using the command, **guestshell destroy**. If an instance of NDB exists in the guestshell and you need to install a new NDB instance, you need to destroy the existing Guestshell and re-install the Guestshell and the NDB.

```
N9K-switch# guestshell destroy
```

Note After you disable and enable NXAPI mode, you should reconfigure **nxapi use-vrf management** command on the node.

Important Ensure that you have sufficient space available in the bootflash. The **ndb1000-sw-app-emb-i6-plus-k9-3.9.0.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 7.0(3)I6(1), with memory greater than 8GB.

Step 1 switch# **copy** ftp:/10.10.10.1 **NDBActivator3.0_I5_Plus.py bootflash:vrf management**

Copies the NDBActivator3.0_I5_Plus.py from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Step 2 switch# **copy** ftp://10.10.10.1 **ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip bootflash:vrf management**

Copies the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Step 3 switch# **show virtual-service list**

Monitors the status of the copy processes.

Step 4 switch# **guestshell enable**

Enables the guestshell.

Step 5 switch# **python bootflash:NDBActivator3.0_I5_Plus.py -v guestshell+ /bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip**

Installs the Cisco Nexus Data Broker Embedded package on the switch.

Step 6 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.

Note To stop the NDB application, use the **guestshell disable** command. Use **guestshell enable** command to enable NDB.

Note Do not continue until installation completes successfully. NDB application starts after it is installed successfully.

Step 7 switch# **show processes cpu sort | grep java**

Example:

switch# show processes cpu sort | grep java

19587 3 6 551 0.00% java

switch#

Verify whether NDB installed and initiated successfully.

Adding a Device

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

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 3.8 for Cisco NXOS Releases Upto 7.0(3)I4(x)

Upgrading the Cisco NDB to release 3.8 (or later) for Cisco NXOS releases upto 7.0(3)I4(x) involves using the GUI to download the configuration, performing the upgrade, and then uploading the configuration.

Note For detailed information about upgrading Cisco Nexus OS from I4(6) to I6(1) and I7(1), see *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 7.x*.

Step 1 Navigate to the **System** tab under **Administration**.

The **System Administration** window is displayed.

Step 2 Click **Download Configuration** to download 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.x release version.

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

a) configure terminal

Example:

```
device# configure terminal
```

b) virtual-service **virtual-services-name**

Example:

```
device(config)# virtual-service vsn1
```

c) no activate

Example:

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

d) no virtual-service **<virtual-services-name>**

Example:

```
device(config)# no virtual-service vsn1
```

e) **end**

Example:

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

f) virtual-service uninstall name **virtual-services-name**

Example:

```
# virtual-service uninstall name vsn1
```

Step 5 Install and activate Cisco NDB 3.9 using the following steps:

a) **virtual-service install name <virtual-services-name> package bootflash: NDB_OVA**

```
device# virtual-service install name vsn1 package bootflash: ndb1000-sw-app-emb-k9-3.9.0.ova
```

b) **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.

```
device# show virtual-service list
```

c) **configure terminal**

```
device(config)# conf terminal
```

d) **virtual-service virtual-service-name**

```
device(config)# virtual-service vsn1
```

e) **activate**

```
device(config)# activate
```

f) **end**

```
device(config)# end
```

g) **copy running-config startup-config**

```
device(config)# copy running-config startup-config
```

Step 6 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_I3_I4.py: For Cisco NXOS versions I3 and I4.

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

repro-9372-2# python bootflash:NDBActivator2.0_I3_I4_working.py -vndb_emb
```

Step 7 Upload Cisco NDB 3.9 configuration that you downloaded in Step 1 in the Cisco NDB user interface (UI) Navigate to **Administration** -> **System** -> **Backup Restore** -> **Restore Locally**.

Upgrading to Release 3.8 for Cisco NXOS Release 7.0(3)I6(1) or Later

This process involves using the GUI to download the configuration, perform the upgrade, and then upload the configuration. The process is applicable for Release 3.8 and later.

Note For detailed information about upgrading Cisco Nexus OS from I4(6) to I6(1) and I7(1), see *Cisco Nexus 9000 Series NX-OS Software Upgrade and Downgrade Guide, Release 7.x*.

Step 1 Navigate to the **System** tab under **Administration** in the running Embedded NDB 3.x or earlier version. Click **Download Configuration** to download the configuration in a zip file format.

OR

Navigate to the **System** tab under **Administration** > **Backup/Restore** tab. Click **Backup and Backup Locally**.

It downloads the configuration in a zip file format. The name of the zip file is configuration_startup.zip

Step 2 Copy the NDBActivator3.0_I5_Plus.py from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Example:

```
switch# copy scp://10.10.10.1 NDBActivator3.0_I5_Plus.py bootflash:vrf management
```

Step 3 Copy the Cisco Nexus Data Broker Embedded package from the directory where you downloaded it to the switch. You can download the file from different sources such as HTTP, FTP, or SSH.

Example:

```
switch# copy scp://10.10.10.1 ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip bootflash:vrf management
```

Step 4 Monitor the status of the copy processes.

Example:

```
switch# show virtual-service list
```

Step 5 Enable the guestshell.

Example:

```
switch# guestshell enable
```

Step 6 Install the Cisco Nexus Data Broker Embedded package on the switch.

Example:

```
switch# python bootflash:NDBActivator3.0_I5_Plus.py -v guestshell+
/bootflash/ndb1000-sw-app-emb-i6-plus-k9-3.9.0.zip
```

Step 7 Monitor the status of installation process.

Example:

```
switch# show virtual-service list
```

To stop the NDB application, use the guestshell disable command.

Note Do not continue until installation completes successfully. NDB application starts after it is installed successfully.

Step 8 Verify whether NDB installed and initiated successfully.

Example:

```
switch# show processes cpu sort | grep java
Example:
switch# show processes cpu sort | grep java
19587 3 6 551 0.00% java
```

Step 9 Log in to the NDB application using the credentials. You need to manually add a device in NDB.

Step 10 To add a new device, click **Administration**-> **DEVICE CONNECTIONS** Tab.

Step 11 Click **Add Devices**, the **Add Device** dialog box appears.

Step 12 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 13 Click **Add Device** in the **Add Device** dialog box to add the device with the provided credentials.

Step 14 Copy the running configuration to the startup configuration.

Example:

```
switch(config)# copy running-config startup-config
```

Step 15 Upload Cisco NDB 3.9 configuration that you downloaded in Step 1 in the Cisco NDB user interface (UI). Navigate to **Administration** > **System** > **Backup Restore** > **Restore Locally**.

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:

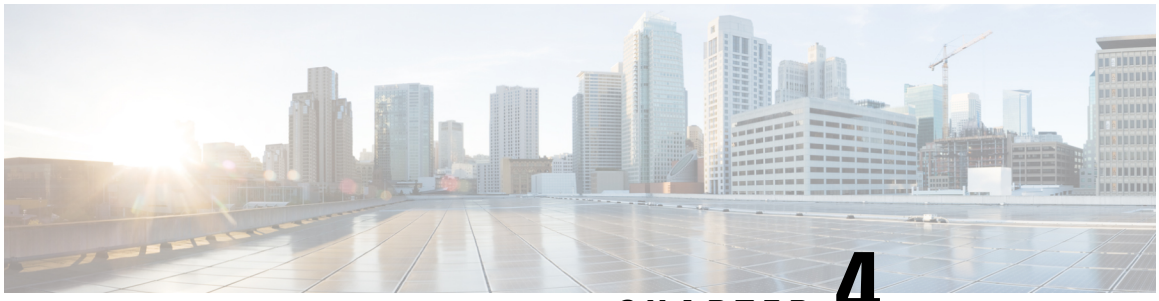
a) Enter your username and password.

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 details of procedures for installing and upgrading NDB in centralized mode.

Before your proceed with the upgrade/ install procedures in this chapter, compare the **md5sum** between the NDB CCO image and image file copied to linux. Use the following command to check (linux):

```
cisco@NDB-virtual-machine:~/3.9/$ md5sum ndb1000-sw-app-k9-3.9.0.zip
Displayed output: c2d273dce4abbba03c06aeee8774b901 ndb1000-sw-app-k9-3.9.0.zip
```

This chapter contains the following topics:

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

Note To add a device to NDB in centralized mode, use the device IP address or the device hostname.

- 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

Complete these steps to install 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 3.9 is displayed: Cisco Nexus Data Broker Software Application: ndb1000-sw-app-k9-3.9.0.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.
- `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.

Note To migrate from OVA-based Openflow to Native Openflow, see the Uninstalling Cisco Plug-in for OpenFlow chapter.

Upgrading the Application Software in Centralized Mode Using CLI

Use the **upgrade** command to upgrade to Cisco NDB Release 3.9.0.

Note

- Once you upgrade to Cisco NDB Release 3.9, you cannot use the downgrade option to rollback to a previous release. You have to use the configuration archive that is created during the upgrade process to rollback the software.
- When you upgrade the software to Cisco Nexus Data Broker Release 3.2 or later release, the hostname should not be changed during the upgrade process. If the hostname is changed during the upgrade process, the upgrade might fail. If you are upgrading from release 2.x, 3.0 and 3.1, the domain name configuration in the switch should be removed before upgrading the software.
- When you run the **upgrade** command, the installation and the configuration are upgraded. However, any changes you made to the shell scripts or configuration files, for example, `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.
- For NDB configuration upload or Backup/Restore process, first bring up the NDB instance where configuration is uploaded or where Backup/Restore is done, then start rest of the nodes in the cluster.
- Backup up the NDB configuration. For more information, see *Backing Up or Restoring the Configuration Using NDB GUI* section.
- 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` file.

Important You should manually backup your `config.ini` file 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, 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.

SUMMARY STEPS

1. In a web browser, navigate to Cisco.com.
2. Under **Support**, click **All Downloads**.
3. In the center pane, click **Cloud and Systems Management**.
4. In the right pane, click **Network Controllers and Applications**, and then click **Cisco Nexus Data Broker.**

5. Download the Cisco NDB Release 3.9 applicable bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.9.0.zip
6. Create a temporary directory in your Linux machine where you plan to upgrade to Cisco NDB.
7. Unzip the Cisco NDB Release 3.9 zip file into the temporary directory that you created.
8. Navigate to the `xnc` directory that was created when you installed the Cisco Nexus Data Broker release earlier.
9. Backup your Cisco Nexus Data Broker release installation using your standard backup procedures.
10. Stop running all Cisco Nexus Data Broker release processes.
11. Navigate to the `xnc/bin` directory in the temporary directory that you created for Cisco NDB Release 3.9 upgrade software.
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.
13. Navigate to the `xnc` directory where you originally installed Cisco XNC Monitor Manager.
14. If TLS certification is enabled between NDB server and NXOS switch, copy the tlsTrustStore and tlsKeyStore files to /xnc/configuration from the old xnc backup.
15. Start the application processes that you previously stopped.
16. If the secondary/cluster NDB server is configured, start the server.

DETAILED STEPS

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 NDB Release 3.9 applicable bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.9.0.zip

Step 6 Create a temporary directory in your Linux machine where you plan to upgrade to Cisco NDB.

Step 7 Unzip the Cisco NDB Release 3.9 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 Cisco NDB Release 3.9 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 intallation to Cisco NDB.

Option	Description
--perform --target-home {*xnc_directory_to_be_upgraded*} **--backupfile** {*xnc_backup_location_and_zip_filename*}	Upgrades the Cisco XNC Monitor Manager installation to Cisco NDB 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. • The backup file should not be saved in the xnc directory with current NDB installation or its subdirectory.
--verbose	Displays detailed information to the console. This option can be used with any other option and is disabled by default.
--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 If TLS certification is enabled between NDB server and NXOS switch, copy the tlsTrustStore and tlsKeyStore files to /xnc/configuration from the old xnc backup.

Step 15 Start the application processes that you previously stopped.

Note
- Clear the browser cache. Use Shift+Ctrl+Delete keys to clear the cache.
- Press Ctrl–F5, or press the Cmd, Shift, and R keys simultaneously when you access through a web UI following an upgrade.

Step 16 If the secondary/cluster NDB server is configured, start the server.

Note If TLS certification is enabled, start the secondary/cluster using the commands as shown below:

```
./runxnc.sh -tls -tlskeystore ./configuration/tlsKeyStore -tlstruststore
./configuration/tlsTrustStore
cd bin
./xnc config-keystore-passwords --user <NDB_username> --password <NDB_password> --url
https://<Cluster_NDB_IP>:8443 --verbose --prompt --keystore-password <keystore-password>
 --truststore-password <truststore-password>
```

Upgrading the Application Software in Centralized Mode Using GUI

Complete the following steps to upgrade the application software in the Centralized mode using GUI:

Step 1 Log into NDB.

Step 2 Navigate to the **System** tab under **Administration**.

The **System Administration** window is displayed.

Step 3 Click **Download Configuration** to download the switch configuration file in a .zip file format.

The default name of the zip file is **configuration_startup.zip**.

OR

Navigate to the **Backup/Restore** tab under **Administration** > **System** tab. Click **Backup and Backup Locally** to download the configuration in zip file format.

Step 4 Stop the current NDB instance using the **runxnc.sh -stop** command.

Example:

```
./runxnc.sh -stop
```

Step 5 If TLS certification is enabled between NDB server and NXOS switch, copy the tlsTrustStore and tlsKeyStore files to /xnc/configuration from the old xnc backup.

Step 6 Start the new NDB installation using the **runxnc.sh -start** command.

Example:

```
./runxnc.sh -start
```

Step 7 Navigate to the **Backup/Restore** tab under **Administration** > **System** tab.

Step 8 Click **Restore Locally** and upload the configuration_startup.zip

Step 9 Restart the new NDB instance using the **runxnc.sh -restart** command.

Example:

```
./runxnc.sh -restart
```

GUI Notifications during Install/ Upgrade

Beginning with Release 3.9.2, the GUI behavior has changed while installing or upgrading the NDB controller software. The GUI will be in *read-only* state until the whole installation or upgradation procedure is completed. You will see relevant messages at the top of the NDB GUI indicating the current background operation/ process/ event that is in progress. Wait for a *Ready* message to appear at the top of the GUI screen before you make any configuration changes. This change is behavior is to facilitate smooth install and upgrade as NDB is not stabilized while the install or upgrade is in progress. This is applicable to both the upgrades—HA and standalone.

Some of the messages that appear at the top of the screen indicating the completed events or background processes are:

-

For HA upgrade, when the Primary is ready, a small green tick-mark appears at the cluster information (see illustration, below); the corresponding message displayed at the top is, *Primary is Ready, bring up the members* . You can hover over to see the members of the cluster.

Figure 1: GUI enhancement - Primary is Ready Notification

For standalone, wait for the *NDB is Ready for Configurations* message to be displayed at the top of the screen to perform configurations.

The configuration buttons are either disabled, or are temporarily removed, until the installation / upgradation is complete. Some examples are provided here.

Under **Connections** > **User Connections**, the configuration buttons are temporarily removed.

Figure 2: GUI enhancement - Connections (without configuration buttons)

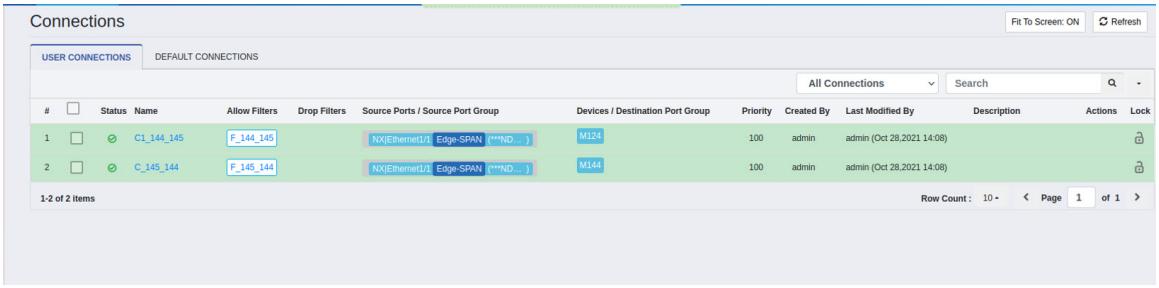
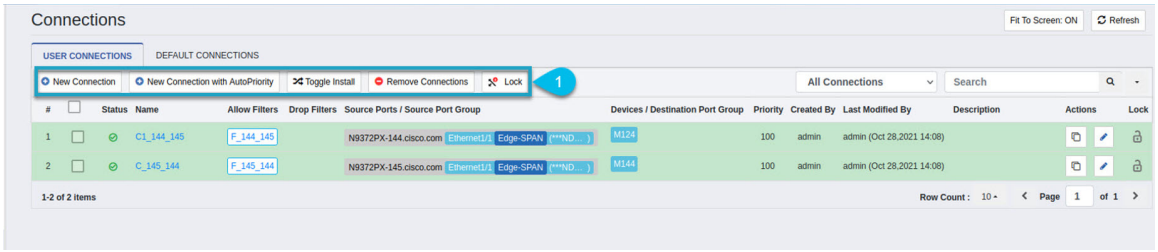


Figure 3: GUI enhancement - Connections (with configuration buttons)



Under **Devices** > **Device Conenctions**, the configuration buttons are temporarily disabled.

Figure 4: GUI enhancement - Devices (configuration buttons are disabled)

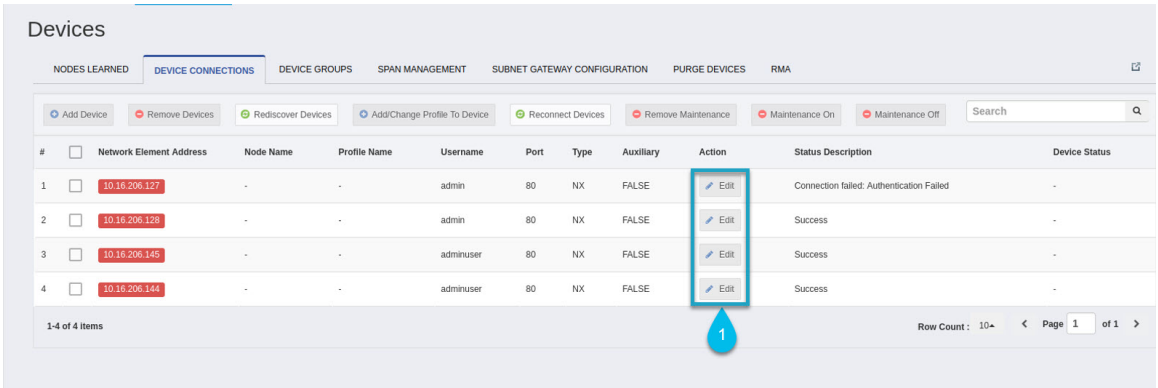
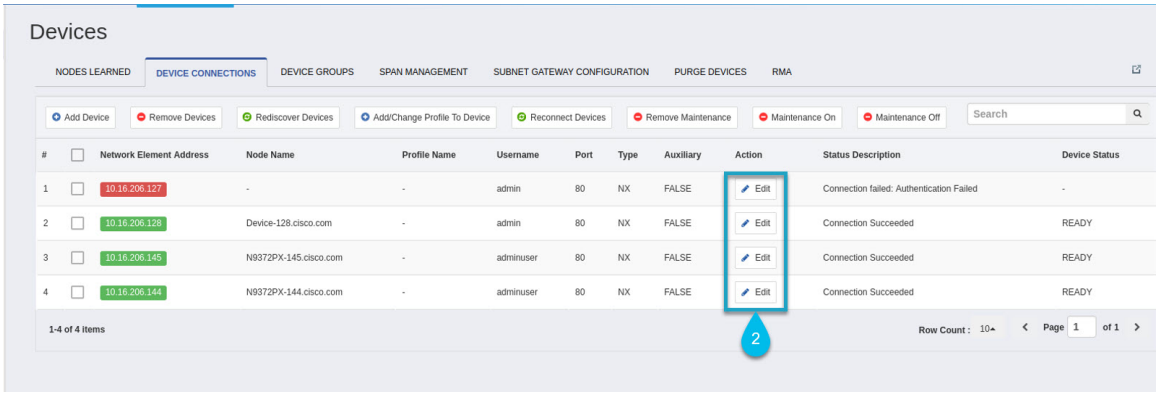


Figure 5: GUI enhancement - Devices (configuration buttons are enabled)



Upgrading NDB Using the Hitless Method

You can upgrade Cisco NDB using the either the upload or the CLI upgrade hitless methods.

Upgrading Cisco NDB - Hitless Method (Using Upload)

You can upgrade Cisco NDB to Release 3.9.0 with the hitless method using upload.

Before you begin

If the Cisco NDB version is earlier than Release 3.8, you must edit the config.ini file and update the **skipConfigurtionStateDBfiles** key to false on both the controllers, and restart all the earlier version controllers.

Step 1 Log into NDB.

Step 2 Navigate to the location (`/home/3.9.0/xnc`) of the xnc for Release 3.9 in both, server 1 and server 2.

Step 3 Navigate to the **System** tab under **Adminstration** to view the **System Administration** window.

Step 4 Navigate to **Administration** > **system** > **Backup/Restore** > **Backup** > **Backup now locally** to download the configuration in zip file format and save it on your local desk.

Note The server that is started first will become the primary server, while the second server will become the member.

Step 5 Verify the versions of the servers to confirm that it displays Release 3.9.0. Also, verify that the primary server and member is assigned.

Step 6 If TLS certification is enabled between NDB server and NXOS switch, copy the tlsTrustStore and tlsKeyStore files to /xnc/configuration from the old xnc backup.

Step 7 Navigate to **Administration** > **system** > **Backup/Restore** > **Restore** > **Restore locally** to upload the configuration to the primary server. Stop Cisco NDB on the second server and restart the first server. After you restart the server, Release 3.9.0 configurations are successfully uploaded in Cisco NDB Release 3.9.0. Verify all the configurations.

Step 8 If secondary / cluster NDB server is configured, start the server.

Note If TLS certification is enabled, start the secondary/ cluster using the commands as shown below:

```
./runxnc.sh -tls -tlskeystore ./configuration/tlsKeyStore -tlstruststore
./configuration/tlsTrustStore
cd bin
./xnc config-keystore-passwords --user <NDB_username> --password <NDB_password> --url
https://<Cluster_NDB_IP>:8443 --verbose --prompt --keystore-password <keystore-password>
--truststore-password <truststore-password>
```

Upgrading NDB - Hitless Method (Using CLI)

You can upgrade Cisco NDB to Release 3.9.0 with the hitless method using CLI.

Before you begin

If the Cisco NDB version is earlier than Release 3.8, you must edit the config.ini file and update the **skipConfigurtionStateDBfiles** key to false on both the controllers, and restart all the earlier version controllers.

Step 1 Stop both the servers.

Step 2 Navigate to the the s server location `/home/3.9.0/xnc/bin` and enter the **. /xnc upgrade --perform --target-home {xnc directory to be upgraded} --verbose** command.

Note You must provide the location of the XNC directory in the target home. For example, provide the location of the 3.9.0 XNC directory which is /home/3.9.0/xnc.

Step 3 Navigate to the the secondary server location `/home/3.9.0/xnc/bin` and enter the **. /xnc upgrade --perform --target-home {xnc directory to be upgraded} --verbose** command.

Note You must provide the location of the XNC directory in the target home. For example, provide the location of the 3.9.0 XNC directory which is /home/3.9.0/xnc.

Step 4 If TLS certification is enabled between NDB server and NXOS switch, copy the tlsTrustStore and tlsKeyStore files to /xnc/configuration from the old xnc backup in the primary and secondary servers.

Step 5 Navigate to the Cisco NDB Release 3.9.0 XNC directory in the primary server and start Cisco NDB using the **./runxnc.sh –start** command.

Step 6 Login to Cisco NDB and verify that the Cisco NDB version is displayed as Release 3.9.0. Verify that the primary configuration and the other configurations are retained.

Step 7 If secondary / cluster NDB server is configured, start the server.

Note If TLS certification is enabled, start the secondary/ cluster using the commands as shown below:

```
./runxnc.sh -tls -tlskeystore ./configuration/tlsKeyStore -tlstruststore
./configuration/tlsTrustStore
cd bin
./xnc config-keystore-passwords --user <NDB_username> --password <NDB_password> --url
https://<Cluster_NDB_IP>:8443 --verbose --prompt --keystore-password <keystore-password>
--truststore-password <truststore-password>
```

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	
-jmxport *port_number*	Enables JMX remote access on the specified JVM port.
-debugport *port_number*	Enables debugging on the specified JVM port.
-start	
-start *port_number*	
-stop	
-restart	
-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 The Application Status

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

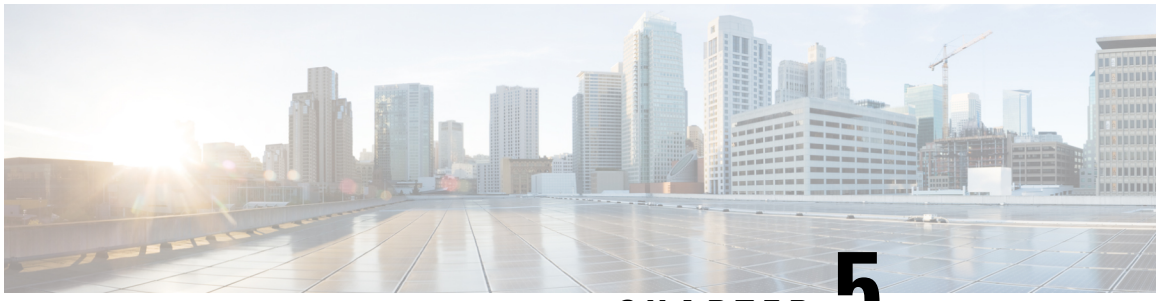
Step 2 Verify that the application is running by entering the **./runxnc.sh -status** command.

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.



CHAPTER 5

Migrating Cisco NDB OpenFlow to NXAPI Implementation

This chapter contains the following sections:

NDB Migration Overview

Starting with Cisco Nexus Data Broker, release 3.7, you can now migrate centralized NDB OpenFlow implementation to NXAPI implementation using the NDB migration tool. NDB migration occurs on the same virtual machine where the existing OpenFlow instance exists. The NDB migration process involves:

- Upgrading to NDB version 3.6 or later
- Exporting the device configuration in NDB 3.6
- NDB configuration cleanup
- Device conversion from OpenFlow to NXAPI by removing OpenFlow virtual service instances.
- Importing the NXAPI device configuration in NDB 3.6

NDB migration tool provides the following features:

- Single Touch Migration from OpenFlow to NXAPI devices.
- Supports NDB version from NDB 3.6.
- Supports all NDB Platform devices.
- Supports Atomic & Non-Atomic operations.
- Multiple Device Upgrade in a single Migration job.

NDB Migration Limitations

Follow these limitations and usage guidelines while migrating NDB from OpenFlow to NXAPI implementation:

- Port groups are not supported for NDB migration. If NDB has port groups, you need to manually reconfigure the port groups after migrating NDB to NXAPI.
- Port group description does not support special characters. Ensure that you remove all the special characters from port group description before starting the migration process.

Prerequisites for Migrating NDB

- You should have administrative access to mingrate NDB implementation from OpenFlow o NXAPI.
- You have following packages installed on the device:
  - Python (version 2.7)
  - Pip (version 10.0.1)
  - Open SSL
  - pexpect
  - YAML
  - Requests
  - ExScript
  - Configobj
  - paramika
  - Git

Note You need to install Python and Pip. For the rest of the packages, you can use the requirement.txt file with pip install command. For more information about package installation, see Installing Packages on Linux, on page 56.

Installing Packages on Linux

You can install required packages on Linux Ubuntu or Linux Redhat flavors:

- Installing Packages on Linux Ubuntu
- Installing Packages on Red Hat Linux, on page 57

Installing Packages on Linux Ubuntu

Complete these steps to install the following packages on Linux Ubuntu (version 10.0.1):

- Open SSL
- pexpect
- YAML
- Requests
- ExScript
- Configobj
- Paramiko

Step 1 Install Git using the **sudo** command.

Example:

```
sudo apt-get install git
```

Step 2 Install Python using the sudo command.

Example:

```
sudo apt-get install python2.7
```

Step 3 Install Pip using the sudo command.

Example:

```
Sudo apt-get install pip
```

You can also install a specific pip version using the **pip install pip==**<version> command.

Step 4 Update the `requirement.txt` file with the packages to install.

Example:

```
pexpect==4.6.0
pyyaml==3.12
Requests==2.18.4
ExScript==2.5.7
configobj==5.0.6
```

Step 5 Use the **pip install** command to install packages listed in the requirement.txt file.:

Example:

```
# pip install -r requirements.txt
```

Installing Packages on Red Hat Linux

Complete these steps to install the following packages on Red Hat Linux:

- Open SSL

- pexpect
- YAML
- Requests
- ExScript
- Configobj
- Paramiko

Step 1 Install Git using the **yum** command.

Example:

```
yum install git
```

Step 2 Install Python using the **sudo** command.

Example:

```
sudo yum install python27
```

Step 3 Install Pip using the **sudo** command.

Example:

```
Sudo yum install python-pip
```

Step 4 Install the Pur package using the pip install command, which is required if the older version of packages exist in the Red Had Linux.

Example:

```
pip install pur
```

Step 5 Update the `requirement.txt` file with the packages to install.

Example:

```
pexpect==4.6.0
pyyaml==3.12
Requests==2.18.4
ExScript==2.5.7
configobj==5.0.6
```

Step 6 Use the **pur** command to install packages listed in the requirement.txt file.:

Example:

```
# pur -r requirements.txt
```

Migrating Cisco NDB from OpenFlow to NXAPI

Complete the following steps to migrate NDB from OpenFlow to NXAPI.

Step 1 Download the migration script available at GIThub server (https://github.com/datacenter/nexus-data-broker). For example:

Example:

```
git clone http://ndb-build.cisco.com/gerrit/NDBMigration
```

The migration script is available in the `datacenter\nexus-data-broker` folder

Step 2 Open the input.yaml file and update the following fields:

Table 2:

Field Name	Description
NDB Server	
host_name/IP	Host Name or IP address of the server.
username	Username to log in to NDB Server.
password	Password to log into the NDB Server
ndb_gui_username	NDB Server GUI login username.
ndb_gui_password	NDB Server GUI login password.
old_path_ndb_build	Location of current NDB server xnc folder
new_path_ndb_build	Location where new NDB server xnc folder will be created after migrating to NXAPI.
Device Details	
host_name/IP	Host name or IP address of the switch.
username	Username to log in to the switch.
passoword	Password to log in to the switch.
mode	Switch mode for the switch after migration, ensure that it is configured to NXAPI.
tcam_ifacl	TCAM regions to create after device conversion to NXAPI.
tcam_mac-ifacl	TCAM regions to create after device conversion to NXAPI on MAC.
nxos	NXOS image to which device needs to be migrated. In case of Nexus 3000 series switches, if the current NXOS version is below u6, then first you need to upgrade the device to U6 and then to I46 or I47.

Step 3 Use the **python** command to run the migration script.

Example:

```
python NDBMigration.py
```

A unique jobid folder is created every time the migration script is run and contains three folders:

- Backup: Contains the old NDB zip file, export JSON file, import JSON file, and the state file. The state file contains detailed information about the status of every step involve in the migration process. Insert diagram for the state file screen shot.
- Log: Contains the migration log information
- Report: Contains information about the migration script result

Successful completion of migration process will result in NDB NXAPI implementation in the virtual machine. If the migration process fails, the resultant behavior depends on the revertFlat attribute configured in the input.yaml file.

- If revertFlag is set to 1, NDB and device configurations are reverted to old NDB version along with OF device configurations.
- If revertFlag is set to 0, the revert behavior depends on the stage where the failure occurs
  - Failure during NDB upgrade – NDB and device configurations are reverted to old NDB version along with OF device configurations.
  - Failure during NDB export – NDB and device configurations are reverted to old NDB version along with OF device configurations.
  - Failure during NDB clean up – NDB and device configurations are reverted to old NDB version along with OF device configurations.
  - Failure during device conversion – Migration script will continue to the next device and the state of the failed device is set to FAIL.
  - Failure during NDB import – Migration script will continue to the next device and the state of the failed device is set to FAIL.

Note You can rerun the migration script on failure to proceed the migration from the point of failure. Use the **python NDBMigration.py –rerun** failedjobid command to start the migration process from the point of failure. For example:

```
python NDBMigration.py –rerun  job.2018Aug07_04:49:39
```

Troubleshooting NDB Migration Issues

NDB migration script may fail during the migration process. You can look for the log file and migration report in the `jobid` folder that is created every time the migration script is run for troubleshooting information. The `jobid` folder contains three folders:

- Backup: Contains the old NDB zip file, export JSON file, import JSON file, and the state file. The `state` file contains detailed information about the status of every step involve in the migration process. Every step is represented by either of these three states:

- Pass
- Fail
- Skip

- Log: Contains the migration log information
- Report: Contains information about the migration script result.

NDB Proxy Issues

Migration process may fail due to proxy issues. To resolve any proxy issues, you need to unset the proxy.

Table 3: Proxy Issue Related Error Messages

Error Message	Command
`HTTPSConnectionPool(host='10.16.206.197', port=8443): Max retries exceeded with url: /monitor (Caused by ProxyError('Cannot connect to proxy.', error('Tunnel connection failed: 504 Gateway Timeout',)))`	`#unset http_proxy` `#unset https_proxy`

NDB Import Issues

NDB import process may fail on Cisco Nexus 3000 series switches. Check the switching mode for the Nexus 3000 switch using the **show system switch-mode** command. The switchport mode should be n3k.

```
N3K-123# show system switch-mode
       system switch-mode n3k
```

Note The Cisco Nexus N3K-3132Q-40GX and N3K-3172PQ-10GE switches support both n3k and n9k switching mode. For NDB migration, ensure that the switching mode is set to n3k.

Reverting to Previous Configuration in case of Script Failure

Follow these steps to revert to the previous configuration in case of migration script failure:

Step 1 Remove all the interface configurations from all the devices using the no feature command.

Example:

```
(config)# no feature openflow
```

Step 2 Load previous NXOS image that was loaded before running migration script on all the devices using the **install all system** command or **install all nxos** command.

Example:

```
install all system  n3000-uk9.6.0.2.U6.4a.bin kickstart n3000-uk9-kickstart.6.0.2.U6.4a.bin

install all  nxos nxos.7.0.3.I4.6.bin
```

Step 3 Install and activate openflow ova on all the devices using the **virtual-service install** command.

Example:

```
virtual-service install name ofa package bootflash:<openflow-ova>
```

Step 4 Configure open flow configuration on all the devices. Old configuration is available in the `Migration_backup` file available on each device.

Example:

```
#openflow
#switch 1
#pipeline 201
#probe-interval 5
#controller ipv4 10.16.206.136 port 6653 vrf   management security none
#of-port interface Ethernet1/1-54
```

Step 5 Configure tcam region which was present before running migration script.

Example:

```
#sh file Migration_backup_2019Feb03_01:48:52 | grep hardware
hardware access-list tcam region ifacl 256
hardware access-list tcam region openflow 256

// Current tcam configuration:
#sh run |grep  hardware
hardware access-list tcam region ifacl 256

// Command to configure Tcam region:
#config t
#hardware access-list tcam region openflow 256
```

Step 6 Stop and start the NDB to apply the new configuration.

Example:

```
./runxnc -stop
./runxnc -start
```

FAQs - NDB Migration

Q. Where should I run NDB migration script?

A. You can run the migration script from a VM where NDB is running or from a new VM (Ubuntu/Redhat). Cisco recommends that you run the migration script from a new VM.

Q. What happens if a user already has python packages with older version in Redhat?

A. You need to install the Pur package using the **pip** command and then use the **pur** command to install the packages listed in the `requirement.txt`.

Q. How to check Python version?

A. Use the **python -V** commamnd to check the current Python version..

Q. How to check Pip version?

A. Use the **pip -V** commamnd to check the current Pip version..

Q. How to check Pip packages?

A. Use the **pip list** commamnd to check the Pip packages installed along with the version.