Cisco Nexus Data Broker Deployment Guide, Release 3.9.2

First Published: 2022-02-23

Last Modified: 2023-03-01

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.

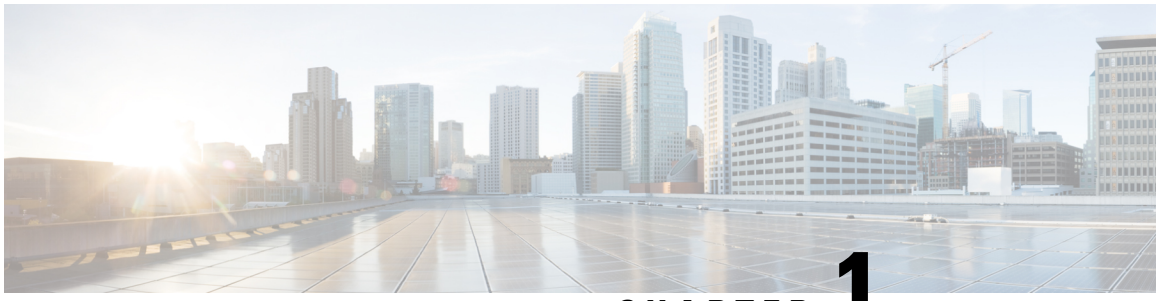
Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability 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-k9-3.7.0.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-k9-3.7.0.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-3.7.0-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-3.7.0-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-3.7.0-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-k9-3.7.0.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-k9-3.7.0.zip

Nexus Data Broker Hardware and Software Interoperability Matrix

The following table lists the hardware and software ineteroperability matrix for Cisco NDB.

Note Cisco Nexus 9200 Series switches support only one switch deployment.

Nexus Switch Model(s)	Implementation Type	Supported NX-OS Versions	Open Flow Agent
Nexus 3048 / 3064 / 3172	OpenFlow	6.0(2)U6	1.1.5
Nexus 3048 / 3064 / 3172	OpenFlow	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)	2.1.4
Nexus 3048 / 3064	NXAPI	6.0(2)U6(x), and 7.0(3)I4(1) to 7.0(3)I4(8b)	NA
Nexus 3172	NXAPI	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)	NA
Nexus 3164	OpenFlow	Not Supported	Not Supported
Nexus 3164	NXAPI	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)	NA
Nexus 3232	OpenFlow	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)	2.1.4
Nexus 3232	NXAPI	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)	NA
Nexus 3548	OpenFlow	6.0(2)A6(x) and 6.0(2)A8(x) I7(5) and I7(5a) (OF agent is not required) 7.0(3)I7(2) to 7.0(3)I7(7)	1.1.5
Nexus 3548	NXAPI	Not Supported	Not Supported
Nexus 92160 / 92304	OpenFlow	Not Supported	Not Supported
Nexus 92160 / 92304	NXAPI	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)	NA
Nexus 9372 / 9396 / 93128	OpenFlow	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)	2.1.4
Nexus 9372 / 9396 / 93128	NXAPI	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)	NA

Nexus Switch Model(s)	Implementation Type	Supported NX-OS Versions	Open Flow Agent
Nexus 93180LC-EX	OpenFlow	Not Supported	Not Supported
Nexus 93180LC-EX	NXAPI	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)	NA
Nexus 93108TC-EX / 93180YC-EX	OpenFlow	Not Supported	Not Supported
Nexus 93108TC-EX / 93180YC-EX	NXAPI	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)	NA
Nexus 93108TC-FX / 93180YC-FX	OpenFlow	Not Supported	Not Supported
Nexus 93108TC-FX/ 93180YC-FX	NXAPI	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)	NA
Nexus 9504 / 9508 / 9516	OpenFlow	Not Supported	Not Supported
Nexus 9504 / 9508 / 9516	NXAPI	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)	NA
Nexus 31108TC-V / 31108PC-V	NXAPI	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)	NA
Nexus 31108TC-V / 31108PC-V	OpenFlow	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)	NA
Nexus 9336C-FX2 / 93240YC-FX2	NXAPI	7.0(3)I7(5), 7.0(3)I7(5a), 7.0(3)I7(6), 7.0(3)I7(7), 9.2(1) to 9.2(4), 9.3(1) to 9.3(4).	NA
Nexus C93360YC-FX2	NXAPI	9.3(1) to 9.3(4)	NA

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.

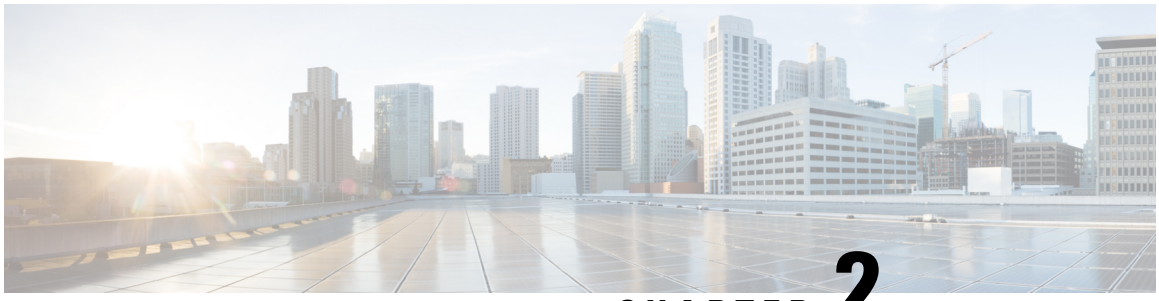
Supported Deployment Profiles

Cisco NDB controller supports three deployment profiles:

Deployment Profile	No. of NDB Devices	Memory Requirement(in GB)
Small	25*	4
Medium	50*	8
Large	75*	16

* Nexus switches with 52 ports.

After an upgrade/ first install, run the **./runxnc.sh -start** command. The supported deployment profiles are, *small*, *medium*, *large*). The memory requirement for each profile is different (as indicated above). By default, the *medium* profile is selected. If sufficient memory is not available (to support the *medium* profile), then the *small* profile is selected. You will be prompted to indicate **yes** or **no** before the profile is switched from the default *medium* to *small*. You can change the profile to *large* at a later time, as required. Run the **./runxnc.sh -start -profile** *type* command, where the *type* is *large.*



CHAPTER 2

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:~/3.9.2/$ md5sum ndb1000-sw-app-k9-3.9.2.zip
Displayed output: c2d273dce4abbba03c06aeee8774b901 ndb1000-sw-app-k9-3.9.2.zip
```

This chapter contains the following topics:

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

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 3.9.2 is displayed: Cisco Nexus Data Broker Software Application: ndb1000-sw-app-k9-3.9.2.zip

Step 6 Download the Cisco Nexus Data Broker application bundle.

Step 7 Create a directory in your Linux machine where you plan to install Cisco Nexus Data Broker.

For example, in your Home directory, create `CiscoNDB`.

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

Step 9 Unzip the Cisco Nexus Data Broker zip file.

The Cisco Nexus Data Broker software is installed in a directory called `xnc`. The directory contains the following:

- `runxnc.sh` file—The file that you use to launch Cisco Nexus Data Broker.
- `version.properties` file—The Cisco Nexus Data Broker build version.
- `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 3.9.2.

Note

- Once you upgrade to Cisco NDB Release 3.9.2, 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 3.9.2 applicable bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.9.2.zip
6. Create a temporary directory in your Linux machine where you plan to upgrade to Cisco NDB.
7. Unzip the Cisco NDB Release 3.9.2 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 3.9.2 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 3.9.2 applicable bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.9.2.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 3.9.2 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 3.9.2 upgrade software.

Step 12 Upgrade the application by entering the **./xnc upgrade --perform --target-home** {*xnc_directory_to_be_upgraded*} **[--verbose] [--backupfile** {*xnc_backup_location_and_zip_filename*}**]** command.

You can use one of the following options:

Option	Description
--perform --target-home {*xnc_directory_to_be_upgraded*}	Upgrades the Cisco XNC Monitor Manager 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
```

Upgrading the Application Software when TLS is enabled in the Standalone Controller

Use this procedure for upgrading the application software in centralized mode, using the GUI, when the TLS certification is enabled in the standalone controller.

Step 1 Log in to the existing NDB GUI instance using *https://<server IP>:8443*.

Step 2 Navigate to **Administration** > **System**> **Backup/ Restore** > **Backup** tab.

Step 3 Click **Backup now Locally** to download the configuration as a zip file.

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

Step 5 After the NDB instance is stopped, navigate to */xnc/configuration* folder, and copy the tlsTrustStore and tlsKeyStore files to *local/common* folder.

Step 6 Download the NDB 3.9.2 software from the standard Downloads page and start the new NDB 3.9.2 installation using the `runxnc.sh -start` command.

Step 7 Log in to the new instance of NDB GUI using *https://<server IP>:8443*.

Step 8 Navigate to **Administration** > **System**> **Backup/ Restore** > **Backup** tab.

Step 9 Click **Restore Locally** to upload the configuration file which you have downloaded earlier (see Step 3, above).

After the configuration is uploaded successfully, you will see a *success* message on the GUI.

Step 10 Connect using SSH to the NDB server, and copy the tlsTrustStore and tlsKeyStore files to NDB 3.9.2 */xnc/configuration/startup* folder (which is copied to *local/common* folder in step 5).

Step 11 Stop the NDB 3.9.2 instance using the `runxnc.sh -stop` command.

Step 12 Start the NDB 3.9.2 instance again using the following command:

```
./runxnc.sh -tls -tlskeystore ./configuration/startup/tlsKeyStore -tlstruststore
./configuration/startup/tlsTrustStore
```

Upgrading the Application Software when TLS is enabled in the HA-Clustered Controller

Use this procedure for upgrading the application software in centralized mode, using the GUI, when the TLS certification is enabled in the HA-clustered controller.

Step 1 Log in to the existing NDB GUI instance using *https://<server IP>:8443*.

Step 2 Navigate to **Administration** > **System**> **Backup/ Restore** > **Backup** tab.

Step 3 Click **Backup now Locally** to download the configuration as a zip file.

Step 4 Stop all the current NDB instance(s) using the `runxnc.sh -stop` command.

Step 5 After the NDB instance is stopped, navigate to */xnc/configuration* folder, and copy the tlsTrustStore and tlsKeyStore files to *local/common* folder.

Step 6 Download the NDB 3.9.2 software from the standard Cisco downloads page and configure the cluster mode with "supernodes" configuration in the *config.ini* file and start the new NDB 3.9.2 cluster using the `runxnc.sh -start` command on all the controllers.

Step 7 On the primary controller, navigate to **Administration** > **Syste,** > **Backup/Restore** > **Backup** tab.

Step 8 Click **Restore Locally** to upload the configuration file which you have downloaded earlier (see Step 3, above).

After the configuration is uploaded successfully, you will see a *success* message on the GUI.

Step 9 Connect using SSH to the NDB server, and copy the tlsTrustStore and tlsKeyStore files to NDB 3.9.2 */xnc/configuration/startup* folder (which is copied to *local/common* folder in step 5).

Step 10 Stop the NDB 3.9.2 instances on all the controllers of the cluster, using the `runxnc.sh -stop` command.

Step 11 Start the NDB 3.9.2 instance on the primary controller using the following command.

```
./runxnc.sh -tls -tlskeystore ./configuration/startup/tlsKeyStore -tlstruststore
./configuration/startup/tlsTrustStore
```

Wait for a few minutes; a *ready* message is displayed.

Step 12 Start the NDB 3.9.2 instance on the other controllers of the cluster using the following command:

```
./runxnc.sh -tls -tlskeystore ./configuration/startup/tlsKeyStore -tlstruststore
./configuration/startup/tlsTrustStore
```

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 3.9.2 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/3.9.2/xnc`) of the xnc for Release 3.9.2 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 3.9.2. 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 3.9.2 configurations are successfully uploaded in Cisco NDB Release 3.9.2. 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 3.9.2 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/3.9.2/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 3.9.2 XNC directory which is /home/3.9.2/xnc.

Step 3 Navigate to the the secondary server location `/home/3.9.2/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 3.9.2 XNC directory which is /home/3.9.2/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 3.9.2 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 3.9.2. 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>
```

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 is in a *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 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:

- Message indicating the GUI is ready to accept configuration(s)— *NDB is ready for configuration* .
- (for HA) Message indicating that the primary is loaded and it is time for the members in the cluster— *Primary is ready and bring up the members.*
- During cluster rehashing, when members are joining/ leaving the quorum — *Cluster is rehashing.*

Note Messages are displayed in red until NDB is ready. After NDB is ready, the message, *NDB is ready for configurations* is displayed in green.

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 configuration* 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)

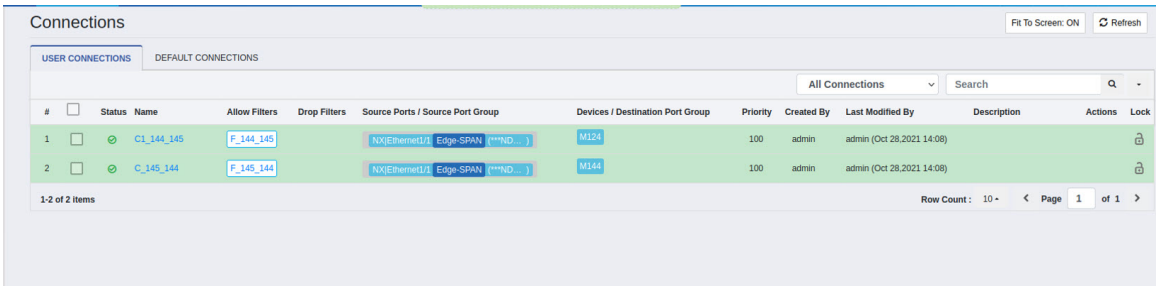
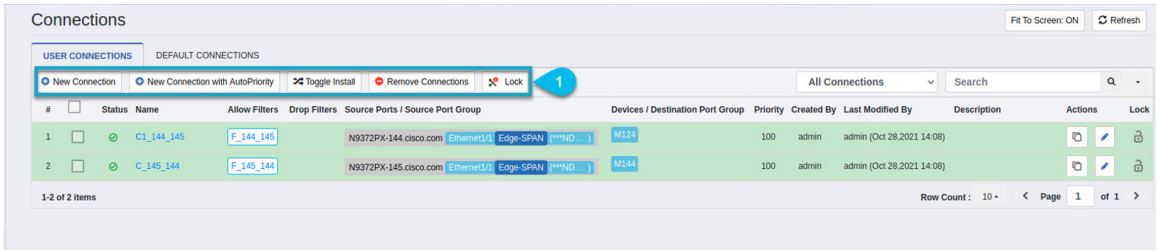


Figure 3: GUI enhancement - Connections (with configuration buttons)



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

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

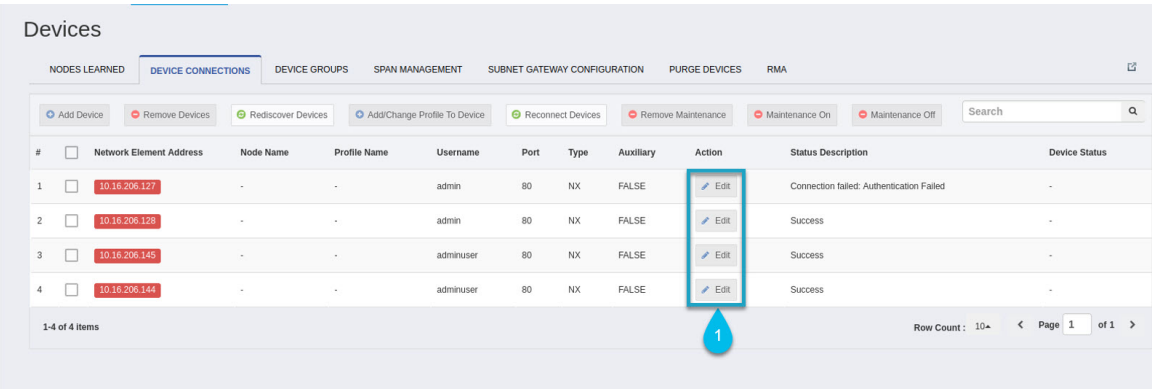
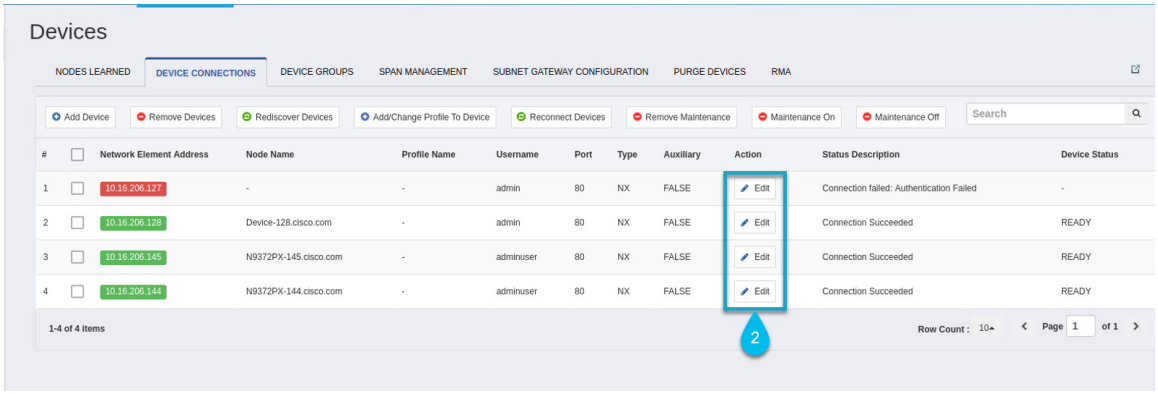


Figure 5: GUI enhancement - Devices (configuration buttons are enabled)



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
-jmxport *port_number*	Enables JMX remote access on the specified JVM port.
-debugport *port_number*	Enables debugging on the specified JVM port.
-start	Starts NDB.
-start *port_number*	Starts NDB on the specified port.
-stop	Stops NDB.

Option	Description
-restart	Restarts NDB.
-status	Displays the NDB status with process ID.
-console	Starts NDB with the login 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 `ndb` directory that was created when you installed the software.

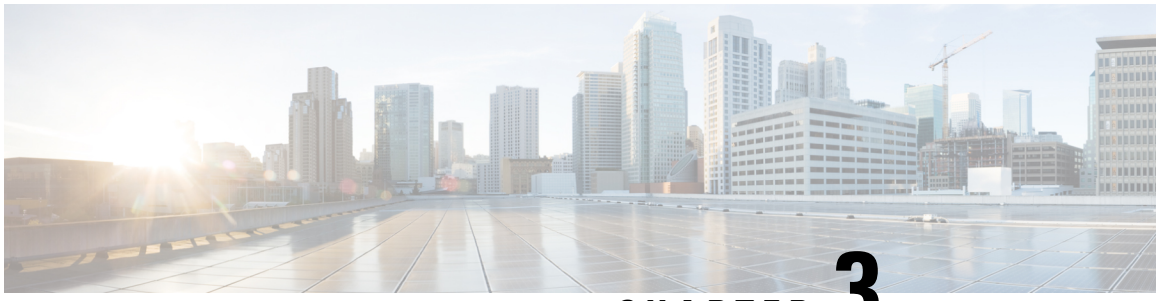
Step 2 Verify that the application is running by entering the **./runndb.sh -status** command.

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 3

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 30.

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 31

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.