



## **Cisco Nexus Data Broker Configuration Guide, Release 3.7.1**

**First Published:** 2019-02-26

**Last Modified:** 2019-09-27

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

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

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

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

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

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

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at [www.cisco.com/go/offices](http://www.cisco.com/go/offices).

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](http://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)

© 2019 Cisco Systems, Inc. All rights reserved.



The Java logo is a trademark or registered trademark of Sun Microsystems, Inc. in the U.S. or other countries.

© 2019 Cisco Systems, Inc. All rights reserved.





# CHAPTER 1

## Cisco Nexus Data Broker Overview

---

This chapter has an overview of the Cisco Nexus Data Broker. It also has details of the filename and interoperability matrices. See the prerequisites section for important details before configuring NDB.

This chapter contains the following sections:

- [About Cisco Nexus Data Broker, on page 1](#)
- [Prerequisites for Cisco Nexus Series Switches, on page 6](#)
- [Supported Web Browsers, on page 11](#)
- [System Requirements, on page 11](#)
- [Guidelines and Limitations for Nexus Data Broker, on page 11](#)
- [Cisco Nexus Data Broker Software Release Filename Matrix, on page 12](#)
- [Nexus Data Broker Hardware and Software Interoperability Matrix, on page 14](#)

## 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.




---

**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, the `mm.addDefaultISLDenyRules` attribute is not present in `config.in` file. To disable this feature, you need to add the `mm.addDefaultISLDenyRules` attribute to `config.ini` file and 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 3.6.2, 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.
- 

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 and MPLS tag stripping.



- 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 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.

## 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 17(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 <b># hardware profile openflow</b> command at the prompt.	
Cisco Nexus 3164Q switches	The OpenFlow mode is not supported on the Nexus 3164Q switches.	Enter the following commands at the prompt: <ul style="list-style-type: none"> <li>• <b># hardware profile tcam region qos 0</b></li> <li>• <b># hardware profile tcam region racl 0</b></li> <li>• <b># hardware profile tcam region vacl 0</b></li> <li>• <b># hardware profile tcam region ifacl 1024 double-wide</b></li> <li>• <b># hardware access-list tcam region mac-ifacl 512</b></li> <li>• <b>#feature nxapi</b></li> <li>• <b>#feature lldp</b></li> </ul>
Cisco Nexus 3172 Series switches	Enter the <b># hardware profile openflow</b> command at the prompt.	Use the <b>hardware profile mode tap-aggregation [ l2drop ]</b> 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 <b>hardware access-list tcam region openflow 256</b> command at the prompt.	Enter the following commands at the prompt: <ul style="list-style-type: none"> <li>• <b># hardware access-list tcam region e-racl 0</b></li> <li>• <b># hardware access-list tcam region span 0</b></li> <li>• <b># hardware access-list tcam region redirect 0</b></li> <li>• <b># hardware access-list tcam region vpc-convergence 0</b></li> <li>• <b># hardware access-list tcam region racl-lite 256</b></li> <li>• <b># hardware access-list tcam region l3qos-intra-lite 0</b></li> <li>• <b># hardware access-list tcam region ifacl 256 double-wide</b></li> <li>• <b># hardware access-list tcam region mac-ifacl 512</b></li> <li>• <b># hardware access-list tcam region ipv6-ifacl 256</b></li> <li>• <b>#feature nxapi</b></li> <li>• <b>#feature lldp</b></li> </ul>
Cisco Nexus 3500 series switches	Enter either of the following commands at the prompt to configure OpenFlow TCAM: <ul style="list-style-type: none"> <li>• <b># hardware profile forwarding-mode openflow-hybrid</b></li> <li>• <b>#hardware profile forwarding-mode openflow-only</b></li> </ul>	

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 9300 Series switches	<p>Enter the <b>hardware access-list tcam region openflow 512 double-wide</b> command at the prompt to configure the MAC filters.</p> <p>For IPv4 and IPv6, enter the <b>hardware access-list tcam region openflow 512</b> command.</p> <p><b>Note</b> IPv6 and IPv4 dual stack is not supported in I6 and I7.</p>	<p>Enter the following commands at the prompt:</p> <ul style="list-style-type: none"> <li>• <b># hardware access-list tcam region qos 0</b></li> <li>• <b># hardware access-list tcam region vacl 0</b></li> <li>• <b># hardware access-list tcam region racl 0</b></li> <li>• <b># hardware access-list tcam region redirect 0</b></li> <li>• <b># hardware access-list tcam region vpc-convergence 0</b></li> <li>• <b># hardware access-list tcam region ifacl 1024 double-wide</b></li> <li>• <b># hardware access-list tcam region mac-ifacl 512</b></li> <li>• <b># hardware access-list tcam region ipv6-ifacl 512</b></li> <li>• <b>#feature nxapi</b></li> <li>• <b>#feature lldp</b></li> </ul>

Device Models	OpenFlow Mode	NX-API Mode
Cisco Nexus 9200, 9300-EX, 9336C-FX2, 93240YC-FX2, and N9K-C93360YC-FX2 switches	The OpenFlow mode is not supported on the 9200, 9300-EX, 9336C-FX2, 93240YC-FX2, and N9K-C93360YC-FX2 switches.	Enter the following commands at the prompt: <ul style="list-style-type: none"> <li>• <b>#hardware access-list tcam region ing-l2-span-filter 0</b> (For Cisco Nexus 93108 series switch only)</li> <li>• <b>#hardware access-list tcam region ing-l3-span-filter 0</b> (For Cisco Nexus 93108 series switch only)</li> <li>• <b># hardware access-list tcam region ing-racl 0</b></li> <li>• <b>hardware access-list tcam region ing-l3-vlan-qos 0</b></li> <li>• <b># hardware access-list tcam region egr-racl 0</b></li> <li>• <b># hardware access-list tcam region ing-ifacl 1024</b></li> <li>• <b>#feature nxapi</b></li> <li>• <b>#feature lldp</b></li> </ul>
Cisco Nexus 9500-EX and 9500-FX Series switches (9504, 9508 and 9516)	The OpenFlow mode is not supported on the Cisco Nexus 9500-EX and 9500-FX Series switches.	Enter the following commands at the prompt: <ul style="list-style-type: none"> <li>• <b># hardware access-list tcam region ing-racl 0</b></li> <li>• <b># hardware access-list tcam region ing-l3-vlan-qos 0</b></li> <li>• <b># hardware access-list tcam region egr-racl 0</b></li> <li>• <b># hardware access-list tcam region ing-ifacl 1024</b></li> <li>• <b>#feature nxapi</b></li> <li>• <b>#hardware acl tap-agg</b></li> <li>• <b>#feature lldp</b></li> </ul>

## Supported Web Browsers

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

- Firefox 45.x and later versions.
- Chrome 45.x and later versions.
- Internet Explorer 11 or later versions.
- Microsoft Edge 42 or later versions.



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

## System Requirements

The following table lists the system requirements as per the deployment size for Cisco Nexus Data Broker 3.8.

**Table 1: System Requirements per Deployment Size**

Description	Small	Medium	Large
CPUs (virtual or physical)	6-core	12-core	18-core
Memory	8 GB RAM	16 GB RAM	24 GB RAM
Hard disk	Minimum of 40 GB of free space available on the partition on which the Cisco Nexus Data Broker software is installed.		
Operating System	A recent 64-bit Linux distribution that supports Java, preferably Ubuntu, Fedora, or Red Hat.		
Other	Java Virtual Machine 1.8 or later.		

## Guidelines and Limitations for Nexus Data Broker

Cisco Nexus Data Broker runs in a Java Virtual Machine (JVM). As a Java-based application, Cisco Nexus Data Broker can run on any x86 server. For best results, we recommend the following:

- Java Virtual Machine 1.8.0\_45 and higher.
- Python 2.7.3 and a higher version is required for the backup and restore script. This is also required to do the TLS configuration if Cisco Nexus Data Broker needs to use TLS for the device communication.
- A \$JAVA\_HOME environment variable in your profile that is set to the path of the JVM.

- JConsole and VisualVM that are both part of JDK are the recommended (but not required) additions for troubleshooting.
- During OpenFlow configuration for Cisco NXOS Release 7.0(3)I5(1) software image, virtual service ofa should not be installed and the following configuration should be used:

```
switch#
conf t
feature openflow
openflow
  switch 1 pipeline 201
  controller ipv4 10.16.206.162 port 6653 vrf management security none
of-port interface ethernet1/1-30
```

See the following link for further details on NXOS configuration for OpenFlow:[http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus/openflow/b\\_openflow\\_agent\\_nxos\\_1\\_3/Cisco\\_Plug\\_in\\_for\\_OpenFlow.html#reference\\_B6284F508CC6461B8EF30DCF870C809F](http://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus/openflow/b_openflow_agent_nxos_1_3/Cisco_Plug_in_for_OpenFlow.html#reference_B6284F508CC6461B8EF30DCF870C809F)

- You should not configure the same name for more than one switch in the topology to avoid unpredictable behavior in the link discovery by Cisco Nexus Data Broker.
- Starting with Cisco NDB Release 3.7, the following special characters are not allowed in description field for Port Definitions, Port Groups, Connections, Redirections, Monitoring Devices, and Service Nodes: Apostrophe (‘), Less Than (<), Greater Than (>), Double Quotation (“), Back Slash (\), Vertical Bar (|), and Question Mark (?).
- When the domain name is enabled in the switch, it does not reflect the change in the LLDP neighbors and the links get removed for that particular switch. The workaround for this issue is to disable the LLDP feature and then to enable it again by using **no feature lldp** and **feature lldp** CLI commands respectively.

### Global Updates with Cisco Nexus Data Broker, Release 3.3

See the following global updates that are available with Cisco Nexus Data Broker, Release 3.3:

- The ports in the Graphical User Interface (GUI) are listed in a sorted order.
- A new field, **Row Count** is added in the GUI to display the rows in the multiples of 10, 25, 50, and 100.
- Cisco Nexus 92XX switches do not support the QnQ, you cannot use 92XX switches in the multi-switch environment.

## 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
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_nmemb-2.1.4-r2-nxos-SPA-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-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 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 Switch Model(s)	Implementation Type	Supported NX-OS Versions	Open Flow Agent
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



## CHAPTER 2

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

---

This chapter contains the following sections:

- [Installing Cisco Nexus Data Broker in Centralized Mode, on page 17](#)

## Installing Cisco Nexus Data Broker 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:

#### Procedure

---

- 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 3.7.1 is displayed: Cisco Nexus Data Broker Software Application:  
ndb1000-sw-app-k9-3.7.1.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 Nexus Data Broker Release 3.7.1 or later release. When you are upgrading from Release 2.2.0 and/or Release 2.2.1, you first need to upgrade to Release 3.0.0 and only then you can upgrade to Cisco Nexus Data Broker Release 3.7.1 or later release. This upgrade is an in-place upgrade, which means that the product bits are replaced. A backup archive is created to restore your original installation, if necessary.



**Note** Once you upgrade to Cisco Nexus Data Broker Release 3.7.1 or later release, 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.



**Note** 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.
- 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.
- All the HA cluster members send keepalive message every 3 seconds. If a peer does not respond to keepalive message, peers attempt three retries to determine whether the peer is alive or dead. If a peer does not respond even after three retries, split-brain situation occurs where multiple HA members act as coordinators.



#### 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.

### Procedure

- Step 1** In a web browser, navigate to [Cisco.com](https://www.cisco.com).
- Step 2** Under **Support**, click **All Downloads**.
- Step 3** In the center pane, click **Cloud and Systems Management**.
- Step 4** In the right pane, click **Network Controllers and Applications**, and then click **Cisco Nexus Data Broker**.
- Step 5** Download the Cisco Nexus Data Broker Release 3.7.1 application bundle: Cisco Nexus Data Broker Software Application—`ndb1000-sw-app-k9-3.7.1.zip`
- Step 6** Create a temporary directory in your Linux machine where you plan to upgrade to Cisco Nexus Data Broker. For example, in your `Home` directory, create `CiscoNDB_Upgrade`.
- Step 7** Unzip the Cisco Nexus Data Broker Release 3.7.1 zip file into the temporary directory that you created.
- Step 8** Navigate to the `xnc` directory that was created when you installed the Cisco Nexus Data Broker release earlier.
- Step 9** Backup your Cisco Nexus Data Broker release installation using your standard backup procedures.

- Step 10** Stop running all Cisco Nexus Data Broker release processes.
- Step 11** Navigate to the `xnc/bin` directory in the temporary directory that you created for the Cisco Nexus Data Broker Release 3.7.1 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
<code>--perform --target-home {xnc_directory_to_be_upgraded}</code>	Upgrades the Cisco XNC Monitor Manager installation to Cisco Nexus Data Broker.
<code>--perform --target-home {xnc_directory_to_be_upgraded} --backupfile {xnc_backup_location_and_zip_filename}</code>	Upgrades the Cisco XNC Monitor Manager installation to Cisco Nexus Data Broker and creates a backup .zip file in the directory path that you set.  <b>Note</b> You must provide the name of the backup file and the .zip extension.  <b>Note</b> The backup file should not be saved in the xnc directory with current NDB installation or its subdirectory.
<code>--verbose</code>	Displays detailed information to the console. This option can be used with any other option and is disabled by default.
<code>--validate --target-home {xnc_directory_to_be_upgraded}</code>	Validates the installation.
<code>./xnc help upgrade</code>	Displays the options for the <code>upgrade</code> command.

- Step 13** Navigate to the `xnc` directory where you originally installed Cisco XNC Monitor Manager.
- Step 14** Start the application processes that you previously stopped.

**Note** Clear the browser cache. Use Shift+Ctrl+Delete keys to clear the cache.

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



## Starting the Application

### Procedure

**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	
<b>-jmxport</b> <i>port_number</i>	Enables JMX remote access on the specified JVM port.
<b>-debugport</b> <i>port_number</i>	Enables debugging on the specified JVM port.
<b>-start</b>	
<b>-start</b> <i>port_number</i>	
<b>-stop</b>	
<b>-restart</b>	
<b>-status</b>	
<b>-console</b>	
<b>-help</b>	Displays the options for the <code>./runxnc.sh</code> command.
<b>-tls</b>	To enable TLS, start the controller by entering the <code>./runxnc.sh -tls -tlskeystore keystore_file_location -tlstruststore truststore_file_location</code> command.
<b>-osgiPasswordSync</b>	To set the OSGi web console password same as the XNC password if the XNC password is changed.  <b>Note</b> 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

### Procedure

---

**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 3

# Managing TLS Certificate, KeyStore, and TrustStore Files

---

This chapter contains the following sections:

- [Generating TLS Self-Signed Certification Between NDB Server and NDB Switch for NXAPI, on page 23](#)
- [Generating TLS 3rd Party Certification Between NDB Server and NDB Switch for NXAPI, on page 29](#)
- [Generating TLS Self-Signed Certification Between NDB Server and NDB Switch for OpenFlow, on page 37](#)
- [Generating TLS Self-Signed Certification Between WebUI Browser and NDB Server, on page 45](#)
- [Generating TLS 3rd Party Certification Between WebUI Browser and NDB Server, on page 57](#)

## Generating TLS Self-Signed Certification Between NDB Server and NDB Switch for NXAPI

This section describes how to generate TLS self-signed certification between NDB server and NDB Switch. You need to generate certificates and keys for each switch to enable TLS. TLS communication between NDB switch and NDB server uses port 443 only.

Complete the following steps to generate TLS self-signed certification between NDB Server and NDB Switch for NXAPI:

- [Generating Self-Signed Certificate and Key, on page 24](#)
- [Creating the TLS TrustStore File, on page 26](#)
- [Starting NDB with TLS, on page 27](#)
- [Configuring TLS KeyStore and TrustStore Passwords on NDB, on page 28](#)



---

**Note** You cannot configure a controller to communicate using port 80 after configuring TLS.

---

## Generating Self-Signed Certificate and Key

This section describes how to generate self-signed certificate and key.

### Before you begin

Ensure that you have domain name configured on the switch using **ip domain-name** command for each NDB switch that acts as the Fully Qualified Domain Name (FQDN) for the switch. For example:

```
conf t
ip domain-name cisco.com
hostname N9k-117
end
```

The FQDN for the switch is configured to N9K-117.cisco.com.

### Procedure

#### Step 1

Log in to the server.

#### Step 2

Generate the private key and self-signed certificate using the **openssl req** command.

#### Example:

```
docker@docker-virtual-machine:~/TLS$ openssl req -x509 -nodes -days 3650 -newkey rsa:2048
-out sw1-ca.pem -outform PEM -keyout sw1-ca.key
```

```
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to 'sw1-ca.key'
```

-----

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

-----

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:SJ
Organization Name (eg, company) [Internet Widgits Pty Ltd]:cisco
Organizational Unit Name (eg, section) []:insbu
Common Name (e.g. server FQDN or YOUR name) []:N9K-117.cisco.com
Email Address []:myname@cisco.com
```

**Note** If you have multiple switches, generate the certificate file and private key for each switch.

This command creates a certificate file (sw1-ca.pem) and a private key (sw1-ca.key).

#### Step 3

Log in to the NDB switch.

#### Step 4

Copy the certificate file, sw1-ca.pem, and keyfile, sw1-ca.key, to the switch using the **copy** command.

**Example:**

```
N9K-117# copy scp://docker@10.16.206.250/home/docker/Mallik/TLS_CA_june_23/sw1-ca.pem
bootflash:
Enter vrf (If no input, current vrf 'default' is considered): management
docker@10.16.206.250's password:
server.cer

100% 4676      4.6KB/s   00:00
Copy complete, now saving to disk (please wait)...
```

```
N9K-117# copy scp://docker@10.16.206.250/home/docker/Mallik/TLS_CA_june_23/sw1-ca.key
bootflash:
Enter vrf (If no input, current vrf 'default' is considered): management

docker@10.16.206.250's password:
cert.key

100%
Copy complete, now saving to disk (please wait)...
```

**Note** If you have multiple switches, repeat this step for all the switches.

**Step 5** Configure the certificate file, sw1-ca.pem, and keyfile, sw1-ca.key in the switch using the **nxapi** command.

**Example:**

```
N9K-117 (config)# nxapi certificate httpskey keyfile bootflash:sw1-ca.key
Upload done. Please enable. Note cert and key must match.
N9K-117 (config)#
N9K-117 (config)# nxapi certificate httpsCRT certfile bootflash:sw1-ca.pem
Upload done. Please enable. Note cert and key must match.
N9K-117 (config)#
```

**Note** If you have multiple switches, configure the corresponding certificate and private key to each switch.

**Step 6** Enable self-signed certificates on the switch using the **nxapi certificate** command.

**Example:**

```
N9K-117 (config)# nxapi certificate enable
N9K-117 (config)#
```

**Note** Ensure that there is no error while enabling self-signed certificates on the switch.

**Step 7** Log in to the server.

**Step 8** Copy and convert the sw1-ca.key and sw1-ca.pem files to .PEM format using the **copy** command.

**Example:**

```
cp sw1-ca.key sw1-xnc-privatekey.pem
cp sw1-ca.pem sw1-xnc-cert.pem
```

**Step 9** Concatenate the private key and the certificate file using **cat** command.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ cat sw1-xnc-privatekey.pem sw1-xnc-cert.pem > sw1-xnc.pem
```

- Step 10** Convert the .pem file to .p12 file format using the **openssl** command. Enter the export password when prompted to create a password protected .p12 certificate file.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ openssl pkcs12 -export -out sw1-xnc.p12 -in sw1-xnc.pem
Enter Export Password: cisco123
Verifying - Enter Export Password: cisco123
Enter a password at the prompt. Use the same password that you entered in the previous Step
(cisco123)
```

- Step 11** Convert the sw1-xnc.p12 to a password protected Java KeyStore (tlsKeyStore) file using the **keytool** command. Use the relative path to jre/bin in the xnc directory to run the keytool command.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ ./relativePath/keytool -importkeystore -srckeystore
sw1-xnc.p12 -srcstoretype pkcs12 -destkeystore tlsKeyStore -deststoretype jks
Enter Destination Keystore password:cisco123
Re-enter new password:cisco123
Enter source keystore password:cisco123
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled.
```

**Note** By default an alias named “1” is stored in tlsKeyStore for the first switch. If the NDB controller is managing multiple switches, repeat this step for all the switches. When you add the second switch, the utility allows you to rename the first switch alias and also provides a provision to rename alias for the second switch. Refer examples as shown below.

```
keytool -importkeystore -srckeystore sw2-xnc.p12 -srcstoretype pkcs12 -destkeystore
tlsKeyStore -deststoretype jks
keytool -importkeystore -srckeystore sw3-xnc.p12 -srcstoretype pkcs12 -destkeystore
tlsKeyStore -deststoretype jks
```

- Step 12** List and verify content in the java tlsKeyStore using the **keytool** command.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ keytool -list -v -keystore tlsKeyStore | more
```

**What to do next**

Proceed to the subsequent task, *Creating the TLS TrustStore File*.

## Creating the TLS TrustStore File

TrustStore is created from the self-signed certificates that are generated for one or more switches. It holds certificates for one or more switches in the controller. This section describes how to create a Truststore using the self-signed certificate created in [Generating Self-Signed Certificate and Key](#) section. If you have multiple switches in the controller, each switch will have separate certificate file (For example, sw1-xnc-cert.pem, sw2-xnc-cert.pem)

## Procedure

---

- Step 1** Log in to the server.
- Step 2** Convert the certificate file (For example, sw1-xnc-cert.pem) to a Java TrustStore (tlsTrustStore) file using the **keytool** command. Enter a password when prompted to create a password protected Java TrustStore (tlsTrustStore) file. The password should be at least six characters.

### Example:

```
docker@docker-virtual-machine:~/TLS$keytool -import -alias sw1 -file sw1-xnc-cert.pem
-keystore tlsTrustStore
Enter Export Password: cisco123
Verifying - Enter Export Password: cisco123
Enter a password at the prompt. Use the same password that you entered in the previous Step
(cisco123)
```

- Note** If a NDB controller manages multiple switches, repeat this step for all the switches to add all switch keys into the same TrustStore. For example:

```
docker@docker-virtual-machine:~/TLS$ keytool -import -alias sw2 -file
sw2-xnc-cert.pem -keystore tlsTrustStore
docker@docker-virtual-machine:~/TLS$ keytool -import -alias sw3 -file
sw3-xnc-cert.pem -keystore tlsTrustStore
// Here sw2 and sw3 are alias for switch 2 and switch 3 for identification purpose.
```

- Step 3** List and verify keys for multiple switches in the same tlsTrustStore using the **keytool** command.

### Example:

```
docker@docker-virtual-machine:~/TLS$ keytool -list -v -keystore tlsTrustStore | more
```

---

## Starting NDB with TLS

To start NDB with TLS, complete these steps:

## Procedure

---

- Step 1** Log in to the NDB server.
- Step 2** Stop the NDB application, if running, using the **runxnc.sh** command
- Example:**
- ```
./runxnc.sh -stop
Controller with PID: 17426 -- Stopped!
```
- Step 3** Copy the tlsKeystore and tlsTruststore files that you created to configuration folder of NDB (xnc/configuration).

### Example:

```
cp tlskeystore /root/xnc/configuration
cp tlsTrustStore /root/xnc/configuration
```

- Step 4** Start the NDB application with TLS using the **runxnc.sh**.

**Example:**

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

**Note** To disable TLS, run the `./runxnc.sh -notls` command. To disable TLS and start NDB, run the `./runxnc.sh -notls -start` command. Before disabling TLS, ensure to *stop* NDB. After TLS is disabled, the port number for the devices connected to the NDB server should be changed to 80.

---

## Configuring TLS KeyStore and TrustStore Passwords on NDB

You need to configure TLS KeyStore and TrustStore passwords to enable NDB to read password protected TLS KeyStore and TrustStore files. To configure TLS KeyStore and TrustStore passwords on NDB, complete these steps:

**Procedure**

**Step 1** Log in to the NDB server.

**Step 2** Navigate to bin directory.

**Example:**

```
cd xnc/bin
```

**Step 3** Configure the TLS KeyStore and TrustStore passwords using the `xnc config-keystore-passwords` command.

**Example:**

```
./xnc config-keystore-passwords --user admin --password admin --url https://10.16.206.250:8443
--verbose --prompt --keystore-password cisco123 --truststore-password cisco123
```

In case NDB is configured with AAA (Tacacs/LDAP/Radius), and if the above command, `ndb config-keystore-passwords` fails, and you see a `401 unauthorized` error, then:

- a. Go to the `xnc` directory.
- b. Stop the NDB server using `./runxnc.sh -stop`.
- c. Enable the flag `enable.LocalUser.Authentication` by changing the value from `false` to `true` in the NDB `config.ini` file.
- d. Start the NDB server using `./runxnc.sh -start`.
- e. Run the `ndb config-keystore-passwords` command again.

**Note** In a HA environment, you need to run the above procedure for all the NDB servers in the cluster.

After the TLS is enabled on NDB, all the connections between NDB server and NDB switch are established using port 443. Ensure that you change device connections in NDB to use port 443.

Up on successfully completing these steps, you can add nexus switch in the controller using port 443. Use FQDN of the switch to add the device to the NDB controller.



You can verify the Certificate information using the WebUI Sandbox of the switch.

---

## Generating TLS 3rd Party Certification Between NDB Server and NDB Switch for NXAPI

This section describes how to generate TLS 3rd party certification between NDB server and NDB Switch. You need to request for a separate certificate and key for each switch in your network. TLS communication between NDB switch and NDB server uses port 443 only.

Complete the following steps to generate TLS 3rd party certification between NDB Server and NDB Switch for NXAPI:

- [Obtaining Certificates from a Certification Authority](#)
- [Creating TLS Keystore and Truststore Files for NDB Controller](#)
- [Starting NDB with TLS, on page 27](#)
- [Configuring TLS KeyStore and TrustStore Passwords on NDB, on page 28](#)



---

**Note** Complete all the steps under both the sections to ensure successful communication between the controller and the switch over TLS.

---

### Obtaining Certificates from a Certification Authority

You can obtain certificate from a Certification Authority (CA) in two ways. You can either directly approach a CA for both the private key and certificate. The CA will generate a private key on your behalf along with the certificate that contains the public key with issuing CA's signature.

In the other approach, you can generate a private key using tools such as openssl and generate a Certificate Signing Request (CSR) to a certificate issuing authority. The CA generates the certificates with public key using the user identity information from CSR.

#### Before you begin

Ensure that you have domain name configured in the switch using **ip domain-name** command for each NDB switch that acts as the Fully Qualified Domain Name (FQDN) for the switch. For example:

```
conf t
ip domain-name cisco.com
hostname N9k-117
end
```

The FQDN for the switch is configured to N9K-117.cisco.com.

## Procedure

---

### Step 1

Log in to the server.

### Step 2

Generate the private key (cert.key) and certificate signing request (cert.req) using openssl command.

#### Example:

```
docker@docker-virtual-machine:~/Mallik/TLS_CA$ openssl req -newkey rsa:2048 -sha256 -keyout
cert.key -keyform PEM -out cert.req -outform PEM
```

```
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'cert.key'
Enter PEM pass phrase:                               □ cisco123
Verifying - Enter PEM pass phrase:                   □ cisco123
-----
You are about to be asked to enter information that will be incorporated into your certificate
request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:CA
Locality Name (eg, city) [Newbury]:SJ
Organization Name (eg, company) [My Company Ltd]:cisco
Organizational Unit Name (eg, section) []:insbu
Common Name (eg, your name or your server's hostname) []:N9K-117.cisco.com
Email Address []:myname@cisco.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []: □ cisco123
An optional company name []:□ cisco123
```

```
docker@docker-virtual-machine: # ls
cert.key cert.req
```

### Step 3

Verify the CSR using the openssl command.

#### Example:

```
docker@docker-virtual-machine:~/Mallik/TLS_CA$ openssl req -noout -text -in cert.req
```

### Step 4

The private key is generated with a security passphrase. You may need to unencrypt the private key. To remove the passphrase from the private key, use the openssl command.

#### Example:

```
docker@docker-virtual-machine:~/Mk/TLS_CA$ ls
cert.key cert.req
docker@docker-virtual-machine:~/Mk/TLS_CA$ cp cert.key cert.keybkp
docker@docker-virtual-machine:~/Mk/TLS_CA$ rm cert.key
docker@docker-virtual-machine:~/Mk/TLS_CA$ openssl rsa -in cert.keybkp -out cert.key
```

```
Enter pass phrase for cert.keybkp: cisco123
```

**Note** Repeat this step to remove passphrase from private keys for all the switches.

**Note** Depending on the tier of the CA you choose, you can get up to three certificates (certificate chain) for each CSR. This means you get three certificates (root, intermediate and domain) from CA for each NDB switch. You need to check with CA to identify each type of certificate. Certificate naming convention might be different for different certifying authorities. For example: test-root-ca-2048.cer (root), test-ssl-ca.cer (intermediate), N9K-117.cisco.com.cer (domain).

Certificates are mostly shared in .PEM file format.

The cert.req file data needs to be submitted to 3rd party certification authority. Follow the relevant procedures and get the three (certificate) files.

**Step 5** Create a single certificate file from the three certificate files using the **cat** command. The concatenation should be done in the following order, domain certificate, root certificate, and intermediate certificate. Syntax for **cat** command: **Cat domain certificate root certificate intermediate certificate > server.cer.**

**Example:**

```
$cat N9K-117.cisco.com.cer test-root-ca-2048.cer test-ssl-ca.cer > server.cer
```

**Step 6** Edit the newly created server.cer file to separate the concatenated END and BEGIN lines. Do not delete anything in the file.

**Example:**

```
-----END CERTIFICATE-----BEGIN CERTIFICATE-----
```

```
///// Modify the above line like this by adding a line feed between the two.
```

```
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

**Note** Repeat this step all the switches.

**Step 7** Log into the NDB switch.

**Step 8** Copy the private key (cert.key) and the certificate from CA (server.cer) to the switch using the copy command.

**Example:**

```
N9K-117# copy scp://docker@10.16.206.250/home/docker/Mallik/TLS_CA_june_23/server.cer
bootflash:
Enter vrf (If no input, current vrf 'default' is considered): management
docker@10.16.206.250's password:
server.cer
```

```
100% 4676      4.6KB/s   00:00
```

```
Copy complete, now saving to disk (please wait)...
```

```
N9K-117# copy scp://docker@10.16.206.250/home/docker/Mallik/TLS_CA_june_23/cert.key bootflash:
```

```
Enter vrf (If no input, current vrf 'default' is considered): management
docker@10.16.206.250's password:
cert.key
```

```
100%
```

```
Copy complete, now saving to disk (please wait)...
```

**Note** Repeat this step for all the switches.

**Step 9** Configure the certificate file, sw1-ca.pem, and keyfile, sw1-ca.key in the switch using the **nxapi** command.

**Example:**

```
N9K-117 (config)# nxapi certificate httpskey keyfile bootflash:cert.key
```

```
Upload done. Please enable. Note cert and key must match.
```

```
N9K-117 (config)#
N9K-117 (config)# nxapi certificate httpsctrl certfile bootflash:server.cer
Upload done. Please enable. Note cert and key must match.
N9K-117 (config)#
```

**Note** If you have multiple switches, configure the corresponding certificate and private key to each switch.

**Step 10** Enable self-signed certificates on the switch using the **nxapi certificate** command.

**Example:**

```
N9K-117 (config)# nxapi certificate enable
N9K-117 (config)#
```

**Note** Ensure that there is no error while enabling self-signed certificates on the switch.

## Creating TLS Keystore and Truststore Files for NDB Controller

NDB uses certificates and keys to secure communication between switches. It stores the keys and certificates in keystores. These files are stored as `tlsTruststore` and `tlsKeystore` files in NDB. Complete the following steps to generate the Java `tlsKeyStore` and `tlsTrustStore` files for NDB Controller:

### Procedure

**Step 1** Create a TLS directory and navigate to it.

**Example:**

```
mkdir -p TLS
cd TLS
```

**Step 2** Create three directories under `mypersonalca` and two prerequisite files.

**Example:**

```
mkdir -p mypersonalca/certs
mkdir -p mypersonalca/private
mkdir -p mypersonalca/crl
echo "01" > mypersonalca/serial
touch mypersonalca/index.txt
```

Generate the TLS private key and Certification Authority (CA) files for each switch connected to NDB using the command

**Step 3** Generate the TLS private key and Certification Authority (CA) files for each switch connected to NDB using the **openssl req -x509 -nodes -days 3650 -newkey rsa:2048 -out mypersonalca/certs/sw1-ca.pem -outform PEM -keyout mypersonalca/private/sw1-ca.key** command.

This step generates the TLS private key in PEM format with a key length of 2048 bits, and the CA file (`mypersonalca/certs/sw1-ca.pem`, `mypersonalca/private/sw1-ca.key`). If you have multiple switches then you need to create `sw1-ca.pem` and `sw1-ca.key` file for all the switches with the exact values that were provided when generating CSR for these switches.

**Note** Use the same inputs which were provided when generating cert.key in the *Obtaining Certificates from a Certification Authority* section. Any mismatch in the inputs will generate a new key.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ openssl req -x509 -nodes -days 3650 -newkey rsa:2048
-out mypersonalca/certs/sw1-ca.pem -outform PEM -keyout mypersonalca/private/sw1-ca.key
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to 'mypersonalca/private/sw1-ca.key'
-----
You are about to be asked to enter information that will be incorporated into your certificate
request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:CA
Locality Name (eg, city) []:SJ
Organization Name (eg, company) [Internet Widgits Pty Ltd]:cisco
Organizational Unit Name (eg, section) []:insbu
Common Name (e.g. server FQDN or YOUR name) []:N9K-117.cisco.com
Email Address []:myname@cisco.com
```

Copy the cert.key and server.cer created in Obtaining Certificates from a Certification Authority section into the current directory (TLS). Select the certificate and key files for a single switch. These files were earlier generated for all the switches connecting to the controller. Using the server.cer and cert.key for the current switch, create the TLS KeyStore File.

**Step 4** Copy the cert.key and server.cer created in the *Obtaining Certificates from a Certification Authority* section into the current directory (TLS). Select the certificate and key files for a single switch. These files were earlier generated for all the switches connecting to the controller. Using the server.cer and cert.key for the current switch, create the TLS KeyStore File.

If multiple switches are connected, repeat this step for each switch separately.

**Step 5** Copy and convert the server.cer and cert.key files to .PEM format using the **copy** command.

**Example:**

```
cp cert.key sw1-xnc-privatekey.pem
cp server.cer sw1-xnc-cert.pem
```

**Step 6** Concatenate the private key (sw1-xnc-privatekey.pem) and certificate file (sw1-xnc-cert.pem) into a single .PEM file using the **cat** command.

**Example:**

```
cat sw1-xnc-privatekey.pem sw1-xnc-cert.pem > sw1-xnc.pem
```

**Step 7** Convert the .PEM file to .P12 format using the **openssl** command. Enter the export password when prompted. The password must contain at least 6 characters, for example, cisco123. The sw1-xnc.pem file is converted to a password-protected sw1-xnc.p12 file.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ openssl pkcs12 -export -out sw1-xnc.p12 -in sw1-xnc.pem
Enter Export Password: cisco123
Verifying - Enter Export Password: cisco123
Enter a password at the prompt. Use the same password that you entered in the previous Step
(cisco123)
```

- Step 8** Convert the sw1-xnc.p12 to a password protected Java KeyStore (tlsKeyStore) file using the **keytool** command. This command converts the sw1-xnc.p12 file to a password-protected tlsKeyStore file.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ keytool -importkeystore -srckeystore sw1-xnc.p12
-srcstoretype pkcs12 -destkeystore tlsKeyStore -deststoretype jks
Enter Destination Keystore password:cisco123
```

- Note** By default an alias named “1” is stored in tlsKeyStore for the first switch. If the NDB controller is managing multiple switches, repeat this step for all the switches. When you add the second switch, the utility allows you to rename the first switch alias and also provides a provision to rename alias for the new switch. For example, see below.

```
keytool -importkeystore -srckeystore sw2-xnc.p12 -srcstoretype pkcs12 -destkeystore
tlsKeyStore -deststoretype jks
keytool -importkeystore -srckeystore sw3-xnc.p12 -srcstoretype pkcs12 -destkeystore
tlsKeyStore -deststoretype jks
```

- Step 9** List and verify content in the java tlsKeyStore using the keytool command.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ keytool -list -v -keystore tlsKeyStore | more
```

- Step 10** Convert the certificate file (sw1-xnc-cert.pem) to a Java TrustStore (tlsTrustStore) file using the **keytool** command. Enter a password when prompted to create a password protected Java TrustStore (tlsTrustStore) file. The password should be at least six characters.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ keytool -import -alias sw1 -file sw1-xnc-cert.pem
-keystore tlsTrustStore
Enter keystore password: cisco123
Re-enter new password: cisco123
Owner: EMAILADDRESS=myname@cisco.com, CN=localhost, OU=insbu, O=cisco, L=SJ, ST=CA, C=US
Issuer: EMAILADDRESS=myname@cisco.com, CN=localhost, OU=insbu, O=cisco, L=SJ, ST=CA, C=US
Serial number: c557f668a0dd2ca5
Valid from: Thu Jun 15 05:43:48 IST 2017 until: Sun Jun 13 05:43:48 IST 2027
Certificate fingerprints:
MD5: C2:7B:9E:26:31:7A:74:25:55:DF:A7:91:C9:5D:20:A3
SHA1: 3C:DF:66:96:72:12:CE:81:DB:AB:58:30:60:E7:CC:04:4D:DF:6D:B2
SHA256:
DD:FB:3D:71:B4:B8:9E:CE:97:A3:E4:2D:D3:B6:90:CD:76:A8:5F:84:77:78:BE:49:6C:04:01:84:62:2C:2F:EB
Signature algorithm name: SHA256withRSA
Version: 3

Extensions:

#1: ObjectId: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 0D B3 CF 81 66 4A 33 4E EF 86 7E 26 C3 50 9B 73 ....fJ3N...&.P.s
0010: 38 EF DF 40 8..@
]
]
```

```
#2: ObjectId: 2.5.29.19 Criticality=false
BasicConstraints:[
CA:true
PathLen:2147483647
]

#3: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 0D B3 CF 81 66 4A 33 4E EF 86 7E 26 C3 50 9B 73 ....fJ3N...&.P.s
0010: 38 EF DF 40 8..@
]
]

Trust this certificate? [no]: yes
Certificate was added to keystore
```

**Note** If a NDB controller manages multiple switches, repeat this step for all the switches to add all switch keys into the same TrustStore. For example:

```
keytool -import -alias sw2 -file sw2-xnc-cert.pem -keystore tlsTrustStore
keytool -import -alias sw3 -file sw3-xnc-cert.pem -keystore tlsTrustStore
```

**Step 11** List and verify keys for multiple switches in the same `tlsTrustStore` using the `keytool` command.

**Example:**

```
docker@docker-virtual-machine:~/TLS$ keytool -list -v -keystore tlsTrustStore | more
```

## Starting NDB with TLS

To start NDB with TLS, complete these steps:

### Procedure

**Step 1** Log in to the NDB server.

**Step 2** Stop the NDB application, if running, using the `runxnc.sh` command

**Example:**

```
./runxnc.sh -stop
Controller with PID: 17426 -- Stopped!
```

**Step 3** Copy the `tlsKeystore` and `tlsTruststore` files that you created to configuration folder of NDB (`xnc/configuration`).

**Example:**

```
cp tlskeystore /root/xnc/configuration
cp tlsTrustStore /root/xnc/configuration
```

**Step 4** Start the NDB application with TLS using the `runxnc.sh`.

**Example:**

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

**Note** To disable TLS, run the `./runxnc.sh -notls` command. To disable TLS and start NDB, run the `./runxnc.sh -notls -start` command. Before disabling TLS, ensure to `stop` NDB. After TLS is disabled, the port number for the devices connected to the NDB server should be changed to 80.

## Configuring TLS KeyStore and TrustStore Passwords on NDB

You need to configure TLS KeyStore and TrustStore passwords to enable NDB to read password protected TLS KeyStore and TrustStore files. To configure TLS KeyStore and TrustStore passwords on NDB, complete these steps:

### Procedure

**Step 1** Log in to the NDB server.

**Step 2** Navigate to bin directory.

**Example:**

```
cd xnc/bin
```

**Step 3** Configure the TLS KeyStore and TrustStore passwords using the `xnc config-keystore-passwords` command.

**Example:**

```
./xnc config-keystore-passwords --user admin --password admin --url https://10.16.206.250:8443
--verbose --prompt --keystore-password cisco123 --truststore-password cisco123
```

In case NDB is configured with AAA (Tacacs/LDAP/Radius), and if the above command, `ndb config-keystore-passwords` fails, and you see a `401 unauthorized` error, then:

- a. Go to the `xnc` directory.
- b. Stop the NDB server using `./runxnc.sh -stop`.
- c. Enable the flag `enable.LocalUser.Authentication` by changing the value from `false` to `true` in the NDB `config.ini` file.
- d. Start the NDB server using `./runxnc.sh -start`.
- e. Run the `ndb config-keystore-passwords` command again.

**Note** In a HA environment, you need to run the above procedure for all the NDB servers in the cluster.

After the TLS is enabled on NDB, all the connections between NDB server and NDB switch are established using port 443. Ensure that you change device connections in NDB to use port 443.

Up on successfully completing these steps, you can add nexus switch in the controller using port 443. Use FQDN of the switch to add the device to the NDB controller.

You can verify the Certificate information using the WebUI Sandbox of the switch.



# Generating TLS Self-Signed Certification Between NDB Server and NDB Switch for OpenFlow

Complete the following steps to generate TLS self-signed certification between NDB Server and NDB Switch for OpenFlow:

## Procedure

---

**Step 1** Create a TLS directory using the **mkdir** **directory\_name** and navigate to the new directory.

### Example:

```
[ ]# mkdir -p TLS
[ ]# cd TLS
```

**Step 2** Create 3 directories under mypersonalca folder for CA system.

### Example:

```
[ ]# mkdir -p mypersonalca/certs
[ ]# mkdir -p mypersonalca/private
[ ]# mkdir -p mypersonalca/crl
```

**Step 3** Initialize the serial file using the **echo** command. The serial file and the index.txt file are used by the CA to maintain its database of the certificate files.

### Example:

```
[ ]# echo "01" > mypersonalca/serial
```

**Step 4** Initialize the index file using the **touch** command.

### Example:

```
[ ]# touch mypersonalca/index.txt
```

**Step 5** Create a CA configuration file and configure the **alt\_names** section with relevant IP Addresses.

### Example:

```
[ ca ]
default_ca = mypersonalca
[ mypersonalca ]
#
# WARNING: if you change that, change the default_keyfile in the [req] section below too
# Where everything is kept
dir = ./mypersonalca
# Where the issued certs are kept
certs = $dir/certs
# Where the issued crl are kept
crl_dir = $dir/crl
# database index file
database = $dir/index.txt
# default place for new certs
```

```

new_certs_dir = $dir/certs
#
# The CA certificate
certificate = $dir/certs/ca.pem
# The current serial number
serial = $dir/serial
# The current CRL
crl = $dir/crl/crl.pem
# WARNING: if you change that, change the default_keyfile in the [req] section below too
# The private key
private_key = $dir/private/ca.key
# private random number file
RANDFILE = $dir/private/.rand
# The extensions to add to the cert
x509_extensions = usr_cert
# how long to certify for
default_days = 365
# how long before next CRL
default_crl_days= 30
# which md to use; people in comments indicated to use sha1 here
default_md = sha1
# keep passed DN ordering
preserve = no
# Section names
policy = mypolicy
x509_extensions = certificate_extensions
[ mypolicy ]
# Use the supplied information
commonName = supplied
stateOrProvinceName = optional
countryName = optional
emailAddress = optional
organizationName = optional
organizationalUnitName = optional
[ certificate_extensions ]
# The signed certificate cannot be used as CA
basicConstraints = CA:false
[ req ]
# same as private_key
default_keyfile = ./mypersonalca/private/ca.key
# Which hash to use
default_md = sha1
# No prompts
prompt = no
# This is for CA
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid:always,issuer
string_mask = utf8only
basicConstraints = CA:true
distinguished_name = root_ca_distinguished_name
x509_extensions = root_ca_extensions
[ root_ca_distinguished_name ]
commonName = Controller
stateOrProvinceName = Mass
countryName = US
emailAddress = root_ca_userid@cisco.com
organizationName = Cisco
[ root_ca_extensions ]
basicConstraints = CA:true

```

- Step 6** Generate the TLS private key, certificate, and Certification Authority (CA) files using the **openssl req** command. The TLS private key is created in PEM format with a key length of 2048 bits and the CA file.

**Example:**

```

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

```

**Note** This step generates the TLS private key in PEM format with a key length of 2048 bits, and the CA file.

**Step 7** Generate the certificate key and certificate request files, using the **openssl req** command.

**Example:**

```

openssl req -newkey rsa:2048 -keyout cert.key -keyform PEM -out cert.req -outform PEM
Generating a 2048 bit RSA private key
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'cert.key'
Enter PEM pass phrase:(Enter pwd123 here)
Verifying - Enter PEM pass phrase:(Enter pwd123 here)
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Mass
Locality Name (eg, city) []:San
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Cisco

```

```

Organizational Unit Name (eg, section) []:NDB
Common Name (e.g. server FQDN or YOUR name) []:Controller
Email Address []:masavanu@cisco.com
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:pwd123
An optional company name []:

```

**Note** This step generates the controller key (cert.key) and certificate request (cert.req) files in PEM format.

**Step 8** Generate the certificate file, using the **openssl ca** command.

**Example:**

```

openssl ca -batch -notext -in cert.req -out cert.pem -config ca.cnf
Using configuration from ca.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName :PRINTABLE:'US'
stateOrProvinceName :ASN.1 12:'Mass'
localityName :ASN.1 12:'San'
organizationName :ASN.1 12:'Cisco'
organizationalUnitName:ASN.1 12:'NDB'
commonName :ASN.1 12:'Controller'
emailAddress :IA5STRING:'masavanu@cisco.com'
Certificate is to be certified until May 17 02:59:40 2017 GMT (365 days)
Write out database with 1 new entries
Data Base Updated

```

**Note** This step generates the certificate (cert.pem) file in PEM format using the certificate request (cert.req) and the certificate configuration (ca.cnf) files.

**Step 9** Configure the Cryptographic Keys on the Switch.

**Example:**

```

switch(config)# ip domain-name domain-name
//Configures the domain name for the switch
switch(config)# crypto key generate rsa label myKey2 exportable modulus 2048
//Generates the cryptographic key.
switch(config)# crypto ca trustpoint myCA
//Enters the trustpoint configuration mode and installs the trustpoint file on the switch
switch(config-trustpoint)# rsakeypair myKey2
//Installs the key files on the switch
switch(config-trustpoint)# exit
//Exits trustpoint configuration mode
switch# show crypto ca trustpoints
//(Optional) Verifies creation of the trustpoint files.
switch# show crypto key mypubkey rsa
//(Optional) Verifies creation of the key files.
cat mypersonalca/certs/ca.pem
//Displays the certificate file on the machine hosting the generated TLS certificates
switch(config)# crypto ca authenticate myCA
-----BEGIN CERTIFICATE-----
-----END CERTIFICATE-----
END OF INPUT:

```

```

Fingerprint(s): SHA1
Fingerprint=56:0F:56:85:6A:07:A1:44:6C:F4:4C:45:CF:CC:BA:47:22:17:1D:93
Do you accept this certificate [yes/no]:yes

switch(config)# crypto ca enroll myCA
//Generates the certificate request on the switch

Create the certificate request ..
Create a challenge password. You will need to verbally provide this
password to the CA Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.
Password:pwd123
The subject name in the certificate will be the name of the switch.
Include the switch serial number in the subject name [yes/no]:no
Include an IP address in the subject name [yes/no]:no
Include the Alternate Subject Name [yes/no]:no
The certificate request will be displayed...
-----BEGIN CERTIFICATE REQUEST-----
.....
-----END CERTIFICATE REQUEST-----
openssl ca -in n3k-cert.req -out newcert.pem -config ./ca.cnf
//Copies the certificate request from the switch to the file n3k-cert.req on your Linux
machine, and then uses it to generate the switch certificate.
Using configuration from ./ca.cnf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
commonName :PRINTABLE:'ndb-3172-4.cisco.com'
Certificate is to be certified until May 17 04:27:57 2017 GMT (365 days)
Sign the certificate [y/n]:y
out of 1 certificate requests certified, commit [y/n]y
Write out database with 1 new entries
Data Base Updated

cat newcert.pem
//Copies the certificate (newcert.pem) to the switch

switch(config)# crypto ca import myCA certificate
switch(config)# crypto ca import myCA certificate
input (cut & paste) certificate in PEM format:

switch# show crypto ca certificates
//Displays the certificates on the switch

```

## Step 10 Enable the TLS OpenFlow Switches

### Example:

```

switch(config)# openflow
//Enters OpenFlow agent configuration mode on the switch.
switch(config-ofa)# switch 1
//Enters OpenFlow agent configuration mode for switch 1.
switch(config-ofa)# tls trust-point local myCA remote myCA
//Enables TLS certificate authority on the switch.
switch(config-ofa-switch)# pipeline{201/203}
//Configures the pipeline

```

```
switch(config-ofa-switch)#controller ipv4 {A.B.C.D} port 6653 vrf management security tls
//Enables TLS for OpenFlow switches
```

**Step 11** Create the TLS KeyStore File**Example:**

```
cp cert.key xnc-privatekey.pem
//Copy cert.key to xnc-privatekey.pem
cp cert.pem xnc-cert.pem
//Copy cert.pem to xnc-cert.pem under TLS folder
cat xnc-privatekey.pem xnc-cert.pem > xnc.pem
Creates the xnc.pem file, which contains the private key and certificate.

openssl pkcs12 -export -out xnc.p12 -in xnc.pem
//Convert the PEM file xnc.pem file to the file xnc.p12
Enter a password at the prompt
The xnc.pem file is converted to a password-protected .p12 file.
openssl pkcs12 -export -out xnc.p12 -in xnc.pem
Enter pass phrase for xnc.pem: (enter pass phrase as pwd123)
Enter Export Password: (Enter Export password as pwd123)
Verifying - Enter Export Password: (Enter as pwd123)
```

**Step 12** Convert the xnc.p12 file to password protected TLS KeyStore**Example:**

```
keytool -importkeystore -srckeystore xnc.p12 -srcstoretype pkcs12 -destkeystore tlsKeyStore
-deststoretype jks
Enter destination keystore password: (Enter pwd123)
Re-enter new password: (Enter pwd123)
Enter source keystore password: (Enter pwd123)
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries
failed or cancelled
```

**Step 13** Enter the TLS KeyStore password.**Step 14** Create the TLS TrustStore File.**Example:**

```
cp mypersonalca/certs/ca.pem sw-cacert.pem
//Copies the mypersonalca/certs/ca.pem file to sw-cacert.pem
keytool -import -alias swca1 -file sw-cacert.pem -keystore tlsTrustStore
Enter keystore password: (Enter pwd123)
Re-enter new password: (Enter pwd123)
Owner: EMAILADDRESS=masavanu@cisco.com, CN=Controller, OU=NDB, O=Cisco,
L=San, ST=Mass, C=US
Issuer: EMAILADDRESS=masavanu@cisco.com, CN=Controller, OU=NDB, O=Cisco,
L=San, ST=Mass, C=US
Serial number: d764c5b1e5e6b531
Valid from: Mon May 16 22:49:13 EDT 2016 until: Thu May 14 22:49:13 EDT
2026
Certificate fingerprints:
MD5: BD:C8:21:13:D0:7F:ED:A4:B4:FA:97:9A:D0:EA:12:78
SHA1: 56:0F:56:85:6A:07:A1:44:6C:F4:4C:45:CF:CC:BA:47:22:17:1D:93
SHA256:
09:32:74:12:BF:56:04:07:42:8C:D8:1B:78:AD:7A:40:0D:51:AA:56:91:B1:1A:18:90:6A:A5:A0:44:04:6A:EC
Signature algorithm name: SHA256withRSA
```

```

Version: 3
Extensions:
#1: ObjectID: 2.5.29.35 Criticality=false
AuthorityKeyIdentifier [
KeyIdentifier [
0000: 78 C5 2B 09 7F AF EC 86 FE 50 EA 6C 8A 56 B3 BE x.+.....P.l.V..
0010: BE F2 97 98 ....
]
]
#2: ObjectID: 2.5.29.19 Criticality=false
BasicConstraints:[
CA:true
PathLen:2147483647
]
#3: ObjectID: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 78 C5 2B 09 7F AF EC 86 FE 50 EA 6C 8A 56 B3 BE x.+.....P.l.V..
0010: BE F2 97 98 ....
]
]
Trust this certificate [no]: yes
Certificate was added to keystore

```

**Note** The sw-cacert.pem file is converted into a password-protected Java TrustStore (tlsTrustStore) file.

## Starting the NDB Application with TLS Enabled

Complete the following steps to start NDB application with TLS enabled.

### Before you begin

Copy TLS Truststore and TLS Keystore files created under TLS folder to the configuration directory of Cisco Nexus Data Broker.

### Procedure

**Step 1** Start the NDB application using the **runxnc.sh** script.

#### Example:

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

#### Example:

To start NDB with default username (admin) and a non-default password (for example, pwd123):

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

If XNC password is changed, OSGi webconsole password needs to be changed,

to set non-default OSGi webconsole password Enter XNC Admin Password  
[default] :( Type the non-default password which was set.

**Example:**

To start NDB with default username (admin) and password (admin):

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

**Step 2**

Configure the TLS KeyStore and TrustStore passwords in Cisco NDB. You need to configure TLS KeyStore and TrustStore Passwords to enable NDB to read the password-protected TLS KeyStore and TrustStore files.

**Example:**

```
xnc/bin directory# ./xnc config-keystore-passwords --user admin --password admin --url
https://NDB_URL:8443
--verbose --prompt --keystore-password pwd123 --truststore-password pwd123
```

If the TLS KeyStore and TrustStore passwords configuration fails with Failed to connect to the controller, you need to change the protocol to HTTP.

```
./xnc config-keystore-passwords --user admin --password admin --url https://localhost:8443
--verbose --prompt --keystore-password pwd123 --truststore-password pwd123
[Info] Sending request: https://10.16.206.189:8443/controller/osgi/system/console/vmstat
---- REQUEST HEADERS ----
GET https://10.16.206.189:8443/controller/osgi/system/console/vmstat HTTP/1.1
-----
[Error] Failed to connect to the controller at "https://10.16.206.189:8443". Controller may
not be running.
javax.net.ssl.SSLHandshakeException: java.security.cert.CertificateException:
No subject alternative names present
at sun.security.ssl.Alerts.getSSLException(Alerts.java:192)
at sun.security.ssl.SSLSocketImpl.fatal(SSLSocketImpl.java:1949)
at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:302)
at sun.security.ssl.Handshaker.fatalSE(Handshaker.java:296)
at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1509)
at sun.security.ssl.ClientHandshaker.processMessage(ClientHandshaker.java:216)
at sun.security.ssl.Handshaker.processLoop(Handshaker.java:979)
at sun.security.ssl.Handshaker.process_record(Handshaker.java:914)
at sun.security.ssl.SSLSocketImpl.readRecord(SSLSocketImpl.java:1062)
at sun.security.ssl.SSLSocketImpl.performInitialHandshake(SSLSocketImpl.java:1375)
at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1403)
at sun.security.ssl.SSLSocketImpl.startHandshake(SSLSocketImpl.java:1387)
at sun.net.www.protocol.https.HttpsClient.afterConnect(HttpsClient.java:559)
at
sun.net.www.protocol.https.AbstractDelegateHttpsURLConnection.connect(AbstractDelegateHttpsURLConnection.java:185)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream0(HttpURLConnection.java:1513)
at sun.net.www.protocol.http.HttpURLConnection.getInputStream(HttpURLConnection.java:1441)
at java.net.HttpURLConnection.getResponseCode(HttpURLConnection.java:480)
at
sun.net.www.protocol.https.HttpsURLConnectionImpl.getResponseCode(HttpsURLConnectionImpl.java:338)
at com.cisco.csdn.cli.online.HttpClient$HttpResponse.<init>(HttpClient.java:191)
at com.cisco.csdn.cli.online.HttpClient.sendRequest(HttpClient.java:108)
at com.cisco.csdn.cli.online.HttpClient.get(HttpClient.java:92)
at com.cisco.csdn.cli.online.OnlineCommand.isRunning(OnlineCommand.java:88)
at
com.cisco.csdn.cli.online.ConfigKeyStorePasswordCommand.processCommand(ConfigKeyStorePasswordCommand.java:46)
```



```

at com.cisco.csdn.cli.Cli.processCommand(Cli.java:70)
at com.cisco.csdn.cli.Main.main(Main.java:33)
Caused by: java.security.cert.CertificateException: No subject alternative names present
at sun.security.util.HostnameChecker.matchIP(HostnameChecker.java:144)
at sun.security.util.HostnameChecker.match(HostnameChecker.java:93)
at sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:455)
at sun.security.ssl.X509TrustManagerImpl.checkIdentity(X509TrustManagerImpl.java:436)
at sun.security.ssl.X509TrustManagerImpl.checkTrusted(X509TrustManagerImpl.java:200)
at sun.security.ssl.X509TrustManagerImpl.checkServerTrusted(X509TrustManagerImpl.java:124)

at sun.security.ssl.ClientHandshaker.serverCertificate(ClientHandshaker.java:1491)
... 20 more

//Change the protocol to HTTP.
./xnc config-keystore-passwords --user admin --password admin --url http://localhost:8080
--verbose --prompt --keystore-password pwd123 --truststore-password pwd123

```

## Generating TLS Self-Signed Certification Between WebUI Browser and NDB Server

You can secure communication between a Web browser and NDB server running in centralized mode using self-signed certificates. This section describes how to generate a self-signed certificate to secure communication between a WebUI browser and NDB application. By default Cisco NDB is shipped with default certificate which is issued to Cisco NDB and issued by Cisco NDB with default validity. You can use the **generateWebUICertificate.sh** script under configuration folder to create self-signed certificates. For Cisco NDB releases 3.5 and earlier, these certificates are valid for 6 months. Starting with Cisco NDB release 3.6, default validity of a certificate is 6 months but you can configure the validity of a certificate.



**Note** You can create self-signed TLS certificates for NDB in Centralized mode only.

- Generating TLS Self-Signed Certification Between WebUI Browser and NDB Server Running in Centralized Mode

## Generating TLS Self-Signed Certification Between WebUI Browser and NDB Server Running in Centralized Environment

Complete the following steps to generate TLS self-signed certification between WebUI Browser and NDB Server running in Centralized mode:

### Procedure

**Step 1** Log into the NDB server and change the current directory `\xnc\configuration`.

#### Example:

```
[root@RHEL-VM-NDB-ACI]# cd \xnc\configuration
```

**Step 2** Generate the TLS self-signed certificate using the `generateWebUICertificate.sh` script.

**Example:**

```
[root@RHEL-VM-NDB-ACI configuration]# ./generateWebUICertificate.sh

*****
Enter Fully qualified domain name :
*****
NDB-browser This can be FQDN of the NDB java application as well
*****
Enter Organizational unit :
*****
INSBU
*****
Enter Organization :
*****
cisco
*****
Enter Location :
*****
SJ
*****
Enter State :
*****
CA
*****
Enter Country :
*****
USA
*****
Enter keypass :
*****
cisco123
*****
Enter storepass :
*****
cisco123
*****
Enter the validity in number of days :
*****
365 in NDB 3.5 this script will let you to specify the certificate
validity.
*****
Below process will rename the existing key file to <old_keystore>, will
generate a new key file. Do you want to continue (y/n) ?
*****
Y
*****
Self-Signed Certificate Created
*****
Alias name: cisco
```

```

Creation date: Jan 6, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Issuer: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Serial number: b404be5
Valid from: Sun Jan 06 20:22:05 PST 2019 until: Mon Jan 06 20:22:05 PST
2020
Certificate fingerprints:
    MD5: 71:07:F6:4E:57:6A:08:3A:AD:06:32:B3:6C:5F:8F:52
    SHA1: 04:08:B9:D5:B7:EB:ED:E0:F9:22:49:14:FA:C6:09:39:22:32:43:A2

    SHA256:
34:D9:EB:34:0A:52:D1:4A:DD:F1:8B:14:D0:84:E4:1C:57:8B:2B:99:9B:E5:A1:4C:C7:8C:CD:AE:24:31:49:75

    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 63 8A 92 8F 6F 0F 45 BD   EE 55 C5 A8 99 3B F6 F7   c...o.E..U...;...
0010: AC FA 4A 21                               ..J!
]
]

*****
Displayed the generated keystore
*****
*****
Configured the keystore details on tomcat-server.xml
*****
*****
The newly generated key will used on next NDB restart. Do you want to
restart NDB now (y/n) ?
*****
Y
Doesn't seem any Controller daemon is currently running
Running controller in background with PID: 13573, to connect to it please
SSH to this host on port 2400
NDB GUI can be accessed using below URL:
[https://10.16.206.160:8443]
[https://[fe80::250:56ff:fe90:b764]:8443]
[https://10.16.206.159:8443]
[https://192.168.1.123:8443]
[https://[fe80::250:56ff:fe90:9c79]:8443]

*****

```

```
NDB Restarted
*****
```

**Note** The `generateWebUICertificate.sh` script reloads the NDB application to ensure that the browser starts using this certificate when we access NDB java application from the browser.

**Step 3** Decode the generated certificate using the `keytool -list -v -keystore keystore_Name` command. Enter the store password when prompted.

**Example:**

```
[root@RHEL-VM-NDB-ACI configuration]# keytool -list -v -keystore keystore
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: cisco
Creation date: Jul 6, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Issuer: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Serial number: b404be5
Valid from: Sun Jan 06 20:22:05 PST 2019 until: Mon Jan 06 20:22:05 PST 2020
Certificate fingerprints:
    MD5: 71:07:F6:4E:57:6A:08:3A:AD:06:32:B3:6C:5F:8F:52
    SHA1: 04:08:B9:D5:B7:EB:ED:E0:F9:22:49:14:FA:C6:09:39:22:32:43:A2
    SHA256:
34:D9:EB:34:0A:52:D1:4A:DD:F1:8B:14:D0:84:E4:1C:57:8B:2B:99:9B:E5:A1:4C:C7:8C:CD:AE:24:31:49:75

    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 63 8A 92 8F 6F 0F 45 BD EE 55 C5 A8 99 3B F6 F7 c...o.E..U...;..
0010: AC FA 4A 21 ..J!
]
]

*****
*****
```

**Step 4** The self-signed certificates are generated in JKS format which is not compatible with the browsers. Hence, you need to convert these certificates into PKCS12 format before importing the certificate in the browser. Complete the following steps to convert JKS format certificate to PKCS12 format. Convert JKS format certificate into PKCS12 format using the `keytool` command.

**Note** Ensure that you keep a copy of the original certificates before proceeding with the conversion.

**Example:**

```
keytool -importkeystore -srckeystore keystore -srcstorepass cisco123 -srckeypass cisco123
```

```
-destkeystore keystore.p12 -deststoretype PKCS12 -srcalias cisco -deststorepass cisco123
-destkeypass cisco123
```

**Note** The inputs in the **keytool** command should match the inputs provided during UI certificate generation.

**Note** The resulting certificate file (keystore.p12) is in PKSC12 format.

- Step 5** Add this certificate to Trusted Root certificate store on the browser. See help for respective Web browsers about how to add the certificate to the Trusted Root certificate store.

## Generating TLS Self-Signed Certification Between Web Browser and NDB Server Running in Embedded Mode

You can generate TLS self-signed certification between a Web browser and NDB server running in Embedded mode in the Guestshell Environment.

### Generating TLS Self-Signed Certificate Between Web Browser and NDB Server Running in Embedded Mode Using Guest Shell Environment

To generate TLS self-signed certificate between Web browser and NDB server running in embedded mode using Guest Shell environment, complete the following steps.

#### Procedure

- Step 1** Connect to Guest Shell using **guestshell** command.

**Example:**

```
N9K-C93108TC-EX-108# guestshell
[admin@guestshell ~]$
[admin@guestshell ~]$
```

- Step 2** Change the current directory to `\ndb\configuration`.

**Example:**

```
[admin@guestshell ~]$ cd \ndb\configuration
```

- Step 3** Edit the `generateWebUICertificate.sh` script. Update the Keytool attribute in the file (two instances) with the NDB OVA JRE path.

**Example:**

```
cmd1=echo /home/admin/ndb/jre/bin/keytool -genkey -noprompt -trustcacerts -keyalg RSA -alias
cisco -dname "CN=$CN, OU=$OU, O=$O, L=$L, ST=$ST, C=$C" -keypass $Keypass -keystore $eig
-storepass $Storepass -validity $validityVal
```

```
cmd2=echo /home/admin/ndb/jre/bin/keytool -alias cisco -list -v -keystore keystore -storepass
$Storepass;
```

- Step 4** Set Java home path using the **export** command.

**Example:**

```
[admin@guestshell ~]$ export JAVA_HOME='/usr/lib/jvm/i586-wrs-jre'
```

**Step 5** Generate the TLS self-signed certificate using the `/home/admin/ndb/configuration/generateWebUICertificate.sh` script.

**Example:**

```
[root@RHEL-VM-NDB-ACI configuration]# ./generateWebUICertificate.sh

*****
Enter Fully qualified domain name :
*****
NDB-browser  This can be FQDN of the NDB java application as well
*****
Enter Organizational unit :
*****
INSBU
*****
Enter Organization :
*****
cisco
*****
Enter Location :
*****
SJ
*****
Enter State :
*****
CA
*****
Enter Country :
*****
USA
*****
Enter keypass :
*****
cisco123
*****
Enter storepass :
*****
cisco123
*****
Enter the validity in number of days :
*****
365  in NDB 3.5 this script will let you to specify the certificate
validity.
*****
Below process will rename the existing key file to <old_keystore>, will
generate a new key file. Do you want to continue (y/n) ?
*****
Y
*****
Self-Signed Certificate Created
```

```

*****
Alias name: cisco
Creation date: Jan 6, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Issuer: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Serial number: b404be5
Valid from: Sun Jan 06 20:22:05 PST 2019 until: Mon Jan 06 20:22:05 PST
2020
Certificate fingerprints:
    MD5: 71:07:F6:4E:57:6A:08:3A:AD:06:32:B3:6C:5F:8F:52
    SHA1: 04:08:B9:D5:B7:EB:ED:E0:F9:22:49:14:FA:C6:09:39:22:32:43:A2

    SHA256:
34:D9:EB:34:0A:52:D1:4A:DD:F1:8B:14:D0:84:E4:1C:57:8B:2B:99:9B:E5:A1:4C:C7:8C:CD:AE:24:31:49:75

    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 63 8A 92 8F 6F 0F 45 BD EE 55 C5 A8 99 3B F6 F7 c...o.E..U...;...
0010: AC FA 4A 21 ..J!
]
]

*****
Displayed the generated keystore
*****
*****
Configured the keystore details on jetty-ssl-context.xml
*****
*****
The newly generated key will used on next NDB restart. Do you want to
restart NDB now (y/n) ?
*****
y
Doesn't seem any Controller daemon is currently running
Running controller in background with PID: 13573, to connect to it please
SSH to this host on port 2400
NDB GUI can be accessed using below URL:
[https://10.16.206.160:8443]
[https://[fe80::250:56ff:fe90:b764]:8443]
[https://10.16.206.159:8443]
[https://192.168.1.123:8443]
[https://[fe80::250:56ff:fe90:9c79]:8443]

```

```
*****
NDB Restarted
*****
```

**Note** The `generateWebUICertificate.sh` script reloads the NDB application to ensure that the browser starts using this certificate when we access NDB java application from the browser.

**Step 6** Decode the generated certificate using the `keytool -list -v -keystore keystore_Name` command. Enter the store password when prompted.

**Example:**

```
[root@RHEL-VM-NDB-ACI configuration]# /home/admin/ndb/jre/bin/keytool -list -v -keystore
keystore
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: cisco
Creation date: Jul 6, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Issuer: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Serial number: b404be5
Valid from: Sun Jan 06 20:22:05 PST 2019 until: Mon Jan 06 20:22:05 PST 2020
Certificate fingerprints:
    MD5: 71:07:F6:4E:57:6A:08:3A:AD:06:32:B3:6C:5F:8F:52
    SHA1: 04:08:B9:D5:B7:EB:ED:E0:F9:22:49:14:FA:C6:09:39:22:32:43:A2
    SHA256:
34:D9:EB:34:0A:52:D1:4A:DD:F1:8B:14:D0:84:E4:1C:57:8B:2B:99:9B:E5:A1:4C:C7:8C:CD:AE:24:31:49:75

    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 63 8A 92 8F 6F 0F 45 BD EE 55 C5 A8 99 3B F6 F7 c...o.E..U...;..
0010: AC FA 4A 21 ..J!
]
]

*****
*****
```

**Step 7** The self-signed certificates are generated in JKS format which is not compatible with the browsers. You need to convert these certificates into PKCS12 format before importing the certificate in the browser. Complete the following steps to convert JKS format certificate to PKCS12 format. Convert JKS format certificate into PKCS12 format using the `keytool` command.

**Note** Ensure that you keep a copy of the original certificates before proceeding with the conversion.

**Example:**



```
/home/admin/ndb/jre/bin/keytool -importkeystore -srckeystore keystore -srcstorepass cisco123
-srckeypass cisco123 -destkeystore keystore.p12 -deststoretype PKCS12 -srcalias cisco
-deststorepass cisco123 -destkeypass cisco123
```

**Note** The inputs in the **keytool** command should match the inputs provided during UI certificate generation.

**Note** The resulting certificate file (keystore.p12) is in PKSC12 format.

- Step 8** Upload the CA certificate into Trusted Root certificate store of Web browser. See help for respective Web browsers about how to add the certificate to the Trusted Root certificate store. Use the password that you created while creating the certificate when prompted while uploading the certificate to the Web browser.
- Step 9** Restart the Guest Shell to restart the NDB.

## Generating TLS Self-Signed Certificate Between Web Browser and NDB Server Running in Embedded Mode Using OVA Environment

To generate TLS self-signed certificate between Web browser and NDB server running in embedded mode using OVA environment, complete the following steps.

### Procedure

- Step 1** Connect to virtual service console using the **virtual-service connect** command.

**Example:**

```
N3K-3172# virtual-service connect name ova console
Connecting to virtual-service. Exit using ^c^c
Entering character mode
Escape character is '^]'.
root@N3K-3172:~#
root@N3K-3172:~#
```

- Step 2** Change current directory to /xnclite/xnc/configuration.

**Example:**

```
root@ N9396TX-116:~# cd /xnclite/xnc/configuration
root@ N9396TX-116:/xnclite/xnc/configuration#
```

- Step 3** Edit the generateWebUICertificate.sh script. Update the Keytool attribute in the file(two instances) with the NDB OVA JRE path.

**Example:**

```
cmd1=echo /usr/lib/jvm/i586-wrs-jre/bin/keytool -genkey -noprompt -trustcacerts -keyalg RSA
-alias cisco -dname "CN=$CN, OU=$OU, O=$O, L=$L, ST=$ST, C=$C" -keypass $Keypass -keystore
$eig -storepass $Storepass -validity $validityVal
```

```
cmd2=echo /usr/lib/jvm/i586-wrs-jre/bin/keytool -alias cisco -list -v -keystore keystore
-storepass $Storepass;
```

- Step 4** Set Java home path using the **export** command.

**Example:**

```
export JAVA_HOME='/usr/lib/jvm/i586-wrs-jre'
```

**Step 5** Generate the TLS self-signed certificate using the `/home/admin/xnc/configuration/generateWebUICertificate.sh` script.

**Example:**

```
[root@RHEL-VM-NDB-ACI configuration]# ./generateWebUICertificate.sh

*****
Enter Fully qualified domain name :
*****
NDB-browser [ ] This can be FQDN of the NDB java application as well
*****
Enter Organizational unit :
*****
INSBU
*****
Enter Organization :
*****
cisco
*****
Enter Location :
*****
SJ
*****
Enter State :
*****
CA
*****
Enter Country :
*****
USA
*****
Enter keypass :
*****
cisco123
*****
Enter storepass :
*****
cisco123
*****
Enter the validity in number of days :
*****
365 [ ] in NDB 3.5 this script will let you to specify the certificate
validity.
*****
Below process will rename the existing key file to <old_keystore>, will
generate a new key file. Do you want to continue (y/n) ?
*****
Y
*****
Self-Signed Certificate Created
```

```

*****
Alias name: cisco
Creation date: Jan 6, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Issuer: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Serial number: b404be5
Valid from: Sun Jan 06 20:22:05 PST 2019 until: Mon Jan 06 20:22:05 PST
2020
Certificate fingerprints:
    MD5: 71:07:F6:4E:57:6A:08:3A:AD:06:32:B3:6C:5F:8F:52
    SHA1: 04:08:B9:D5:B7:EB:ED:E0:F9:22:49:14:FA:C6:09:39:22:32:43:A2

    SHA256:
34:D9:EB:34:0A:52:D1:4A:DD:F1:8B:14:D0:84:E4:1C:57:8B:2B:99:9B:E5:A1:4C:C7:8C:CD:AE:24:31:49:75

    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 63 8A 92 8F 6F 0F 45 BD   EE 55 C5 A8 99 3B F6 F7   c...o.E..U...;...
0010: AC FA 4A 21                               ..J!
]
]

*****
Displayed the generated keystore
*****
*****
Configured the keystore details on tomcat-server.xml
*****
*****
The newly generated key will used on next NDB restart. Do you want to
restart NDB now (y/n) ?
*****
y
Doesn't seem any Controller daemon is currently running
Running controller in background with PID: 13573, to connect to it please
SSH to this host on port 2400
NDB GUI can be accessed using below URL:
[https://10.16.206.160:8443]
[https://[fe80::250:56ff:fe90:b764]:8443]
[https://10.16.206.159:8443]
[https://192.168.1.123:8443]
[https://[fe80::250:56ff:fe90:9c79]:8443]

```

```
*****
NDB Restarted
*****
```

**Note** The `generateWebUICertificate.sh` script reloads the NDB application to ensure that the browser starts using this certificate when we access NDB java application from the browser.

**Step 6** Decode the generated certificate using the `keytool -list -v -keystore keystore_Name` command. Enter the store password when prompted.

**Example:**

```
[root@RHEL-VM-NDB-ACI configuration]# /usr/lib/jvm/i586-wrs-jre/bin/keytool -list -v -keystore
keystore
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 1 entry

Alias name: cisco
Creation date: Jul 6, 2019
Entry type: PrivateKeyEntry
Certificate chain length: 1
Certificate[1]:
Owner: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Issuer: CN=NDB-browser, OU=INSBU, O=cisco, L=SJ, ST=CA, C=USA
Serial number: b404be5
Valid from: Sun Jan 06 20:22:05 PST 2019 until: Mon Jan 06 20:22:05 PST 2020
Certificate fingerprints:
    MD5:  71:07:F6:4E:57:6A:08:3A:AD:06:32:B3:6C:5F:8F:52
        SHA1:  04:08:B9:D5:B7:EB:ED:E0:F9:22:49:14:FA:C6:09:39:22:32:43:A2
        SHA256:
34:D9:EB:34:0A:52:D1:4A:DD:F1:8B:14:D0:84:E4:1C:57:8B:2B:99:9B:E5:A1:4C:C7:8C:CD:AE:24:31:49:75

    Signature algorithm name: SHA256withRSA
    Version: 3

Extensions:

#1: ObjectId: 2.5.29.14 Criticality=false
SubjectKeyIdentifier [
KeyIdentifier [
0000: 63 8A 92 8F 6F 0F 45 BD   EE 55 C5 A8 99 3B F6 F7   c...o.E..U...;..
0010: AC FA 4A 21                               ..J!
]
]

*****
*****
```

**Step 7** Convert the `ndb-server-xnc.p12` to a password protected Java KeyStore (`ndb-server-keystore`) file using the `keytool` command. This command converts the `swl-xnc.p12` file to a password-protected `ndb-server-keystore` file. Create a new password for the destination JKS store and enter the source keystore password when prompted.

**Example:**

```
root@N9396TX-116:/xnclite/xnc/configuration# /usr/lib/jvm/i586-wrs-jre/bin/keytool keytool
-importkeystore -srckeystore keystore -srcstorepass cisco123 -srckeypass cisco123
```

```
-destkeystore keystore.p12 -deststoretype PKCS12 -srcalias cisco -deststorepass cisco123
-destkeypass cisco123
```

```
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
```

**Note** This command creates final self-signed certificate (keystore.p12) in PKCS12 format.

- Step 8** Upload the self-signed certificate into Trusted Root certificate store of your Web browser. See help for respective Web browsers about how to add a certificate to the Trusted Root certificate store. Use the password that you used while creating the certificate when prompted while uploading the certificate to the Web browser.
- Step 9** Deactivate and activate the virtual service to restart the NDB.

## Generating TLS 3rd Party Certification Between WebUI Browser and NDB Server

You can secure communication between a Web browser and NDB server running in centralized mode. This section describes how to generate CA certificates, convert the certificates into JKS format, and upload the certificates into a Web browser. To generate a CA certificate, you need to generate a Certificate Signing Request (CSR) and send it to a Certificate issuing authority (CA). You can use an open source tool to generate a CSR.

- Generating TLS 3rd Party Certification Between WebUI Browser and NDB Server Running in Centralized Mode

## Generating TLS 3rd Party Certification Between WebUI Browser and NDB Server Running in Centralized Mode

Complete these steps to generate TLS 3rd party certification between WebUI browser and NDB server running in centralized mode:

### Procedure

- Step 1** Generate Certificate Signing Request (CSR) using the **openssl req** command.

#### Example:

```
[root@NDB-server ~]# openssl req -newkey rsa:2048 -sha256 -keyout ndb-server.key -keyform
PEM -out ndb-server.req -outform PEM
```

```
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to 'ndb-server.key'
Enter PEM pass phrase: [ ] cisco123
Verifying - Enter PEM pass phrase: [ ] cisco123
```

```

-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----

```

```

Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:CA
Locality Name (eg, city) [Newbury]:SJ
Organization Name (eg, company) [My Company Ltd]:cisco
Organizational Unit Name (eg, section) []:insbu
Common Name (eg, your name or your server's hostname)
[]:ndb-server.cisco.com
Email Address []:chburra@cisco.com

```

```

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:cisco123
An optional company name []:cisco123

```

```

[root@NDB-server ~]# ls
ndb-server.req  ndb-server.key

```

**Note** The ndb-server.req (CSR) file is submitted to the certificate issuing authority (CA).

**Note** You need to use the same information when exporting the CA provided certificate into browser. The CSR file, cert.req, is submitted to CA.

**Step 2** To verify or view the CSR request, use the **openssl req** command.

**Example:**

```

[root@NDB-server ~]# openssl req -noout -text -in ndb-server.req
Certificate Request:
  Data:
    Version: 0 (0x0)
    Subject: C=US, ST=CA, L=SJ, O=cisco, OU=insbu,
    CN=ndb-server.cisco.com/emailAddress=chburra@cisco.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (2048 bit)
      Modulus (2048 bit):
        00:b5:30:75:e8:c8:5f:05:3b:0e:4f:aa:00:d9:64:
        8d:bf:b2:80:20:56:c3:be:b0:4c:e0:52:e5:be:d8:
        d2:74:85:4e:8a:ba:d3:1e:30:76:bf:e5:de:7d:51:
        11:79:8e:bc:96:38:7a:23:5a:26:31:50:50:fa:29:
        44:ab:56:b6:0d:41:38:ba:d1:d5:b4:e3:ba:a3:6c:
        4a:35:73:27:d9:fd:5c:4b:21:85:1a:f9:4d:b0:9e:

```

```

f3:ae:ce:49:98:ef:a2:f8:11:ab:bd:7e:64:ee:68:
68:19:6e:8f:3c:54:30:0f:28:01:13:b0:3d:34:b8:
f9:f5:cc:4a:84:d8:e5:d2:27:47:cc:83:76:92:ad:
92:62:f3:a3:35:be:14:ce:38:af:2a:c5:2e:fa:b8:
31:6b:71:cd:56:00:1f:0d:cc:b0:f8:fc:b0:52:91:
f8:9c:cf:45:13:c9:b5:86:fa:30:dd:88:78:01:15:
fb:5c:c9:6f:5b:b7:80:28:6c:86:54:c0:f2:5f:35:
70:82:49:5c:79:1c:f2:23:dd:50:d5:47:12:37:a3:
3f:f9:1d:90:8f:c0:e8:18:09:2e:66:8d:c3:72:17:
7f:7d:27:da:b1:cc:26:2d:8c:6b:ee:c5:e8:b5:78:
31:7c:bb:ba:6d:2c:e5:a3:29:7e:c1:4a:93:19:ed:
9a:e7
Exponent: 65537 (0x10001)
Attributes:
    unstructuredName          :cisco123
    challengePassword         :cisco123
Signature Algorithm: sha256WithRSAEncryption
9c:9a:51:e0:1d:e4:0b:8f:c1:c6:f5:e0:d2:f6:30:0e:18:af:
a7:b2:a4:4a:57:d7:07:44:cd:9c:fa:2d:0e:8b:c9:31:5b:16:
6b:84:42:0b:ed:06:5c:ed:30:d8:9b:ee:5d:79:f4:8a:e3:52:
3c:b3:4a:eb:6c:22:a2:f4:35:80:28:3a:67:62:7f:5f:dc:80:
e0:74:f0:3c:39:26:39:3a:76:6a:6a:98:e9:68:f9:b7:58:bf:
e7:44:2e:e7:73:0a:9c:62:28:b2:c6:09:41:81:b2:53:46:14:
e6:e4:dc:ca:90:81:5a:5e:dc:1b:dc:36:2c:86:5f:37:29:4c:
b0:ee:85:2b:34:f2:82:8a:d4:fc:a0:ce:10:e4:44:4e:d0:7a:
37:6d:3e:f9:ff:a1:19:8c:db:06:bf:be:87:57:a1:cb:05:15:
0b:9f:6c:8b:c2:ad:22:25:10:f0:4d:0f:4d:b7:be:71:87:f7:
85:24:e7:2d:f9:59:86:1a:b7:88:57:16:93:31:1f:d7:e5:07:
42:77:00:f9:ac:44:3b:6c:35:0f:80:5d:00:6f:ea:be:fe:e7:
28:53:0c:6b:5f:0c:76:bf:8c:a7:60:57:63:05:06:ff:ac:3d:
f1:63:54:d0:d0:13:44:b1:e9:53:6b:32:11:e2:83:26:04:f5:
23:67:6b:de

```

**Step 3** The private key, `ndb-server.key`, is secured with the passphrase. You need to unencrypt the certificate private key. Unencrypt the private key using the `openssl rsa` command.

**Example:**

```

[root@NDB-server ~]# cp ndb-server.key ndb-server.keybkp
[root@NDB-server ~]# rm ndb-server.key
[root@NDB-server ~]# openssl rsa -in ndb-server.keybkp -out ndb-server.key

```

Enter pass phrase for `ndb-server.keybkp`:

writing RSA key

**Note** Depending on the tier of the CA you choose, you can get up to three certificates (certificate chain) for each CSR. This means you get three certificates (root, intermediate and domain) from CA for each NDB switch. You need to check with CA to identify each type of certificate. Certificate naming convention might be different for different certifying authorities. For example: qvrca2.cer (root), hydssl2.cer (intermediate), ndb-server.cisco.com-39891.cer (domain).

Certificates are mostly shared in .PEM file format.

**Step 4** Create a single certificate file from the three certificate files using the cat command. The concatenation should be done in the following order, domain certificate, root certificate, and intermediate certificate. Syntax for cat command: `cat domain certificate root certificate intermediate certificate > ndb-server.cer`.

**Example:**

```
[root@NDB-server ~]# cat ndb-server.cisco.com-39891.cer qvrca.cer hydssl2.cer >
ndb-server.cer
```

**Step 5** Edit the newly created server.cer file to separate the concatenated END and BEGIN lines. Do not delete anything in the file.

**Example:**

```
-----END CERTIFICATE-----BEGIN CERTIFICATE-----

///// Modify the above line like this by adding a line feed between the two.
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

**Step 6** Create the TLS NDB Server Keystore file using the ndb-server.cer and ndb-server.key files. Copy the files to switch using the copy command.

**Example:**

```
cp ndb-server.key ndb-server-xnc-privatekey.pem
cp ndb-server.cer ndb-server-xnc-cert.pem
```

**Step 7** Combine the private key and certificate file into a single .PEM file using the cat command.

**Example:**

```
cat ndb-server-xnc-privatekey.pem ndb-server-xnc-cert.pem > ndb-server-xnc.pem
```

**Step 8** CA provides certificates in PEM format and extension of the certificate is .pem. You need to convert the PEM format certificate to PKCS12 format. Convert the PEM file, ndb-server-xnc.pem, to .P12 file format using the **openssl pkcs12** command. Enter the export password when prompted. The password must contain at least 6 characters, for example, cisco123. The ndb-server-xnc.pem file is converted to a password-protected ndb-server-xnc.p12 file.

**Example:**

```
[root@NDB-server ~]# openssl pkcs12 -export -out ndb-server-xnc.p12 -in ndb-server-xnc.pem
Enter Export Password: [cisco123
Verifying - Enter Export Password: [cisco123
```

**Step 9** Convert the ndb-server-xnc.p12 to a password protected Java KeyStore (ndb-server-keystore) file using the **keytool** command. This command converts the sw1-xnc.p12 file to a password-protected ndb-server-keystore file. Create a new password for the destination JKS store and enter the source keystore password when prompted.

**Example:**



```
[root@NDB-server ~]# keytool -importkeystore -srckeystore ndb-server-xnc.p12 -srcstoretype
pkcs12 -destkeystore ndb-server-keystore -deststoretype jks
```

```
Enter destination keystore password: [cisco123]
```

```
Re-enter new password: [cisco123]
```

```
Enter source keystore password: --cisco123
```

```
Entry for alias 1 successfully imported.
```

```
Import command completed: 1 entries successfully imported, 0 entries
failed or cancelled
```

```
[root@NDB-server ~]#
```

**Step 10** List and verify content in the java tlsKeyStore using the **keytool** command.

**Example:**

```
[root@NDB-server ~]#keytool -list -v -keystore ndb-server-keystore
```

**Step 11** Configure the tomcat-server.xml (stored in configuration folder) with key store password that was provided while generating the certificate. You can use VI editor and edit the following lines with keystore and keystorepass.

**Example:**

```
keystoreFile="configuration/ndb-server-keystore"
keystorePass="cisco123" server="Cisco XNC"
```

**Step 12** Restart the NDB.

**Step 13** Upload the CA certificate into Trusted Root certificate store of Web browser. See help for respective Web browsers about how to add the certificate to the Trusted Root certificate store. Use the password that you created while creating the certificate when prompted while uploading the certificate to the Web browser.

## Generating TLS 3rd Party Certification Between Web Browser and NDB Server Running in Embedded Mode

You can generate TLS 3rd party certificate between Web browser and NDB server running in embedded mode for the Guestshell Environment.

### Generating TLS 3rd Party Certificate Between Web Browser and NDB Server Running in Embedded Mode Using Guest Shell Environment

To generate TLS 3rd party certificate between Web browser and NDB server running in embedded mode using guest shell environment, complete the following steps.

**Procedure**

**Step 1** Enable bash-shell feature on the switch using the feature command.

**Example:**

```
N9396TX-116(config)# feature bash-shell
```

**Step 2** Enter bash-shell mode on the switch using the run command.

**Example:**

```
N9396TX-116(config)# run bash
bash-4.2$
```

**Step 3** Generate Certificate Signing Request (CSR) using the **openssl req** command. Enter the required information when prompted.

**Example:**

```
bash-4.2$ openssl req -newkey rsa:2048 -sha256 -keyout ndb-server.key -keyform PEM -out
ndb-server.req -outform PEM
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to 'ndb-server.key'
Enter PEM pass phrase: [ ] cisco123
Verifying - Enter PEM pass phrase: [ ] cisco123
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:CA
Locality Name (eg, city) [Newbury]:SJ
Organization Name (eg, company) [My Company Ltd]:cisco
Organizational Unit Name (eg, section) []:insbu
Common Name (eg, your name or your server's hostname) []:ndb-server.cisco.com
Email Address []:chburra@cisco.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:cisco123
An optional company name []:cisco123
```

```
bash-4.2$ ls

ndb-server.req  ndb-server.key
```

**Note** The openssl command creates a private key, ndb-server.key, and a certificate signing request file, ndb-server.req. The ndb-server.req (CSR) file is submitted to the certificate issuing authority (CA).

**Note** You need to use the same information when exporting the CA provided certificate into browser. The CSR file, cert.req, is submitted to CA.

**Step 4** To view the content or verify the CSR request, use the **openssl req** command.

**Example:**

```
bash-4.2$ openssl req -noout -text -in ndb-server.req
Certificate Request:
  Data:
    Version: 0 (0x0)
    Subject: C=US, ST=CA, L=SJ, O=cisco, OU=insbu,
    CN=ndb-server.cisco.com/emailAddress=chburra@cisco.com
```

```

Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (2048 bit)
    Modulus (2048 bit):
      00:b5:30:75:e8:c8:5f:05:3b:0e:4f:aa:00:d9:64:
      8d:bf:b2:80:20:56:c3:be:b0:4c:e0:52:e5:be:d8:
      d2:74:85:4e:8a:ba:d3:1e:30:76:bf:e5:de:7d:51:
      11:79:8e:bc:96:38:7a:23:5a:26:31:50:50:fa:29:
      44:ab:56:b6:0d:41:38:ba:d1:d5:b4:e3:ba:a3:6c:
      4a:35:73:27:d9:fd:5c:4b:21:85:1a:f9:4d:b0:9e:
      f3:ae:ce:49:98:ef:a2:f8:11:ab:bd:7e:64:ee:68:
      68:19:6e:8f:3c:54:30:0f:28:01:13:b0:3d:34:b8:
      f9:f5:cc:4a:84:d8:e5:d2:27:47:cc:83:76:92:ad:
      92:62:f3:a3:35:be:14:ce:38:af:2a:c5:2e:fa:b8:
      31:6b:71:cd:56:00:1f:0d:cc:b0:f8:fc:b0:52:91:
      f8:9c:cf:45:13:c9:b5:86:fa:30:dd:88:78:01:15:
      fb:5c:c9:6f:5b:b7:80:28:6c:86:54:c0:f2:5f:35:
      70:82:49:5c:79:1c:f2:23:dd:50:d5:47:12:37:a3:
      3f:f9:1d:90:8f:c0:e8:18:09:2e:66:8d:c3:72:17:
      7f:7d:27:da:b1:cc:26:2d:8c:6b:ee:c5:e8:b5:78:
      31:7c:bb:ba:6d:2c:e5:a3:29:7e:c1:4a:93:19:ed:
      9a:e7
    Exponent: 65537 (0x10001)
Attributes:
  unstructuredName          :cisco123
  challengePassword        :cisco123
Signature Algorithm: sha256WithRSAEncryption
  9c:9a:51:e0:1d:e4:0b:8f:c1:c6:f5:e0:d2:f6:30:0e:18:af:
  a7:b2:a4:4a:57:d7:07:44:cd:9c:fa:2d:0e:8b:c9:31:5b:16:
  6b:84:42:0b:ed:06:5c:ed:30:d8:9b:ee:5d:79:f4:8a:e3:52:
  3c:b3:4a:eb:6c:22:a2:f4:35:80:28:3a:67:62:7f:5f:dc:80:
  e0:74:f0:3c:39:26:39:3a:76:6a:6a:98:e9:68:f9:b7:58:bf:
  e7:44:2e:e7:73:0a:9c:62:28:b2:c6:09:41:81:b2:53:46:14:
  e6:e4:dc:ca:90:81:5a:5e:dc:1b:dc:36:2c:86:5f:37:29:4c:
  b0:ee:85:2b:34:f2:82:8a:d4:fc:a0:ce:10:e4:44:4e:d0:7a:
  37:6d:3e:f9:ff:a1:19:8c:db:06:bf:be:87:57:a1:cb:05:15:
  0b:9f:6c:8b:c2:ad:22:25:10:f0:4d:0f:4d:b7:be:71:87:f7:
  85:24:e7:2d:f9:59:86:1a:b7:88:57:16:93:31:1f:d7:e5:07:
  42:77:00:f9:ac:44:3b:6c:35:0f:80:5d:00:6f:ea:be:fe:e7:
  28:53:0c:6b:5f:0c:76:bf:8c:a7:60:57:63:05:06:ff:ac:3d:
  f1:63:54:d0:d0:13:44:b1:e9:53:6b:32:11:e2:83:26:04:f5:
  23:67:6b:de

```

**Step 5** The private key, `ndb-server.key`, is secured with the passphrase. You need to unencrypt the certificate private key. Unencrypt the private key using the `openssl rsa` command.

**Example:**

```

bash-4.2$ cp ndb-server.key ndb-server.keybkp
bash-4.2$ rm ndb-server.key
bash-4.2$ openssl rsa -in ndb-server.keybkp -out ndb-server.key
Enter pass phrase for ndb-server.keybkp: cisco123
writing RSA key

```

**Note** Depending on the tier of the CA you choose, you can get up to three certificates (certificate chain) for each CSR. This means you get three certificates (root, intermediate and domain) from CA for each NDB switch. You need to check with CA to identify each type of certificate. Certificate naming convention might be different for different certifying authorities. For example: `qvrca2.cer` (root), `hydssl2.cer` (intermediate), `ndb-server.cisco.com-39891.cer` (domain).

Certificates are mostly shared in .PEM file format.

**Step 6** Create a single certificate file from the three certificate files using the `cat` command. The concatenation should be done in the following order, domain certificate, root certificate, and intermediate certificate. Syntax for `cat` command: `cat domain certificate root certificate intermediate certificate > ndb-server.cer`.

**Example:**

```
bash-4.2$ cat ndb-server.cisco.com-39891.cer qvrca.cer hydssl2.cer > ndb-server.cer
```

**Step 7** Edit the newly created `server.cer` file to separate the concatenated END and BEGIN lines. Do not delete anything in the file.

**Example:**

```
-----END CERTIFICATE-----BEGIN CERTIFICATE-----

///// Modify the above line like this by adding a line feed between the two.
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

**Step 8** Create the TLS NDB Server Keystore file using the `ndb-server.cer` and `ndb-server.key` files. Copy the files to switch using the `copy` command.

**Example:**

```
cp ndb-server.key ndb-server-ndb-privatekey.pem
cp ndb-server.cer ndb-server-ndb-cert.pem
```

**Step 9** Combine the private key and certificate file into a single `.PEM` file using the `cat` command.

**Example:**

```
cat ndb-server-ndb-privatekey.pem ndb-server-ndb-cert.pem > ndb-server-ndb.pem
```

**Step 10** CA provides certificates in PEM format and extension of the certificate is `.pem`. You need to convert the PEM format certificate to PKCS12 format. Convert the PEM file, `ndb-server-ndb.pem`, to `.P12` file format using the `openssl pkcs12` command. Enter the export password when prompted. The password must contain at least 6 characters, for example, `cisco123`. The `ndb-server-ndb.pem` file is converted to a password-protected `ndb-server-ndb.p12` file.

**Example:**

```
bash-4.2$ openssl pkcs12 -export -out ndb-server-ndb.p12 -in ndb-server-ndb.pem
Enter Export Password: [cisco123
Verifying - Enter Export Password: [cisco123
```

**Step 11** Copy the certificate file to the NDB configuration folder.

**Example:**

```
bash-4.2$ cp ndb-server-ndb.p12
/isan/vdc_1/virtual-instance/guestshell+/rootfs/home/admin/ndb/configuration/
```

**Step 12** Exit the bash shell mode using the `exit` command.

**Example:**

```
bash-4.2$ exit
exit
N9396TX-116#
```

**Step 13** Connect to guest shell using `guestshell` command.

**Example:**

```
N9396TX-116# guestshell
[admin@guestshell ~]$
```

**Step 14** Change current directory to `ndb\configuration`.

**Example:**

```
[admin@guestshell ~]$ cd ndb\configuration
```

**Step 15** Set Java home path using the `export` command.

**Example:**

```
[admin@guestshell ~]$ export JAVA_HOME='/usr/lib/jvm/i586-wrs-jre'
```

**Step 16** Convert the `ndb-server-ndb.p12` to a password protected Java KeyStore (`ndb-server-keystore`) file using the `keytool` command. This command converts the `sw1-ndb.p12` file to a password-protected `ndb-server-keystore` file. Create a new password for the destination JKS store and enter the source keystore password when prompted.

**Example:**

```
[admin@guestshell configuration]$ /home/admin/ndb/jre/bin/keytool -importkeystore -srckeystore
ndb-server-ndb.p12 -srcstoretype pkcs12 -destkeystore ndb-server-keystore -deststoretype
jks
Enter destination keystore password: [cisco123
Re-enter new password: [cisco123
Enter source keystore password: [cisco123
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
```

**Step 17** List and verify content in the `java tlsKeyStore` using the `keytool` command.

**Example:**

```
[admin@guestshell configuration]$ /home/admin/ndb/jre/bin/keytool -list -v -keystore
ndb-server-keystore
```

**Step 18** Configure the `tomcat-server.xml` (stored in `configuration` folder) with key store password that was provided while generating the certificate. You can use VI editor and edit the following lines with `keystore` and `keystorepass`.

**Example:**

```
keystoreFile="configuration/ndb-server-keystore"
keystorePass="cisco123" server="Cisco ndb"
```

**Step 19** Upload the CA certificate into Trusted Root certificate store of Web browser. See help for respective Web browsers about how to add the certificate to the Trusted Root certificate store. Use the password that you created while creating the certificate when prompted while uploading the certificate to the Web browser.

**Step 20** Restart the NDB.

## Generating TLS 3rd Party Certificate Between Web Browser and NDB Server Running in Embedded Mode Using OVA Environment

To generate TLS 3rd party certificate between Web browser and NDB server running in embedded mode using OVA environment, complete the following steps.

## Procedure

---

**Step 1** Enable bash-shell feature on the switch using the feature command.

**Example:**

```
N9396TX-116(config)# feature bash-shell
```

**Step 2** Enter bash-shell mode on the switch using the run command.

**Example:**

```
N9396TX-116(config)# run bash
bash-4.2$
```

**Step 3** Generate Certificate Signing Request (CSR) using the **openssl req** command. Enter the required information when prompted.

**Example:**

```
bash-4.2$ openssl req -newkey rsa:2048 -sha256 -keyout ndb-server.key -keyform PEM -out
ndb-server.req -outform PEM
```

```
Generating a 2048 bit RSA private key
...+++
.....+++
writing new private key to 'ndb-server.key'
Enter PEM pass phrase:  cisco123
Verifying - Enter PEM pass phrase:  cisco123
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:US
State or Province Name (full name) [Berkshire]:CA
Locality Name (eg, city) [Newbury]:SJ
Organization Name (eg, company) [My Company Ltd]:cisco
Organizational Unit Name (eg, section) []:insbu
Common Name (eg, your name or your server's hostname) []:ndb-server.cisco.com
Email Address []:chburra@cisco.com
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:cisco123
An optional company name []:cisco123
```

```
bash-4.2$ ls
```

```
ndb-server.req  ndb-server.key
```

**Note** The openssl command creates a private key, ndb-server.key, and a certificate signing request file, ndb-server.req. The ndb-server.req (CSR) file is submitted to the certificate issuing authority (CA).

**Note** You need to use the same information when exporting the CA provided certificate into browser. The CSR file, cert.req, is submitted to CA.

**Step 4** To view the content or verify the CSR request, use the **openssl req** command.

**Example:**

```
bash-4.2$ openssl req -noout -text -in ndb-server.req
Certificate Request:
  Data:
    Version: 0 (0x0)
    Subject: C=US, ST=CA, L=SJ, O=cisco, OU=insbu,
    CN=ndb-server.cisco.com/emailAddress=chburra@cisco.com
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      RSA Public Key: (2048 bit)
        Modulus (2048 bit):
          00:b5:30:75:e8:c8:5f:05:3b:0e:4f:aa:00:d9:64:
          8d:bf:b2:80:20:56:c3:be:b0:4c:e0:52:e5:be:d8:
          d2:74:85:4e:8a:ba:d3:1e:30:76:bf:e5:de:7d:51:
          11:79:8e:bc:96:38:7a:23:5a:26:31:50:50:fa:29:
          44:ab:56:b6:0d:41:38:ba:d1:d5:b4:e3:ba:a3:6c:
          4a:35:73:27:d9:fd:5c:4b:21:85:1a:f9:4d:b0:9e:
          f3:ae:ce:49:98:ef:a2:f8:11:ab:bd:7e:64:ee:68:
          68:19:6e:8f:3c:54:30:0f:28:01:13:b0:3d:34:b8:
          f9:f5:cc:4a:84:d8:e5:d2:27:47:cc:83:76:92:ad:
          92:62:f3:a3:35:be:14:ce:38:af:2a:c5:2e:fa:b8:
          31:6b:71:cd:56:00:1f:0d:cc:b0:f8:fc:b0:52:91:
          f8:9c:cf:45:13:c9:b5:86:fa:30:dd:88:78:01:15:
          fb:5c:c9:6f:5b:b7:80:28:6c:86:54:c0:f2:5f:35:
          70:82:49:5c:79:1c:f2:23:dd:50:d5:47:12:37:a3:
          3f:f9:1d:90:8f:c0:e8:18:09:2e:66:8d:c3:72:17:
          7f:7d:27:da:b1:cc:26:2d:8c:6b:ee:c5:e8:b5:78:
          31:7c:bb:ba:6d:2c:e5:a3:29:7e:c1:4a:93:19:ed:
          9a:e7
        Exponent: 65537 (0x10001)
    Attributes:
      unstructuredName          :cisco123
      challengePassword         :cisco123
    Signature Algorithm: sha256WithRSAEncryption
    9c:9a:51:e0:1d:e4:0b:8f:c1:c6:f5:e0:d2:f6:30:0e:18:af:
    a7:b2:a4:4a:57:d7:07:44:cd:9c:fa:2d:0e:8b:c9:31:5b:16:
    6b:84:42:0b:ed:06:5c:ed:30:d8:9b:ee:5d:79:f4:8a:e3:52:
    3c:b3:4a:eb:6c:22:a2:f4:35:80:28:3a:67:62:7f:5f:dc:80:
    e0:74:f0:3c:39:26:39:3a:76:6a:6a:98:e9:68:f9:b7:58:bf:
    e7:44:2e:e7:73:0a:9c:62:28:b2:c6:09:41:81:b2:53:46:14:
    e6:e4:dc:ca:90:81:5a:5e:dc:1b:dc:36:2c:86:5f:37:29:4c:
    b0:ee:85:2b:34:f2:82:8a:d4:fc:a0:ce:10:e4:44:4e:d0:7a:
    37:6d:3e:f9:ff:a1:19:8c:db:06:bf:be:87:57:a1:cb:05:15:
    0b:9f:6c:8b:c2:ad:22:25:10:f0:4d:0f:4d:b7:be:71:87:f7:
    85:24:e7:2d:f9:59:86:1a:b7:88:57:16:93:31:1f:d7:e5:07:
    42:77:00:f9:ac:44:3b:6c:35:0f:80:5d:00:6f:ea:be:fe:e7:
    28:53:0c:6b:5f:0c:76:bf:8c:a7:60:57:63:05:06:ff:ac:3d:
    f1:63:54:d0:d0:13:44:b1:e9:53:6b:32:11:e2:83:26:04:f5:
    23:67:6b:de
```

**Step 5** The private key, `ndb-server.key`, is secured with the passphrase. You need to unencrypt the certificate private key. Unencrypt the private key using the **openssl rsa** command.

**Example:**

```
bash-4.2$ cp ndb-server.key ndb-server.keybkp
bash-4.2$ rm ndb-server.key
bash-4.2$ openssl rsa -in ndb-server.keybkp -out ndb-server.key
Enter pass phrase for ndb-server.keybkp: [cisco123]
writing RSA key
```

**Note** Depending on the tier of the CA you choose, you can get up to three certificates (certificate chain) for each CSR. This means you get three certificates (root, intermediate and domain) from CA for each NDB switch. You need to check with CA to identify each type of certificate. Certificate naming convention might be different for different certifying authorities. For example: qvrca2.cer (root), hydssl2.cer (intermediate), ndb-server.cisco.com-39891.cer (domain).

Certificates are mostly shared in .PEM file format.

**Step 6** Create a single certificate file from the three certificate files using the `cat` command. The concatenation should be done in the following order, domain certificate, root certificate, and intermediate certificate. Syntax for `cat` command: `cat domain certificate root certificate intermediate certificate > ndb-server.cer`.

**Example:**

```
bash-4.2$ cat ndb-server.cisco.com-39891.cer qvrca.cer hydssl2.cer > ndb-server.cer
```

**Step 7** Edit the newly created server.cer file to separate the concatenated END and BEGIN lines. Do not delete anything in the file.

**Example:**

```
-----END CERTIFICATE-----BEGIN CERTIFICATE-----
```

```
///// Modify the above line like this by adding a line feed between the two.
```

```
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
```

**Step 8** Create the TLS NDB Server Keystore file using the `ndb-server.cer` and `ndb-server.key` files. Copy the files to switch using the `copy` command.

**Example:**

```
cp ndb-server.key ndb-server-xnc-privatekey.pem
cp ndb-server.cer ndb-server-xnc-cert.pem
```

**Step 9** Combine the private key and certificate file into a single .PEM file using the `cat` command.

**Example:**

```
cat ndb-server-xnc-privatekey.pem ndb-server-xnc-cert.pem > ndb-server-xnc.pem
```

**Step 10** CA provides certificates in PEM format and extension of the certificate is .pem. You need to convert the PEM format certificate to PKCS12 format. Convert the PEM file, `ndb-server-xnc.pem`, to .P12 file format using the `openssl pkcs12` command. Enter the export password when prompted. The password must contain at least 6 characters, for example, `cisco123`. The `ndb-server-xnc.pem` file is converted to a password-protected `ndb-server-xnc.p12` file.

**Example:**

```
bash-4.2$ openssl pkcs12 -export -out ndb-server-xnc.p12 -in ndb-server-xnc.pem
Enter Export Password: cisco123
Verifying - Enter Export Password: cisco123
```

**Step 11** Copy the certificate file to the NDB configuration folder.

**Example:**

```
bash-4.2$ cp ndb-server-xnc.p12
/isan/vdc_1/virtual-instance/guestshell+/rootfs/home/admin/xnc/configuration/
```

**Step 12** Exit the bash shell mode using the `exit` command.



**Example:**

```
bash-4.2$ exit
exit
N9396TX-116#
```

**Step 13** Connect to virtual service console using the **virtual-service connect** command.

**Example:**

```
N9396TX-116# virtual-service connect name ova console
Connecting to virtual-service. Exit using ^c^c^c
Entering character mode
Escape character is '^]'.
root@ N9396TX-116:~#
root@ N9396TX-116:~#
```

**Step 14** Change current directory to /xnclite/xnc/configuration.

**Example:**

```
root@ N9396TX-116:~# cd /xnclite/xnc/configuration
root@ N9396TX-116:/xnclite/xnc/configuration#
```

**Step 15** Set Java home path using the **export** command.

**Example:**

```
[admin@guestshell ~]$ export JAVA_HOME='/usr/lib/jvm/i586-wrs-jre'
```

**Step 16** Convert the ndb-server-xnc.p12 to a password protected Java KeyStore (ndb-server-keystore) file using the **keytool** command. This command converts the sw1-xnc.p12 file to a password-protected ndb-server-keystore file. Create a new password for the destination JKS store and enter the source keystore password when prompted.

**Example:**

```
root@ N9396TX-116:/xnclite/xnc/configuration# /usr/lib/jvm/i586-wrs-jre/bin/keytool
-importkeystore -srckeystore ndb-server-xnc.p12 -srcstoretype pkcs12 -destkeystore
ndb-server-keystore -deststoretype jks
Enter destination keystore password: [cisco123
Re-enter new password: [cisco123
Enter source keystore password: [cisco123
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
```

**Step 17** List and verify content in the java tlsKeyStore using the **keytool** command.

**Example:**

```
root@ N9396TX-116:/xnclite/xnc/configuration# /usr/lib/jvm/i586-wrs-jre/bin/keytool -list
-v -keystore ndb-server-keystore
```

**Step 18** Configure the tomcat-server.xml (stored in configuration folder) with key store password that was provided while generating the certificate. You can use VI editor and edit the following lines with keystore and keystorepass.

**Example:**

```
keystoreFile="configuration/ndb-server-keystore"
keystorePass="cisco123" server="Cisco XNC"
```

- Step 19** Upload the CA certificate into Trusted Root certificate store of Web browser. See help for respective Web browsers about how to add the certificate to the Trusted Root certificate store. Use the password that you created while creating the certificate when prompted while uploading the certificate to the Web browser.
- Step 20** Deactivate and activate the virtual service to restart the NDB.
-



## CHAPTER 4

# Logging in and Managing Cisco Nexus Data Broker

This chapter contains the following sections:

- [Configuring Cisco Nexus Data Broker, on page 71](#)
- [Logging in to the Cisco Nexus Data Broker GUI, on page 75](#)
- [Changing the Controller Access to HTTP, on page 75](#)
- [Cisco Nexus Data Broker GUI Overview, on page 76](#)
- [Saving Configuration Changes , on page 78](#)

## Configuring Cisco Nexus Data Broker

### Configuring High Availability Clusters

Cisco Nexus Data Broker supports high availability clustering in active/active mode with up to five controllers. To use high availability clustering with Cisco Nexus Data Broker, you must edit the `config.ini` file for each instance of Cisco Nexus Data Broker.



**Note** IPv6 is supported in centralized NDB mode only, it is not supported in Embedded mode.



**Note** Cisco NDB supports only 2 node configuration or odd number node configuration. If you configure even number of nodes, the last node is not included in the cluster formation, ensuring odd number of nodes in a setup.

*Table 2: Cluster Operation Status*

| Cluster Indicator | Cluster Status | Recommendation |
|-------------------|----------------|----------------|
| Green             | Operational    |                |

| Cluster Indicator | Cluster Status                              | Recommendation                                                                                                                                                                              |
|-------------------|---------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Yellow            | Some of the cluster nodes are not available | Do not make any changes or add to the existing NDB configuration.                                                                                                                           |
| Red               | The node is isolated from the cluster.      | Do not make any changes or add to the existing NDB configuration.<br><br>Note: For two node cluster, you need to override in any one of the cluster node only, to ensure regular operation. |

### Before you begin

- All IP addresses must be reachable and capable of communicating with each other.
- All switches in the cluster must connect to all of the controllers.
- All controllers must have the same HA clustering configuration information in the `config.ini` files.
- All controllers must have the same information in the `xnc/configuration/startup` directory.
- If using cluster passwords, all controllers must have the same password configured in the `xncjgroups.xml` file. See [Password Protecting the High Availability Clusters](#), on page 73.

### Procedure

**Step 1** Open a command window on one of the instances in the cluster.

**Step 2** Navigate to the `xnc/configuration` directory that was created when you installed the software.

**Step 3** Use any text editor to open the `config.ini` file.

**Step 4** Locate the following text:

```
# HA Clustering configuration (semi-colon-separated IP addresses of all controllers that
are part of the cluster.)
# supernodes=<ip1>;<ip2>;<ip3>;<ipn>
```

**Step 5** **Example:**

IPv4 example.

```
# HA Clustering configuration (semi-colon-separated IP addresses of all controllers that
are part of the cluster.)
supernodes=10.1.1.1;10.2.1.1;10.3.1.1;10.4.1.1;10.5.1.1
```

**Example:**

IPv6 example.

```
# HA Clustering configuration (semi-colon-separated IP addresses of all controllers that
are part of the cluster.)
supernodes=2001:22:11::1;2001:33::44::1;2001:55:66::1
```

**Step 6** Save the file and exit the editor.

## Password Protecting the High Availability Clusters

### Procedure

**Step 1** Open a command window on one of the instances in the cluster.

**Step 2** Navigate to the `xnc/configuration` directory.

**Step 3** Use any text editor to open the `xncjgroups.xml` file.

**Step 4** Locate the following text:

```
<!-- <AUTH auth_class="org.jgroups.auth.MD5Token" auth_value="ciscoXNC"
token_hash="MD5"></AUTH> -->
```

**Step 5** Remove the comments from the AUTH line.

#### Example:

```
<AUTH auth_class="org.jgroups.auth.MD5Token" auth_value="ciscoXNC" token_hash="MD5"></AUTH>
```

**Step 6** (Optional) Change the password in the `auth_value` attribute.

By default, the cluster is protected with the password "ciscoXNC". You can change this password to whatever value you want, you need make the similar changes on all machines in the cluster.

**Step 7** Save the file and exit the editor.

## Editing Cisco Nexus Switch Configuration

Cisco Nexus Data Broker periodically verifies the Cisco Nexus Switch inventory and the topology so that the topology and inventory is in sync. Cisco Nexus data broker periodically rediscovers the switch inventory and the topology interconnection and status. This information is updated in the GUI depending on the status. You can configure the rediscovery interval and the default value is 60 seconds.

### Procedure

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

**Step 2** Use any text editor to open the `config.ini` file.

**Step 3** Update the following parameters:

| Name                     | Predefined Value in Seconds | Minimum Value in Seconds | Recommended Value in Seconds |
|--------------------------|-----------------------------|--------------------------|------------------------------|
| of.messageResponseTimer  | 60                          | 2                        | 60                           |
| of.switchLivenessTimeout | 120.5                       | 60.5                     | 120.5                        |
| of.flowStatsPollInterval | 240                         | 10                       | 240                          |
| of.portStatsPollInterval | 240                         | 5                        | 240                          |
| of.descStatsPollInterval | 240                         | 60                       | 240                          |

| Name                                    | Predefined Value in Seconds | Minimum Value in Seconds | Recommended Value in Seconds |
|-----------------------------------------|-----------------------------|--------------------------|------------------------------|
| of.barrierMessagePriorCount             | 50                          | 100                      | 50                           |
| of.discoveryInterval                    | 300                         | 30                       | 300                          |
| of.discoveryTimeoutMultiple             | 2                           | 2                        | 2                            |
| <b>NX-API related system parameters</b> |                             |                          |                              |
| nx.connectionDelayTimer                 | 300                         | —                        | 300                          |
| nx.flowStatsPollInterval                | 120                         | —                        | 120                          |
| nx.tableStatsPollInterval               | 120                         | —                        | 120                          |
| nx.portStatsPollInterval                | 120                         | —                        | 120                          |
| nx.descStatsPollInterval                | 120                         | —                        | 120                          |
| nx.lldpPollingTimer                     | 10                          | —                        | 10                           |
| nx.portPollingTimer                     | 20                          | —                        | 20                           |

**Note** Predefined values are the values that Cisco includes in the `config.ini` file that is shipped with Cisco Nexus Data Broker. An em dash ("—") in this column of the table means that unless you explicitly update the value, the minimum value will be used.

- Step 4** Save the file and exit the editor.
- Step 5** Restart Cisco Nexus Data Broker.

## Configuring Inactivity Timeout

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.

### Procedure

- Step 1** Log into NDB UI.
- Step 2** Navigate to **Administration > System > Session Timeout**.
- Step 3** Enter inactivity timeout value in the **Session Timeout** text field and click **Submit**.
- Step 4** Log out and log into NDB to apply the changes.

## Configuring User Roles for Edge Ports

To enable RBAC for the App-User role, follow these steps:

### Procedure

---

- Step 1** Open the `config.ini` file for editing.
  - Step 2** Locate the line `# Enforce restriction on edge/tap ports user can capture` (default `false`).
  - Step 3** Remove the comment character from the following line:  
`monitor.strictAuthorization=true`
  - Step 4** Save your work and close the file.
- 

## Logging in to the Cisco Nexus Data Broker GUI

You can log into the Cisco Nexus Data Broker using HTTPS. The default HTTPS web link for the Cisco Nexus Data Broker GUI is `https://Nexus_Data_Broker_IP:8443/monitor`.



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

### Procedure

---

- Step 1** In your web browser, enter the Cisco Nexus Data Broker web link.
  - 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**.
- 

## Changing the Controller Access to HTTP

Starting with Cisco Nexus Data Broker Release 2.1, an unencrypted (HTTP) access to the GUI and the API to the controller access is disabled by default. You cannot access the controller with the URL `http://<host>:8080`.

If you want to change the controller access to HTTP, complete the following steps:

## Procedure

---

**Step 1** Remove the comment character from the connector for port 8080 in the tomcat-server.xml file in the configuration directory as displayed in the following example:

**Example:**

```
<Service name="Catalina">
  <!--
    <Connector port="8080" protocol="HTTP/1.1"
      connectionTimeout="20000"
      redirectPort="8443" server="Cisco XNC" enableLookups="false" />
  -->
  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="configuration/keystore"
    keystorePass="ciscoxnc" server="Cisco XNC"
    connectionTimeout="60000" enableLookups="false" />
```

**Example:**

Remove the comment character as displayed in the following example:

```
<Service name="Catalina">
  <Connector port="8080" protocol="HTTP/1.1"
    connectionTimeout="20000"
    redirectPort="8443" server="Cisco XNC" enableLookups="false" />

  <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"
    scheme="https" secure="true"
    clientAuth="false" sslProtocol="TLS"
    keystoreFile="configuration/keystore"
    keystorePass="ciscoxnc" server="Cisco XNC"
    connectionTimeout="60000" enableLookups="false" />
```

**Step 2** Restart the controller.

---

# Cisco Nexus Data Broker GUI Overview

The Cisco Nexus Data Broker Release GUI contains the following tabs:

- Cisco Nexus Data Broker, Release Version
- **Configuration** tab at the top of the screen
- **Administration** tab at the top of the screen
- **Default** tab displaying the switches in use
- **Save** button—Enables you to save any additions or changes you make in Cisco Nexus Data Broker.
- The **Online help** button—Provides access to the online help for the current page.
- Bookmarks
- Administrator Details



The **Configuration** tab contains the following items:

- Topology
- Port Definitions
- Port Groups
- Monitoring Devices
- Service Nodes
- Filters
- Connections
- Redirections
- Statistics
- SPAN Sessions

The **Administration** tab contains the following items:

- Device Management
- Devices
- Flows
- Troubleshoot
- Consistency Check
- System Management
- User Management
- System

### Topology Tools

The left side of the topology pane contains a zoom slider that allows you increase or decrease the size of the topology diagram. You can also increase or decrease the size of the topology diagram by scrolling up or down, respectively, with your mouse wheel.

You can move the entire topology diagram, a single topology element, or a node group. To move the diagram, an element, or a node group, click it and drag it.

To view information about a node or an edge port, hover over the node or edge port icon with your mouse. The information displayed depends on the device you choose.

To view information about a path, hover over the path in the topology diagram.

To view information about a filter, hover over the **Name** of the filter in the **Filters** tab.

### What's New Utility Tool

Starting with Cisco NDB Release 3.7, a new utility tool is added to the NDB user interface. This utility appears when you log into NDB and lists all the new features introduced for the release. You can also open this utility using the What's New icon on the NDB GUI.

## Saving Configuration Changes

In Cisco Nexus Data Broker, Release 3.2.0 the auto-save configuration option is added. You can save the configuration changes, but it is not required. For example, if you configure Edge-SPAN, monitor the device, or configure any other functionality in Cisco Nexus Data Broker, it is saved automatically.

### Procedure

---

On the menu bar, click **Save**.

---



## CHAPTER 5

# Viewing and Adding Devices

---

This chapter contains the following sections:

- [Viewing and Adding Devices](#) , on page 79
- [Managing a Device in NDB](#), on page 82
- [Device Prerequisites](#), on page 85
- [Profile Management](#), on page 86

## Viewing and Adding Devices

On the **Devices** screen, the following tabs are displayed:

- Nodes Learned
- Device Connections
- Device Groups
- SPAN Management
- Subnet Gateway Configuration

On the **Nodes Learned** tab, the following details are displayed for each node:

- The name of the node
- The ID of the node
- IP Address of the node
- The number of ports on the node

When you click the node name under the tab **Node Name**, the **Update Node Information** window is displayed. Update the following fields in the window:

- **Node ID**: Enter the node ID.
- **Node Name**: The name of the node.
- **Tier**: Select the tier of the node from the following options in the drop-down list: Unknown, Access, Distribution, and Core.
- **Operation Mode**: Choose how the traffic is handled based on the flows. This can be one of the following:

Allow reactive forwarding—No default flows are programmed. How traffic that does not match a flow is treated depends upon the switch implementation.

Proactive forwarding only—The following default flows are programmed on the switch:

- Punt Link Layer Discovery Protocol (LLDP) packets.
- Drop all other traffic.

On the **Device Connections** tab, click **Add Device** to add a device, click **Remove Devices** to remove a device, or click **Rediscover Devices** to rediscover a device. When you click **Rediscover Devices** tab, the **Rediscover Device** window is displayed. Click **Rediscover Device** so that the device gets deleted and rediscovered again.

In each device window, click **View**, **Edit**, or **Delete** to add a device, edit an existing device, or delete a device. The following details are displayed for each device in each device window:

- The name of the device and its IP address
- The username on the device
- The type of the mode, for example, NX-API
- The uptime on the device, for example, date and time
- The hardware on the node

On the **Device Groups** tab, click + **Group** to add a group of devices. In each group window, click **View**, **Edit**, or **Delete** to add a group of devices, edit an existing group of devices, or delete a group of devices respectively. The following details are displayed in each group window:

- The name of the node group, for example, Node Group Name One
- The names of the nodes in the group, for example, nx-tap-agg-sw1 and nx-tap-agg-sw2

On the **SPAN Management** tab, click + **Add Device** to add an APIC device or the production switch to the network. Click **Remove Devices** to delete the devices or click **Rediscover Devices** to rediscover the devices. The production switch should be a Cisco Nexus 9000 Series switch or Cisco Nexus 3000 Series switch in NXOS mode. The feature NXAPI has to be enabled on these production switches.




---

**Note** If a device is unreachable and disconnects from NDB, NDB tries to locate and connect to the device after every 30 seconds.

---

The following columns are displayed on the **SPAN Management** tab to display the information about the devices:

- IP Address
- Username
- Type: The APIC device is listed as AC and the production switch will belated here is listed as PS.
- Active IP
- Secondary IP Address
- Tertiary IP Address

- Action

You must add an APIC controller before you can set up SPAN session and SPAN destination.

Starting with Cisco NDB release 3.6, Global deny ACLs are automatically added to all non-configured (Edge SPAN/TAP & Monitor) interfaces on a device. The Global deny ACL feature is equivalent to Block Rx feature. By default, Global Deny ACL feature is enabled for a device. To disable the Global Deny ACL feature, you need to add the `configure.global.acls` parameter and set it to *false* in the `config.ini` file. After setting the `configure.global.acls` parameter, you need to restart the system to disable Global Deny ACLs on the newly added devices.



**Note** To disable Global Deny ACL during CLI upgrade, run the CLI upgrade command and then configure the `configure.global.acls` parameter to *false* in the `config.ini` file before restarting the NDB. For example:

```
/xnc upgrade --perform --target-home {xnc_directory_to_be_upgraded} [--verbose] [--backupfile
{xnc_backup_location_and_zip_filename}]
// In the config.ini file//
configure.global.acls=false
```

To disable Global Deny ACL features during configuration upload, set the `configure.global.acls` parameter to *false* in the `config.ini` file before restarting the NDB.

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.



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

```
mm.addDefaultISLDenyRules = false
```



**Note** To disable Default-Deny-ISL Default-Deny-ISL-ICMP features during CLI upgrade, run the CLI upgrade command and then configure the `mm.addDefaultISLDenyRules` parameter to `false` in the `config.ini` file before restarting the NDB. For example:

```
./xnc upgrade --perform --target-home {xnc_directory_to_be_upgraded} [--verbose] [--backupfile
{xnc_backup_location_and_zip_filename}]
// In the config.ini file//
mm.addDefaultISLDenyRules=false
```

To disable Default-Deny-ISL Default-Deny-ISL-ICMP features during configuration upload, set the `mm.addDefaultISLDenyRules` parameter to `false` in the `config.ini` file before restarting the NDB.

## Managing a Device in NDB

You can add, remove, or edit a device using NDB.

- Adding a Device
- Removing a Device
- Rediscovering a Device
- Managing Profile for a Device

## Adding a Device

Complete these steps to add a device.

### Procedure

- Step 1** Navigate to **ADMINISTRATION > Device Connections** tab.
- Step 2** Click **Add Device**, the **Add Device** dialog box opens.
- Step 3** In the **Add Device** dialog box, enter the following details:

**Table 3: New Device Details**

Field	Description
<b>IP address/Hostname</b>	The name or IP address of the device. To add multiple devices, add the hostnames or IP Addresses separated with the comma. For example, <code>ndb1.cisco.com, ndb2.cisco.com, ndb3.cisco.com</code> .
<b>Username/Password</b>	Select this option to add a device using username and password credentials.

Field	Description
<b>Profile</b>	Select this option to add a device using a profile. For more information about adding a device using profiles, see <a href="#">Profile Management, on page 86</a>
<b>Username</b>	Username for authenticating the device.
<b>Password</b>	Password for authenticating the device.
<b>Connection Type</b>	Type of connection supported. Currently, <b>NXAPI</b> is supported.
<b>Port</b>	The device communication port. For example, use port 80 for NX-API over HTTP and 443 for HTTPS.
<b>Set Auxiliary Node</b>	Indicates whether this NX-API connection is Auxiliary for the OpenFlow device.
<b>Device Prerequisites</b>	To set the device to default configuration required for NX-API type of connection. This option is available for NXAPI connection type only without Auxiliary mode. To know more about the Device Prerequisites, see <a href="#">Device Prerequisites</a> section.

**Step 4** Click **Add Device** to create and add the new device to NDB. The new device is listed on the **DEVICE CONNECTION** tab.

Global deny ACLs are automatically added to all non-configured interfaces (Edge SPAN/TAP, Packet Truncation, Remote Source, and Local and Remote Monitor) on a device. By default, Global Deny ACL feature is enabled on all the devices. You can disable the Global Deny ACL feature by setting the `configure.global.acls` parameter to **false** in the config.ini file. Ensure that you restart NDB after making changes in the configuration file.

By default, deny ACL is enabled on all the Inter Switch Links (ISL) interfaces causing all the traffic in the ISL interfaces to be dropped if there is no connection installed. 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 Nexus 9200, 9300EX, 9300FX, 9500EX, and 9500FX switches in NXAPI mode.

You can disable deny ACL on all the ISL interfaces by setting the `configure.global.acls` parameter to **false** in the config.ini file. Ensure that you restart NDB after making changes in the configuration file.

You can disable Global deny ACL or ISL deny ACL during the CLI upgrade or configuration upload by using the CLI upgrade command and setting the `configure.global.acls` parameter to **false** in the config.ini file. For example:

```
xnc upgrade --perform --target-home {xnc_directory_to_be_upgraded} [--verbose] [--backupfile {xnc_backup_location_and_zip_filename}]
```

```
Path:<NDBhome>/configuration/configure.global.acls=false
```

**Note** You can also disable Global deny ACL or ISL deny ACL by uploading the configuration in a Web browser.

---

## Removing a Device

To remove a device from NDB, complete these steps

### Procedure

---

- Step 1** Navigate to **Device Connections** tab.
  - Step 2** Select the device to remove from the table.
  - Step 3** Click **Remove Devices**. The **Remove Devices** dialog box opens.
  - Step 4** Verify the selected device(s) in the **Remove Devices** dialog box and click either of the two options:
    - **Remove Device**: Use this option to remove the device connection from NDB while retaining the device configuration.
    - **Purge & Remove Device**: Use this option to remove the device connection from NDB along with the device configuration.
- 

## Rediscovering a Device

To rediscover a device from NDB, complete these steps:

### Procedure

---

- Step 1** Navigate to **Device Connections** tab.
  - Step 2** Select the device(s) to rediscover from the table under the **Device Connections** tab.
  - Step 3** Click **Rediscover Devices**, the **Rediscover Devices** window appears.
  - Step 4** Verify the selected devices in the **Rediscover Devices** window.
  - Step 5** Click **Rediscover Devices** to rediscover the device(s).
- 

## Managing Profile for a Device

You can attach a profile to an existing device or change the profile attached to a device using NDB. Complete these steps to add or change a profile to a device:





---

**Note** This feature is currently supported in NXAPI mode only.

---

### Procedure

---

- Step 1** Navigate to **Device Connections** tab.
- Step 2** Click **Add/Change Profile to Device**, the **Add/Change Profile to Device** window appears.
- Step 3** Verify the selected devices in the **Add/Change Profile to Device** window. You can edit the following details in the **Add/Change Profile to Device** window:
- **Profile:** Select a profile to attach. For more information about the profiles, see [Profile Management](#) section.
  - **Connection Type:** Specify the supported connection type. Currently, **NXAPI** is supported.
  - **Port:** The device communication port. For example, use port 80 for NX-API over HTTP and 443 for HTTPS.
- 

## Device Prerequisites

Starting with Cisco NDB release 3.8, NDB pushes basic configuration to a newly added switch into NDB. Manual configuration of the NX-API devices to make it ready for NDB is not required. As a part of the adding a new device, the prerequisites are configured by NDB on the devices.

You need to ensure that NX-API is enabled on the new device for NDB to push prerequisite configuration successfully.

Following configurations are pushed into the new switch by NDB.

- TCAM configurations based on the device platform
- MST mode is enabled on the Spanning Tree
- Basic VLAN Configuration
- LLDP feature is enabled (only for the centralized mode of NDB)



---

**Note** Device is rebooted after all the configurations are successfully pushed by NDB. The device reboot is required because of the TCAM configurations. The reboot is supported from NX-OS is 9.2(3) and above

---

The Device Prerequisites can be configured when you add or edit a device, or when you add or change profile to device.

# Profile Management

Starting with Cisco NDB release 3.8, you can add, edit, or delete a profile through NDB. A profile allow you to manage multiple switches attached to a NDB. You can attach multiple switches to a profile. The profile configuration is applied to all the member switches.

## Adding a Profile

Complete these steps to add a profile:

### Procedure

---

- Step 1** Navigate to **ADMINISTRATION > User Management**.
  - Step 2** On the **User Management** page, click **Profile** tab.
  - Step 3** Click **Add Profile** to open the **Add Profile** dialog-box.
  - Step 4** Enter name of the profile in the **Profile Name** text-field.
  - Step 5** Enter the user name to be configured for the member switches in the **User Name** text-field.
  - Step 6** Enter the password to be configured for the member switches in the **Password** text-field.
  - Step 7** Click **Create** to create a new profile.
- 

## Editing a Profile

To edit a profile using NDB, complete these steps:

### Procedure

---

- Step 1** Navigate to **ADMINISTRATION > User Management > Profile** tab.
  - Step 2** Click **Edit** to edit a profile. The **Edit Profile** dialog box opens.
  - Step 3** Enter new name of the profile in the **Profile Name** text-field.
  - Step 4** Enter the new user name to be configured for the member switches in the **User Name** text-field.
  - Step 5** Enter the new password to be configured for the member switches in the **Password** text-field.
  - Step 6** Click **Submit**. All the devices that are part of the profile are rediscovered with the updated credentials.
- 

## Deleting a Profile

Complete these steps to delete a profile:

## Procedure

---

- Step 1** Navigate to **ADMINISTRATION > User Management > PROFILE**.
- Step 2** On the **Profile** page, click **Delete** for the profile to delete it. The deleted profile is removed from the **Profile** page.
-





## CHAPTER 6

# Configuring Cisco Nexus 9000 Series Switches

This chapter contains the following sections:

- [Guidelines and Limitations for Cisco Nexus 9000 Series Switches, on page 89](#)
- [Configuring TCAM Hardware Sizing on Cisco Nexus 9000 Series Switches, on page 90](#)
- [Enabling Cisco NX-API on Cisco Nexus 9000 Series Switches Using CLI, on page 91](#)
- [Enabling Switch Port Mode as Trunk on the Inter-switch Ports and Port Channels, on page 92](#)

## Guidelines and Limitations for Cisco Nexus 9000 Series Switches

See the following guidelines and limitations for configuring Cisco Nexus 9000 Series switches through Cisco Nexus Data Broker.

- Beginning with Cisco NX-OS Release 7.0(3)I7(2), you can enable TAP aggregation for Cisco Nexus 9500 platform switches with N9K-X9700-EX and N9K-X9700-FX line card.
- To enable TAP AGG feature on N9K-X9700-EX and N9K-X9700-FX line card, you need to configure hardware acl tap-agg globally on the Cisco Nexus 9500 switches.
- Cisco Nexus Data Broker supports NX-API protocol for Cisco Nexus 9000 series family of devices starting with Release 7.x.
- The devices that are going to be provisioned by Cisco Nexus Data Broker are assumed to have LLDP enabled and the LLDP feature should not be disabled during the device association with Cisco Nexus Data Broker. If the LLDP feature is disabled, there might be an inconsistency in Cisco Nexus Data Broker that cannot be fixed without device deletion and re-addition.
- Cisco Nexus Data Broker assumes that the device interfaces configured by the port definitions are L2 switch ports and these interfaces have device configurations as switchport trunk by default.
- Cisco Nexus 9200 Series switches do not support Q-in-Q VLAN tagging for the Edge SPAN and Edge TAP port.
- For Cisco Nexus 9000 Series switches, upgrade the Cisco NX-OS software to Cisco NX-OS Release 7.x or above.

- You can now add a Cisco Nexus 9000 Series switch to the Cisco Nexus Data Broker that can be discovered through NX-API protocol. Once the connection is successful, all the line card information for chassis model 9500 is discovered.
- Prior to deploying the Cisco Nexus 9000 Series switches for Tap/SPAN aggregation through Cisco Nexus Data Broker with NX-API mode, the following configurations should be completed:
  - Configure the ACL TCAM region size for IPV4 port ACLs or MAC port ACLs. For more information, see [Prerequisites for Cisco Nexus Series Switches](#).
  - Enable NX-API feature in the switch using the **feature nxapi** command.
  - Configure **switchport mode trunk** on all the inter-switch ports and the port-channels.
- Prior to deploying the Cisco Nexus 9000 Series switches for Tap/SPAN aggregation through Cisco Nexus Data Broker with OpenFlow mode, enter the command **hardware access-list tcam region openflow 512** at the prompt.
- Cisco Nexus data broker periodically rediscovers the switch inventory, the topology interconnection, and the status. This information is updated in the GUI depending on the status. The rediscovery interval can be configured and the default value for the rediscovery interval is every 10 seconds.
- Cisco Nexus Data Broker in OpenFlow mode supports Ethernet MAC source and destination addresses as match capabilities only when the OpenFlow TCAM region is configured as double wide (for example, hardware access-list tcam region openflow 512 double-wide). Only ether type match is supported in non double wide mode.

## Configuring TCAM Hardware Sizing on Cisco Nexus 9000 Series Switches

The TCAM configuration is based on the filtering requirement. You may need to configure multiple TCAM entries based on your filtering requirement. Complete these steps to configure a TCAM:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	Use the <b>hardware access-list tcam region &lt;region&gt; &lt;tcam-size&gt;</b> command to configure the following TCAM regions:	<ul style="list-style-type: none"> <li>• IPV4 PACL [ifacl] size = 1024</li> <li>• IPV6 PACL [ipv6-ifacl] size = 0</li> <li>• MAC PACL [mac-ifacl] size = 512</li> <li>• Egress IPV4 RACL [e-racl] size = 256</li> <li>• Egress IPV6 RACL [e-ipv6-racl] size = 0</li> <li>• Ingress System size = 256</li> <li>• Egress System size = 256</li> <li>• SPAN [span] size = 256</li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>Ingress COPP [copp] size = 256</li> </ul> <p>See the <i>Cisco Nexus 9000 Series NX-OS Security Configuration Guide</i> for the step-by-step TCAM hardware sizing configuration on Cisco Nexus 9000 Series Switches.</p> <p><b>Note</b> Cisco NDB in OpenFlow mode supports Ethernet MAC source and destination addresses as match capabilities only when the OpenFlow TCAM region is configured as double wide (for example, <b>hardware access-list tcam region openflow 512 double-wide</b>). If the OpenFlow TCAM region is configured as non double wide, only ether type match is supported as match capabilities.</p>

## Enabling Cisco NX-API on Cisco Nexus 9000 Series Switches Using CLI

You can now manage multiple Cisco Nexus 9000 Series switches that are connected in a topology. Cisco Nexus Data Broker plugin can discover the switch interconnections using LLDP and update the topology services within Cisco Nexus Data Broker. The switch interconnections can be a physical link or a port-channel interface. The topology displays only the interconnections between Cisco Nexus 9000 Series switches that are added to the Cisco Nexus Data Broker device list. The topology interconnection is displayed in the GUI.

Complete the following steps for enabling Cisco NX-API on Cisco Nexus 9000 Series switches:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	Enable the management interface.	Enable the management interface on the switch.
<b>Step 2</b>	switch# <b>conf t</b>	Enter the configuration mode.
<b>Step 3</b>	switch (config) # <b>feature nxapi</b>	Enable the NX-API feature.
<b>Step 4</b>	switch (config) # <b>nxapi http port 80</b>	Configure the HTTP port.
<b>Step 5</b>	switch (config) # <b>nxapi https port 443</b>	Configure the HTTPS port.
		For the step-by-step configuration information for enabling the NX-API feature on Cisco Nexus 9000 Series switches, see the <i>Cisco</i>

	Command or Action	Purpose
		<i>Nexus 9000 Series NX-OS Programmability Guide.</i>

## Enabling Switch Port Mode as Trunk on the Inter-switch Ports and Port Channels

Complete the following steps to enable the switch port mode on the inter-switch ports and port-channels:

### Procedure

	Command or Action	Purpose
<b>Step 1</b>	switch(config)# <b>config t</b>	Enables the configuration mode.
<b>Step 2</b>	switch(config)# interface {{ <b>type slot/port</b> }   { <i>port-channel number</i> }}	Specifies an interface to configure.
<b>Step 3</b>	switch(config-if)# <b>switchport mode</b> { <i>access</i>   <i>trunk</i> }	Configures the switchport mode as access or trunk on the inter-switch ports and the port-channels.
<b>Step 4</b>	switch(config)# <b>exit</b>	Exits the configuration mode.





## CHAPTER 7

# Configuring the Nexus Data Broker

---

This chapter contains the following sections:

- [Viewing Topology, on page 93](#)
- [Configuring Port Definition, on page 94](#)
- [Configuring Port Groups, on page 99](#)
- [Adding a Remote Monitoring Device, on page 101](#)
- [Adding a Monitoring Device, on page 102](#)
- [Editing In Use Monitoring Device, on page 103](#)
- [Adding a Service Node, on page 103](#)
- [User Defined Field, on page 105](#)
- [Adding Filters, on page 106](#)
- [Adding Connections, on page 115](#)
- [Connection with AutoPriority, on page 119](#)
- [Adding Redirections, on page 121](#)
- [Viewing Statistics, on page 125](#)
- [Viewing Connection Port Statistics, on page 127](#)
- [Deleting Flow and Port Statistics, on page 127](#)
- [Purging Device Configuration, on page 128](#)
- [Exporting and Importing NDB Configuration, on page 128](#)
- [Managing Sampled Flow Configuration , on page 129](#)
- [Configuring Symmetric/Non-Symmetric Load Balancing and MPLS Tag Stripping , on page 130](#)
- [Configuring PTP Using NDB, on page 131](#)
- [Configuring Packet Truncation, on page 132](#)

## Viewing Topology

Click the **Topology** tab in the left frame to view the topology in the network.



---

**Note** Starting with Cisco NDB Release 3.7, additional information such as hardware detail and software detail is displayed for each device along with the port count in the topology diagram.

---

# Configuring Port Definition

When you click **Port Definition** tab in the GUI, the **Port Definition** screen is displayed. Select the switch from the drop-down list to configure the ports.



---

**Note** Starting with Cisco NDB release 3.8, you can search all the switches in a network using the advanced search option.

---

On the **Port Definition** screen, the following two tabs are displayed:

- Port Configuration
- SPAN Destination

Click the **Port Configuration** tab, the following tabs are displayed:

- Configure Multiple Ports
- Remove port Configuration
- Add Service Node
- Add Monitoring Device

When you click **Configure Multiple Ports** tab, the **Configure Multiple Ports** window is displayed. The following details are displayed on the screen: Number, Status, Port Name, Type, In Use, Port ID, and Action.



---

**Note** Beginning with Cisco Nexus Data Broker, Release 3.1, the interface description is updated from the Cisco Nexus Data Broker GUI to the switch and the interface description is also available from the switch into the Cisco Nexus Data Broker GUI. When using in Openflow mode, the NX-API auxiliary connection is required for this functionality to work.

---



---

**Note** On the Port Configuration tab, the port name and the interface are displayed as hyperlinks. When you click the port name, you can view the running configuration for that interface on the tab.

---

If you want to remove any ports, select the port and click **Remove port Configuration** tab.

Click **Add Service Node** to add a service node.

Click **Add Monitoring Device** to add a monitoring device.

On the **Port Configuration** screen, the following port details are displayed for the selected node:

- Serial Number
- Status
- Port name

- Type
- In Use
- Port ID
- Action—When you click **Configure**, the **Configure Ports** window is displayed.

On the **SPAN Destination** tab, the following details are displayed:

- SPAN Destination Name
- SPAN Destinations
- Node Connector
- Monitor Port Type
- Description

## Configuring Ports

Complete the following steps to configure a port.

### Procedure

---

- Step 1** Select the switch for which you want to configure the port details on the **Port Configuration** screen.
- Step 2** Click **Configure** under **Action**.  
The **Configure Ports** window is displayed.
- Step 3** In the **Configure Ports** window, configure the port type from the **Select a port type** drop-down list by selecting one of the following options:
- **Add Monitoring Device**
  - **Edge Port-SPAN**
  - **Edge Port-TAP**
  - **Production Port**

**Monitoring Device**—Creates a monitoring device for capturing traffic and configures the corresponding delivery port.

**Edge Port-SPAN**—Creates an edge port for incoming traffic connected to an upstream switch that is configured as a SPAN destination.

**Edge Port-TAP**—Creates an edge port for incoming traffic connected to a physical TAP port.

**Production Port**—Creates a production port for the ingress and egress traffic.

**Note** Starting with Cisco NDB Release 3.6, you can configure Maximum Transmission Unit (MTU) for all the Cisco Nexus 9xxx devices in NX-API mode using NDB GUI. MTU can be configured at the system level (node level or global level) and the Interface Level (supported only on SPAN and TAP ports). The default value for MTU is 1500 and you can configure MTU between 1500 and 9216.

Jumbo MTU is the maximum MTU that can be configured for a node. When you configure Jumbo MTU at the system level, the same MTU value is applied to all the node ISLs.

- Note** Starting with Cisco NDB Release 3.4, a description can have a leading numerical for Edge-SPAN, Edge-TAP, Monitoring devices, Production ports, Filter names, Connections, and Redirections (NX-API, OpenFlow and NX-AUX mode).
- Note** To receive the traffic from the production network, the production ingress port is configured. After entering the service nodes (multiple security tools), the traffic exits the data center through the production egress port.
- Note** Starting with Cisco Nexus Data Broker, Release 3.2, when Edge-SPAN, Edge-TAP, monitoring device, or production port is defined in NX-API mode of configuration, the CLI command, **spanning-tree bpdudfilter enable** is automatically configured in the interface mode on the ports to filter the BPDU packets. This configuration is applicable for all Cisco Nexus 3000 and 9000 Series switches. The sample configuration is displayed in the example:

```
switch#
show run interface eth1/1
interface ethernet1/1
switchport mode trunk
mode tap-aggregation
spanning-tree bpdudfilter enable
```

- Note** Production port has be enabled for Q-in-Q in Cisco Nexus Data Broker and a unique VLAN should be assigned for each production port. This VLAN should not overlap with any production VLAN numbers.
- Note** The **spanning-tree bpdudfilter enable** CLI command should be configured by the user on all the inter-switch ports for all Cisco Nexus series switches and Cisco Nexus Data Broker does not configure this command.
- Note** Once an interface is configured with Q-in-Q, do not configure multiple VLAN filters for the Q-in-Q configured interface.

When you select the port type, the title of the window changes to **Manage Configure Ports**.

- Step 4** (Optional) In the **Port Description** field, enter the port description.  
Beginning with Cisco Nexus Data Broker, Release 3.1, the interface description is updated from the Cisco Nexus Data Broker GUI to the switch and the interface description is also available from the switch into the Cisco Nexus Data Broker GUI. When using in Openflow mode, the NX-API auxiliary connection is required for this functionality to work.
- Step 5** Required: Enter VLAN ID for the port.  
The port is configured as dot1q to preserve any production VLAN information. The VLAN ID is used to identify the port that the traffic is coming from.
- Step 6** (Optional) If APIC is available, you can select the ACI side port and designate it as the SPAN destination port.
- Step 7** In the **Enable Packet Truncation** field, enter the packet length.
- Step 8** A check box is added for **Block Tx** and it is applicable for both Edge-SPAN and Edge-TAP where you can block the traffic that is being transmitted out of Edge-SPAN AND Edge-TAP interface.

- Step 9** A check box is added for **Drop ICMPv6 Neighbour Solicitation** and by default all the ICMP traffic is blocked for all the Edge-SPAN and Edge-TAP port types for Nexus 9300-EX and 9200 Series switches. For the rest of Nexus 9000 Series switches, user has to manually enable this feature to deny or block all the ICMP traffic. This feature is currently available on NX-API based switches for NX-OS versions I5 and later.
- Step 10** A check box is added for **Enable Timestamp Tagging** to append timestamp tag on the packets using the Timestamp Tagging feature. You can configure this feature on a single device or multiple devices.
- For Nexus 9300-EX and 9200 series switches, this feature is applicable for Edge-SPAN and Edge-TAP ports and on the Monitoring devices. To configure Timestamp Tagging feature, ensure that PTP feature is enabled on the device. You need to enable Timestamp tagging on monitoring device and edge ports. If Timestamp Tagging feature is not configured on either side of the connection, Edge-SPAN/Edge-TAP and Monitor Devices, the packets are not tagged with timestamp. For Nexus 3500 Series switches, the Timestamp Tagging feature is applicable only for the Monitoring devices.
- Step 11** Click **Submit** to save the settings or click **Clear** to clear the details.
- Once you configure a port, you can click **Edit** under **Action** on the **Port Configuration** screen to edit the port details. You can click **Remove** under **Action** on the **Port Configuration** screen to clear the port details.

## Editing In Use Ports

Starting with NDB Release 3.4, you can edit the select fields under Port configuration(Edge-Span, Edge-Tap or Production) while in use. Ports can be edited in all the modes of connection. The following table lists the fields that you can edit for port in use.:

Section	Field	Editable
Port Configuration	Port Description	Yes
	Block Tx	Yes
	Port Type	No
	VLAN Packet Truncation	No
	Drop ICMPv6 Neighbour Solicitation	Yes
	Enable Timestamp Taggin	Yes

## Enabling or Disabling Ports

Starting with Cisco NDB Release 3.4, you can now enable or disable an interface using the NDB GUI. This feature is currently available for NX-API and NX-AUX based switches. A switch based on OpenFlow mode does not support this feature.

### Procedure

- Step 1** Select the switch for which you want to configure the port details on the **Port Configuration** screen.

**Step 2** Click **Enable/Disable** to enable or disable the selected port.

**Note** You can enable or disable only one interface at a time.

## Adding SPAN Destination

When you configure a port as an edge SPAN port and the port is connected to the API side, you can select the pod, node, and port from the ACI side and set the port as SPAN destination.



**Note** You can add SPAN destination only after APIC has been successfully added to the network.

### Procedure

- Step 1** Select the switch for which you want to configure the port details on the **Port Configuration** screen.
- Step 2** Click **Configure** under **Action**.  
The **Configure Ports** window is displayed.
- Step 3** In the **Configure Ports** window, configure the port type from the **Select a port type** drop-down list by selecting **Edge Port-Span**.
- Step 4** Click **SPAN Destination**, the **Select SPAN Destination** window is appears. From the **Select APIC Node** drop-down list, select an APIC node.
- Step 5** From the **Select Node** drop-down list, select an APIC device.
- Step 6** Select corresponding leaf switch, node and port from the **Select Pod** drop-down list, **Select Node** drop-down list, and **Select Port** drop-down list to configure the SPAN Destination.
- Step 7** Click **Apply** to save the settings.
- Step 8** (Optional) In the **Configure Ports** dialog box, under **Destination** section, specify MTU for the device in the **MTU** field. You can configure MTU range between 1 and 9216.
- Step 9** Click **Submit**.

## Configuring Multiple Ports

You can configure multiple ports for a node.

### Procedure

- Step 1** Click **Configure Multiple Ports** on the **Port Configuration** screen. The **Configure Multiple Ports** window is displayed.
- Step 2** Use **CTRL/SHIFT** to select multiple ports in the **Select Ports** field.
- Step 3** Select port type from the drop-down list in the **Select Port Type** field.

- Step 4** Click **Submit** to save the settings.
- 

## Configuring Port Groups

You can create a port group and add the ports to the connection. Starting with Cisco Nexus Data Broker, Release 3.2, you can create port groups for different source ports. The port groups can be a combination of the edge-span and the edge-tap ports across different switches. You can select ports, define port groups, provide a name to the port group, select the port group in a connection screen (only one port group per connection), and use the ports defined in the port group as source ports for creating a connection. Selecting individual ports is disabled when using a port group.

Complete the following steps to configure port groups:

### Procedure

---

- Step 1** Select the switch for which you want to configure the port details on the Port Configuration screen.
- Step 2** Click **Port Groups** tab in the left frame.
- Step 3** Click + **Add Group** to create a port group.
- Step 4** In the **Create Port Group** window, enter the group name in the **Group Name** field.
- Step 5** In the **Select Node** field, select a node, for example, N9K-116.
- Step 6** In the **Select Port** field, select a port, for example, Ethernet1/1 (Ethernet1/1).

You can add only edge-span and edge-tap ports and you cannot add production ports to the port groups.

- Step 7** Click + **Add To Group** to add the port to the group.

You can add multiple ports to the group.

- Step 8** Click **Apply**.

The port group is displayed on the **Port Groups** screen with the following information for the group, for example, **Name**, **Connection Name**, **Ports** and **Action**.

Starting with NDB 3.4 release, you can now configure selected fields under Port Group. This feature is supported on the switches running in NX-API, OpenFlow, or NX-AUX mode.

Starting with Cisco NDB Release 3.7, **Created By** and **Modified By** information is displayed for Port Groups in the NDB UI.

The following table lists the fields that you can configure for a Port Group:

Section	Field	Editable
Port Group	Port Description	Yes
	Port	Yes
	Port Name	Yes (If the port is not part of an active connection)
	Port Group	Yes (If the port group is not part of an active connection)

## Editing In Use Port Groups

You can edit an in-use port group using NDB GUI. To edit an in-use port group, complete the following steps:



**Note** Starting with Cisco NDB release 3.8, you can now rename an in-use port group.



**Note** Starting with NDB 3.4 release, you can edit the port groups that are currently in use in a connection. This feature is supported on the switches running in NX-API, OpenFlow, or NX-AUX mode.

### Procedure

- Step 1** Select the switch for which you want to configure the port details on the **Port Group** pane.
- Step 2** Click **Edit** on the listed table row.

## Editable Attributes for In Use Port Groups

The following table lists the fields that you can edit for a Port Group that is currently in use:

Section	Field	Editable
Port Group	Port Description	Yes
	Port	Yes
	Port Name	Yes.
	Port Group	Yes (If the port group is not part of an active connection)



# Adding a Remote Monitoring Device

Starting with Cisco NDB Release 3.7, you can now use a device outside the network as a monitoring device using the Encapsulated Remote Switch Port Analyzer (ERSPAN) Source Session feature for Cisco Nexus 9300 FX and EX series switches. This feature enables you to direct the traffic for monitoring outside the local network



**Note** You can associate a remote delivery port to only one destination port group.

To add a new remote monitoring device, complete these steps:

## Procedure

- Step 1** Navigate to the **Monitoring Device** tab under **Configuration**.
- Step 2** Click **Add Monitoring Device**.
- Step 3** In the **Monitoring Device** window, complete the following fields:

Name	Description
<b>Monitoring Device Name</b>	Add the service node name.  <b>Note</b> The valid characters for the monitoring devices are the alphanumeric characters and the special characters: period ("."), underscore ("_"), and hyphen ("-").
<b>Select Switch Node</b>	Select the switch node.
<b>Select Port</b>	Select the port.
<b>Port Description</b>	Description for the port.
<b>Block Rx</b>	Not applicable
<b>Remote Monitor Device</b>	Select this option to indicate that the monitoring device is available in the local network.
<b>Enable Timestamp Tagging</b>	Enable Timestamp Tagging for remote monitoring.
<b>Enable Timestamp Strip</b>	Enable Timestamp Strip for remote monitoring.
<b>MTU</b>	Enable MTU for remote monitoring.
<b>Icons</b>	Select a Monitoring Device Icon.
<b>Interface IP</b>	IP address to be assigned to the selected interface.
<b>Destination IP</b>	IP Address where ER-SPAN terminates and should be reachable from the selected port.

**Step 4** Click **Submit** to create the monitoring device.

**Note** You cannot use more than one remote delivery port per switch per connection.

**Note** You cannot share the same source interface across with multiple connections.

**Note** Remote monitor tool involving inter switched links is restricted to only one connection per ISL.

## Adding a Monitoring Device

To add a new monitoring device, complete these steps:

### Procedure

**Step 1** Navigate to the **Monitoring Device** tab under **Configuration**.

**Step 2** Click **Add Monitoring Device**.

**Step 3** In the **Monitoring Device** window, complete the following fields:

Name	Description
<b>Monitoring Device Name</b>	Add the service node name.  <b>Note</b> The valid characters for the monitoring devices are the alphanumeric characters and the special characters: period ("."), underscore ("_"), and hyphen ("-").
<b>Select Switch Node</b>	Select the switch node.
<b>Select Port</b>	Select the port.
Port Description	Description for the port.
<b>Icons</b>	Select a Monitoring Device Icon.
Local Monitor Device	Indicates that the monitoring device is available in the local network.
<b>Block Rx</b>	Block any traffic from being received from the monitoring tools. This option is selected by default. You can turn this option off by unchecking the box.
<b>Enable Timestamp Tagging</b>	Time stamp tagging is supported on Cisco Nexus 3500, 9200 and 93XXX-EX Series switches. Cisco Nexus 3500 Series switches require NX-OS Release 6.0(2)A8(1) or later version.

Name	Description
Enable Timestamp Strip	Select this option to remove timestamp tag from the source packets.
MTU	You can configure MTU at the system level (node level or global level) and the Interface Level. The default value for MTU is 1500 and you can configure MTU between 1500 and 9216. Jumbo MTU is the maximum MTU that can be configured for a node. When you configure Jumbo MTU at the system level, the same MTU value is applied to all the node ISLs.

**Step 4** Click **Submit** to create the monitoring device.

## Editing In Use Monitoring Device

Starting with Cisco NDB, Release 3.4, you can edit a monitoring device configuration using the NDB GUI. Support to edit description of a Monitoring device is available for NX-API, OpenFlow, and NX-AUX based switches. For the OpenFlow devices, the updated descriptions are synchronized to NDB User Interface (UI) only. For NX-API and NX-AUX devices, the updates are synchronized to NDB UI and the switch interface.

The following table lists the fields that you can configure for Monitoring Devices:

Section	Field	Editable
Monitor Devices	Monitor Devices Name	Yes (If the monitoring device is not in use)
	Port Description	Yes
	Block Rx	Yes
	Icons	Yes
	Enable Timestamp Tagging	Yes
	Enable Timestamp Strip	Yes
	MTU	Yes

## Adding a Service Node

Starting with Cisco NDB Release 3.7, **Created By** and **Modified By** information is displayed for Service Nodes in the NDB UI.

Complete the following steps to add a service node:

## Procedure

- 
- Step 1** Navigate to the **Service Nodes** tab under **Configuration** and click + **Service Node**.
  - Step 2** In the **Add Service Node** window, enter the name of the service node.
  - Step 3** Select the ingress port for the service node from the **Service Node Ingress Port** drop-down list.
  - Step 4** Select the egress port for the service node from the **Service Node Egress Port** drop-down list.
  - Step 5** Enable health check on a service node by selecting the **Service Node Health Check** option.

Beginning with Cisco Nexus Data Broker, Release 3.2, you can configure the wait interval in the **config.ini** file before the health check is up. The **ServiceNodeHealthCheckWaitInterval** is the variable in the **config.ini** file to set the wait interval. If you do not specify a value or if the value is 0 for the wait interval in the **config.ini** file, the default value of 5 Seconds is used. The wait interval is not applicable if the port is in shutdown state.

This option works only in the OpenFlow mode. The controller or the NDB injects a packet in the service node ingress port and the packet is received at the egress port. The packets are checked at the interval of every 5 seconds. If five packets are not received in 5 seconds, the health of the service node is considered as down.

For the service node, a new field is displayed in the details: Service Node Status. This field displays the status of the service node.

- Step 6** Select a service node icon from the available options.
  - Step 7** Click **Save**.
- 

## Editable Fields for an Active Service Node

Starting with Cisco NDB, Release 3.4, you can now edit and configure Service Node fields using the NDB GUI. For the OpenFlow devices, the updated descriptions are synchronized to NDB User Interface (UI) only. For NX-API and NX-AUX devices, the updates are synchronized to NDB UI and the switch interface.

The following table lists the fields that you can configure for an active Service Node:

Section	Field	Editable
Service Node	Description	Yes
	Icon	Yes
	Service Node Health Check	Yes
	Service Node Name	Yes (If the service node is not in use)
	Service Node Ingress Port	No
	Service Node Egress Port	No
	Ingress port Description	Yes
	Egress port Description	Yes

## User Defined Field

You can define a User Defined Field (UDF) and use it while creating a filter for traffic management.

Starting with Cisco NDB Release 3.7, **Created By** and **Modified By** information is displayed for User Defined Fields in the NDB UI.



**Note** Starting with Cisco NDB Release 3.6, you can add multiple UDFs while defining a filter. You can add up to four UDFs for each filter.

**Table 4: UDF Support Matrix**

UDF Ethertype	NDB Version	NXOS Version	Platform
IPv4	3.3	7.0(3)I5(2)	9200 & 9300
IPv6	3.6	7.0(3)I6(1)	93xx EX/FX , 95xx EX/FX , 92xx



**Note** Please refer the NXOS documentation for number of UDF support per platform.

To use UDF to manage traffic, you need to:

- Define a UDF, see [Defining a UDF](#).
- Create a filter using the UDF, see [Adding Filters](#).
- Apply the filter (configured with UDF) to a connection to manage traffic, see [Adding Connections](#).

**Table 5: Qualifying Region for UDF for Different Platforms**

Platform	UDF Qualifying TCAM region
Cisco Nexus 9200, 9300-EX/9300-FX and 9500-EX/9500-FX	ing-ifacl
Other platforms	ifacl

## Defining a UDF

Complete the following steps to define a UDF:

### Procedure

**Step 1** Log into NDB application.

**Step 2** Navigate to **Configuration** tab, click **UDF Definition** to define a user defined filter. The **UDF Definition** window is displayed.

**Step 3** In the **UDF Definition** window, complete the following fields:

Name	Description
Name field	The name of the user defined field.
Keyword	If <b>Header</b> option is selected, the Inner (Offset base from inner/outer header) and L3/L4 (Offset base from L3/L4 header) is enabled.
Offset field	Number of characters to offset while using matching criteria.
Devices	Cisco Nexus 9000 Series switch name.
Type	Specify UDFv4 or UDFv6 support.

**Step 4** Click **Add UDF**. The newly added UDF appears in the **UDF Definition** window.

**Note** Any change in a UDF definition requires device reboot.

**Note** By default, NDB generates a UDF named *udfInnerVlan* and *udfInnerVlanv6*, used to match the inner VLAN in the ISL ports.

**Note** The icon for UDF is yellow in color immediately after it is created. After you reboot the device, if the UDF is successfully installed, the UDF icon color changes to green, else it changes to red color.

## Adding Filters

Beginning with Cisco Nexus Data Broker, Release 3.3, the Default-Match-All filter includes the following protocols packet filtering:

- IPv4
- IPv6
- ARP
- MPLS Unicast
- MPLS Multicast
- MAC

**Before you begin**

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



**Note** Beginning with NDB 3.4 release, you can now edit Filter Name using the NDB GUI. This feature is supported on the switches running in NX-API, OpenFlow, or NX-AUX mode.



**Note** Starting with Cisco NDB Release 3.7, **Created By** and **Modified By** information is displayed for filters in the NDB UI.

**Procedure**

**Step 1** On the **Filters** tab, click **Add Filter** to add a filter. The **Add Filter** window is displayed.

**Step 2** In the **Filter Description** section of the **Add Filter** window, complete the following fields:

Name	Description
Name field	The name of the filter.  <b>Note</b> The name cannot be changed once you have saved it.
Bidirectional check box	Check this box if you want the filter to capture traffic information from a source IP, source port, or source MAC address to a destination IP, destination port, or destination MAC address, and from a destination IP, destination port, or destination MAC to a source IP, source port, or source MAC address.

**Step 3** In the **Layer 2** section of the **Add Filter** window, complete the following fields:

<b>Ethernet Type</b> field	<p>Required. The Ethernet type of the Layer 2 traffic. The default value displayed is IPv4, or you can choose one of the following:</p> <ul style="list-style-type: none"> <li>• IPv6</li> <li>• ARP</li> <li>• LLDP</li> <li>• Predefined EtherTypes</li> <li>• All EtherTypes</li> <li>• Enter Ethernet Type—If you choose Enter Ethernet Type as the type, enter the Ethernet type in hexadecimal format. If you choose Predefined EtherTypes, all predefined Ethernet types contained in the config.in file are associated with the rule, and you should not configure any other parameters.</li> </ul>
<b>VLAN Identification Number</b> field	The VLAN ID for the Layer 2 traffic. You can enter a single VLAN ID, a range of VLAN ID values, or comma-separated VLAN ID values and VLAN ID ranges, for example, 1-4,6,8,9-12.
<b>VLAN Priority</b> field	The VLAN priority for the Layer 2 traffic.
<b>Source MAC Address</b> field	The source MAC address of the Layer 2 traffic.
<b>Destination MAC Address</b> field	The destination MAC address of the Layer 2 traffic.

**Step 4** In the **Layer 3** section of the **Add Filter** window, update the following fields:



Name	Description
Source IP Address field	<p>The source IP address of the Layer 3 traffic. This can be one of the following:</p> <ul style="list-style-type: none"> <li>• The host IP address, for example, 10.10.10.10</li> <li>• Discontiguous source IP address, for example, 10.10.10.10, 10.10.10.11, 10.10.10.12</li> <li>• An IPv4 address range, for example, 10.10.10.10-10.10.10.15</li> <li>• An IPv4 subnet, for example, 10.1.1.0/24</li> <li>• The host IP address in IPv6 format, for example, 2001::0</li> <li>• Combination of range and simple IP addresses, for example, 4.4.4.1,4.4.4.2-4.4.4.4,4.4.4.5.</li> </ul> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• When a switch is used in NX-API mode, you can now select an IPv6 filter and setup a connection. You can enter a single IPv6 address, comma separated multiple IPv6 addresses, an IPv6 address range, and/or IPv6 subnet in the <b>Source IP Address</b> field.</li> <li>• If you configure a range of Layer 3 source IP addresses, you cannot configure ranges of Layer 4 source or destination ports.</li> <li>• If you configure a range of Layer 3 source IP addresses, you cannot configure ranges of Layer 2 VLAN identifiers.</li> </ul> <p><b>Note</b> When using IPv6 address in the filter, the <b>Ethernet Type</b> should be set to IPv6.</p>

Name	Description
<b>Destination IP Address</b> field	<p>The destination IP address of the Layer 3 traffic. This can be one of the following:</p> <ul style="list-style-type: none"> <li>• The host IP address, for example, 10.10.10.11</li> <li>• An IPv4 address range, for example, 10.10.10.11-10.10.10.18</li> <li>• An IPv4 subnet, for example, 10.1.1.0/24</li> <li>• The host IP address in IPv6 format, for example, 2001::4</li> <li>• The subnet, for example, 10.0.0.0/25</li> </ul> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• When a switch is used in NX-API mode, you can now select a IPv6 filter and setup a connection. You can enter a single IPv6 address only in the <b>Destination IP Address</b> field. The comma separated multiple IPv6 addresses, an IPv6 address range, and/or IPv6 subnets are not supported. The hardware command that is a pre-requisite is for using the IPv6 feature is <b>hardware access-list tcam region ipv6-ifacl 512</b> .</li> <li>• If you configure a range of Layer 3 source IP addresses, you cannot configure ranges of Layer 4 source or destination ports.</li> <li>• If you configure a range of Layer 3 source IP addresses, you cannot configure ranges of Layer 2 VLAN identifiers.</li> </ul>
<b>Protocol</b> drop-down list	<p>Choose the Internet protocol of the Layer 3 traffic. This can be one of the following: If you choose Enter Protocol as the type, enter the protocol number in decimal format.</p> <ul style="list-style-type: none"> <li>• ICMP</li> <li>• TCP</li> <li>• UDP</li> <li>• Enter Protocol</li> </ul>

Name	Description
UDF drop-down list	<p>User Defined Field.</p> <ul style="list-style-type: none"> <li>• UDF: UDF name.</li> <li>• UDF value: Value to be matched in decimal notation (0-65535). Example: if you want to match 0x0806, enter 2054 which is 0x0806 in decimal notation.</li> <li>• UDF Mask: Mask value in packet to be matched. Mask is applied to the value for matching purposes. Example: to exactly match 2054 (0x0806) enter 65535 (0xffff), to match 2048-2063 (0x0800-0x080f) use 65520 (0xffff0).</li> </ul> <p><b>Note</b> Starting with Cisco NDB Release 3.6, you can add upto four UDFs to a filter.</p> <p><b>Note</b> If Add default udf for inner vlan option is selected, you can add only one UDF (UDF that matches outervlan offset). The udfInnerVlan is applied to the ISL links along with QinQ vlan specified on the SPAN port.</p> <p><b>Note</b> UDF option is enabled for IPv4 and IPv6 ethertypes.</p>
ToS Bits field	The Type of Service (ToS) bits in the IP header of the Layer 3 traffic. Only the Differentiated Services Code Point (DSCP) values are used.
Advanced Filter field	Advanced filter, combination of Ethernet type and attributes to manage traffic.

**Step 5** In the **Layer 4** section of the **Add Filter** dialog box, complete the following fields:

Name	Description
<b>Source Port</b> drop-down list	<p>Choose the source port of the Layer 4 traffic. This can be one of the following:</p> <ul style="list-style-type: none"> <li>• FTP (Data)</li> <li>• FTP (Control)</li> <li>• SSH</li> <li>• TELNET</li> <li>• HTTP</li> <li>• HTTPS</li> <li>• Enter Source Port</li> </ul> <p><b>Note</b> Beginning with Cisco Nexus Data Broker Release 3.2 , you can enter comma separated single port numbers and a range of the source port numbers in the <b>Enter Source Port</b> field.</p> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• If you configure a range of Layer 4 source ports, you cannot configure ranges of Layer 3 IP source or destination addresses.</li> <li>• If you configure a range of Layer 4 source ports, you cannot configure ranges of Layer 2 VLAN identifiers</li> </ul>

Name	Description
<b>Destination Port</b> drop-down list	<p>Choose the destination port of the Layer 4 traffic. This can be one of the following:</p> <ul style="list-style-type: none"> <li>• FTP (Data)</li> <li>• FTP (Control)</li> <li>• SSH</li> <li>• TELNET</li> <li>• HTTP</li> <li>• HTTPS</li> <li>• Enter Destination Port</li> </ul> <p><b>Note</b> Beginning with Cisco Nexus Data Broker Release 3.2 , you can enter comma separated single port numbers and a range of the source port numbers in the <b>Enter Destination Port</b> field.</p> <p><b>Note</b></p> <ul style="list-style-type: none"> <li>• If you configure a range of Layer 4 destination ports, you cannot configure ranges of Layer 3 IP source or destination addresses.</li> <li>• If you configure a range of Layer 4 destination ports, you cannot configure ranges of Layer 2 VLAN identifiers</li> </ul>

**Step 6** In the **Layer 7** section of the **Add Filter** dialog box, complete the following fields:

Name	Description
<b>HTTP Method field</b>	<p>You can configure matching on the HTTP methods and redirect the traffic based on that method. Select one or more methods to match within a single filter. This option is available only when the destination port is HTTP or HTTPS.</p> <ul style="list-style-type: none"> <li>• Connect</li> <li>• Delete</li> <li>• Get</li> <li>• Head</li> <li>• Post</li> <li>• Put</li> <li>• Trace</li> </ul> <p><b>Note</b> Layer 7 match is supported only with the NX-API mode only and it is not supported in OpenFlow.</p> <p><b>Note</b> The TCP option length is enabled when you select any one of the methods from Layer 7 traffic.</p>
<b>TCP Option Length field</b>	<p>You can extend the filter configuration to specify the TCP option length in the text box. The default value on the text box is 0. All methods within the filter have the same option length.</p> <p>Enter the TCP option length in a decimal format.</p> <p><b>Note</b> The value on the text box should be in the multiples of 4 and it can range from 0-40.</p>

**Step 7** Click **Add Filter**.

## Advanced Filter support

Starting with Cisco Nexus Data Broker, Release 3.3, advanced filtering option is available to manage the traffic. Advanced filtering provides multiple options to filter (permit or deny) the traffic based on Ethernet type and attributes such as Acknowledgment, FIN, Fragments, PSH, RST, SYN, DSCP, Precedence, TTL, packet-length, and NVE. Advanced filtering is available for the following Ethernet types and options:

Table 6: Advanced Filtering Support

Data Type	Supported Options
IPv4	DSCP, Fragment, Precedence, and TTL
IPv4 with TCP	Acknowledgment, DSCP, Fragment, FIN, Precedence, PSH, RST, SYN, and TTL
IPv4 with UDP	DSCP, Fragment, Precedence, and TTL
IPv6	DSCP and Fragment
IPv6 with TCP	Acknowledgment, DSCP, Fragment, FIN, PSH, RST, and SYN
IPv6 with UDP	DSCP and Fragment

**Important**

Advanced Filtering is available only for NX-API on Cisco Nexus 9000 platform.

**Important**

The value of Time to Live (TTL) attribute ranges from 0 to 255.

- For Nexus 9200 devices, the maximum value of TTL that can be set is 3.
- For rest of the Nexus 9000 series devices, the maximum TTL value can be 3 for NX-OS version 7.0(3)I6(1) and above. For NXOS versions 7.0(3)I4(1) and below, you can configure any value within the range.

While configuring advanced filtering support, you cannot:

- Configure DSCP and Precedence together in advance filtering.
- Configure fragments and ACK or SYN or FIN or PSH or RST together in advance filtering.
- Configure fragments and port numbers with UDP and IPv4 or IPv6 Combination
- Configure Precedence and HTTP Methods with IPv4 and TCP Combination.

## Adding Connections

Starting with Cisco Nexus Data Broker Release 3.7, you can lock a connection while creating it using the **Lock Connection** option on the **Add Connection** page. Locking a connection prevents unauthorized changes to a connection.

### Before you begin

- Add a filter to be assigned to the connection.
- Configure a monitoring device (optional).

- Configure an edge port or multiple edge ports (optional).

### Procedure

**Step 1** On the **Connections** tab, click + **Connection**. The **Add Connections** window is displayed.

**Step 2** In the **Add Connections** window, you can add the **Connection Name** and the **Priority** of the connection in the **Connection Details** area:

Field	Description
<b>Connection Name</b>	The name of the connection.
<b>Description</b>	Enter the description when creating a new connection.
<b>Priority</b>	The priority that you want to set for the connection. Connection by default has priority of 100. It can be changed in the range of <1-10000>.
<b>Lock Connection</b>	Select this option to lock the connection you are creating to prevent any unauthorized modification.

**Step 3** In the **Allow Matching Traffic** area, modify the following fields:

Field	Description
<b>Allow Filters</b> drop-down list	Choose a filter to use to allow matching traffic. <b>Note</b> You cannot choose the same filter for Allow Filters that you choose for Drop Filters.
<b>Set VLAN</b> field	The VLAN ID that you want to set for the connection. <b>Note</b> This functionality is available only in Openflow mode.
<b>Select Destination Port Group(s)</b> field	Select <b>Port Group</b> option and then select the destination port group from the drop-down list for the new connection.
<b>Strip VLAN at delivery port</b> check box	Check this box to strip the VLAN tag from the packet before it reaches the delivery port. <b>Note</b> The Strip VLAN at delivery port action is only valid for connections with a single edge port and one or more delivery devices for a single, separate node. This functionality is available only in Openflow mode.



Field	Description
Destination Devices list	The monitoring devices that you want to associate with the filter. You can choose one or more devices by checking the boxes next to their names.

**Step 4** In the **Drop Matching Traffic** area, complete the following fields:

Field	Description
Drop Filters	Choose the default filter <b>Default-Match-all</b> or use other filters to drop the matching traffic.  <b>Note</b> You cannot choose the same filter for Drop Filters that you choose for Allow Filters.

**Step 5** In the **Source Ports (Optional)** area, complete the following fields:

Field	Description
Select Source Node drop-down list	Choose the source node that you want to assign.  <b>Note</b> If you do not choose a source node, the any-to-multipoint loop-free forwarding path option is used, and traffic from all nondelivery ports is evaluated against the filter.  <b>Note</b> When setting up a new redirection, you can see the number of flows that are part of each input port. When you click the port number, the flow details are displayed.
Select Source Port drop-down list	Choose the port on the source node that you want to assign.  <b>Note</b> Only edge ports can be used as source ports.  <b>Note</b> If you do not select a source port while adding a new connection, the following warning message is displayed: No source port is selected. Connection will be setup from all configured Edge-SPAN and Edge-TAP ports. Click OK to continue with the connection installation/creation. It ensures that you do not install any to multi point connection and disrupt any existing traffic. Click Cancel to take you to the connection setup page.

Field	Description
In the <b>Source Ports (Optional)</b> area, select <b>Port Group</b> instead of <b>Source Ports</b> .	Select a port group from <b>Select Port Group</b> drop-down list. If you do not have any port groups configured, click + <b>Port Group</b> to add a port group.

**Note** Similar to the number of Edge-TAP or Edge-SPAN ports are displayed on top of each switch in the topology, the number of forwarding rules that a particular monitoring tool is part of are displayed when you hover the mouse over a switch. A popup table displays the rule (connection) names within which the monitoring tool is being used.

**Note** In Cisco Nexus Data Broker, Release 3.2.0, you can also select a port group in which case the individual ports cannot be selected.

**Step 6** Do one of the following:

- Click **Save Connection** to save the connection, but not to install it until later.
- Click **Install Connection** to save the connection and install it at the same time.
- Click **Dry Run** to estimate the amount of traffic that will be generated on the new connection.
- Click **Close** to exit the connection without saving it.

**Note** Starting with Cisco NDB Release 3.6, you can estimate the amount of traffic generated for a new connection using the Dry Run feature. This feature samples the traffic for 30 seconds for the new connection and estimates the approximate traffic generated for the connection. You can use the Dry Run feature before adding a new connection.

You can manage the Dry Run feature using the `mm.dryrun.timer` parameter in the `config.ini` file. To enable the Dry Run feature, set the `mm.dryrun.timer` parameter to a value greater than zero. If the `mm.dryrun.timer` parameter is set to zero, the Dry Run feature is disabled.

The Dry Run feature shows the topology for the new connection with information about the estimated traffic. The feature samples the traffic for few (`mm.dryrun.timer` value in `config.ini` file) seconds for the new connection and estimates the approximate traffic generated for the connection. You can use the Dry Run feature before adding a new connection.

The following fields are displayed on the **Connection Setup** screen:

- Name
- Allow Filters
- Drop Filters
- Source Ports/Port Groups
- Devices
- Priority
- Last Modified By
- Description
- Status

- Action



**Note** Beginning with Cisco Nexus Data Broker, Release 3.2, if you have added two or more interfaces (source ports) using the Connections tab, two interfaces (source ports) are displayed by default. If you have more than two interfaces (source ports) in the **Connections** tab, you can expand or collapse the source ports by using **more...** or **less...** options that are provided in the GUI.

Click **i Search Connections** tab in the Connections screen to search for the connections using the keywords, **Success**, **Installing**, **Creating**, **Partial**, and **Failed**.



**Note** If a remote monitoring tool is selected, same sources or remote monitoring tool cannot be shared across connections. This condition applies to ISL links also.

## Connection with AutoPriority

Starting with Cisco NDB Release 3.4, you can configure filters with overlapping IP address. The traffic from the interfaces with overlapping IP addresses is forwarded to all the monitors configured for the IP address. This feature is supported for the switches running NX-API, OpenFlow, and AUX mode.

You can also configure rules with common IP addresses. You can configure IP address and port overlapping in the same filter.

Beginning with Cisco Nexus Data Broker, Release 3.3, you can now add a new connection with AutoPriority. This functionality provides flexibility to map filters to multiple destination devices in a connection. The priority of a connection with Auto-Priority is set to the value configured in **config.ini** file. You can configure the *connection.autopriority.priorityValue* attribute in the **config.ini** file with a priority value to be used for all the new connections with auto-priority. The connection information lists the allowed filters along with the destination devices.

## Restrictions and Usage Guidelines

Follow these restrictions and usage guidelines for creating a connection with auto-priority:

- To add a new connection with AutoPriority across devices (with multiple hops), the QinQ VLAN configuration is required.
- You can configure only one connection with Auto-Priority mode for each source port/port group.

## Adding a New Connection With Auto-Priority

To add a new connection with Auto-Priority, complete these steps:

**Before you begin**

Ensure that you have configured the monitoring device, destination device, and filters before adding a new connection.

**Procedure**

- Step 1** Log into the NDB application.
- Step 2** Navigate to **Configuration** -> **Connection**, and click **New Connection with AutoPriority** to add a new connection. The **New Connection with AutoPriority** window is displayed.
- Step 3** In the **New Connection with AutoPriority** window, complete the following fields:

<b>Name</b>	<b>Description</b>
<b>Connection Name</b> field	The name of the connection.
<b>Description</b> field	Short description of the connection.
<b>Devices/Port Groups</b> drop-down list	The name of the destination device or the destination port group.  Select <b>Devices</b> and then select a destination device from the Destination Device drop-down list and select corresponding filter from the Allow Filter drop-down list. You can add multiple destination devices with filters for a connection with AutoPriority mode.  Select <b>Port Group</b> and then select destination port group.
<b>Allow Filters</b> list	Filter to apply to the destination device.
<b>Set VLAN</b> field	VLAN ID range to override the incoming tagged VLAN traffic.
(Optional) <b>Source Ports</b> Section	Select Source Ports or Port Group button to create or modify a connection.
<b>Select Source Node</b> drop-down list	Source Node Id.
<b>Select Source port</b> drop-down list	Source Node port number.
<b>Select Port Group</b>	If Port Group Button is enabled, select a Port Group from the drop down for creating or modifying a connection.

# Adding Redirections



**Note** The redirection setup feature is supported on Cisco Nexus 3000 Series and Cisco Nexus 9300 switches with Release 7.x.

Cisco Nexus Data Broker lets you configure redirection policies that match specific traffic, redirecting it through multiple security tools before it enters or exits your data center using redirection.

## Before you begin

- Add a filter to be assigned to the redirection.
- Configure a monitoring device (optional).
- Configure an edge port or multiple edge ports (optional).
- The production ingress port, the production egress port, and the service node should be on the same redirection switch.

## Procedure

**Step 1** On the **Redirections** tab, click + **Redirection**. The **Add Redirection** window is displayed.

**Step 2** In the **Add Redirection** window, you can add the **Redirection Name** and the **Priority** of the redirection in the **Redirection Details** area:

Field	Description
<b>Redirection Name</b>	The name of the redirection. <b>Note</b> The name of the redirection cannot be changed once you have saved it.
<b>Description</b>	Enter the description when creating a new redirection.

Field	Description
<b>Set Auto Priority</b> checkbox	<p>Check this option to enable the auto-priority for redirection, The priority of the redirection is set based on the existing redirections that are installed on the selected ingress ports.</p> <p>If auto-priority is enabled, redirection has a default priority of 10000. Next redirection with auto-priority enabled will have the priority value as the last priority minus 1.</p> <p>Without the auto-priority feature, the default value is 100. It can be changed in the range of &lt;2-10000&gt;.</p> <p>Priority value 1 is reserved for the backup bypass flows.</p> <p><b>Note</b> The priority of the redirection should not be configured as 1. Also, if the last priority is configured as 2, you cannot clone the redirection with auto-priority enabled. You have to manually clone the redirection.</p>
<b>Priority</b>	The priority that you want to set for the redirection. The valid range of the values is 0–10000. The default is 100.
<b>Automatic Fail-safe</b> checkbox	Check this option to enable the fail-safe feature of redirection. When you enable this feature, the direct flow from the production ingress port and the egress port is created that matches all ethertype traffic of low priority.

**Step 3** In the **Matching Traffic** area, modify the following fields:

Field	Description
<b>Filters</b> drop-down list	<p>Choose a filter to use to allow matching traffic.</p> <p><b>Note</b> You cannot choose the same redirection for the filter.</p>

**Step 4** In the **Redirection Switch** area, modify the following fields:

Field	Description
<b>Select Redirection Switch</b> drop-down list	Select the redirection switch that you want to assign.

**Note** You can have only one ingress port and one egress port per one redirection switch.

**Step 5** In the **Service Nodes (OPTIONAL)** area, complete the following fields:

Field	Description
<b>Select Service Node</b> drop-down list	Select the redirection service node that you want to assign and click <b>Add Service Node</b> .

**Note** If you want to add multiple service nodes, you should add them in an order in which you want the packets to travel.

Starting with Cisco Nexus Data Broker, Release 3.2.0, the order of the service nodes is maintained. For example, if you have added the service nodes s1, s2, and s3 to redirection in an order. The service nodes become operationally down and therefore, they get removed from the redirection. Once the nodes become operationally up, they are added to the redirection in the same order.

**Step 6** Select the **Reverse ServiceNode Direction** option to enable reverse direction on the service node.

When you enable this option and click **Submit**, the ingress and egress ports of the service node are swapped and reverse redirection is enabled on the service node. The option is also displayed as enabled in the **Redirections** tab.

**Step 7** In the **Production Ports** area, complete the following fields:

Field	Description
Select Production Ingress Port drop-down list	<p>Select the production ingress port that you want to assign.</p> <p><b>Note</b> You can select only one ingress port. Multiple ingress ports are not allowed. You cannot use the same ports as the ingress and the egress ports.</p> <p><b>Note</b> When setting up a new redirection, you can see the number of flows that are part of each input port. When you click the port number, the flow details are displayed.</p>
Select Production Egress Port drop-down list	Select the production egress port that you want to assign.

**Step 8** In the **Delievery Devices to copy traffic (OPTIONAL)** area, complete the following fields:

Field	Description
Select Device drop-down list	<p>Select a device, for example, a switch from the drop-down list, that you want to assign and click <b>Add Device</b>.</p> <p><b>Note</b> You can select multiple delivery devices for the redirection.</p>

Field	Description
Select <b>Monitor Traffic</b> drop-down list	<p>When creating inline redirection with copy, the monitoring port receives one flow from the production ingress port and another from the egress port of service node.</p> <p>Starting with Cisco Nexus Data broker Release 3.2, a filtering mechanism is added in the GUI to filter out the traffic. Use the drop down list to select the traffic to copy device in redirection.</p> <p>The following options are displayed in the drop-down list:</p> <ul style="list-style-type: none"> <li>• Production Ingress-- Flow from the production ingress port</li> <li>• Production Egress-- Flow from the egress port of the service node</li> <li>• Both-- Flow from both the ports (the ingress and the egress ports)</li> </ul>

**Step 9** Do one of the following:

- Click **Save Redirection** to save the redirection, but not to install it until later.
- Click **Install Redirection** to save the redirection and install it at the same time.
- Click **Close** to exit the redirection without saving it.

**Step 10** When you click **Install Redirection** to save the redirection and install it at the same time, the redirection path on the redirection switch is displayed on the production ingress ports, service nodes, and the production egress ports.

**Step 11** Click **Flow Statistics** to view the flow statistics for the redirection switch.

The following fields provide information on the flow statistics:

- In Port field—The Input port(s) from which the traffic is matched. An asterisk ("\*") indicates any input port.
- DL Src field—The source MAC address to be matched for the incoming traffic. An asterisk ("\*") indicates any source MAC address.
- DL Dst field—The destination MAC address to be matched for the incoming traffic. An asterisk ("\*") indicates any destination MAC address.
- DL Type field—The Ethertype to be matched for the incoming traffic. For example, "IPv4" or "IPv6" is used for all IP traffic types.
- DL VLAN field—The VLAN ID to be matched for the incoming traffic. An asterisk ("\*") indicates any VLAN ID.
- VLAN PCP field—The VLAN priority to be matched for the incoming traffic. An asterisk ("\*") is almost always displayed in this field.



- NW Src field—The IPv4 or IPv6 source address for the incoming traffic. An asterisk ("\*") indicates any source address based on IPv4 or IPv6 Ethertypes.
- NW Dst field—The IPv4 or IPv6 destination address for the incoming traffic. An asterisk ("\*") indicates any destination address based on IPv4 or IPv6 Ethertypes.
- NW Proto field—The network protocol to be matched for the incoming traffic. For example, "6" indicates the TCP protocol.
- TP Src field—The source port associated with the network protocol to be matched for the incoming traffic. An asterisk ("\*") indicates any port value.
- TP Dst field—The destination port associated with the network protocol to be matched for the incoming traffic. An asterisk ("\*") indicates any port value.
- Actions field—The output action to be performed for the traffic matching the criteria specified, for example, "OUTPUT = OF|2".
- Byte Count field—The aggregate traffic volume shown in bytes that match the specified flow connection.
- Packet Count field—The aggregate traffic volume shown in packets that match the specified flow connection.
- Duration Seconds field—The amount of time, in milliseconds, that the specific flow connection has been installed in the switch.
- Idle Timeout field—The amount of time, in milliseconds, that the flow can be idle before it is removed from the flow table.
- Priority field—The priority assigned to the flow. The flows with higher priority numbers take precedence.

**Step 12** Click **Close** to close the flow statistics display window.

---

## Viewing Statistics

View the flow and port statistics for the switches on the Statistics tab.



**Note** When you select a switch on the statistics page, the **Auto Refresh** tab for the switch is ON by default. Click **Auto Refresh: Off** to disable auto refresh on the Statistics tab. The screen is refreshed every 30 seconds and the updated statistics for the switch are displayed on the screen.

---

### Procedure

---

**Step 1** Navigate to the **Statistics** tab under **Configuration** and click a node from the drop-down list to check and view the flow and port statistics of that node.

You can also navigate to the statistics of another switch by selecting the switch in the drop down box.

You can view the flow statistics, for example:

- Flow Name
- In Port
- DL Source
- DL Destination
- DL Type
- DL VLAN
- VLAN PCP
- NW Source
- NW Destination
- NW Proto
- TP Source
- TP Destination
- AP HttpMd
- AP TcpOptLn
- Actions
- Byte Count
- Packet Count
- Duration Seconds
- Idle Timeout
- Priority

**Step 2** Click the **Ports** tab to check the ports statistics.

You can view the ports statistics as displayed in the following fields.

**Note** If you are programming the switches with OpenFlow, when you navigate to the **Statistics** tab, select a switch, and select **Ports** tab, the statistics gathered from the switches for the **Rx Frame Errs** and **Collisions** are not supported. The value of -1 is displayed rather than N/A because the variable needs to be an integer.

- Port Name
- Rx Packets
- Tx Packets
- Rx Bytes
- Tx Bytes
- Rx Rate (kbps)

- Tx rate (kbps)
  - Rx Drops
  - Tx Drops
  - Rx Errors
  - Tx Errors
  - Rx Frame Errors
  - Rx Overrun Errors
  - Rx CRC Errors
  - Collisions
- 

## Viewing Connection Port Statistics

Starting with Cisco NDB Release 3.4, port statistics are shown along with the connection path information in the NDB GUI. This feature is supported for Nexus 9K and Nexus 3K Series switches based on NX-API, OpenFlow, and NX-AUX mode.

To view the port statistics for a connection, complete the following steps:

### Procedure

---

- Step 1** Navigate to **CONFIGURATION** -> **Connections** .
  - Step 2** On the **Connection** page, click a connection name for which you want to view the port statistics.
  - Step 3** Click **Port Statistics** to open the **Flow Statistics** page.
  - Step 4** Click **Port** tab to view the port statistics for the selected connection.
- 

## Deleting Flow and Port Statistics

Starting with Cisco NDB release 3.4, you can now clear port and flow statistics using the NDB GUI. You can either clear all the port related statistics for a switch or clear statistics for a specific port on the switch. For This feature is currently available only for NXAPI based Nexus 9K and Nexus 3K switches.

To clear flow statistics, complete the following steps:

### Procedure

---

- Step 1** Navigate to the **CONFIGURATION** → **Statistics** and click the **Flows** tab to clear flow statistic. Click **Delete ALL** to clear all the flow statistics such as byte count and packet count for the switch.

- Step 2** Click the **Ports** tab to clear port statistics.
- Select a port and click **Delete** to delete statistics for the selected port.
  - Click **Delete All** to clear statistics for all the ports (interfaces) on the switch.
- 

## Purging Device Configuration

Starting with Cisco NDB release 3.6, you can now remove and purge all the configuration information (such as connection and redirection) associated with a device that has been removed from the NDB.

To remove device configuration, complete the following steps:

### Procedure

---

- Step 1** Navigate to the **ADMINISTRATION > Devices > Purge Devices**.
- Step 2** Select the devices for which you want to remove all the configuration information and click **Purge Devices**. All the configurations associated with the removed device will be deleted from NDB database.
- 

## Exporting and Importing NDB Configuration

Starting with Cisco Nexus Data Broker, Release 3.4, you can now export and import the device configuration in JSON file format. The configuration file includes information about connected as well as disconnected devices with configuration information such as filters, ports, connections, and redirections.

### Exporting NDB Configuration

Complete the following steps to export a configuration from NDB:

#### Procedure

---

- Step 1** Navigate to **Administration -> System -> Backup/Restore -> Node**, and click **Export** tab.
- Step 2** Click **Refresh** to list the latest list of NDB devices.
- Step 3** Select a device for exporting the configuration from the **Configuration** Pane.
- Step 4** (Optional) Select **Include Connections** check box to include connection information such as filters and connections.
- Step 5** (Optional) Select **Include Redirections** check box to include connection information such as filters, service nodes, and redirections.
- Step 6** Click **Generate new Configuration** to create and save the configuration in JSON format.
-

## Importing NDB Configuration

Complete the following steps to import a configuration into NDB:

### Procedure

- 
- Step 1** Navigate to **Administration -> System -> Backup/Restore-> Node**, and click **Import** tab.
  - Step 2** Click **Select Configuration**, the **File Upload** dialog box appears.
  - Step 3** Select a JSON file and click **Open**. The selected configuration appears in the **Import** section.
  - Step 4** Click **Edit** to enter device password (applicable only for NXAPI and NX-AUX mode).
  - Step 5** (Optional) Select **Include Connections** check box to include connection information such as filters and connections.
  - Step 6** (Optional) Select **Include Redirections** check box to include connection information such as filters, service node, and redirections.
  - Step 7** Click **Apply** to apply the configuration to NDB. The **Compatibility Matrix** page appears.
  - Step 8** Select **Accept and continue** to import the configuration.
- 

## Managing Sampled Flow Configuration

Starting with Cisco NDB Release 3.4, you can now manage the Sampled Flow (sFlow) on NDB switches that are based on NX-API. This feature is currently not available for OpenFlow and NX-AUX based switches. sFlow allows you to monitor real-time traffic in data networks that contain switches and routers. It uses the sampling mechanism in the sFlow agent software on switches and routers to monitor traffic and to forward the sample data to the central data collector.

To enable sFlow on a port, complete the following steps:

### Procedure

- 
- Step 1** Log into the NDB GUI.
  - Step 2** Navigate to **CONFIGURATION -> Port Definition** tab.
  - Step 3** Click **Configure Node** to open the **Node Configuration** pane. The **Node Configuration** window is displayed.
  - Step 4** Click **Configure sFlow** to open the **Configure sFlow** pane.
  - Step 5** Select **Enable sFlow** from the **Enable/Disable sFlow** drop-down list to open the **Configure sFlow** pane.
  - Step 6** In the **Configure sFlow** pane, enter the following details and click Submit.

Field	Description
Agent IP address	sFlow agent IP address.
Select a VRF	VRF to use to reach the SFlow collector IP address.
Collector IP address	SFlow collector address.

Field	Description
Collector UDP	SFlow collector UDP.
Counter Poll Interval	SFlow counter poll interval.
Max Datagram Size	Maximum sampling data size.
Sampling rate	Data sampling rate.
Select Data Source(s)	SFlow datasource interface (Edge-ports)

**Note** Use **Add to Group** option to add the configured port to a Group of ports.

**Note** In Sflow pane, the **Select Data Source** field displays only those ports that are configured either as a Edge-SPAN or as a Edge-TAP .

To verify SFLow configuration on a switch, use the show sflow command:

```
RU-29-2003(config)# show sflow
sflow sampling-rate : 4096
sflow max-sampled-size : 128
sflow counter-poll-interval : 20
sflow max-datagram-size : 1400
sflow collector-ip : 0.0.0.0 , vrf : default
sflow collector-port : 6343
sflow agent-ip : 10.16.206.122
sflow data-source interface Ethernet1/1
```

## Configuring Symmetric/Non-Symmetric Load Balancing and MPLS Tag Stripping

From the Cisco Nexus Data Broker GUI and the REST API interfaces, you can now configure symmetric load balancing and enable MPLS tag stripping on the Cisco Nexus 3000 Series and Cisco Nexus 9000 Series switches using NX-API as the configuration mode.

### Before you begin

Add device to Cisco Nexus Data Broker using NX-API.

### Procedure

- Step 1** In the topology diagram, click the node for which you wish to configure MPLS tag stripping.
- Step 2** In the **Port Configuration** window, click **Configure Node**. The **Node Configuration** window is displayed.
- Step 3** In the **Load Balancing Type Configuration** drop-down list, select the type and corresponding **Hashing Option**.
- Step 4** In the **MPLS Strip Configuration** drop-down list, choose one of the following:

- Enable MPLS Strip.
- Disable MPLS Strip.

**Step 5** When you select **Enable MPLS Strip** option, the **Label Age** field is displayed. In the field, enter a value for the MPLS strip label age. The range for MPLS strip label age configuration is 61-31622400.

**Step 6** Click **Submit**.

## Symmetric/Non-Symmetric Load Balancing Options

The following table lists the symmetric and non-symmetric load balancing options:

*Table 7: Symmetric / Non-Symmetric Load Balancing Port Channel Support*

Configuration type	Hashing Configuration	Platforms	Options
Symmetric	SOURCE_DESTINATION	N9K*, N3K-C3164XXX, N3K-C32XXX	IP, IP-GRE, IP-L4PORT, IP-L4PORT-VLAN, IP-VLAN, L4PORT, MAC
		Rest	IP, IP-GRE, PORT, MAC, IP-ONLY, PORT-ONLY
Non-symmetric	SOURCE DESTINATION	N9K*, N3K-C3164XXX, N3K-C32XXX	IP, IP-GRE, IP-L4PORT, IP-L4PORT-VLAN, IP-VLAN, L4PORT, MAC
		Rest	IP, IP-GRE, PORT, MAC

## Configuring PTP Using NDB

A PTP system can consist of a combination of PTP and non-PTP devices. PTP devices include ordinary clocks, boundary clocks, and transparent clocks. Non-PTP devices include ordinary network switches, routers, and other infrastructure devices.

PTP is a distributed protocol that specifies how real-time PTP clocks in the system synchronize with each other. These clocks are organized into a master-member synchronization hierarchy with the grandmaster clock, the clock at the top of the hierarchy, determining the reference time for the entire system. Synchronization is achieved by exchanging PTP timing messages, with the members using the timing information to adjust their clocks to the time of their master in the hierarchy. PTP operates within a logical scope called a PTP domain.

Starting with Cisco NDB Release 3.4, you can configure PTP Timestamping feature using the NDB GUI. PTP is a time synchronization protocol for nodes distributed across a network. Its hardware timestamp feature provides greater accuracy than other time synchronization protocols such as Network Time Protocol (NTP).



**Note** Starting with Cisco NDB release, 3.8, you can configure PTP on Nexus 3548 switches.



---

**Note** For Cisco NDB 3.4 release and later, PTP Timestamp Tagging feature is supported on the Cisco Nexus 93XXX-EX and 92XX Series switches.

---



---

**Note** Starting with Cisco NDB release 3.8, Timestamp Tagging is supported on Cisco Nexus 9500 Series switches.

---



---

**Note** You need to enable PTP for all the devices in the network to ensure PTP clock time synchronization.

---



---

**Note** After PTP is configured, the default PTP configuration is synchronized with all the ISL ports of the corresponding device.

---

To configure PTP using NDB GUI, complete these steps:

### Procedure

---

- Step 1** Log into Cisco NDB GUI.
  - Step 2** Navigate to **CONFIGURATION** -> **Port Definition** tab.
  - Step 3** Click **Configure Node** to open the **Node Configuration** pane.
  - Step 4** Click **Configure PTP** to open the **Configure PTP** pane.
  - Step 5** Select **Enable PTP** from the **Enable/Disable PTP** drop-down list.
  - Step 6** Enter the PTP source IP address in the **Source IP Address** text field.
  - Step 7** Select the interfaces on which you want to enable PTP from the **Select Port(s)** list.
  - Step 8** Click **Submit** to enable PTP on the selected interfaces.
- 

## Configuring Packet Truncation

Starting with Cisco NDB Release 3.5, you can configure packet truncation on egress ports for Cisco Nexus 9300 FX and EX series switches. Packet truncation involves discarding bytes from a packet starting at a specified byte position. All the data after the specified byte position is discarded. Packet truncation is required when the main information of interest is in the header of a packet or in the initial part of the packet.



---

**Note** Starting with Cisco NDB release 3.8, you can now configure packet truncation on Cisco Nexus 9500 Series switches.

---





---

**Note** You can configure a maximum of four monitoring devices with packet truncation on a switch.

---

To configure packet truncation on a device, you need to:

1. [Configuring a Packet Truncation Interface](#)
2. [Defining a Monitoring Device with Packet Truncation Interface](#)

## Configuring a Packet Truncation Interface

To configure a packet truncation interface, complete these steps:

### Procedure

---

- Step 1** Log into NDB.
- Step 2** Navigate to the **CONFIGURATION > Port Definition** and select the switch for which you plan to configure packet truncation.
- Step 3** Click **PORT CONFIGURATION** tab.
- Step 4** Click **Configure** for the interface selected for configuration.
- Step 5** In the **Configure Ports** pane, click **Select a port type** and then click **Packet Truncation Port**.
- Step 6** (Optional) Enter description for the port in the **Port Description** text field.
- Step 7** Click **Submit** to create a packet truncation port.

By default a packet truncation port is blocked for ingress traffic.

**Note** Ensure that the status of the packet truncation port is Administratively Up (green icon) and that the other end of the link is not connected to the same NDB switch. To change the port Layer 2 status to Up, you need to connect to another switch or create a loopback using a third party loopback fiber optic.

---

### What to do next

After the packet truncation port is created, you need to create a monitoring device with the packet truncation port. For more information, see [Defining a Monitoring Device with Packet Truncation Interface](#) section.

## Defining a Monitoring Device with Packet Truncation Interface

Complete the following steps to define a monitoring device with a packet truncation interface:

### Procedure

---

- Step 1** Navigate to the **CONFIGURATION > Port Definition** and select the switch for which you plan to configure packet truncation.

- Step 2** Click **PORT CONFIGURATION** tab.
- Step 3** Click **Configure** for the interface selected for configuration.
- Step 4** In the **Configure Ports** pane, click **Add Monitoring Device**.
- Step 5** In the **Monitoring Device** window, complete the following fields:

Name	Description
<b>Monitoring Device Name</b>	Name of the monitoring device.
<b>Select Switch Name</b>	Name of the switch to add the monitoring device to.
<b>Select Port</b>	Packet truncation port you configured.
<b>Port Description</b>	Description of the port.

- Step 6** Select **Packet Truncation**.
- Step 7** Enter maximum packet size in the **MTU Size** text field. The MTU size can be between 320 and 1518 bytes.
- Step 8** From the **Select Packet Truncation Port** drop-down list, select the packet truncation port you created on the same switch.
- Step 9** (Optional) Select device icon for the monitoring device.
- Step 10** Click **Submit** to create the monitoring device.

---

### What to do next

Create a new connection using the monitoring device to implement the packet truncation feature. For more information, see Adding Connections.



## CHAPTER 8

# Integrating Cisco Nexus Data Broker With Cisco ACI

---

This chapter contains the following sections:

- [SPAN Management, on page 135](#)
- [Viewing the SPAN Management Tab, on page 137](#)
- [Viewing the SPAN Destination Tab, on page 138](#)
- [Adding SPAN Destination, on page 139](#)
- [Creating Copy Devices Using Copy Sessions \(BETA\), on page 140](#)
- [Adding SPAN Sessions, on page 141](#)

## SPAN Management

Switch port Analyzer (SPAN) is an efficient and high performance traffic monitoring system. It duplicates the network traffic and routes the packets to an the analyzer for monitoring. SPAN is used for troubleshooting connectivity issues and calculating network utilization, and performance monitoring. You can add, edit, remove, and rediscover a device to SPAN using NDB.

## Adding a Device to SPAN

Complete these steps to add a device to Switch Port Analyzer (SPAN).



---

**Note** Starting with Cisco NDB release 3.8, you can add multiple APIC in NDB.

---

### Procedure

---

- Step 1** Navigate to **ADMINISTRATION > Devices > SPAN MANAGEMENT** tab.
- Step 2** Click **Add Device**, the **Connect to Devices** dialog box appears.
- Step 3** Select **ACI** option to add a new APIC device to SPAN. To add a new NX-OS based device, select the **NXOS** option.
- Step 4** In the **Connect to Devices** dialog box, enter the following details:

Table 8: New Device Details

Field	Description
APIC IP Addresses (Applicable for adding APIC device)	
<b>APIC IP Address / HostName (Primary)</b>	Primary APIC IP address or hostname.
<b>APIC IP Address (Secondary)</b>	Secondary (backup) APIC IP address or hostname. This field is optional.
<b>APIC IP Address (Tertiary)</b>	Tertiary (backup) APIC IP address or hostname. This field is optional.
NXOS Device Details (Applicable for adding NX-OS based device)	
<b>Address</b>	IP address of the NX-OS device.
<b>Port</b>	The device communication port.
<b>Username</b>	Username for authenticating the device.
<b>Password</b>	Password for authenticating the device.

- Step 5** Click **Connect** to add the new device to SPAN. The new device is listed under the **SPAN MANAGEMENT** tab.

## Editing a SPAN Device

Complete these steps to edit details of a SPAN device:

### Procedure

- Step 1** Navigate to **ADMINISTRATION > Devices > SPAN MANAGEMENT** tab .
- Step 2** Click **Edit** for a device to edit device details. The **Edit Device** dialog box appears. You can update the following details for a device:

Field	Description
<b>Port</b>	Specify new port number for communication. This field is available only for NX-API devices.
<b>User Name</b>	Specify new user name for the device authentication.
<b>Password</b>	Specify new password for the device authentication.

- Step 3** Click **Edit Device** to apply the updated information and close the **Edit Device** dialog box.

## Rediscovering a SPAN Device

You can rediscover a device after updating device details using the Rediscover Devices feature. To rediscover a device, complete these steps:

### Procedure

---

- Step 1** Navigate to **ADMINISTRATION > Devices > SPAN MANAGEMENT** tab.
  - Step 2** Select a device to rediscover from the list of devices under the **SPAN MANAGEMENT** tab.
  - Step 3** Click **Rediscover Devices**, the **Rediscover Devices** dialog box appears.
  - Step 4** Verify the selected device for rediscovering and click **Rediscover Device**.
  - Step 5** Click **Yes** in the dialog box to complete the rediscovery process.
- 

## Removing a Device from SPAN

You can remove a device from SPAN while retaining its configuration or you can remove a device from SPAN and delete all the corresponding device configuration from a SPAN. Complete these steps to remove a device from SPAN.

### Procedure

---

- Step 1** Navigate to **ADMINISTRATION > Devices > SPAN MANAGEMENT** tab.
- Step 2** From the list of devices under the **SPAN MANAGEMENT** tab, select the device(s) to remove.
- Step 3** Click **Remove Devices**. The **Remove Devices** dialog box appears.
- Step 4** Verify the selected device(s) in the **Remove Devices** dialog box and click either of the two options:
  - **Remove Device**: Use this option to remove the device from SPAN while retaining the device configuration.
  - **Purge & Remove Device**: Use this option to remove the device and device configuration from SPAN.

The device is removed from the **SPAN MANAGEMENT** tab.

---

## Viewing the SPAN Management Tab

The **SPAN Management** tab is displayed on the **Devices** screen under the **Administration** tab in the GUI.



**Note** Starting with Cisco Nexus Data Broker, release 3.8, you can configure multiple APIC devices on NDB.

---

On the **SPAN Management** tab, click + **Add Device**. The **Connect to Device** window is displayed. Complete the following steps to connect to the device:

**Before you begin**

For APIC and production switches, the centralized deployment of Cisco Nexus Data Broker is mandatory.

**Procedure**

- 
- Step 1** Choose **ACI** device to add an APIC device.
- Step 2** In the **APIC IP Addresses** panel, add the **APIC IP Address (Primary)**, **APIC IP address (Secondary)**, and **APIC IP address (Tertiary)**.
- Step 3** In the **User Details** panel, add **Username** and **Password**.
- Step 4** After an ACI device has been added, the ACI radio button is disabled. Then you can add a NXOS production switch. Click **NXOS** in the first step to add a NXOS production switch.
- The NX-API feature has to be enabled for the NXOS production switch to be added. To add a NXOS production switch in the **SPAN Management** tab, one NX-API device should already exist. This is a pre-requisite.
- Step 5** Click **Connect**.
- 

The NXOS production switch is displayed with the **Type** as **PS** in the **SPAN Management** tab. The **APIC IP Address (Primary)**, **APIC IP address (Secondary)**, and **APIC IP address (Tertiary)** do not apply to the NXOS production switch. Therefore, those fields are blank. You can also edit the credentials of the NXOS production switch. Once the production switch is added, it is displayed in the Configuration tab in green. In the Port Configuration window, you can configure SPAN Destination in the production Nexus switches that are NX-API enabled.

## Viewing the SPAN Destination Tab

When you click **Port Definition** tab in the GUI, the **Port Definition** screen is displayed. Select the switch from the drop-down list to configure the ports.

On the **Port Definition** screen, the following two tabs are displayed:

- Port Configuration
- SPAN Destination

On the **SPAN Destination** tab, the following details are displayed:

- SPAN Destination Name
- SPAN Destinations
- Node Connector
- Monitor Port Type
- Description

# Adding SPAN Destination

When you configure a port as an edge SPAN port and the port is connected to the API side, you can select the APIC device, pod, node, and port from the ACI side and set the port as SPAN destination. SPAN destination can now be configured on the Cisco Nexus 9000 or Cisco Nexus 3000 Series production switches.



**Note** For APIC SPAN destination, when you configure a port as an Edge SPAN port and the port is connected to the API side, you can select the pod, the node, and the port from the ACI side and set the port as SPAN destination. For production switch SPAN destination, when you configure a port as an Edge SPAN port and the port is connected to the production switch side, you can select the node and the port from the production switch side and set the port as SPAN destination.

You can add SPAN destination only after either an APIC or the production switch has been successfully added to the network.

## Procedure

- Step 1** Select the switch for which you want to configure the port details on the **Port Configuration** screen.
- Step 2** Click **Configure** under **Action**.  
The **Configure Ports** window is displayed.
- Step 3** In the **Configure Ports** window, configure the port type from the **Select a port type** drop-down list by selecting one of the following options:
  - **Add Monitoring Device**
  - **Edge Port-SPAN**
  - **Edge Port-TAP**
  - **Production Port**

**Monitoring Device**—Creates a monitoring device for capturing traffic and configures the corresponding delivery port.

**Edge Port-SPAN**—Creates an edge port for incoming traffic connected to an upstream switch that is configured as a SPAN destination.

**Edge Port-TAP**—Creates an edge port for incoming traffic connected to a physical TAP port.

**Production Port**—Creates a production port for the ingress and egress traffic.

When you select the port type, the title of the window changes to **Manage Configure Ports**.
- Step 4** In **Manage Configure Ports** window, the details of the selected node are displayed.
- Step 5** In the **Destination** panel, if the APIC device is added, it is listed in the drop-down list. Select the **Node Type** as **APIC** from the drop-down list.  
The **SPAN Destination** and **Copy Device** tabs are displayed.
- Step 6** When you click the **SPAN Destination** tab, the **Select SPAN Destination** window is displayed. From the Select APIC Node select an APIC node.

- Step 7** From the **Select Node** drop-down list, select an APIC device.
- Step 8** Select corresponding leaf switch, node and port from the **Select Pod** drop-down list, **Select Node** drop-down list, and **Select Port** drop-down list to configure the SPAN Destination.
- Step 9** Click **Apply**.  
The port is now configured as SPAN destination part and it is displayed on the Port Definition screen.
- Step 10** (Optional) In the **Configure Ports** dialog box, under **Destination** section, specify MTU for the SPAN Destination in the **MTU** text-field. You can configure MTU range between 1 and 9216.
- Step 11** Click **Submit**.

## Creating Copy Devices Using Copy Sessions (BETA)

When you configure a port as an edge-SPAN port, you can create copy devices using Copy Sessions (BETA) functionality.



**Note** You can add SPAN destination and copy devices only after an APIC device has been successfully added to the network.

### Procedure

- Step 1** Select the switch for which you want to configure the port details using the **Port Configuration** screen.
- Step 2** Click **Configure** under **Action**.

The **Configure Ports** window is displayed.

- Step 3** In the **Configure Ports** window, configure the port type from the **Select a port type** drop-down list by selecting one of the following options:

- **Add Monitoring Device**
- **Edge Port-SPAN**
- **Edge Port-TAP**
- **Production Port**

**Monitoring Device**—Creates a monitoring device for capturing traffic and configures the corresponding delivery port.

**Edge Port-SPAN**—Creates an edge port for incoming traffic connected to an upstream switch that is configured as a SPAN destination.

**Edge Port-TAP**—Creates an edge port for incoming traffic connected to a physical TAP port.

**Production Port**—Creates a production port for the ingress and egress traffic.

When you select the port type, the title of the window changes to **Manage Configure Ports**.

- Step 4** In **Manage Configure Ports** window, the details of the selected node are displayed.



- Step 5** In the **Destination** panel, if the APIC device is added, it is listed in the drop-down list. Select the **Node Type** as **APIC** from the drop-down list.
- The **SPAN Destination** and **Copy Device** tabs are displayed. See *Adding SPAN Destination* section for adding SPAN destination.
- Step 6** When you click the **Copy Device** tab in the same window, the **Create Copy Device (BETA)** window is displayed.
- Step 7** In the General panel, enter the name of the device in the **Name** field. The values for the fields, **Device Type** and **Physical Domain** are hard-coded.
- Step 8** In the Device Interface panel, enter the details in the following fields: **Name**, **Pod**, **Node**, and **Port**. The value for the field, **Path Type** is hard-coded.
- Step 9** In the Cluster panel, enter the details in the following fields, **Name** and **VLAN Encap**. The value for the field, **Interface** is hard-coded.
- Step 10** Click **Submit** to save the settings.
- The name and the path of the copy device is displayed in the destination panel.
- Step 11** When you click **Submit** in **Manage Configure Ports** window, the device is displayed in the **Destination** column in the **Port Configuration** screen. When you hover over the device name in the GUI, the name of the **Copy Device** is displayed.
- Step 12** Once the **Copy Device** is added, it is displayed in the **APIC Copy Session (BETA)** screen under the **Copy Device** tab.
- The following fields are displayed under the **Copy Device** tab: **Cluster Name**, **Managed**, **Device Type**, and **Service Type**.
- Step 13** In the **APIC Copy Session (BETA)** screen, the **Service Graph** tab is displayed. When you click **+Add Service Graph**, the **Add Service Graph (BETA)** window is displayed.
- Step 14** Add name for the service graph in the **Name** field.
- Step 15** Select the copy device for the service graph in the **Copy Device** field.
- The copy devices that are created by Cisco Nexus Data Broker are listed in the **Copy Device** field.
- Step 16** Click **Submit** to save the settings.
- Once the service graph is added, it is displayed in the **APIC Copy Session (BETA)** screen under the **Service Graph** tab. The fields that are displayed on the tab are **Name**, **Copy Device**, **Function Nodes**, and **Action**. The parameters that can be edited for the service graph are **Name** and **Copy Device** only. You can click **Remove** under **Action** column in the **APIC Copy Session (BETA)** screen to remove the service graph.
- Note** By default, the copy device and the service graph get created under the common tenant.

---

## Adding SPAN Sessions

On the SPAN Sessions tab, the following fields are displayed:

- SPAN Session
- Filter

- Devices
- SPAN Source
- SPAN Destination

You can add a SPAN session in ACI. Complete the following steps to add a SPAN session.



**Note** Starting with Cisco NDB release 3.8, a new column named, **Status**, is added on the SPAN Session tab that displays the status of each session. The status of a SPAN session depends on Operational status of the session in APIC and status of the connection attached to it (if a connection is attached to the session).



**Note** You can create a maximum of 4 SPAN sessions on a switch.

### Procedure

- Step 1** Navigate to **ADMINISTRATION > Devices > SPAN MANAGEMENT** tab.
- Step 2** Click **Add SPAN Session** to add a SPAN session. The **Add SPAN Session** window is displayed.
- Step 3** In the **Add SPAN Session** window, add a session name in the **SPAN Session Name** field.
- Step 4** Under **SPAN Sources**, select **ACI** as device type from the **Select Device Type** option list.
- Step 5** Select an APIC node from the drop-down list on which the SPAN session is to be configured.
- Step 6** Click **Apply SPAN Source**.
- Step 7** In the **SPAN SOURCES** pane, click + **Apply SPAN Source**. In the pane, click + **Apply Leaf Ports** to add a leaf port to capture the traffic from multiple leaf ports. Or optionally, you can click +**Apply EPG/AAEP** to add an EPG source. Enter the values in the following fields:
- If + **Apply Leaf Ports** is clicked.
    - In the **Apply Leaf Ports** window, select a pod using the drop-down list in the **POD** field.
    - Select a node using the drop-down list in the **Node** field.
    - Select a port using the drop-down list in the **Port** field.
  - Click **Apply Leaf Ports**.
  - In the **SPAN SOURCES** pane, select a direction from the **Incoming**, **Outgoing**, or **Both** options.
 

The selected Span source is displayed in the **Span Source** field.
  - If +**Apply EPG / AAEP** is clicked.
 

**Note** Starting with Cisco NDB Release 3.7, you can now add multiple EPGs in the same SPAN session.
  - To add EPG source, select a tenant from the **Tenant** drop-down list in the **Add EPG** window.
 

**Note**

    - All EPG interfaces work only when all the ports are within the same leaf switch.
    - If an EPG is spread across multiple switches, select the corresponding SPAN destination on all the leaf switches.

- i) Select a profile using the drop-down list in the **Profile** field.
- j) Select EPG associated with the tenant using the **EPG** drop-down list.

The selected **SPAN Source** is displayed.

- k) Select EPG or AAEP member from the **EPG Members** drop-down list.
- l) Click **+Add**.
- m) Click **Apply EPG/AAEP**.

**Note** If the EPG is selected, by default, Cisco Nexus Data Broker listens for the changes in the statically or dynamically configured interfaces of the selected EPG. If there is any change, it is applied to the SPAN session. The web socket connection is not secured with the certificates. To disable the event listening, add **enableWebSocketHandle=false** in the **config.ini** file under **xnc/configuration** folder.

**Note** When new EPG members are added in APIC, if there is no SPAN destination on the leaf switch that matches the newly added EPG member as part of the configured SPAN session, NDB ignores this event and the new EPG member are not shown in NDB.

**Step 8** In the **SPAN Destination** field, select SPAN destination.

If you install ACI SPAN session, it lists the SPAN destination that is created in ACI.

If you install NXOS SPAN session, it lists the SPAN destination that is created in NXOS.

**Note** Ensure that each leaf switch in the SPAN source has at least one corresponding SPAN destination.

**Note** Starting with Cisco NDB Release 3.7, addition of multiple SPAN Destinations in the same SPAN session is supported.

**Step 9** (Optional) Select a connection in the **Select Connections** field.

**Note** Starting with Cisco NDB Release 3.7, attaching a connection to the SPAN session is optional.

**Step 10** (Optional) In the **Action** pane, select a priority for the SPAN session.

**Step 11** (Optional) Select a rule using the drop-down list in the **Rule Filter** field.

**Step 12** (Optional) Select a destination device to which the traffic is sent.

**Step 13** Do one of the following:

- Click **Save SPAN Session** to save the session without installing it on ACI.
- Click **Install SPAN Session** to save and install the session on ACI.

**Note** Starting with Cisco NDB release 3.8, you can install a saved SPAN session on ACI using the **Toggle Install** button. Select the saved SPAN session that you want to install and click **Toggle Install** button to install the session on ACI. You can also uninstall a SPAN session without removing it from NDB using the **Toggle Install** button. The SPAN session is uninstalled from ACI but remains saved on the NDB for future use.

**Step 14** Click **OK**.

As a result, a SPAN session is set up in ACI. It also sets up a connection automatically on the Cisco Nexus Data Broker with the same SPAN session name and this connection redirects the traffic from that source port to the monitoring device.

**Note** Each leaf can have a maximum of 4 SPAN sessions.

You can set up additional SPAN sessions. You can append a new SPAN session to the existing connection. In that case, you can select the new SPAN session in the Add SPAN Session window, use the same connection that is previously created, select new SPAN sources from different leaf ports, select the SPAN destination, and add the SPAN session.

It creates a new session in ACI, but it appends an existing connection to include the new traffic on the Cisco Nexus Data Broker side.

You can edit or clone the existing SPAN sessions. If you want to remove a SPAN session, click the session and click **Remove SPAN Session(s)**. A message box is displayed asking you to confirm, **Remove the following sessions?**, if you want to remove the displayed SPAN session. Click **Remove SPAN Sessions** to confirm. If the SPAN session is using an existing connection, the connection is updated automatically with the changes. If it is the last connection associated with the SPAN session, the connection is deleted.

---



## CHAPTER 9

# Viewing and Adding Flows

---

This chapter contains the following sections:

- [Viewing Flows, on page 145](#)
- [Adding a Flow, on page 145](#)

## Viewing Flows



---

**Note** This functionality is applicable only for OpenFlow mode of deployment.

---

On the **Flows** tab, the following fields are displayed:

- Serial Number
- Status
- Flow Name
- Node

### What to do next

Click + **Flow** to add a flow.

## Adding a Flow

### Procedure

---

- Step 1** Navigate to the **Flows** tab under **Administration**, click + **Flow** to add a flow.
- Step 2** On the **Add Flow Description** window, update the following fields:

Name	Description
Name field	<p>The name that you want to assign to the flow.</p> <p><b>Note</b> You cannot change the name of the flow entry after it is saved.</p>
Select a Node drop-down list	<p>Select a node name for the device.</p> <p><b>Note</b> The node you choose cannot be changed one you save the flow entry.</p>
Input Port drop-down list	<p>Choose the port on the node where traffic enters the flow.</p>
Priority field	<p>The priority that you want to apply to the flow. The default priority is 500. Flows with a higher priority are given precedence over flows with a lower priority.</p> <p><b>Note</b> The priority is considered only when all of the Layer 2, Layer 3, and Layer 4 match fields are equal.</p>
Hard Timeout field	<p>The amount of time in milliseconds for the flow to be installed before it is removed from the flow table.</p>
Idle Timeout field	<p>The amount of time in milliseconds that the flow can be idle before it is removed from the flow table.</p>
Layer 2	
Ethernet Type field	<p>The Ethernet type for the Layer 2 traffic. The Ethernet type for IPv4, in hexadecimal format, is displayed by default. Either accept the default value, or enter one of the following, in hexadecimal format:</p> <ul style="list-style-type: none"> <li>• IPv6</li> <li>• ARP</li> <li>• LLDP</li> </ul>
VLAN Identification Number field	<p>The VLAN ID for the Layer 2 traffic.</p>
VLAN Priority field	<p>The VLAN priority for the Layer 2 traffic.</p>
Source MAC Address field	<p>The source MAC address for the Layer 2 traffic.</p>
Destination MAC Address field	<p>The destination MAC address for the Layer 2 traffic.</p>
Layer 4	
Source Port field	<p>The source port of the Layer 4 traffic.</p>
Destination Port field	<p>The destination port of the Layer 4 traffic.</p>

Name	Description
Protocol field	The Internet protocol number of the Layer 4 traffic. Enter the IP protocol number in decimal, hexadecimal, or octal format.
Actions drop-down list	Select an action from the drop-down list.

**Step 3** Click **Install Flow** to install the flow into the device OR click **Save Flow** to save the flow to the **Flow Entries** table, but the system does not install the flow in the flow table of the device.

---







# CHAPTER 10

## Viewing Consistency Check

---

This chapter contains the following sections:

- [Consistency Check, on page 149](#)
- [Viewing Consistency Check, on page 149](#)

### Consistency Check

Consistency check shows the number of controller or node inconsistencies for each device and provides option to resolve the inconsistency issues. The consistency check feature shows two types of inconsistencies:

- Controller flows inconsistency: Flows are present in Cisco NDB, but missing from the device.
- Node flows inconsistency: Flows are present in the switch, but missing from Cisco NDB.



---

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

---

### Viewing Consistency Check

To check for inconsistency for an OpenFlow or NX-API based device, complete the following steps:

On the **Consistency Check** tab, the following details are displayed:

#### Procedure

---

- Step 1** Navigate to **ADMINISTRATION > Consistency Check**.
- Step 2** Click **FLOW CHECK NX-API/OpenFlow** tab to view the summary of inconsistencies for the NX-API/OpenFlow based devices. On the **Consistency Check** tab, the following details are displayed:
- Node Name
  - Inconsistent Controller Flow

- Inconsistent Node Flow
- Non NDB Flows

**Note** To fix an inconsistent flow, select the devices from the list and click **Fix Inconsistent Flow**.

**Step 3** To view detailed inconsistency information:

- Click **Inconsistent Controller Flows** to view the controller inconsistencies.
- Click **Inconsistent Node Flows** to view the node inconsistencies.
- Click **Non NDB Flows** (available only for NX-API) to view the ACLs present in the device by default or added manually.

**Step 4** To resolve the inconsistency issues:

- Click **Fix Inconsistent Flows** on the **Controller Inconsistent** page, to add the missing controller flows to the device.
  - Click **Fix Inconsistent Flows** on the **Node Inconsistent** page, to remove the stale flows from the device.
-



# CHAPTER 11

## Managing Users

This chapter contains the following sections:

- [Adding a User, on page 151](#)
- [Adding a Role, on page 152](#)
- [Adding a Group, on page 153](#)

## Adding a User

After creating a user, you can change the password, but you cannot change the roles assigned to the user.

### Procedure

**Step 1** Navigate to the **User Management** tab under **Administration** and click + **User** to add a user.

**Step 2** In the **Add User** window, complete the following fields:

Name	Description
<b>Username</b> field	The name that you want to assign to the user.
<b>Password</b> field	The password for the user. Passwords must be between 8 and 256 characters long, contain uppercase and lowercase characters, have at least one numeric character, and have at least one nonalphanumeric character.
<b>Verify Password</b> field	Verify the password by re-entering it.

Name	Description
Choose <b>Role(s)</b> drop-down list	Choose the role that you want to assign to the user. You can assign more than one role. It can be one of the following: <ul style="list-style-type: none"> <li>• <b>Application User</b>—Provides privileges that are defined in the specified application.</li> <li>• <b>Security</b>—Provides privileges that are defined in the security application.</li> <li>• <b>Network Administrator</b>—Provides full administrative privileges to all applications.</li> <li>• <b>Network Operator</b>—Provides read-only privileges to all applications.</li> <li>• <b>Slice User</b>—Provides access to a specified slice.</li> </ul>
Enter a <b>Role Name</b> field	If you choose <b>Application User</b> , enter the name that you want to assign to the role.

**Step 3** Click **Save** in the **User Management** window or click **Cancel** to cancel the action.

## Adding a Role

### Procedure

**Step 1** Navigate to the **User Management** tab under **Administration** and click + **Role**.

**Step 2** In the **Add Role** window, complete the following fields:

Field	Description
Name field	The name of the role.

Field	Description
Level drop-down list	<p>Choose the level that you want to assign to the role. This can be one of the following:</p> <ul style="list-style-type: none"> <li>• App-Administrator—Has full access to all Cisco Nexus Data Broker resources but the App-Administrator cannot add NXAPI or production devices into NDB because Administration tab is not available in NDB for App-Administrator role .</li> <li>• App-User—Has full access to create, edit, clone, or delete connections and redirections that are assigned to his resource group and resources that are created by another user with similar permissions. An App-User can only view Edge-SPAN, Tap, Monitoring device, and Production ports.</li> </ul> <p>An App-User can view resources that are created by another user with similar permissions in Toplogy page of NDB. But, you can not configure Edge-SPAN or Connections created by another App-User.</p>
Assign Group(s)	Assign groups to the selected role.

**Step 3** Click save.

## Adding a Group

### Procedure

**Step 1** Navigate to the **User Management** tab under **Administration** and click **+ Groups**.

**Step 2** In the **Add Group** window, complete the following field:

Field	Description
Resource Group Name	The name of the resource group.
Select Switch Node	Select a switch node from the drop-down list.
+ Assign Switch and Ports	Click + to add a new switch to the group.
Select Ports	Select the ports associated with the switch.
Assign Group to Roles	Assign a role to the group.

**Step 3** Click **Save**.

---



## CHAPTER 12

# Configuring the Setup for a Use Case in the Centralized Mode

---

This chapter contains the following sections:

- [Configuring Cisco Nexus Data Broker For Centralized Mode Using The CLI, on page 155](#)
- [Configuring Cisco Nexus Data Broker in Centralized Mode Using The GUI, on page 159](#)

## Configuring Cisco Nexus Data Broker For Centralized Mode Using The CLI

Complete the following steps to configure

### Procedure

---

- Step 1** Create two connections.
- Connection 1 aggregates TAP and SPAN port. Apply filters and deliver to two monitor devices in switch-2, that is connected to 1/2 and 1/1.
  - Connection 2 receives the TAP port traffic. After applying HTTP filter, the traffic is directed to only one monitor device.
- Step 2** Run Cisco Nexus Data Broker in Linux server.
- Step 3** Verify that the ofa package is there.
- Step 4** Install ofa.
- virtual-service install name ofa package ofa\_mmemb-1.1.5-r3-n3000-SPA-k9.ova
  - sh virtual-service list
  - configure
  - virtual-service ofa
  - activate
  - show virtual-service list
- Step 5** Configure OpenFlow switch.
- switch-1(config-virt-serv)# openflow

- b) switch-1(config-ofa)# switch 1
- c) switch-1(config-ofa-switch)# pipeline 203
- d) switch-1(config-ofa-switch)# controller ipv4 10.16.206.161 port 6653 vrf management security none
- e) switch-1(config-ofa-switch)# sh int br
- f) switch-1(config-ofa-switch)# of-port interface ethernet1/1-4
- g) switch-1(config-ofa-switch)# of-port interface ethernet1/47
- h) switch-2(config-ofa-switch)# show virtual-service list

## Example

Run Cisco Nexus Data Broker in Linux server.

```
[root@rhel64-ndb-nxapi NDB3.0.0]#
[root@rhel64-ndb-nxapi NDB3.0.0]# ls
ndb1000-sw-app-k9-3.0.0.zip xnc
[root@rhel64-ndb-nxapi NDB3.0.0]#
[root@rhel64-ndb-nxapi NDB3.0.0]# cd xnc/
[root@rhel64-ndb-nxapi xnc]# ls
bin configuration etc lib logs plugins runxnc.cmd runxnc.sh version.properties
work
[root@rhel64-ndb-nxapi xnc]# ./runxnc.sh -start
Running controller in background with PID: 11987, to connect to it please SSH to this host
on port 2400
[root@rhel64-ndb-nxapi xnc]#
```

Configure NDB to run as a service in the Linux server.

1. Download the script file named, `ndb`, based on the operating system (Ubuntu, CentOS, or Redhat). The service script is available at: <https://github.com/datacenter/nexus-data-broker/tree/master/serviceScripts>.
2. Update the Java Home location in the script file for NDB version is 3.2 and earlier. For the NDB version 3.3 and later, comment the line that configures Java Home.

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre
```

3. Change the permissions for the script file to 755. Use the `chmod 755 ndb` command. For example:

```
ndb-inst# chmod 755 ndb
```

4. Update the NDB location in the downloaded script file.

```
NDB_PATH - /home/user/xnc
```

5. Copy the script file to the `/etc/init.d/` folder in the Linux server.
6. Start, stop, and restart the NDB using the following commands

```
ndb-inst # ndb stop
ndb-inst # ndb start
ndb-inst # ndb restart
```

Verify that the ofa package is installed.

```
switch-1 - Switch
=====

switch-1#
```



```

switch-1# dir
 4096   Jun 01 23:55:07 2016  .patch/
 1044   Aug 13 00:15:17 2014  20140813_001215_poap_3799_init.log
 16     Aug 13 00:30:15 2014  cert.err
 9255   Jun 01 23:38:11 2016  clean_config
2885642 May 12 22:11:57 2014  llformtc-dplug-mzg.6.0.2.A3.0.23.bin
4194304 Sep 08 19:24:42 2014  messages
 3752   Mar 18 00:48:03 2014  mts.log
36825088 Apr 19 18:47:44 2016  n3500-uk9-kickstart.6.0.2.A6.5a.bin
37472256 Jun 01 23:43:34 2016  n3500-uk9-kickstart.6.0.2.A8.0.15.bin
180349300 Apr 19 18:49:37 2016  n3500-uk9.6.0.2.A6.5a.bin
190244286 Jun 01 23:42:07 2016  n3500-uk9.6.0.2.A8.0.15.bin
54343680 Apr 24 05:27:43 2016  ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova
 4096   Mar 18 06:08:07 2014  onep/
 3314   Apr 25 18:14:18 2014  sercert.pl2
 1024   Apr 19 18:58:37 2016  sprom_cstruct_2_0_0
 1024   Apr 19 18:59:22 2016  sprom_cstruct_3_0_0
 4096   Jan 01 03:25:17 2011  vdc_2/
 4096   Jan 01 03:25:17 2011  vdc_3/
 4096   Jan 01 03:25:17 2011  vdc_4/
 4096   Jun 01 23:31:49 2016  virt_strg_pool_bf_vdc_1/
 4096   Jun 01 23:31:49 2016  virtual-instance/
 4096   Aug 09 02:20:14 2014  virtual-instance-stby-sync/
243671040 May 09 20:55:18 2016  xnclite_ofa_jdk1877.ova
243732480 May 10 21:51:52 2016  xnclite_ofa_jdk1892.ova

Usage for bootflash://
1124974592 bytes used
770195456 bytes free
1895170048 bytes total
switch-1#

```

### Install ofa.

```

switch-1#
switch-1# virtual-service install name ofa package ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova
Note: Installing package 'bootflash://ofa_mmemb-1.1.5-r3-n3000-SPA-k9.ova' for virtual service
'ofa'. Once the install has finished, the VM may be activated. Use 'show virtual-service
list' for progress.

```

```
switch-1# sh virtual-service list
```

Virtual Service List:

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

```
switch-1# configure
```

Enter configuration commands, one per line. End with CNTL/Z.

```
switch-1(config)# virtual-service ofa
```

```
switch-1(config-virt-serv)# activate
```

Note: Activating virtual-service 'ofa', this might take a few minutes. Use 'show virtual-service list' for progress.

```
switch-1(config-virt-serv)# show virtual-service list
```

Virtual Service List:

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

```
switch-1(config-virt-serv)#
```

## Configure OpenFlow switch.

```

switch-1(config-virt-serv)# openflow
switch-1(config-ofa)# switch 1
switch-1(config-ofa-switch)# pipeline 203
switch-1(config-ofa-switch)# controller ipv4 10.16.206.161 port 6653 vrf management security
none
switch-1(config-ofa-switch)# sh int br

```

Ethernet Interface	VLAN	Type	Mode	Status	Reason	Speed	Port Ch #
Eth1/1	1	eth	access	up	none	10G(D)	--
Eth1/2	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/3	1	eth	access	up	none	10G(D)	--
Eth1/4	1	eth	access	up	none	10G(D)	--
Eth1/5	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/6	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/7	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/8	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/9	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/10	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/11	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/12	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/13	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/14	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/15	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/16	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/17	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/18	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/19	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/20	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/21	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/22	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/23	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/24	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/25	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/26	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/27	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/28	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/29	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/30	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/31	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/32	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/33	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/34	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/35	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/36	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/37	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/38	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/39	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/40	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/41	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/42	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/43	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/44	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/45	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/46	1	eth	access	down	SFP not inserted	10G(D)	--
Eth1/47	1	eth	access	up	none	10G(D)	--
Eth1/48	1	eth	access	down	SFP not inserted	10G(D)	--

```

-----
Port      VRF      Status IP Address      Speed  MTU
-----
mgmt0    --      up      10.16.206.129    1000   1500
switch-1 (config-ofa-switch) #
switch-1 (config-ofa-switch) #
switch-1 (config-ofa-switch) #
switch-1 (config-ofa-switch) # of-port interface ethernet1/1-4
switch-1 (config-ofa-switch) # of-port interface ethernet1/47
switch-1 (config-ofa-switch) #

Switch-2
=====

switch-2 (config-ofa-switch) # show virtual-service list

Virtual Service List:

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

switch-2 (config-ofa-switch) #

```

### What to do next

For centralized mode, complete the steps for configuring Cisco Nexus Data Broker using the GUI as outlined in the next section.

## Configuring Cisco Nexus Data Broker in Centralized Mode Using The GUI

After configuring the Cisco Nexus Data Broker using the CLI, complete the following steps:

### Procedure

- Step 1** Open a new browser window and type `https://<NDB-IP>:8443`.
- Step 2** Configure the TAP and SPAN ports using the GUI.
- Step 3** Select switch 2 and configure the delivery ports.
- Step 4** Add switch 1 and switch 2 in NX-API as in auxiliary mode by enabling the **Set Auxiliary Node** option in the **Add Device** window.
- Step 5** Click **Nodes Learned** to configure the mode.
- Step 6** For switch 1, click on the OpenFlow device ID and change the **Operation Mode** in the **Update Node Information** window to **Proactive forwarding only** option.
- Step 7** For switch 2, click on the OpenFlow device ID and change the **Operation Mode** in the **Update Node Information** window to **Proactive forwarding only** option.
- Step 8** In the **Port Definition** window, click **Edit** for delivery port 1/1.
- Step 9** Check the **Enable Timestamp Tagging** option in the **Configure Ports** window and click **Submit**.

- Step 10** In the **Port Definition** window, click **Edit** for delivery port 1/2.
- Step 11** Check the **Enable Timestamp Tagging** option in the **Configure Ports** window and click **Submit**.  
After you configure the timestamp, the **TS-Tag** field is displayed next to the port under the **Port Configuration** tab. You can view the monitoring devices in the **Monitoring Devices** tab.
- Step 12** Add different traffic filters under the **Filters** tab.
- Step 13** Click **Topology** to understand how the devices are learned.
- Step 14** Click **Connections** to create a connection.
- Step 15** Click **Add Connection** and add filters and the monitoring devices for connection 1.
- Step 16** Add connection 2 in a similar way.  
After the connections are created, view the connections in the **Connections** tab.
- Step 17** View the final topology.

---

### Example of the configuration on switch 1 and switch 2:

Switch 1 Configuration: switch-1

```
hardware profile tcam region racl 512
hardware profile tcam region ifacl 1024 double-wide
hardware profile forwarding-mode openflow-only
hardware internal mtc-usd ttag-eth-type 0x88b5
snmp-server user admin network-admin auth md5 0x188749ba5e1c6af881227235b1b14d04 priv
0x188749ba5e1c6af881227235b1b14d04 localizedkey
```

```
vlan 1
vrf context management
ip route 0.0.0.0/0 10.16.206.1
```

```
interface Ethernet1/1
no lldp transmit
spanning-tree bpdufilter enable
mode openflow
no shutdown
```

```
interface Ethernet1/2
no lldp transmit
spanning-tree bpdufilter enable
mode openflow
no shutdown
```

```
interface Ethernet1/3
no lldp transmit
switchport mode trunk
spanning-tree bpdufilter enable
mode openflow
no shutdown
```

```
interface Ethernet1/4
no lldp transmit
switchport mode trunk
spanning-tree bpdufilter enable
mode openflow
no shutdown
```

```
interface Ethernet1/5
```

```
no shutdown

interface Ethernet1/6
no shutdown

interface Ethernet1/7
no shutdown

interface Ethernet1/8
no shutdown

interface Ethernet1/9
no shutdown

interface Ethernet1/10
no shutdown

interface Ethernet1/11
no shutdown

interface Ethernet1/12
no shutdown

interface Ethernet1/13
no shutdown

interface Ethernet1/14
no shutdown

interface Ethernet1/15
no shutdown

interface Ethernet1/16
no shutdown

interface Ethernet1/17
no shutdown

interface Ethernet1/18
no shutdown

interface Ethernet1/19
no shutdown

interface Ethernet1/20
no shutdown

interface Ethernet1/21
no shutdown

interface Ethernet1/22
no shutdown

interface Ethernet1/23
no shutdown

interface Ethernet1/24
no shutdown

interface Ethernet1/25
no shutdown

interface Ethernet1/26
no shutdown
```

```
interface Ethernet1/27
  no shutdown

interface Ethernet1/28
  no shutdown

interface Ethernet1/29
  no shutdown

interface Ethernet1/30
  no shutdown

interface Ethernet1/31
  no shutdown

interface Ethernet1/32
  no shutdown

interface Ethernet1/33
  no shutdown

interface Ethernet1/34
  no shutdown

interface Ethernet1/35
  no shutdown

interface Ethernet1/36
  no shutdown

interface Ethernet1/37
  no shutdown

interface Ethernet1/38
  no shutdown

interface Ethernet1/39
  no shutdown

interface Ethernet1/40
  no shutdown

interface Ethernet1/41
  no shutdown

interface Ethernet1/42
  no shutdown

interface Ethernet1/43
  no shutdown

interface Ethernet1/44
  no shutdown

interface Ethernet1/45
  no shutdown

interface Ethernet1/46
  no shutdown

interface Ethernet1/47
  no lldp transmit
  spanning-tree bpdufilter enable
```

```
mode openflow
no shutdown

interface Ethernet1/48
no shutdown

interface mgmt0
vrf member management
ip address 10.16.206.129/24
line console
line vty
boot kickstart bootflash:/n3500-uk9-kickstart.6.0.2.A8.0.15.bin
boot system bootflash:/n3500-uk9.6.0.2.A8.0.15.bin
openflow
switch 1
  pipeline 203
  controller ipv4 10.16.206.161 port 6653 vrf management security none
  of-port interface ethernet1/1-4
  of-port interface ethernet1/47
virtual-service ofa
activate
=====

Switch 2 Configuration : switch-2

hardware profile tcam region racl 512
hardware profile tcam region ifacl 1024 double-wide
hardware profile forwarding-mode openflow-only
hardware internal mtc-usd ttag-eth-type 0x88b5
snmp-server user admin network-admin auth md5 0xb7289bc7f348c5044b495f93bac10137 priv
0xb7289bc7f348c5044b495f93bac10137 localizedkey

vlan 1
vrf context management
ip route 0.0.0.0/0 10.16.206.1

interface Ethernet1/1
no lldp transmit
ttag
switchport mode trunk
spanning-tree bpdufilter enable
mode openflow
no shutdown

interface Ethernet1/2
no lldp transmit
ttag
switchport mode trunk
spanning-tree bpdufilter enable
mode openflow
no shutdown

interface Ethernet1/3
no shutdown

interface Ethernet1/4
no shutdown

interface Ethernet1/5
no shutdown

interface Ethernet1/6
no shutdown
```

```
interface Ethernet1/7
  no shutdown

interface Ethernet1/8
  no shutdown

interface Ethernet1/9
  no shutdown

interface Ethernet1/10
  no shutdown

interface Ethernet1/11
  no shutdown

interface Ethernet1/12
  no shutdown

interface Ethernet1/13
  no shutdown

interface Ethernet1/14
  no shutdown

interface Ethernet1/15
  no shutdown

interface Ethernet1/16
  no shutdown

interface Ethernet1/17
  no shutdown

interface Ethernet1/18
  no shutdown

interface Ethernet1/19
  no shutdown

interface Ethernet1/20
  no shutdown

interface Ethernet1/21
  no shutdown

interface Ethernet1/22
  no shutdown

interface Ethernet1/23
  no shutdown

interface Ethernet1/24
  no shutdown

interface Ethernet1/25
  no shutdown

interface Ethernet1/26
  no shutdown

interface Ethernet1/27
  no shutdown
```



```
interface Ethernet1/28
  no shutdown

interface Ethernet1/29
  no shutdown

interface Ethernet1/30
  no shutdown

interface Ethernet1/31
  no shutdown

interface Ethernet1/32
  no shutdown

interface Ethernet1/33
  no shutdown

interface Ethernet1/34
  no shutdown

interface Ethernet1/35
  no shutdown

interface Ethernet1/36
  no shutdown

interface Ethernet1/37
  no shutdown

interface Ethernet1/38
  no shutdown

interface Ethernet1/39
  no shutdown

interface Ethernet1/40
  no shutdown

interface Ethernet1/41
  no shutdown

interface Ethernet1/42
  no shutdown

interface Ethernet1/43
  no shutdown

interface Ethernet1/44
  no shutdown

interface Ethernet1/45
  no shutdown

interface Ethernet1/46
  no shutdown

interface Ethernet1/47
  no lldp transmit
  spanning-tree bpdufilter enable
  mode openflow
  no shutdown

interface Ethernet1/48
```

```
no shutdown

interface mgmt0
  vrf member management
  ip address 10.16.206.130/24
line console
line vty
boot kickstart bootflash:/n3500-uk9-kickstart.6.0.2.A8.0.15.bin
boot system bootflash:/n3500-uk9.6.0.2.A8.0.15.bin
openflow
  switch 1
    pipeline 203
      controller ipv4 10.16.206.154 port 6653 vrf management security none
      controller ipv4 10.16.206.161 port 6653 vrf management security none
      of-port interface ethernet1/1-2
      of-port interface ethernet1/47
virtual-service ofa
  activate
```



## CHAPTER 13

# Managing System

---

This chapter contains the following sections:

- [About Slices, on page 167](#)
- [Adding a Slice, on page 168](#)
- [Adding a Flow Specification, on page 168](#)
- [About AAA Servers, on page 169](#)
- [Adding a AAA Server, on page 169](#)
- [Viewing Cluster Information, on page 171](#)
- [Viewing the OSGi Console, on page 171](#)
- [Viewing the Northbound API Content, on page 171](#)
- [Downloading the System Log Files, on page 172](#)
- [Backing Up or Restoring the Configuration Using NDB GUI, on page 173](#)
- [Recovering the Administrative Password, on page 175](#)
- [Uninstalling the Application Software, on page 176](#)

## About Slices

The slices screen provides a way for you, as a network administrator, to partition networks into many logical networks. This feature allows you to create multiple disjoint networks and assign different roles and access levels to each one. Each logical network can be assigned to departments, groups of individuals, or applications. Multiple disjoint networks can be managed using the Cisco Nexus Data Broker application.

The slices are created based on the following criteria:

- **Network devices**—The devices that can be used in the slice.  
Network devices can be shared between slices.
- **Network device interfaces**—The device interfaces that can be used in the slice.  
Network device interfaces can be shared between slices.
- **Flow Specification**—A combination of source and destination IP, protocol, and source and destination transport ports used to identify the traffic that belongs to the slice.  
Flow specifications can be assigned to different slices if the associated network devices and interfaces are disjointed.




---

**Note** You can also use VLAN IDs to segregate the slice traffic.

---

Slices must be created by a Cisco Nexus Data Broker user with the Network Administrator role. After creation, the slices can be managed by a user with the Slice Administrator role.

Slices can overlap if each slice has at least one unique attribute. For example, a slice can share the same physical switches and ports, but be differentiated by the type of traffic it receives.

## Adding a Slice

### Procedure

---

**Step 1** Navigate to the **System** tab under **Administration** and click + **Slice**.

The **Add Slice** window is displayed.

**Step 2** In the **Add Slice** window, complete the following fields:

Name	Description
Name	The name that you want to assign to the slice.
Select Devices	Select the devices that you want to associate with the slice.

**Step 3** Click **Save**.

---

## Adding a Flow Specification

### Before you begin

Create a slice before you add a flow specification.




---

**Note** By default, a flow specification is bidirectional.

---

### Procedure

---

**Step 1** Navigate to the **System** tab under **Administration** and click + **Flow Spec** to add a flow specification for the selected slice.

**Step 2** In the **Add Flow Spec** dialog box, complete the following fields:

Name	Description
<b>Name</b> field	The name that you want to use for the flow specification.
<b>VLAN</b> field	The VLAN ID or the range of VLAN IDs that you want to use for the flow specification.
<b>Source IP</b> field	The source IP address that you want to use for the flow specification.
<b>Destination IP</b> field	The destination IP address that you want to use for the flow specification.
<b>Protocol</b> field	The IP protocol number in decimal format that you want to use for the flow specification.
<b>Source Port</b> field	The source port that you want to use for the flow specification.
<b>Destination Port</b> field	The destination port that you want to use for the flow specification.

**Step 3** Click **Save**.

OR you can click **Cancel** to cancel the action.

## About AAA Servers

AAA enables the security appliance to determine who the user is (authentication), what the user can do (authorization), and what the user did (accounting). Cisco Nexus Data Broker uses Remote Authentication Dial-In User Service (RADIUS) or Terminal Access Controller Access-Control System Plus (TACACS+) to communicate with an AAA server.

Remote authentication and authorization is supported using the AAA server. To authenticate each user, Cisco Nexus Data Broker uses both the login credentials and an attribute-value (AV) pair that assigns the authorized role for the user as part of the user administration. After successful authentication, the Cisco AV pair is returned to Cisco Nexus Data Broker for resource access authorization.

## Adding a AAA Server

Starting with Cisco NSB Release 3.4, you can configure a AAA server using a Domain Name or an IP address.



**Note** Starting with Cisco Nexus Data Broker Release 3.7, you can now verify status of AAA TACACS server before adding it using the **Check Server** option on the **Add AAA Server** dialog box. The **Check Server** option verifies whether the AAA server that you are configuring is reachable or not and whether the credentials are valid or not.



**Note** When the configured AAA server(s) are not reachable, the user request is authenticated locally. If the AAA server is reachable and the user authentication fails, the user request is not authenticated locally.

**Procedure**

**Step 1** Navigate to the **AAA** tab under **System** and click **Add Server**.  
The **Add AAA Server** window is displayed.

**Step 2** In the **Add AAA Server** window, complete the following fields:

Name	Description
<b>Protocol</b> field	Choose the protocol for the AAA server. This can be one of the following: <ul style="list-style-type: none"> <li>• <b>Radius+</b></li> <li>• <b>TACACS+</b></li> <li>• <b>LDAP</b></li> </ul> <p><b>Note</b> For detailed information about how to configure LDAP for AAA server, see <i>Configuring User Authentication for LDAP</i>.</p>
<b>Server Address</b> field	Server IP address or Domain Name.
<b>Secret</b> field	The shared secret configured on the AAA server.
<b>User Name</b> field	User name for authentication..
<b>Password</b> field	Password for authenticating the user.

**Step 3** (Optional) Click **Check Server** to verify whether the server is reachable and authentication credentials are valid or not.

**Note** The **Check Server** option is available only for TACACS AAA server.

**Step 4** Click **Save**.

## Viewing Cluster Information

### Procedure

---

Navigate to the **Cluster** tab under **System** to view information about the clusters.

The cluster management dialog boxes are read-only. The dialog box lists the IP addresses of all of the Cisco Nexus Data Broker instances in the cluster.

**Note** For the backup and upload features to work properly, all the servers in the cluster should be stopped and then they should be restarted. You should not configure any functionality during this time. Once the upload configuration is done, you should not configure anything from any other nodes in the cluster as it might lead to few inconsistencies in the data.

---

## Viewing the OSGi Console

You can view all of Cisco Nexus Data Broker bundles that comprise the application by viewing the OSGi Web Console.



**Note** This procedure does not provide a step-by-step guide to everything you can do in the OSGi Web Console for **Cisco XNC Bundles** list. It guides you in opening the OSGi Web Console and viewing bundle information.

---

### Procedure

---

- Step 1** Navigate to the **System** tab under **Administration**.  
A new browser tab opens.
- Step 2** Click **OSGI**.
- Step 3** Enter your username and password, and then press **Enter**.  
The **Cisco – XNC Bundles** list is displayed. In this page you can view all of the active packages, filter on the package name to specify bundle names, and complete other tasks.
- Step 4** When you are finished viewing the list, close the **Cisco – XNC Bundles** browser tab.
- 

## Viewing the Northbound API Content

You can view all of Cisco Nexus Data Broker northbound API content for the application by opening a browser tab using the **Northbound API** tool (book icon) in the menu bar.

## Procedure

---

**Step 1** From the menu bar, click the **Northbound API** button.

A new browser tab (Swagger UI) is opened and the complete list of northbound API content used in Cisco Nexus Data Broker is displayed.

From this tab, you can do the following:

- Show or hide the operations for an API.
- List the operations for an API.
- Expand the operations for an API.

**Step 2** When you are finished viewing northbound API content, close the browser tab.

---

# Downloading the System Log Files

You can download log files for Cisco Nexus Data Broker to use for analysis. Log files are saved as a .zip archive.



**Note** Starting with Cisco Nexus Data Broker Release 3.7, naming convention for the System log files has changed. The System log file name now reflects the time stamp when the file is generated. For example, `NDBLogs-21Aug2018_11_15_08.zip`.

---



**Note** Starting with Cisco Nexus Data Broker Release 3.7, the System log file now provides additional details such as the device connection information, number of redirections, and details about all the devices managed by NDB.

---

## Procedure

---

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

**Step 2** Click **Download Logs**.

A dialog box opens in the browser prompting you to either open or save the .zip file.

**Step 3** Do one of the following:

- Save the archive to a location of your choosing, for example, `home/ndbconfig`.
  - Open the archive to view the contents, and then save it.
-



## Backing Up or Restoring the Configuration Using NDB GUI

Starting with Cisco NDB, Release 3.4, you can create and restore a NDB configuration backup instantly during the pre-deployment phase. Support for instant backup is currently available in NX-OS, OpenFlow, and AUX switches.

Using this feature, a backup point is created with current NDB configuration that can be restored to the system. Cisco NDB provides three backup options:

- Schedule backup to NDB Server—Backup is created at the specified time in the NDB server in the backup directory of xnc.
- Backup now to NDB server—Backup is created in the NDB server in the backup directory of xnc.
- Backup now locally—Backup is created and available for download using Web browser.

Cisco NDB provides two restore options:

- Restore from Server—Configuration is restored from a server.
- Restore Locally—Configuration is restored from a local directory.

## Backing Up or Restoring the Configuration Using the CLI

### Procedure

---

- Step 1** Navigate to the `xnc/bin` directory that was created when you installed the software.
- Step 2** Back up the configuration by entering the `./xnc config --backup` command.
- The `--backup` option creates a backup archive (in .zip format) of the startup configuration in the current `xnc` distribution. The backup archive is stored in `{xncHome}/backup/`. A new archive is created each time that the backup command is entered using a filename with the current timestamp.
- Step 3** Restore the configuration by entering the `./xnc config --restore --backupfile {zip_filename}` command.
- The `--restore` option restores the startup configuration of the current `xnc` distribution from an existing backup archive. The restore action requires the absolute path of the backup archive.
- 

## Scheduling Configuration Backup to NDB Server

Beginning with Cisco Nexus Data Broker, Release 3.2, you can schedule automatic configuration backup to a server with a start date and an end date. Backup is created at the specified time in the NDB server in the backup directory of XNC. When any configuration is performed in Cisco Nexus Data Broker, it is saved automatically.



**Note** Beginning with Cisco Nexus Data Broker, Release 3.2, you do not need to use the **Save** option to save the Cisco Nexus Data Broker configurations. Even after you restart the server, the configuration is autosaved.

### Procedure

- Step 1** Navigate to **Administration** -> **System** -> **Backup/Restore** tab.
- Step 2** From the **Backup** drop-down list, select **Schedule backup to NDB Server** to open the **Schedule** page.
- Step 3** In the **Schedule** page, enter the following details:

Field	Description
Start Date	Date to start the configuration backup.
Start Time	Time to start the configuration backup.
Choose Pattern	Select the pattern of the backup. Valid options are: <ul style="list-style-type: none"> <li>• Daily</li> <li>• Weekly</li> <li>• Monthly</li> </ul>
Choose	Select when to stop the backup process. Valid options are: <ul style="list-style-type: none"> <li>• No End Date: Continue taking backup as per the specified frequency.</li> <li>• End Date: Continue taking backup till the specified date.</li> <li>• Occurrences: Indicates the number of times to take the backup.</li> </ul>
Enable	Enables the scheduled backup.

## Restoring Configuration Locally

You can upload the saved system configuration files for Cisco Nexus Data Broker to restore the Cisco Nexus Data Broker application in the case of a failure or other event. After restoring you configuration, you will need to restart Cisco Nexus Data Broker.

Direct upload path to Cisco Nexus Data Broker is available from Cisco Nexus Data Broker, Release 3.0 or above. If you are running a previous release, upload it to Release 3.0 first before uploading to Release 3.2.

### Procedure

---

- Step 1** Navigate to the **System** tab under **Administration** and click **Restore Locally** to upload the configuration.
- Step 2** Navigate to the location of the file `configuration_startup.zip`.
- Step 3** Click on the archive file.

The system configuration is uploaded and the browser displays a message informing you that you need to restart the server.

- Step 4** Restart the server, and then log back in to the Cisco Nexus Data Broker GUI.
- 

## Backing Up Configuration Locally

Configuration backup is created in the local machine in the specified directory. You can backup the system configuration locally in case you need to restore the system after an upgrade or other change. System configuration files are saved in a zipped archive.

### Procedure

---

- Step 1** Navigate to **Administration** -> **System** -> **Backup/Restore** tab.
  - Step 2** From the **Backup** drop-down list, select **backup now locally**.  
The configuration backup is created and downloaded to the local directory.
- 

## Restoring Configuration from a Server

You can restore NDB configuration from a server using the NDB GUI. Complete the following steps to restore a configuration from a server:

### Procedure

---

- Step 1** Navigate to **Administration** -> **System** -> **Backup/Restore** tab.
  - Step 2** Select a backup from the list and click **Restore** to restore the selected configuration.
- 

## Recovering the Administrative Password

The Cisco Nexus Data Broker network administrator user can return the administrative password to the factory default.




---

**Note** The software may or may not be running when this command is used. If the software is not running, the password reset takes effect the next time that it is run.

---

### Procedure

---

- Step 1** Open a command window where you installed Cisco Nexus Data Broker.
- Step 2** Navigate to the `xnc/bin` directory that was created when you installed the software.
- Step 3** Reset the administrative password by entering the `./xnc reset-admin-password [--wait-seconds {wait_time} --password {password}]` command.

Resets the admin password to the default or specified password by restarting the user manager.

- The **wait-seconds** is the length of time, in seconds, to wait for the user manager to restart. The minimum is 5 seconds and the maximum is 60 seconds.
- The **password** is the administrative password.

- Note**
- The password must be from 8 to 256 characters, contain both uppercase and lowercase characters, and have at least one number and one non-alphanumeric character.
  - If you leave the password blank, it is reset to the factory default of "admin".
  - Each time that you reset the administrative password, make sure that the new password meets these requirements or you will not be able to log in to Cisco Nexus Data Broker.
- 

## Uninstalling the Application Software

### Before you begin

Ensure that your Cisco Nexus Data Broker application is stopped before proceeding.

### Procedure

---

- Step 1** Navigate to the directory where you created the Cisco Nexus Data Broker installation.  
For example, if you installed the software in `Home/ CiscoNDB`, navigate to the `Home` directory.
- Step 2** Delete the `CiscoNDB` directory.
-



## INDEX

3rd Party Certification [57](#)

### A

Adding a Device to SPAN [135](#)

### G

Generating Self-Signed Certification [37, 45, 57](#)

Generating TLS [37, 45, 57](#)

### N

network packet broker [57](#)

Network packet broker [135](#)

### S

SPAN [135](#)

