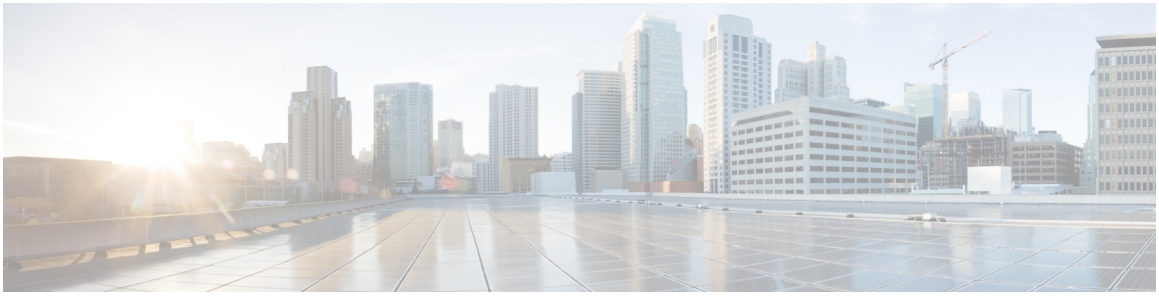
Cisco Nexus Dashboard Data Broker Deployment Guide, Release 3.10.5

First Published: 2025-05-20

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

Trademarks



Trademarks

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

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

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

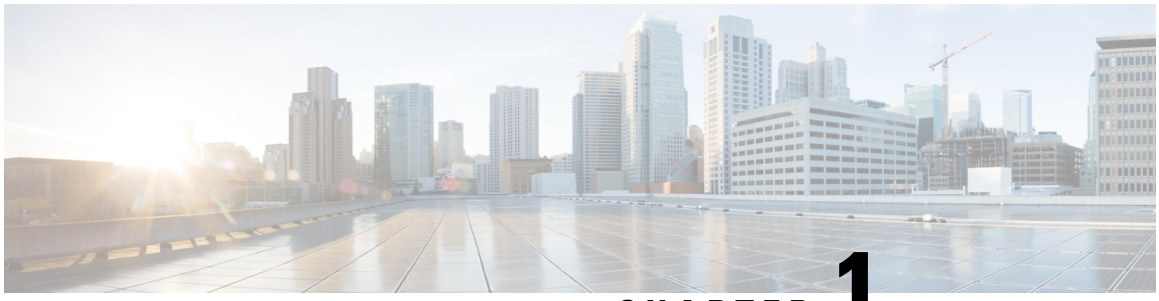
Any products and features described herein as in development or available at a future date remain in varying stages of development and will be offered on a when-and if-available basis. Any such product or feature roadmaps are subject to change at the sole discretion of Cisco and Cisco will have no liability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

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

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

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

CONTENTS



CHAPTER 1

Overview

This chapter gives us an overview of the Cisco Nexus Dashboard Data Broker.

About Cisco Nexus Dashboard 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 Dashboard 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. Using the Cisco NX-API agent to communicate to the switches, Cisco Nexus Dashboard Data Broker provides advance features for traffic management.

Cisco Nexus Dashboard Data Broker provides management support for multiple disjointed Cisco Nexus Dashboard Data Broker networks. You can manage multiple Cisco Nexus Dashboard 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 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.

Note Beginning with Release 3.10.1, Cisco Nexus Data Broker (NDB) has been renamed to Cisco Nexus Dashboard Data Broker. However, some instances of NDB are present in this document, to correspond with the GUI, and installation folder structure. References of NDB/ Nexus Data Broker/ Nexus Dashboard Data Broker can be used interchangeably.

Cisco Nexus Dashboard Data Broker Hardware and Software Interoperability Matrix

See the relevant *Cisco Nexus Dashboard Data Broker Release Notes* for the latest matrix.

Python Activator Scripts for NX-OS Images

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

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

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

Table 1: Python Activator Scripts for NX-OS Images

Python activator script file name	NX-OS Image
NDBActivator4.0_9.3_plus.py	Cisco NXOS version 9.3(1) and above.

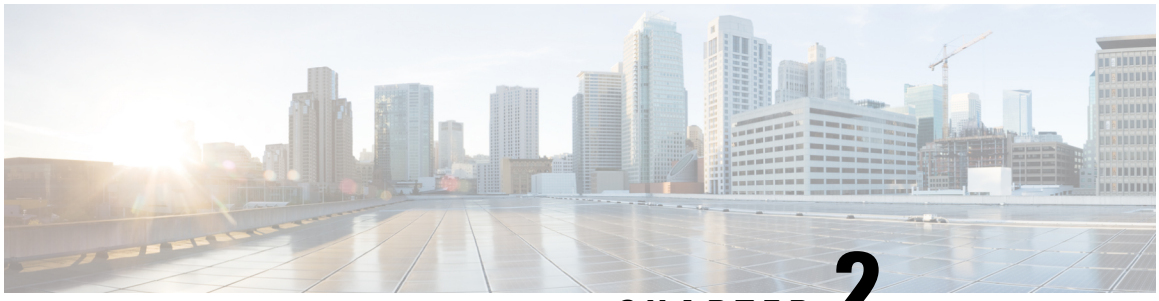
System Requirements

The following table lists the system requirements as per the deployment size:

Table 2: 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

Description	Small	Medium	Large
Number of switches for TAP and SPAN aggregation	Upto 25 switches	Upto 50 switches	75 to 100 switches
Hard disk	Minimum of 40 GB of free space available on the partition on which the 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.		



CHAPTER 2

Deploying Cisco Nexus Dashboard Data Broker Software in Centralized Standalone Mode

This chapter contains details of procedures for installing and upgrading Nexus Dashboard Data Broker in centralized mode.

Beginning with Release 3.10.1, Cisco Nexus Data Broker (NDB) has been renamed to Cisco Nexus Dashboard Data Broker. However, some instances of NDB are present in this document, to correspond with the GUI, and installation folder structure. References of NDB/ Nexus Data Broker/ Nexus Dashboard Data Broker, can be used interchangeably.

Before your proceed with the upgrade/ install procedures in this chapter, compare the **md5sum** between the Nexus Dashboard Data Broker image on Cisco.com, and the image file copied to linux. Use the following command to check (linux):

```
cisco@NDB-virtual-machine:~/3.10/$ md5sum ndb1000-sw-app-k9-3.10.5.zip
Displayed output: 518db25b4a89c996340c0316f72a6287 ndb1000-sw-app-k9-3.10.5.zip
```


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

Before proceeding with the installation, check the System Requirements section in the Overview chapter of this guide.

Use the custom java version by making these changes:

```
Under ndb folder, modify the start.sh file. In this file, comment out the below lines:
a. export JAVA_HOME="$PHYS_DIR$JRE_EXTRACTED_FOLDER"
b. export PATH=$PATH:$JAVA_HOME/bin
```

Restart the data broker service.

Installing the Cisco Nexus Dashboard Data Broker Software in Centralized Mode

Complete these steps to install Cisco Nexus Dashboard Data Broker software in Centralized mode:

Procedure

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

Step 2 Scroll down, and click **Downloads**.

Step 3 In the **Select a Product** field, enter *Nexus Dashboard Data Broker*.

The file information for Release 3.10.5 is displayed: Cisco Nexus Data Broker Software Application: ndb1000-sw-app-k9-3.10.5.zip

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

Step 4 Download the Cisco Nexus Data Broker application bundle.

Step 5 Create a directory in your Linux machine where you plan to install Cisco Nexus Dashboard Data Broker.

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

Step 6 Copy the Cisco Nexus Dashboard Data Broker zip file into the directory that you created.

Step 7 Unzip the Cisco Nexus Dashboard Data Broker zip file.

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

- `runndb.sh` file—The file that you use to launch Cisco Nexus Dashboard Data Broker.
- `version.properties` file—The Cisco Nexus Dashboard Data Broker build version.
- `configuration` directory—The directory that contains the Cisco Nexus Dashboard Data Broker initialization files.

  This directory also contains the etc subdirectory that contains profile information, and startup subdirectory where configurations are saved.
- `bin` directory—The directory that contains the following script:
  - `ndb` file—This script contains the Cisco Nexus Dashboard Data Broker common CLI.
- `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 Dashboard 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 Dashboard Data Broker application is started.

Step 8 Start the data broker application using `runndb.sh -start .`

Upgrading the Application Software in Centralized Mode Using CLI

Use the **upgrade** command to upgrade to release 3.10.5.

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.

  In case, the upgrade was not successful because of the mismatch of the host name, use RMA to correct the configuration of the device. See the *RMA* section in the *Cisco Nexus Data Broker Configuration Guide*.

- 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.

- The latest Nexus Dashboard Data Broker zip file must be extracted in an empty directory.

Hitless Upgrade

For hitless upgrade, configuration backup for releases prior to NDB Release 3.8, will always be the standard upgrade by re-configuring the devices during the upgrade.

Configuration backup taken for releases NDB 3.8 and after, will always be a Hitless upgrade using CLI.

Before you begin

- Backup up the data broker configuration. See the *Backup/ Restore* section in the *Cisco Nexus Dashboard Data Broker Configuration Guide.*
- Back up your `config.ini` file.

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

Note When you run `runndb.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 Backup your Cisco Nexus Data Broker release installation using your standard backup procedures.

Step 2 In a web browser, navigate to Cisco.com.

Step 3 Under **Support**, click **All Downloads**.

Step 4 In the center pane, click **Cloud and Systems Management**.

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

Step 6 Download the applicable bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.10.5.zip

Step 7 Create a temporary directory in your Linux machine where you plan to upgrade to the latest release.

Step 8 Unzip the release 3.10.5 zip file into the temporary directory that you created.

Step 9 Navigate to the `ndb` directory that was created when you installed the Cisco Nexus Data Broker release (created in the previous step).

Step 10 Stop the running Cisco Nexus Data Broker instance.

Step 11 Navigate to the `ndb/bin` directory in the temporary directory that you created for release 3.10.5 upgrade software.

Step 12 Upgrade the application by entering the **./ndb upgrade --perform --target-home** {*ndb_directory_to_be_upgraded*} **[--verbose] [--backupfile** {*ndb_backup_location_and_zip_filename*}**]** command.

You can use one of the following options:

Option	Description
--perform --target-home {*ndb_directory_to_be_upgraded*}	Upgrades the Cisco ndb Monitor Manager intallation to Cisco NDB.
--perform --target-home {*ndb_directory_to_be_upgraded*} **--backupfile** {*ndb_backup_location_and_zip_filename*}	Upgrades the Cisco ndb Monitor Manager installation to Cisco NDB and creates a backup.zip file in the directory path that you set. **Note** • You must provide the name of the backup file and the .zip extension. • The backup file should not be saved in the ndb directory with current NDB installation or its subdirectory.
--verbose	Displays detailed information to the console. This option can be used with any other option and is disabled by default.
--validate --target-home {*ndb_directory_to_be_upgraded*}	Validates the installation.
./ndb help upgrade	Displays the options for the **upgrade** command.

Step 13 Navigate to the older folder (**xnc** in releases prior to Release 3.10) where you originally installed Cisco NDB. Rename the folder from **xnc** to **ndb**.

Step 14 Start the application processes using **runndb.sh -start**.

Upgrading the Application Software in Centralized Mode Using GUI

Note The latest Nexus Dashboard Data Broker zip file must be extracted in an empty directory.

Hitless Upgrade

For hitless upgrade, configuration backup for releases prior to NDB Release 3.8, will always be the standard upgrade by re-configuring the devices during the upgrade.

Configuration backup taken for releases NDB 3.8 and after, will always be a Hitless upgrade using CLI.

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

Procedure

Step 1 Log in to the Nexus Dashboard Data Broker GUI.

Step 2 Navigate to **Administration** > **Backup/Restore** to download the configuration in zip file format.

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

Step 3 Stop the current NDB instance using the **runndb.sh -stop** command.

Example:

```
./runndb.sh -stop
```

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

Step 5 In a web browser, navigate to Cisco.com.

Step 6 Scroll down and click **Downloads**.

The **Software Downloads** page is displayed.

Step 7 In the **Select a Product** field, type in **Nexus Dashboard Data Broker**.

You are taken to a page from where you can download the latest Nexus Dashboard Data Broker software.

Step 8 Download the Cisco NDB Release 3.10.5 applicable bundle: Cisco Nexus Data Broker Software Application—ndb1000-sw-app-k9-3.10.5.zip

Step 9 Create a directory in your Linux machine where you plan to upgrade to Cisco NDB.

Step 10 Unzip the Cisco NDB Release 3.10.5 zip file into the directory that you created.

Step 11 Navigate to the `ndb` directory that was created when you installed the Cisco Nexus Data Broker release (created in the previous step).

Step 12 Start the new NDB installation using the **runndb.sh -start** command.

Example:

```
./runndb.sh -start
```

Step 13 Navigate to **Administration** > **Backup/Restore**.

Step 14 To reconfigure the device during the upgrade, select the **Restore** option (by checking the check-box) during configuration upload.

Step 15 Restart the new NDB instance using the **runndb.sh -restart** command.

Example:

```
./runndb.sh -restart
```

Starting the Application

Procedure

Note
When you are running NDB 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 ./runndb.sh script, the following message is displayed on the screen: Web GUI can be accessed using below URL: *[https://<IP_address>: 8443]*.

Java 8 is required for NDB. Setup JAVA_HOME before NDB is started.

You can use one of the following options:

Option	Description
no option	
-jmxport *port_number*	Enables JMX remote access on the specified JVM port.
-debugport *port_number*	Enables debugging on the specified JVM port.
-start	
-start *port_number*	
-stop	
-restart	
-status	
-console	
-help	Displays the options for the **./runndb.sh** command.
-tls	To enable TLS, start the controller by entering the **./runndb.sh -tls -tlskeystore** *keystore_file_location* **-tlstruststore** *truststore_file_location* command.

Verifying The Application Status

Procedure

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

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

The controller outputs the following, which indicates that the controller is running the Java process with PID 21680:

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

What to do next

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

Upgrade the application software with TLS

Use this procedure to upgrade from a 3.10.x release to 3.10.5 or a 3.9.x release to 3.10.5 with TLS-enabled.

Procedure

Step 1 Log in to the existing NDDB GUI instance using https://server IP:8443.

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

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

If you are upgrading from 3.9.x, your backup file contains the TLS files. The `tlsconf`, `tlsKeyStore`, `tlsTrustStore` files are part of your backup.

If you are upgrading from 3.10.x, your backup file will not contain the `tlsTrustStore`, `tlsKeyStore` files. You will need to manually upload the tls files to the /ndb/configuration folder in the server before restoring the backup. Ensure the files moved are named as "tlsKeyStore" and "tlsTrustStore" before proceeding.

Step 4 Download the NDDB 3.10.5 software from the standard Cisco.com Downloads page and and start the new NDDB 3.10.5 installation using the **./runndb.sh -start** command.

Step 5 Log in to the new instance of the NDDB GUI using https://server IP:8443.

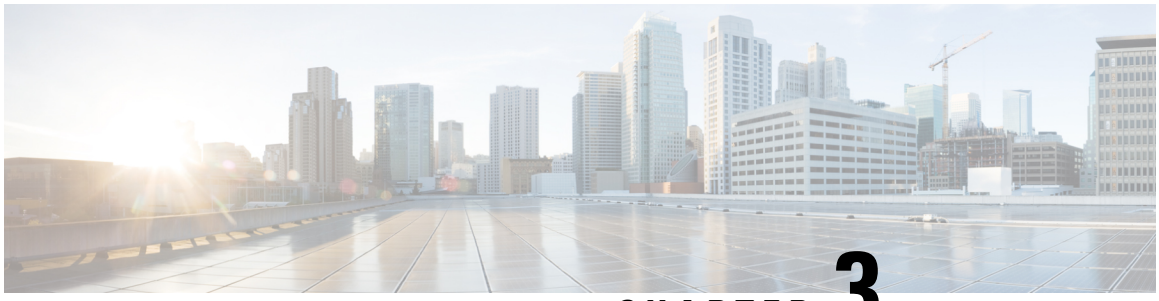
Step 6 Navigate to the **Administration** > **Backup/ Restore** tab.

Step 7 Click **Actions** > **Restore Locally** to upload the configuration file which you had earlier downloaded.

Step 8 On the **Upload Configuration** window, drag and drop the file that you had earlier backed up and click **Restore**.

The TLS files will be automatically synced to other nodes during *restore* for high availability.

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



CHAPTER 3

Deploying Cisco Nexus Dashboard Data Broker Software in Clusters

Beginning with Release 3.10.1, Cisco Nexus Data Broker (NDB) has been renamed to Cisco Nexus Dashboard Data Broker. However, some instances of NDB are present in this document, to correspond with the GUI, and installation folder structure. References of NDB/ Nexus Data Broker/ Nexus Dashboard Data Broker, can be used interchangeably.

This chapter contains the following details:

Installing a Cisco Nexus Dashboard Data Broker Cluster

Use this procedure to install a Cisco Nexus Dashboard Data Broker (NDDB) cluster.

Before you begin

Prerequisites:

- Cisco Nexus Dashboard Data Broker (NDDB) supports 3-node clusters.
- All IP addresses must be reachable and capable of communicating with each other.
- All switches in the cluster must connect to all the controllers.
- All controllers must have the same HA clustering configuration information in the `config.ini` files.
- All the NDDB instances should be of the same NDDB version to form the cluster.
- If using cluster passwords, all controllers must have the same password configured in the **ndbjgroups.xml** file. See *Password Protecting for HA Clusters* section in the *Cisco Nexus Dashboard Data Broker Configuration Guide.*

Note All the NDDB instances to form the cluster should be of the same NDDB version.

Procedure

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

Step 2 Scroll down and click **Downloads**.

Step 3 In the **Select a Product** search box, enter *Nexus Dashboard Data Broker* and you are automatically taken to the latest release **Software Download** screen.

The file information for Release 3.10.5 is displayed: Cisco Nexus Data Broker Software Application: `ndb1000-sw-app-k9-3.10.5.zip`

Step 4 Download the Cisco Nexus Data Broker application bundle. If prompted, enter your Cisco.com username and password to login.

Step 5 Create a directory in your Linux machine where you plan to install the Data Broker.

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

Step 6 Copy the Cisco Nexus Dashboard Data Broker zip file to the created NDDB directory.

Step 7 Unzip the Data Broker zip file.

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

- runndb.sh file—file to launch NDDB.
- version.properties file—NDDB build version.
- configuration directory—contains the NDDB initialization files. This directory also contains the startup subdirectory where configurations are saved.
- bin directory—contains the NDDB file that has the common CLI.
- etc directory—contains profile information.
- lib directory—contains NDDB Java libraries.
- logs directory—contains NDDB logs.

  Note
  The logs directory is created after the NDDB application is started.

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

Step 8 Navigate to the ndb/configuration directory that was created when you installed the software.

Step 9 Use any text editor to open the `config.ini` file and 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>

If a standby node is available:
#supernodes=<ip1>;<ip2>;<ip3>;<ip4>-standby
```

Step 10 Uncomment the line which consists of supernodes and replace <ip*> with NDDB server IPs.

```
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

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 11 Save the file and exit the editor.

Step 12 Repeat steps 5 to 11 in all the Linux machines where the NDDB is installed.

Step 13 Start the primary NDDB server using the **./runndb.sh -start** command.

Step 14 After the GUI of the primary NDDB server is up, start the other NDDB servers using the **./runndb.sh -start** command.

After the primary is up, await the confirmation messsage displayed on the GUI before starting the members of the cluster. The displayed message reads, *Primary is ready, bring up the members*.

Upgrading a Cisco Nexus Dashboard Data Broker Cluster

Before you begin

Prerequisites:

- Cisco Nexus Dashboard Data Broker (NDDB) supports 3-node clusters.
- All IP addresses must be reachable and capable of communicating with each other.
- All switches in the cluster must connect to all the controllers.
- All controllers must have the same HA clustering configuration information in the `config.ini` files.
- All the Nexus Dashboard Data Broker instances should be of the same Nexus Dashboard Data Broker version to form the cluster.
- If using cluster passwords, all controllers must have the same password configured in the **ndbjgroups.xml** file. See *Password Protecting for HA Clusters* section in the *Cisco Nexus Data Broker Configuration Guide.*

Note All the Nexus Dashboard Data Broker instances to form the cluster should be of the same Cisco Nexus Dashboard Data Broker version.

Procedure

Step 1 Login to the Cisco Nexus Dashboard Data Broker primary server.

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

Step 3 Click **Backup Locally** to download the configuration file.

Step 4 Stop all Cisco Nexus Dashboard Data Broker instances using the **runndb.sh -stop** command.

Step 5 If TLS certification is enabled between NDDB server and NDDB Devices, take backup of the tlsTrustStore and tlsKeyStore files from `/ndb/configuration`.

Step 6 Perform the previous step on all the NDDB cluster instances.

Step 7 In a web browser, navigate to www.cisco.com.

Step 8 Navigate to **Support** > **Products** > **Downloads**.

Step 9 In the Find Products and Downloads search box, enter "Nexus Data Broker" and click on 'Downloads' from search response list.

The file information for Release 3.10.5 is displayed: Cisco Nexus Dashboard Data Broker Software Application: `ndb1000-sw-app-k9-3.10.5.zip`

Step 10 Download the Cisco Nexus Dashboard Data Broker application bundle. When prompted, enter your Cisco.com username and password to login.

Step 11 Create a directory in your Linux machine where you plan to install Cisco Nexus Dashboard Data Broker.

For example, in your Home directory, create Cisco Nexus Dashboard Data Broker.

Step 12 Copy the Cisco Nexus Dashboard Data Broker zip file to the directory that you have created.

Step 13 Unzip the Cisco Nexus Dashboard Data Broker zip file.

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

- runndb.sh file—file to launch NDDB.
- version.properties file—NDDB build version.
- configuration directory—contains the NDDB initialization files. This directory also contains the startup subdirectory where configurations are saved.
- bin directory—contains the NDDB file that has the NDDB common CLI.
- etc directory—contains profile information.
- lib directory—contains NDDB Java libraries.
- logs directory—contains NDDB logs.

  Note
  The logs directory is created after the NDDB application is started.

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

Step 14 Navigate to the ndb/configuration directory that was created when you installed the software.

Step 15 Use any text editor to open the `config.ini` file and locate the following text:

Step 16 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>
```

```
If a standby node is available:
#supernodes=<ip1>;<ip2>;<ip3>;<ip4>-standby
```

Step 17 Uncomment the line which consists of supernodes and replace <ip*> with NDDB Server IPs.

```
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

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 18 Save the file and exit the editor.

Step 19 Repeat the steps 7 to 18 in all the Linux machines where the NDDB is installed.

Step 20 Start the Primary NDDB server using the **./runndb.sh -start** command.

Step 21 After the GUI of the Primary NDDB Server is up, start the other NDDB servers using the **./runndb.sh -start** command.

Step 22 Login to the Primary Server NDDB GUI.

Step 23 (applicable for the 3.9.x release train) If the TLS configuration is available, move the TrustStore and KeyStore files to the configuration folder.

Important
If you are upgrading from 3.9.2, your backup file contains the TLS files. The `tlsconf`, `tlsKeyStore`, `tlsTrustStore` files are part of your backup.

If you are upgrading from release 3.10.x or 3.9.0, your backup file will not contain the `tlsTrustStore`, `tlsKeyStore` files. You will need to manually upload the tls files to the /ndb/configuration folder in the server before restoring the backup. Ensure the files moved are named as "tlsKeyStore" and "tlsTrustStore" before proceeding.

Step 24 Navigate to **Administration** > **Backup/Restore** > **Actions** > **Restore Locally** and upload the configuration you had earlier downloaded.

Step 25 Stop all instances of NDDB in the cluster using the **./runndb.sh -stop** command.

Step 26 Start the primary NDDB server using the **./runndb.sh -start** command.

After the primary is up, await the confirmation messsage displayed on the GUI before starting the members of the cluster. The displayed message reads, *Primary is ready, bring up the members*.

After the primary is *up*, the TLS files for the member servers(s) are synced by the NDDB controller.

Step 27 Start the member NDDB server(s) using the **./runndb.sh -start** command.

Upgrading the Application Software with TLS-enabled for HA-clustered Controller

Use this procedure for upgrading the Nexus Dashboard Data Broker (NDDB) application software in centralized mode, using the GUI, when the TLS certification is enabled in a HA-clustered controller.

Procedure

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

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

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

Step 4 Stop the current NDDB instance(s) using the **./runndb.sh -stop** command.

Step 5 After the NDDB instances are stopped, navigate to the `/ndb/configuration` folder, and copy the tlsTrustStore and tlsKeyStore files to `local/common` folder.

Step 6 Download the NDDB 3.10.5 software from the standard *Cisco.com Downloads* page and configure the cluster mode using the "supernodes" configuration in the `config.ini` file and start the new NDDB 3.10.5 cluster using the **./runndb.sh -start** command on all the controllers.

Step 7 (applicable for the 3.9.x release train) If the TLS configuration is available, move the TrustStore and KeyStore files to the configuration folder.

Important
If you are upgrading from 3.9.2, your backup file contains the TLS files. The `tlsconf`, `tlsKeyStore`, `tlsTrustStore` files are part of your backup.

If you are upgrading from release 3.10.x or 3.9.0, your backup file will not contain the `tlsTrustStore`, `tlsKeyStore` files. You will need to manually upload the tls files to the /ndb/configuration folder in the server before restoring the backup. Ensure the files moved are named as "tlsKeyStore" and "tlsTrustStore" before proceeding.

Step 8 On the primary controller, navigate to the **Administration** > **Backup/ Restore** tab.

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

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

Step 10 Stop the NDDB 3.10.5 instances on all the controllers using the **./runndb.sh -stop** command.

Step 11 Start the primary NDDB server using the **./runndb.sh -start** command.

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

After the primary is *up*, the TLS files for the member servers(s) are synced by the NDDB controller.

Step 12 Start the NDDB instance on the other controllers of the cluster using the **./runndb.sh start** command.