



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](#) , on page 1
- [Upgrading a Cisco Nexus Dashboard Data Broker Cluster](#), on page 3

Installing a Cisco Nexus Dashboard Data Broker Cluster

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

Before you begin

Prerequisites:

- Cisco Nexus Dashboard Data Broker (NDDDB) supports 2-node and 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 NDDDB instances should be of the same NDDDB 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 NDDDB instances to form the cluster should be of the same NDDDB version.

- 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 3.10.3 is displayed: Cisco Nexus Data Broker Software Application:
 ndb1000-sw-app-k9-3.10.3.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 Cisco Nexus Data Broker.

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

- Step 6** Copy the Cisco Nexus Dashboard Data Broker zip file to the created NDDDB directory.

- Step 7** Unzip the Cisco Nexus Data Broker zip file.

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

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

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

- `plugins` directory—The directory that contains the NDDDB 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>;<ipn>
```

- Step 10** Uncomment the line which consists of supernodes and replace `<ip*>` with NDDDB 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 NDDDB is installed.
- Step 13** Start the Primary NDDDB server using the `./runndb.sh -start` command.
- Step 14** After the GUI of the primary NDDDB server is up, start the other NDDDB servers using the `./runndb.sh -start` command.

Upgrading a Cisco Nexus Dashboard Data Broker Cluster

Before you begin

Prerequisites:

- Cisco Nexus Dashboard Data Broker (NDDDB) supports 2-node and 3-node clusters. 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.
- 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.

- 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 `runxnc.xh -stop` command.
- Step 5** If TLS certification is enabled between NDDDB server and NDDDB Devices, take backup of the `tlsTrustStore` and `tlsKeyStore` files from `/xnc/configuration`.
- Step 6** Perform the previous step on all the NDDDB 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 3.10.3 is displayed: Cisco Nexus Dashboard Data Broker Software Application:
 ndb1000-sw-app-k9-3.10.3.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>;<ipn>
```

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** Navigate to **Administration > Backup/Restore > Actions > Restore Locally** and upload the configuration you had earlier downloaded.
- Step 24** Stop all instances of NDDB in the cluster using the `./runndb.sh -stop` command.
- Step 25** If TLS certification is enabled between NDDB server and NDDB switches, copy the `tls TrustStore` and `tlSKeyStore` files to `ndb/configuration` taken from the step 5 for all NDDB instances.
- Step 26** Start the primary NDDB server using the `./runndb.sh -start` command.
- If TLS certification is enabled, use below commands to start NDDB servers.

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

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

If TLS certification is enabled, use below commands to start NDDB servers.

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