



## Configuring DCNM Native High Availability

---

This chapter describes the DCNM Native High Availability (HA) configuration and troubleshooting. This chapter contains the following sections:

- [DCNM HA Overview, page 6-1](#)
- [DCNM Native HA Installation, page 6-1](#)
- [DCNM License Usage and Limitations, page 6-2](#)
- [Native HA Failover and Split-Brain, page 6-2](#)
- [Disk File Replication, page 6-2](#)
- [Replace HA Hosts, page 6-2](#)
- [DCNM Native HA with Scaled Up Test, page 6-3](#)
- [AAA Configuration, page 6-3](#)
- [Troubleshooting DCNM Native HA, page 6-3](#)

### DCNM HA Overview

DCNM Native HA provides a high availability solution for the DCNM. It consists of two DCNM nodes in which one node assumes the role of the active node and the other node assumes the role of the standby node.

The native HA is supported on Linux platform with ISO and OVA installation. For standalone installation, we will not support native HA as there might be missing Linux packages which are required for native HA. Native HA is also not supported on Windows platform.

By default, DCNM is bundled with an embedded database engine PostgreSQL. The DCNM native HA is achieved by two DCNM's running as Active / Warm Standby, with their embedded databases synchronized in real time. So once the active DCNM is down, the standby will take over with the same database data and resume the operation.

### DCNM Native HA Installation

For detailed DCNM native HA setup process, please refer to Cisco DCNM Installation Guide, Release 10.0(x).

## DCNM License Usage and Limitations

Cisco DCNM license is tied to host Mac Address. In DCNM native HA setup, there are two hosts with different Mac addresses. Here is how it works:

In DCNM native HA, only primary DCNM (node 1) is allowed to load license, the secondary (node 2) can only apply the licenses. This is similar to DCNM Federation where DCNM with Id 0 could load licenses, all others can only apply the licenses.



### Note

DCNM recommends having licenses on one instance and a spare matching license on the second instance.

## Native HA Failover and Split-Brain

DCNM failover can be manually triggered, or if the standby DCNM detected the active DCNM is not responsive, the standby will then takeover and act as active.

In DCNM native HA, the VIP(s) are always associated with active host. When failover occurs, the active host shall disassociate the VIP(s) and shutdown the DCNM process; and the standby shall associate the VIP(s) with the host, change the database from stream replication mode to normal mode, and start up the DCNM process.

Split-Brain syndrome occurs when the communication on enhanced fabric interface between two HA peer is lost. As the result, both hosts will act as Active. When the communication resumes, both hosts shall negotiate and eventually one will become active, the other standby.

## Disk File Replication

In addition to database real-time synchronization between two DCNM HA peers, there are also bunch of disk files which need to be replicated.

The disk files which need replication include POAP templates, performance data (RRD files), etc.

## Replace HA Hosts

If you need to replace an HA host machine, please follow the procedures:



### Note

The IP addresses or VIPs are assumed not to be changed.  
Hosts that having "Deployed role: Standby" can only be replaced.

- Step 1** Stop the DCNM on the standby host (no IP change).
- Step 2** Stop the DCNM on the active host (no IP change).
- Step 3** Take backup of Standby DCNM.
- Step 4** Take a local copy of ha-properties file from **/root/-packaged-files/properties/ path.**

- Step 5** On the new host which is supposed to replace the old host, configure the IP addresses on eth0 and eth1 to be identical to the old host.
- Step 6** If the host is a virtual machine, configure the mac address to be identical to the old host, so there will be no need to get new licenses for the new host.
- Step 7** On the new host which will join the HA setup, run the HA setup script, just like in the normal HA setup procedure.
- Step 8** Restart the DCNM on the active host, then restart the DCNM on the standby host.
- 

## DCNM Native HA with Scaled Up Test

Different HA scale limits have been mentioned under DCNM 10.x release. Please refer to Cisco DCNM Release Notes, Release 10 for scale requirement and scale limits.

## AAA Configuration

For AAA configuration, you need to install Cisco DCNM native HA with local user credentials. Once the installation is done, please log into the DCNM web client and go to **Administrator > Management Users > Remote AAA** and select the required authentication mode.

**Note**

When doing remote AAA authentication, Cisco DCNM is sending out request using its own eth0 IP rather than VIP. Therefore, on the AAA server, we need to put two entries for DCNM IP, one for active DCNM, the other for standby IP, but not VIP.

---

## Troubleshooting DCNM Native HA

When Cisco DCNM native HA setup is in an uncertain situation, stop both hosts and resolve the problem. Start only one host and ensure that it is fully functional, and the device data is correct before you bring up the second host as standby.

**Note**

Throughout this Troubleshooting procedure, **dcnm1** is considered as the Active host and **dcnm2** is considered for Secondary host.

---

This contains the following sections:

- [“Recovering DCNM when both hosts are Powered Down” section on page 6-4](#)
- [“Recovering from Split-Brain syndrome” section on page 6-4](#)
- [“Checking Cisco DCNM Native HA Status” section on page 6-5](#)
- [“Verifying if the Active and Standby Hosts are Operational” section on page 6-6](#)
- [“Verifying HA Database Synchronization” section on page 6-7](#)
- [“Resolving HA Status Failure condition” section on page 6-7](#)
- [“Bringing up Database on Standby Host” section on page 6-7](#)

## Recovering DCNM when both hosts are Powered Down

Perform the following to troubleshoot the DCNM Native HA setup when both the hosts are powered down.

- 
- Step 1** Power on **dcnm1**.
- Step 2** Wait for all the applications to be operational.  
Use the **appmgr status all** command to check the status of the applications.  
`dcnm1# appmgr status all`
- Logon to DCNM. Verify if it is fully functional. Check if the device data is correct. If success, power on **dcnm2** as Secondary host. Terminate the troubleshooting procedure.
- Step 3** If the host fails to bring up all the applications, or if the device data is incorrect, use the **appmgr stop all** command to stop the process.  
Wait for all the applications to stop.
- Step 4** Power on **dcnm2**.  
Wait for all the applications to be operational.
- Step 5** Use the **appmgr status all** command to check the status of the applications.  
`dcnm2# appmgr status all`
- Logon to DCNM. Verify if it is fully functional. Check if the device data is correct. If success, power on **dcnm1** as Secondary host. Terminate the troubleshooting procedure.
- Step 6** If **dcnm2** fails to bring up all the applications, or if the device data is incorrect, use the **appmgr stop all** command to stop the process.
- Step 7** Restore both hosts from backup.
- 

## Recovering from Split-Brain syndrome

Perform the following to recover Cisco DCNM from the split brain syndrome.

- 
- Step 1** Stop both Active and Standby Cisco DCNM hosts.  
Use the **appmgr stop all** command, to stop the applications  
`dcnm1# appmgr stop all`  
`dcnm2# appmgr stop all`
- Step 2** Wait for all the applications to stop.  
Use the **appmgr status all** command to check the status of the applications.  
`dcnm1# appmgr status all`  
`dcnm2# appmgr status all`
- Resolve the communication problem between two hosts which causes the Split-Brain Syndrome.
- Step 3** Ping the peer host eth1 IP address from both hosts and make sure it is reachable.
- Step 4** Start all the applications on **dcnm1**. Wait for all the applications to be operational.

Use the **appmgr status all** command to check the status of the applications.

```
dcnm1# appmgr status all
```

**Step 5** Logon to **dcnm1** and verify if it is fully functional and if all the data is correct.

If all the data is correct, proceed to [Step 7](#).

If data loss is seen, proceed to [Step 6](#).

**Step 6** Use the **appmgr stop all** command, to stop the applications

```
dcnm1# appmgr stop all
```

**Step 7** Start all the applications on **dcnm2**. Wait for all the applications to be operational.

Use the **appmgr status all** command to check the status of the applications.

```
dcnm2# appmgr status all
```

**Step 8** Logon to DCNM. Verify if it is fully functional.

Check if the device data is correct. If success, power on **dcnm1** as Secondary host. Terminate the troubleshooting procedure.

**Step 9** If data loss is seen on **dcnm2**, stop all the applications.

Use the **appmgr stop all** command, to stop the applications

```
dcnm2# appmgr stop all
```

**Step 10** Restore both hosts from backup.

## Checking Cisco DCNM Native HA Status

Perform the following to determine the status of the Cisco DCNM Native HA.

**Step 1** Login into Cisco DCNM Web Client.

**Step 2** Navigate to **Web Client > Administration > Native HA**.

**Step 3** Check for HA Status.

The status of the Native HA and their description is as depicted in the table below.

HA Status	Description
OK	Implies that the Native HA is operational. Both the hosts on the Native HA are synchronized.
Stopped	Implies that the Standby host is not operation. The database is not synchronized.
Failed	Implies that the Active host is unable to synchronize with the Standby host. Check the log files for more information. The log file is located at: <code>/usr/local/cisco/dcm/fm/logs/fms_ha.log</code>
Not Ready	Implies that the Standby host is not setup or not configured.

## Verifying if the Active and Standby Hosts are Operational

Perform the following to determine if the hosts are operational.

**Step 1** Check the HA role on the host.

**Step 2** Use the **appmgr show ha-role** command to view the current role of the host.

```
dcnm1# show ha-role
Active
```

```
dcnm2# show ha-role
Standby
```

**Step 3** Check the VIP, using the **ip address** command.

On the Active host, both eth0 and eth1 must have two IP addresses configured, with VIP assigned as the secondary IP address; on standby host, only one IP address for both eth0 and eth1 interfaces

**Step 4** Check the DCNM java process.

Use **ps -ef | grep java** command to verify the java process associated with the DCNM.

```
dcnm1# ps -ef | grep java
```

The results must show one Java process, appended with **standalone-san.xml**.

```
dcnm2# ps -ef | grep java
```

There should no be any Java process, appended with **standalone-san.xml**.

**Step 5** Check the heartbeat of the DCNM hosts.

```
dcnm1# /etc/init.d/heartbeat status
heartbeat OK
```

```
dcnm2# /etc/init.d/heartbeat status
heartbeat OK
```

**Step 6** Check if the database engine PostgreSQL is operational.

```
dcnm1# /etc/init.d/postgresql-9.4 status
server is running .....
```

```
dcnm2# /etc/init.d/postgresql-9.4 status
server is running .....
```

**Step 7** Check the HA cluster information.

```
dcnm1# cl_status listnodes
dcnm2# cl_status listnodes
```

The two hostnames of the HA cluster will be displayed.

**Step 8** Check the HA heartbeat status.

```
dcnm1# cl_status nodestatus <hostname>
dcnm2# cl_status nodestatus <hostname>
```

If this command returns “active”, the heartbeat on the host is OK.

If the command returns “dead”, the heartbeat on the host is not running or not recognized.

## Verifying HA Database Synchronization

Perform the following to verify if the databases synchronization on both hosts is in progress.

When running DCNM Native HA, both the host database must be operational, one host as Active and another host as Standby. Any changes made in the Active database must synchronize with the Standby database in real time.

To verify if the database is synchronizing, use `ps -ef | grep post` command.

```
dcnm1# ps -ef | grep post
postgres: wal sender process postgres 172.23.244.222(40826) streaming 0/9A846C04
```

```
dcnm2# ps -ef | grep post
postgres: wal receiver process streaming 0/9A84E00
```

## Resolving HA Status Failure condition

Perform the following to resolve if the HA status check results in failure.

- 
- Step 1** Logon to Cisco DCNM Web UI.
- Step 2** Navigate to **Administration > Native HA**.  
Click the **Test** icon.  
Check if there are errors. Click **Detailed Logs** for more information.
- Step 3** Check log file at the location.  
`/usr/local/cisco/dcm/fm/logs/fms_ha.log`  
There should be some log messages indicating why the HA status is Failed.
- Step 4** Verify if Standby host is running is operational.  
See [Verifying if the Active and Standby Hosts are Operational, page 6-6](#), for more information. Check is any applications are not operational.  
Generally, the HA status shows Failed due to Standby database being down or rejected connection.  
If the connection to standby database is rejected, the HA status shows as Failed. Check the file located at:  
`/usr/local/cisco/dcm/db/data/pg_hba.conf`  
The configuration file must contain entries for all IP addresses listed on active host **ip address**.  
If not, we recommend that you contact the Technical Support for further assistance.
- Step 5** If Standby database is completely down, see [Bringing up Database on Standby Host, page 6-7](#).
- 

## Bringing up Database on Standby Host

Normally, the database must be running on both Active or Standby host, regardless of DCNM being operational or stopped. However, the database could be down mostly because of the initial database synchronization failure.

Perform the following to bring up the database on the Standby host.

---

**Step 1** Start the Standby database, using the `/etc/init.d/postgresql-9.4 start` command.

**Step 2** If the return value is PostgreSQL 9.4 started successfully, the Standby database is OK. The HA status shows OK within a few minutes.

If the database is not started successfully, the database files may be corrupted. This condition occurs due to initial synchronization failure. In such a condition, navigate to the located at:

```
/usr/local/cisco/dcm/db/replication
```

**Step 3** Check for the file `pgsql-standby-backup.tgz`.

If the file exists, perform the following to restore database files, and start database again:.

- a. Enter the `ps -ef | grep post` command and ensure that the Postgres process is not running.
- b. If the Postgres process is running, stop by using the `kill <pid>` command.
- c. Remove all the database files by using the following commands:

```
cd /usr/local/cisco/dcm/db  
rm -rf data/*
```

- d. Restore the database files from the backup by using `tar xzf replication/pgsql-standby-backup.tgz data` command.
  - e. Restart the database by using the `/etc/init.d/postgresql-9.4 start` command.  
Check if the database has started successfully.
-