Deployment and Management Guide for Cisco IoT FND on OVA with Postgres and InfluxDB

Cisco IoT Field Network Director
Updated September 22, 2026

Topics included

1 Cisco IoT FND deployments on OVA with Postgres and Influx DB....4

Cisco IoT FND OVA deployments....5

2 OVA Images and Upgrade Scripts....6

Cisco IoT FND deployment packages....7

Verify the OVA signature....9

Verify RPM signatures....9

Export OVF files....12

3 OVA Deployments Using a Virtualization Software....14

Install OVA using VMware ESXi server....15

Configure Hostnames....17

Configure NTP Server and IP Host Entries on HER Devices....18

4 Keys and Keystore....20

Default keystore values and certificate management....21

Build the cgms_keystore file....22

Encrypt the cgms_keystore password....24

Use keytool to create the cgms_keystore file....25

Custom browser certificates....26

Upload a custom web certificate....37

Properties of Cisco IoT FND and TPS configuration....39

TPSPROXY properties....41

5 Configure Cisco IoT FND on OVA....42

Configure IPv6 tunnel provisioning and registration....43

Start and stop Cisco IoT FND on OVA....44

Start and stop Fog Director....45

Obtain statuses of services running on the host....46

Configure time and timezone using NTP service....47

Configure NTP services....48

6 Upgrade Cisco IoT FND on OVA....50

Device compatibility check....51

Check the Postgres DB status....51

Cisco IoT FND on OVA upgrade matrix....54

Verify and reset the expired root password 55

Upgrade the database and docker server image....56
Upgrade Cisco IoT FND container images....61
Check DB and status post upgrade....64
Upgrade RHEL version....67

7 HA and DR....68
HA and DR for Cisco IoT FND with Postgres....69
Prerequisites....70
Restrictions....70
HA and DR architecture....71
Configure HA and DR....72
Configure HA and DR on DC1 FND....75
Configure HA and DR on DC2 FND....78
Database and Cisco IoT FND Migration....80
Migrate Influx DB to Postgres....80
Verify the data in the Postgres DB....85
Migrate data from standalone to EDB Postgres....87
Monitor servers....90
Configure self-signed certificates....90
Example Influx DB to Postgres migration log....91
Troubleshoot stale server entries....97
Troubleshoot CPU and memory issues....97

1 Cisco IoT FND deployments on OVA with Postgres and Influx DB

Topics:

- Cisco IoT FND OVA deployments

This chapter helps you understand Cisco IoT FND deployments on OVA with Postgres and Influx DB, including the key components and attributes of this deployment model.

Cisco IoT FND OVA deployments

Cisco IoT FND OVA deployments simplify installation, enhance scalability, and provide robust data handling and monitoring capabilities for Cisco IoT networks.

Cisco IoT FND is a management platform designed to facilitate the deployment, monitoring, and management of Cisco IoT networks. Deploying Cisco IoT FND on OVA with Postgres and Influx DB offers several advantages:

- easy deployment: Using an OVA simplifies the Cisco IoT FND installation process with pre-configured settings, reducing complexity and setup time.
- scalability: OVA-based deployments make Cisco IoT FND more scalable and capable of supporting more Cisco IoT devices.
- data handling and monitoring: Postgres ensures reliable storage and retrieval of structured data, while Influx DB enables efficient processing and analysis of real-time time-series data streams.

Virtualization Platforms and Installation Overview

Virtualization platforms allow you to create and manage VMs on a physical hardware system, enabling multiple operating systems to run simultaneously by abstracting hardware resources. Examples include VMware vSphere, VMware Workstation, Oracle VirtualBox, and Microsoft Hyper-v.

Table 1: Comparison of OVA and Bare Metal Deployments

Aspect	Cisco IoT FND Bare Metal Deployment (Custom Office Building)	Cisco IoT FND OVA Deployment (Serviced Office Space)
Setup	Constructing a custom office building from the ground up.	Moving into a pre-furnished serviced office space.
Customization and Control	Complete control over design, layout, and infrastructure.	Limited customization; pre-configured and ready to use.
Resource Investment	Significant upfront investment in time and resources.	Lower initial costs with minimal setup required.
Performance and Optimization	Can be finely tuned for optimal performance.	Offers decent performance, with flexibility for scaling.
Ease and Speed	Requires extensive planning and time to set up.	Rapid installation and deployment.
Scalability and Flexibility	Adjustments require more effort and time.	Easily scalable and flexible to adjust resources.
Cost Efficiency	Higher costs due to custom setup and maintenance.	Cost-effective with reduced need for hardware investment.

2 OVA Images and Upgrade Scripts

Topics:

- Cisco IoT FND deployment packages
- Verify the OVA signature
- Verify RPM signatures
- Export OVF files

This chapter explains how OVA images and upgrade scripts are used to deploy and update Cisco IoT Field Network Director. It provides an overview of the available images and scripts, as well as guidelines for their application during system installation and upgrades.

Cisco IoT FND deployment packages

This topic explains the purpose and contents of Cisco OVA images and upgrade scripts, detailing the files included in each .zip package and how they support secure installation and upgrade of Cisco IoT FND solutions.

You must verify the integrity of OVA images and upgrade scripts before you install or upgrade Cisco IoT FND. This verification ensures that the files are authentic and untampered.

Contents of OVA images and upgrade scripts

Table 2: Contents of the .zip files

Zip File Name	Contents
CISCO-IOTFND-VPI-K9-<release>-<build number>.zip	Contains Postgres/Influx for gateway management. • iot-fnd-<release>-<build number>.iso • iot-fnd-<release>-<build number>.iso.signature

Zip File Name	Contents
CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip **Note** The CGMS tools file is bundled with CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip.	Includes cgms tools rpm for Postgres deployments. • cgms-tools-<release>-<build number>.x86_64.rpm • FND_RPM_SIGN-CCO_RELEASE.pem – Cisco signed x.509 end-entity certificate containing public key that is used to verify the signature. This certificate is chained to Cisco root CA and sub CA posted on https://www.cisco.com/security/pki/ . • cisco_openpgp_verify_release.py – Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate. • cisco_openpgp_verify_release.py.signature – Signature generated for the script cisco_openpgp_verify_release.py. • FND-rel-binary.gpg – Open-pgp public key is used for verification of signed RPM. • FND-rel-ascii.gpg – Open-pgp public key is used for verification of signed RPM.
CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip	Includes upgrade scripts for upgrading FND-Postgres/Influx OVA. • upgrade-ova-<release>-<build number>.rpm – Signature embedded RPM image. • FND_RPM_SIGN-CCO_RELEASE.pem – Cisco signed x.509 end-entity certificate containing public key that is used to verify the signature. This certificate is chained to Cisco root CA and sub CA posted on https://www.cisco.com/security/pki/ . • cisco_openpgp_verify_release.py – Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate. • cisco_openpgp_verify_release.py.signature – Signature generated for the script cisco_openpgp_verify_release.py. • FND-rel-binary.gpg – Open-pgp public key is used for verification of signed RPM. • FND-rel-ascii.gpg – Open-pgp public key is used for verification of signed RPM.

Verify the OVA signature

Use this procedure to verify the OVA signature for your deployment, ensuring the integrity and authenticity of the OVA file before installation. This process helps confirm that the file has not been tampered with and is safe to use.

Use the instructions provided in this section to verify the OVA signature:

Here are the contents of the OVA zip file for different use cases:

Use case	Contents
Oracle-based deployments	**iot-fnd-oracle-<release>-<build number>_SHA1_signed.ova**
TPS-related deployments	**iot-tps-<release>-<build number>_SHA1_signed.ova**

Before you begin

The following knowledge is required to verify OVA signature:

- Python 2.7.x
- OpenSSL
- Internet connection

Procedure

Use the `ovftool` command on your terminal to verify the signed OVA file.

Example

```
ovftool iot-fnd-<release>-<build number>_SHA256_signed.ova
```

This command checks the integrity and authenticity of the OVA file by verifying its signature.

```
Opening OVA source: iot-fnd-<release>-<build number>_SHA256_signed.ova
        The manifest validates
        Completed successfully
```

What to do next

Verify the upgrade RPM signature

Verify RPM signatures

Use this procedure to verify the integrity and authenticity of RPM packages before installation or upgrade. These steps help ensure your files are secure and have not been tampered with prior to deployment.

The following steps collectively ensure the security and integrity of the files before proceeding with any installations or upgrades.

Before you begin

The following knowledge is required to verify OVA signature:

- Python 2.7.x
- OpenSSL
- Internet connection

Procedure

1. Unzip the file **iot-fnd-<release>-<build number>-signed.zip** and change directory to the folder **iot-fnd-<release>-<build number>-signed**

   Example

   ```
   unzip iot-fnd-<release>-<build number>-signed.zip
               cd iot-fnd-<release>-<build number>-signed
   ```

2. Extract the public key from the public certificate:

   Example

   ```
   openssl x509 -pubkey -noout -in FND_RPM_SIGN-CCO_RELEASE.pem >
   FND-EE-cert.pubkey
   ```

   ```
   FND-EE-cert.pubkey is created under the same folder
   ```

3. Verify the scripts using the public key and signature files:

   Example

   ```
   openssl dgst -sha512 -verify FND-EE-cert.pubkey -signature
   cisco_x509_verify_release.py.signature cisco_x509_verify_release.py
               openssl dgst -sha512 -verify FND-EE-cert.pubkey -signature
   cisco_openpgp_verify_release.py.signature cisco_openpgp_verify_release.py
   ```

   Expected Result:

   ```
   Verified OK
   ```

4. Verify the ISO file:

   Example

   ```
   ./cisco_x509_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -s
   iot-fnd-<release>-<build number>.iso.signature -i iot-fnd-<release>-<build
   number>.iso -v dgst -sha512
   ```

   ```
   Retrieving CA certificate from
   http://www.cisco.com/security/pki/certs/crcam2.cer ...
               Successfully retrieved and verified crcam2.cer.
               Retrieving SubCA certificate from
   ```

```
http://www.cisco.com/security/pki/certs/innerspace.cer ...
            Successfully retrieved and verified innerspace.cer.
            Successfully verified root, subca and end-entity certificate
chain.
            Successfully fetched a public key from
FND_RPM_SIGN-CCO_RELEASE.pem.
           Successfully verified the signature of iot-fnd-<release>-<build
number>.iso using FND_RPM_SIGN-CCO_RELEASE.pem
```

5. Install the ISO image file:

   Example

```
cd /mnt
            mkdir iso
            mount -t iso9660 -o loop <path>/iot-fnd-<release>-<build
number>.iso /mnt/iso
            mkdir /tmp/ISO
            cp -pRf /mnt/iso /tmp/ISO
            umount /mnt/iso/
```

6. Verify the delivered binary and ascii keys have matching fingerprints:

   Example

```
gpg FND-rel-binary.gpg
```

```
gpg FND-rel-ascii.gpg
```

```
pub 2048R/F7D5ED29 2017-01-01 user (FND.rel)
```

7. Verify the binary GPG key against EE cert:

   Example

```
./cisco_openpgp_verify_release.py -e FND_RPM_SIGN-CCO_RELEASE.pem -G
FND-rel-binary.gpg
```

```
Downloading CA certificate from
http://www.cisco.com/security/pki/certs/crcam2.cer ...
            Successfully downloaded crcam2.cer.
            Downloading SubCA certificate from
http://www.cisco.com/security/pki/certs/innerspace.cer ...
            Successfully downloaded innerspace.cer.
          Successfully verified Cisco root, subca and end-entity certificate
 chain.
            Successfully fetched a public key from
FND_RPM_SIGN-CCO_RELEASE.pem.
            Successfully authenticated FND-rel-binary.gpg key using Cisco
X.509 certificate trust chain.
```

8. Verify RPM Signature using the GPG ASCII key:

Example

```
sudo rpm --import FND-rel-ascii.gpg
            rpm -K /tmp/ISO/iso/cgms-<release>-<build number>.x86_64.rpm
```

```
 /tmp/ISO/iso/cgms-<release>-<build number>.x86_64.rpm: rsa sha1 (md5) pgp
md5 OK
```

9. Repeat the step 8 to verify all the RPM signatures. For example, CGMS tools RPM for Postgres Signature.

You've successfully verified the RPM signatures.

What to do next

Proceed to installing or upgrading the RPMs.

Export OVF files

Use this procedure to export OVF files from a virtual machine or vApp using VMware vSphere. This process allows you to create a portable package of the virtual machine or vApp for backup, migration, or deployment purposes.

Before you begin

An OVF template captures the state of a virtual machine or vApp into a self-contained package. The disk files are stored in a compressed sparse format.

Procedure

To export an OVF file using VMware vSphere see, Export an OVF Template .

---

3 OVA Deployments Using a Virtualization Software

Topics:

- Install OVA using VMware ESXi server
- Configure Hostnames
- Configure NTP Server and IP Host Entries on HER Devices

This chapter explains what OVA files are and describes their key characteristics and applications in virtualization environments. OVA files are used for distributing virtual machines as a single package.

Install OVA using VMware ESXi server

Use this procedure to install an OVA file on a VMware ESXi server. This task guides you through preparing your environment, deploying the OVA, and performing the initial configuration to set up Cisco IoT FND on your virtualized platform

Before you begin

This procedure focuses on installing the OVA using VMware ESXi server. You can use any virtualization software of your choice.

- Log in to the IP address of a VMware ESXi server running 6.5 and above via a web browser with your user credentials.
- Before deploying Cisco IoT FND, ensure that ESXi host and datastore have sufficient capacity for the FND virtual machine.
  - 24 GB memory
  - 4 vCPUs
  - Hard disk: 540 GB

Note

The VM resource requirements for non-HA and HA deployments are:

Deployment scale	vCPUs	Memory	Virtual disk space
Up to 10,000 routers	10	32 GB	600 GB
10,000 to 50,000 routers	24 (Cores per socket: 4 and sockets: 6)	96 GB	800 GB

- Use the following credentials for SSH access after installing OVA
  - Username: fnduser
  - Password: C!sco123

Procedure

1. Use the steps provided in Broadcom's documentation to Deploy a Virtual Machine From an OVF or OVA File in the VMware Host Client .

> **Note**
>
> - After choosing the data store, choose the provisioning type and enable the **Power on automatically** option. This ensures to power on the virtual machine once the deployment process is complete. Click **Next** .
> - Thick Provisioning – Absolute reservation on the disk space. For the Cisco IoT FND OVA deployment, the disk space required is 600 GB on the ESXi server.
>
>   For deployments supporting up to 10,000 routers, allocate a 600-GB virtual disk. For deployments supporting 10,000 to 50,000 routers, allocate an 800-GB virtual disk.
> - Thin Provisioning – The disk space grows on demand. For the Cisco IoT FND OVA deployment, the disk space is approximately 50 GB initially and the disk space occupied by VM will grow as per the scale of deployment.
>
>   For a 50,000-router deployment, ensure that the datastore can ultimately support the full 800-GB virtual disk plus operational overhead.
> - If the selected storage location doesn't have sufficient storage for the largest file installation option, a message displays noting insufficient storage. If the warning message appears, select another storage resource with greater capacity and click **Next** .

2. The deployed VM gets listed in the left pane. Choose the IoT FND machine name.
3. Click **Console** and login with root/cisco123 once the OS is up.
4. Once you enter the default password, you are prompted to reset your password.
   - The default root user password is C!sco123.
   - The following conditions are applicable to reset the default password:
     - The password must be at least 8 characters in length
     - The password must have at least 1 uppercase character
     - The password must have at least 1 lowercase character
     - The password must have at least 1 special character
     - The password must have at least 1 digit
     - The password cannot be the same as any of the previous 5 passwords used

   After you complete the password reset, Cisco IoT FND is fully deployed.
5. Assign a static IP address and a DNS server on a Red Hat Enterprise Linux system for managing and configuring the Cisco IoT FND.

   > **Note**
   >
   > Follow the same steps for TPS OVA installation as well. In order to upgrade the TPS OVA, delete the existing TPS and reinstall the TPS OVA `iot-tps-version_number.ova` with the updated version number.

6. Open a terminal window, and set up Health Monitoring for the Fog Director Container from FND.

Example

```
[root@iot-fnd ~]# cd /opt/monitor/
            [root@iot-fnd monitor]# .setup.sh
```

```
Starting setup...
            Configuring environment...
            Setup completed successfully.
```

Cisco IoT FND starts monitoring the Fog Director container on the **ADMIN** > **SERVERS** page.

What to do next

Configure hostnames in Cisco IoT FND and TPS servers.

Configure Hostnames

Use this task to configure domain names in the /etc/hosts file for enabling Zero Touch Deployment (ZTD) and Plug and Play (PNP) of FAR devices with Cisco IoT FND and TPS servers. This ensures hostname resolution and proper communication within the IoT network environment.

When onboarding Cisco IoT FND, configure domain names in the **/etc/hosts** file. Use this configuration for the Zero Touch Deployment (ZTD) and Plug and Play (PNP) of FAR devices. The domain name resolves the FND or TPS hostname to an IP address, facilitating communication with Cisco IoT FND.

Use the following instructions to configure hostnames in FND and TPS servers:

Procedure

1. Edit hostname in the **/etc/hosts file** .

   Example

   ```
   root@iot-fnd ~]# nano /etc/hosts
   ```

   This displays the /etc/hosts file, allowing you to edit it by adding or updating hostnames with their corresponding IP addresses.

2. Enter the domain name for the host IP address separated by a space or a tab.

   Example

   ```
   209.165.200.225  fnd.iot.cisco.com
               209.165.201.1  tps.iot.cisco.com
   ```

3. Save the changes by selecting **Y** and exit.
4. Validate the updated hostname using the following command:

   Example

   ```
   [root@iot-fnd-oracle ~]# ping fnd.iot.cisco.com
               64 bytes from fnd.iot.cisco.com (209.165.200.225): icmp_seq=1
   ```

```
ttl=64 time=0.052 ms
            64 bytes from fnd.iot.cisco.com (209.165.200.225): icmp_seq=2
ttl=64 time=0.058 ms
            64 bytes from fnd.iot.cisco.com (209.165.200.225): icmp_seq=3
ttl=64 time=0.048 ms
```

The updated hostname is displayed along with the corresponding IP address.

5. To view the hostnames with the corresponding IP addresses, use the following command:

Example

```
[root@iot-fnd ~]# cat /etc/hosts
            127.0.0.1  iot-fnd-oracle localhost localhost.localdomain
localhost4 localhost4.localdomain4
            ::1        iot-fnd-oracle localhost localhost.localdomain
localhost6 localhost6.localdomain6
            209.165.200.225  fnd.iot.cisco.com
            209.165.201.1  tps.iot.cisco.com
```

You've successfully configured hostnames on Cisco IoT FND and TPS servers.

What to do next

Configure NTP servers and IP host entries.

Configure NTP Server and IP Host Entries on HER Devices

Use this task to configure the NTP server and IP host entries on a HER device to ensure accurate time synchronization and proper hostname resolution within your network.

Follow these instructions to configure the NTP server and IP host entries on HER devices:

Procedure

---

1. Access the HER device using the SSH terminal
2. Enter the global configuration mode

Example

```
config t
```

3. Use the `ntp server` command to specify the IP address or hostname of the NTP server.

Example

```
ntp server 192.168.1.100
            ntp server ntp.example.com
```

4. Exit the configuration.

Example

```
config t
            ntp server 192.168.1.100
            exit
```

5. To remove an existing IP host entry, use the no form of the `ip host` command:

   Example

   ```
   no ip host caserver.fnd.iot.com 192.168.1.100
   ```

6. Add the new IP address for the hostname:

   Example

   ```
   ip host caserver.fnd.iot.com 192.168.1.100
   ```

7. Exit the configuration.

   Example

   ```
   config t
               ip host fndserver.fndiot.com 192.168.1.100
               exit
   ```

You've successfully configured the NTP servers and IP host entires.

4 Keys and Keystore

Topics:

- Default keystore values and certificate management
- Properties of Cisco IoT FND and TPS configuration

This chapter provides instructions for managing security certificates in Cisco IoT FND, including the creation, encryption, and password management of the cgms_keystore file. It also details the installation of custom browser certificates and the steps required to retain them during system upgrades.

Default keystore values and certificate management

This topic explains how Cisco IoT FND OVA stores cryptographic keys and certificates in a keystore for secure device authentication and data encryption. This content outlines default keystore locations, passwords, and key management best practices.

The Cisco IoT FND OVA includes keys and certificates stored in a keystore. Keys are cryptographic components used for encrypting and decrypting data, while certificates are digital documents that verify the authenticity of a device or entity within the network.

The following table lists the default values for the keys and certificates in different environments:

Table 3: Default keystore values

Environment	Keystore Location	Keystore Name	Default Password
Cisco IoT FND OVA Linux Host	`/opt/fnd/data/`	`cgms_keystore.selfsigned`	N/A
Cisco IoT FND container	`/opt/cgms/server/cgms/conf/`	`cgms_keystore`	`Public123!`

Note

When you restart the Cisco IoT FND container, the values of `/opt/cgms/server/cgms/conf/cgms_keystore` file in Cisco IoT FND container are overwritten by `/opt/fnd/data/cgms_keystore` file. If `/opt/fnd/data/cgms_keystore` file is not present in the host, then the `/opt/fnd/data/cgms_keystore.selfsigned` file is used.

When you install Cisco IoT FND OVA for the first time, each certificate/key entry is referenced by an alias name in the keystore. The default aliases are:

- cisco_sudi (cisco root CA certificate with 2029 expiry)
- jmarconi (cisco certificate)
- cgms (self signed certificate that is used by Cisco IoT FND when communicating with devices it has to manage)

The cgms certificate in the `/opt/cgms/server/cgms/conf/cgms_keystore` file on the Cisco IoT FND container and the `/opt/fnd/data/cgms_keystore.selfsigned` file on the Linux host initially contain a self-signed certificate from Cisco IoT FND. To create a custom cgms_keystore at `/opt/fnd/data` on the Linux host, you can either copy the existing `cgms_keystore.selfsigned` file or build it from scratch. The `/opt/fnd/data/cgms_keystore` file takes precedence if both files are present.

Note

- Ensure NTP is synchronized across the issuing Certificate Authority (CA) server, Cisco IoT FND, TPS, and FAR/HER, as it's crucial for Public Key Infrastructure.
- Change the hostname or IP address of the Cisco IoT FND host before issuing the certificate and building the cgms_keystore.
- The SAN field in the Cisco IoT FND certificate must include the hostname. If it includes the IP address, reissue the certificate if the hostname or IP changes.

The following table lists the default certificates and keys in the cgms_keystore:

Table 4: Default certificates and keys in the cgms_keystore

Certificate/Key	Description
Issuing CA Certificate	The certificate of the organization's issuing CA server, which can be a root or intermediate CA. If intermediate, import both root and intermediate CA certificates into the keystore.
Cisco IoT FND Device Certificate	Issued for Cisco IoT FND by the organization's issuing CA server.
Cisco SUDI with 2029 Expiry	The Cisco manufacturer certificate for Cisco IoT FND, issued by Cisco, expiring in 2029.
Cisco SUDI with 2099 Expiry	The Cisco manufacturer certificate for Cisco IoT FND, issued by Cisco, expiring in 2099.

Build the cgms_keystore file

This procedure provides instructions to build the cgms_keystore file from scratch, including required certificates and keys.

You can build a custom cgms_keystore file if you don't want to copy the existing self-signed keystore file. To build a custom cgms_keystore file from scratch, follow these steps:

Procedure

1. Change directory to /opt/fnd/data on linux host.

   Example

   ```
   # cd /opt/fnd/data
   ```

2. To create the keystore file with the keytool command, ensure it is named **cgms_keystore** , as Cisco IoT FND requires this specific name. First, copy your organization issuing CA certificate to a location on the server, such as /root/rootca.pem, using a file transfer method like SCP. The certificate can be in .cer, .crt, or .pem format. In this example, since the issuing CA is the root CA, use the alias name "root."

Example

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore -alias

            root -file /root/rootca.pem
```

Convert the keystore from jks to pkcs12

```
# keytool -importkeystore -srckeystore /opt/fnd/data/cgms_keystore
            -destkeystore /opt/fnd/data/cgms_keystore -deststoretype pkcs12
```

Verify that the file has been created by listing the contents of the keystore.

```
# keytool -list -keystore /opt/fnd/data/cgms_keystore
```

3. Import a Cisco IoT FND certificate into your system. Use the following command to create a key pair and a Certificate Signing Request (CSR). The key pair is like a lock and key for your system, and the CSR is a request to get a digital certificate that matches this key.

   Example

   ```
   keytool -genkeypair -keyalg RSA -keysize 2048 -alias cgms -ext
   "SAN=dns.labfnd.cisco.com, ip:1.0.0.1" -keystore /opt/fnd/data/cgms_keystore
    -dname "CN=labfnd, OU=iotescblr, O=cisco, L=Bengaluru, ST=Karnataka, C=IN"
   ```

   Note

   The key size here is set to 2048, but you can use 4096 for extra security.

4. Use the following command to create the CSR file required by the CA to issue your certificate.

   Example

   ```
   keytool -certreq -file labfnd.csr -keystore /opt/fnd/data/cgms_keystore -alias
    cgms -ext "SAN=dns:labfnd.cisco.com,ip:1.0.0.1"
   ```

5. Send this CSR file to the CA. They will give you back a certificate for your server.
6. Once you receive the certificate, for example, as a file named `labfnd.pem` , copy it to your server.
7. Use the following command to import the certificates:

   Example

   ```
   keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore -alias
   cgms -file /root/labfnd.pem
   ```

8. If your Cisco IoT FND certificate is in a .pfx format, you need to know the alias name used in your .pfx file before importing it. Use the following command to find the alias name:

Example

```
keytool -list -v -keystore /opt/fnd/data/nms.pfx -srcstoretype pkcs12 | grep
 Alias
```

9. Once you have the alias name from the .pfx file, you need to import it into your keystore using the alias cgms.

Example

```
keytool -importkeystore -v -srckeystore /opt/fnd/data/nms.pfx -destkeystore
/opt/fnd/data/cgms_keystore -srcalias le-IoT
FND-8f0908aa-dc8d-4101-a526-93b4eaad9481 -destalias cgms
```

You have built the custom cgms_keystore file.

What to do next

Import SUDI with 2029 and 2000 expiries.

Encrypt the cgms_keystore password

This topic covers information on how to encrypt and update the **cgms_keystore** password in the `cgms.properties` file for Cisco IoT FND application access.

Ensure that the `cgms.properties` file contains the encrypted password for `cgms_keystore` so the Cisco IoT FND application can access it. When you create the `cgms_keystore` for the first time, encrypt its password and include this encrypted password in the `cgms.properties` file. If you ever change the password for `cgms_keystore`, encrypt the new password and update it in the `cgms.properties` file.

Use the following instructions to encrypt and set a new password:

Procedure

---

1. Run the following command to encrypt the password for the new **cgms_keystore** .

Example

```
# docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh
            encrypt <keystore password>
            # docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh

            encrypt cisco123
            #2bVvZsq+vsq94YxuAKdaag--
```

2. Modify the **cgms.properties** file in the /opt/fnd/data folder, and edit the following line to set the new encrypted **cgms_keystore** password:

Example

```
cgms-keystore-password-hidden=<encrypted new cgms_keystore password>
```

You have now successfully encrypted and generated a new cgms_keystore password.

Use keytool to create the cgms_keystore file

This section guides you through the steps involved in creating the cgms_keystore file for both Cisco IoT FND and the TPS proxy.

Use the instructions provided in this section to create the cgms_keystore file for both Cisco IoT FND and the TPS proxy.

Procedure

1. Enter the following command on the Cisco IoT FND or TPS proxy server as root to view the contents of the .PFX file.

   Example

   ```
   [root@ tps_server ~]# keytool -list -v -keystore nms_cert.pfx -srcstoretype
   pkcs12
   ```

   > **Note**
   >
   > You can get the alias name from the .pfx file during import.

2. Enter the keystore password when prompted. The password is the same as the one you used to create the .pfx file.
3. Enter the following command to import the certificates into the cgms_keystore file:

   Example

   ```
   keytool -importkeystore -v -srckeystore
               filename.pfx -srcstoretype pkcs12
             -destkeystore cgms_keystore -deststoretype jks -srcalias alias_name

               -destalias cgms
               -destkeypass
               keystore_password
   ```

4. Enter the destination keystore password when prompted.
5. Re-enter the keystore password when prompted.
6. Enter the password you used earlier while creating the .pfx file (either *nms_cert.pfx* or *tps_cert.pfx*) when prompted for the source keystore password.

   Example

   ```
   # keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
               Enter keystore password: keystore
               Keystore type: PKCS12
               Keystone provider: SunJSSE
               Your keystore contains 1 entry
               Alias name: le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b
               Creation date: March 29.2018
               Entry type: PrivateKey Entry
               Certificate chain length: 3
               Certificate[1]:
               Owner: CN=nms.sgbu.cisco.com
               Issuer: CN=cisco-RSA-SUBCA-CA, DC=cisco, DC=com
   ```

> **Note**
>
> This example diplays the steps for the *nms_cert . pfx* . To view the details on the *tps_cert.pfx* and import the certificates to the TPS proxy, use the same commands but replace the references to *nms_cert.pfx* with *tps_cert.pfx* , and use the alias name from the *tps_cert.pfx* file.

Example

```
# keytool -importkeystore -v
            -srckeystore nms_cert.pfx -srcstoretype pkcs12
            -destkeystore cgms_keystore -deststoretype jks -srcalias
            le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b -destalias cgms
            -destkeypass
            keystore_password
            Enter destination keystore password: keystore_password
            Re-enter new password: keystore_password
            Enter source keystore password: keystore
            ...Storing cgms_keystore
```

The **storing** *cgms* **_keystore** text indicates successful completion.

Custom browser certificates

This section introduces custom browser certificates and briefly explains the stages involved in installing the custom browser certificates.

Custom web certificates are digital certificates that are specifically created and configured for a particular organization or application rather than using standard, off-the-shelf certificates.

By default, Cisco IoT FND installations use a self-signed certificate for HTTP(S) communication with either a client web browser or the NB API client. If necessary, you can use certificates signed by your CA servers, or install custom browser certificates.

To install custom browser certificates using CLI, you must perform the steps in the following two procedures:

- Prepare and import parameters for custom web certificates on page 27
- Configure and finalize custom web certificates on page 29

Table 5: Feature history

Feature name	Release information	Description
UI-based web certificate upload	Cisco IoT FND Release 26.2.1	Enables web certificate uploads directly through the Cisco IoT FND user interface.

To import a custom browser certificate using options in the Cisco IoT FND interface, refer to the following sections:

- Upload a custom web certificate on page 37
- Restore a web certificate

Prepare and import parameters for custom web certificates

This section guides through preparing your environment and importing custom browser certificates into Cisco IoT FND. It covers prerequisite browser steps, secure file handling, and keystore creation using the provided .PFX certificate file.

Prepare and import the parameters required to install custom browser certificates, which enable secure HTTP(S) communication between Cisco IoT FND and client browsers or NB API clients using CA-signed certificates.

This section provides the intial sequence of installation procedures for these custom certificates, which includes preparing and importing the certificates.

The following instructions help you prepare to install custom certificates in the browser client. These instructions assume that the Java environment has been set to use the Java bundled with Cisco IoT FND in /opt/cgms/jre.

> **Note**
>
> Update the jbossas.keystore in Cisco IoT FND container. The keystore is present in: /opt/cgms/server/cgms/conf/. The name of the keystore is jbossas.keystore.

Before you begin

- Clear the client browser cache.
- Remove existing certificates for the NMS server (by IP and DNS) on the client browser

  In Firefox, for example, select **Preferences** > **Advanced** > **Encryption** > **View Certifications**. Remove the certificates in the list for the respective server.
- Choose a common name to use in the signed certificate.

  This name requires a DNS entry that resolves to the NMS server IP address.
- Generate the new certificates and export them to a.PFX file.

  This file must contain the private keys, public certificate, and CA server certificates.

  See Use Keytool to Create the cgms_keystore File for the procedure to generate the private and public keys for the cgms_keystore file and export them to a .PFX file.

Procedure

1. Copy the certificate and private key *.pfx file from the host to container.

   Example

   ```
   docker cp newcert.pfx fnd-container:/opt/cgms/server/cgms/conf/
   ```

2. Run the following command to enter the container:

   Example

   ```
   docker exec -i -t fnd-container /bin/bash
   ```

3. Change directory to /opt/cgms/server/cgms/conf/ in the container:

Example

```
cd /opt/cgms/server/cgms/conf/
```

4. On the NMS server, copy the existing jbossas.keystore, jbossas.keystore.password, vault.keystore, and VAULT.dat files from the /opt/cgms/server/cgms/conf/ directory to a safe location.
5. Delete the existing jbossas.keystore, jbossas.keystore.password, vault.keystore, and VAULT.dat files from the /opt/cgms/server/cgms/conf/ directory.
6. Determine the alias in the .PFX file that you plan to import into the new jbossas.keystore file:

Example

```
#keytool -list -v -keystore newcert.pfx -storetype pkcs12
            Enter the keystore password:
keystore_password_when_pfx_file_was_created
            Keystore type: PKCS12
            Keystore provider: SunJSSE
            Your keystore contains 1 entry
            Alias name: le-nms-a88ef13a-a519-457f-a2e1-0540f5453ee0
            Creation date: Feb 23, 2018
            Entry type: PrivateKeyEntry
            Certificate chain length: 2
            Certificate[1]:
            …
```

7. Import the new custom certificate, in .PFX file format into a new jbossas.keystore file; and, at the same time change the alias name to **jboss**. Follow the prompts:

Example

```
# keytool -importkeystore -v -srckeystore newcert.pfx -srcstoretype pkcs12
            -destkeystore /opt/cgms/server/cgms/conf/jbossas.keystore
-deststoretype jks
           -srcalias le-nms-a88ef13a-a519-457f-a2e1-0540f5453ee0 -destalias

            jboss -destkeypass your_keystore_password
            Enter destination keystore password: your_keystore_password
            Enter source keystore password:
keystore_password_when_pfx_file_was_created
            [Storing /opt/cgms/server/cgms/conf/jbossas.keystore]
```

8. The keystore password is stored in the /opt/cgms/server/cgms/conf/VAULT.dat file. Perform the following steps to update the keystore password to match the keystore password.
9. Create a new vault.keystore file:

Example

```
/opt/cgms/jre/bin/keytool -genseckey -alias vault -storetype jceks
            -keyalg AES -keysize 256 -storepass keystore -dname "CN=IoTFND,
OU=IoT, O=Cisco Systems,
          L=San Jose, ST=CA, C=US" -keypass keystore -validity 730 -keystore
 vault.keystore
```

What to do next

Configure and finalize custom web certificates on page 29

Configure and finalize custom web certificates

This section guides you through configuring security for the custom web ceritificates and finalizing them.

Configure the keystore security, update necessary configuration files, and restart services to complete the installation of custom browser certificates in Cisco IoT FND. These help ensure that your system uses the new certificates for secure HTTPS connections and guides you through final verification steps.

This section provides the final sequence of installation procedures for these custom certificates, which includes configuring security and finalizing the certificates.

Procedure

1. Update the VAULT.dat file with the new password:

Example

```
/opt/cgms/bin/vault.sh --keystore /opt/cgms/server/cgms/conf/vault.keystore
            --keystore-password keystore --alias vault --vault-block
keystore_pass
            --attribute password --sec-attr keystore --enc-dir
/opt/cgms/server/cgms/conf/
            --iteration 50 --salt 12345678 -n
```

```
================================================================

            JBoss Vault

            JBOSS_HOME: /opt/cgms

            JAVA: /opt/cgms/jre/bin/java


========================================================================

            WFLYSEC0047: Secured attribute value has been stored in Vault.
            Please make note of the following:
            ********************************************
            Vault Block:keystore_pass
            Attribute Name:password
            Configuration should be done as follows:
            VAULT::keystore_pass::password::1
            ********************************************
            WFLYSEC0048: Vault Configuration commands in WildFly for CLI:
            ********************************************
            For standalone mode:
            /core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
            "/opt/cgms/server/cgms/conf/vault.keystore"),
            ("KEYSTORE_PASSWORD" =>
"MASK-0VKsAwH928fwt.3H2qUwOG"),("KEYSTORE_ALIAS" => "vault"),
          ("SALT" => "12345678"),("ITERATION_COUNT" => "50"),("ENC_FILE_DIR"
 =>
            "/opt/cgms/server/cgms/conf/")])
            ********************************************
            For domain mode:

/host=the_host/core-service=vault:add(vault-options=[("KEYSTORE_URL" =>
```

```
            "/opt/cgms/server/cgms/conf/vault.keystore"),
            ("KEYSTORE_PASSWORD" =>
"MASK-0VKsAwH928fwt.3H2qUwOG"),("KEYSTORE_ALIAS" => "vault"),
          ("SALT" => "12345678"),("ITERATION_COUNT" => "50"),("ENC_FILE_DIR"
 =>
            "/opt/cgms/server/cgms/conf/")])
            ****************************************
```

2. Copy /opt/cgms/standalone/configuration/standalone.xml and standalone-cluster.xml to a safe location.
3. Update the /opt/cgms/standalone/configuration/standalone.xml or standalone-cluster.xml file. Depending on whether the Cisco IoT FND server is standalone or clustered, update the respective file accordingly.
4. Replace the keystore password:

   Example

```
<vault>
            <vault-option name="KEYSTORE_URL"
            value="/opt/cgms/server/cgms/conf/vault.keystore"/>
            <vault-option name="KEYSTORE_PASSWORD"
value="MASK-VKsAwH928fwt.3H2qUwOG"/><
            <vault-option name="KEYSTORE_ALIAS" value="vault"/>
            <vault-option name="SALT" value="12345678"/>
            <vault-option name="ITERATION_COUNT" value="50"/>
            <vault-option name="ENC_FILE_DIR"
value="/opt/cgms/server/cgms/conf/"/>
            </vault>
```

5. Edit the standalone.xml or standalone-cluster.xml and replace the existing <vault> section with the above. Save and exit.
6. Run the following command to restart the CGMS service:

   Example

```
/etc/init.d/cgms restart
```

7. Use your browser to connect to the NMS server.
8. Accept and add the new certificates.
9. Use your browser to log in to Cisco IoT FND.

When you restart the container, it replaces the custom web certificates with self-signed certificates. It is recommended to restart or upgrade the container before taking a backup of the following files to the local host:

- jbossas.keystore
- jbossas.keystore.password
- VAULT.dat
- vault.keystore
- standalone.xml
- standalone-cluster.xml

Manage custom web certificates

This procedure covers the steps to back up custom web certificate files before rebooting Cisco IoT FND to prevent loss of certificate details.

Because the web certificate details are not retained after a reboot of Cisco IOT FND, you must follow these steps to perform a backup of the following files before you attempt to reboot.

Procedure

1. Use SSH or direct console access to log into the server where Cisco IoT FND is running.
2. Open a terminal and navigate to the `/opt/cgms/server/cgms/conf/` directory.

   Example

   ```
   cd /opt/cgms/server/cgms/conf/
   ```

3. Decide on a location for your backup files. It could be another directory on the same server or an external storage location.

   Example

   ```
   mkdir -p /path/to/backup_directory
   ```

4. Use the cp command to copy each file to the backup directory. Replace `/path/to/backup_directory` with your actual backup directory path:

   Example

   ```
   cp jbossas.keystore.password /path/to/backup_directory/
              cp jbossas.keystore /path/to/backup_directory/
              cp VAULT.dat /path/to/backup_directory/
              cp vault.keystore /path/to/backup_directory/
              cp standalone.xml /path/to/backup_directory/
              cp cgms.conf /path/to/backup_directory/
   ```

5. Ensure all files are copied correctly by listing the contents of the backup directory:

   Example

   ```
   ls -l /path/to/backup_directory/
   ```

6. Reboot the server for the changes to take effect.

What to do next

Install Custom Browser Certificates.

Install custom certificates in the browser client for Cisco IoT FND release 5.1 and later

Install custom certificates in the browser client for Cisco IoT FND Release 5.1 and later by following these step-by-step instructions to securely configure keystore and vault files. This process ensures secure connectivity between your browser and the Cisco IoT FND system using your chosen certificates.

This task helps you install custom certificates for a fresh install of Cisco IoT FND running Cisco IoT FND Release 5.1 and later releases.

Procedure

1. Copy the certificate and private key *.pfx file from the host to the container:

   Example

   ```
   docker cp custom_ca_cert.pfx fnd-container:/opt/cgms/server/cgms/conf/
   ```

   a) Run `docker exec -i -t fnd-container /bin/bash` to enter the container.
   b) Change the directory to `cd /opt/cgms/server/cgms/conf/`
   c) On the NMS server, copy the existing `jbossas.keystore`, `jbossas.keystore.password`, `vault.keystore`, and `VAULT.dat files` from the `/opt/cgms/server/cgms/conf/` directory to a safe location.
2. Determine the alias in the .PFX file that you plan to import into the new jbossas.keystore file:

   Example

   ```
   #keytool -list -v -keystore custom_ca_cert.pfx -storetype pkcs12
   ```

3. Enter the keystore password that you set when .PFX file was created

   Example

   ```
   Enter the keystore password: keystore_password_when_pfx_file_was_created

   Keystore type: PKCS12
   Keystore provider: SunJSSE
   Your keystore contains 1 entry
   Alias name: lab5-win-2mfnersdbr3-ca
   Creation date: Feb 23, 2025
   Entry type: PrivateKeyEntry
   Certificate chain length: 2
   Certificate[1]:
   …
   ```

   a) Delete the existing jbossas.keystore from the /`opt/cgms/server/cgms/conf/` directory.
   b) Import the new custom certificate, in .PFX file format into a new jbossas.keystore file; and, at the same time change the alias name to jboss. Follow the prompts:

      Example

      ```
      keytool -importkeystore -v -srckeystore custom_ca_cert.pfx -srcstoretype
      pkcs12
      -destkeystore /opt/cgms/server/cgms/conf/jbossas.keystore -deststoretype
      jks
      -srcalias Alias name -destalias jboss -destkeypass your_keystore_password
      ```

      The keystore password is stored in the `/opt/cgms/server/cgms/conf/VAULT.dat` file.

4. Delete the following files:
   a) Delete the **`vault.keystore`** file.

b) Delete the **vault.dat** file.

c) Create a new **vault.keystore** file:

Example

```
/opt/cgms/jre/bin/keytool -genseckey -alias vault -storetype jceks
-keyalg AES -keysize 256 -storepass your_keystore_password -dname
your_domain_name -keypass -validity 730 -keystore vault.keystore
```

d) Update the vault.dat file with the new password:

Example

```
/opt/cgms/bin/vault.sh --keystore /opt/cgms/server/cgms/conf/vault.keystore

 --keystore-password your_keystore_password --alias vault --vault-block
keystore_pass
--attribute password --sec-attr your_keystore_password --enc-dir
/opt/cgms/server/cgms/conf/ --iteration 50 --salt 12345678 -n
```

The vault.keystore contains the reference to vault.dat and vault.dat stores and hides the jboss keystore password. This command creates a new vault.dat file that contains the new jboss.keytsore password.

e) Copy `/opt/cgms/standalone/configuration/standalone.xml` to a safe location.

5. Update the `/opt/cgms/standalone/configuration/standalone.xml`.

a) Replace the keystore password:

Example

```
<vault>
<vault-option name="KEYSTORE_URL"
value="/opt/cgms/server/cgms/conf/vault.keystore"/>
<vault-option name="KEYSTORE_PASSWORD" value="MASK-0VKsAwH928fwt.3H2qUwOG"/>
<vault-option name="KEYSTORE_ALIAS" value="vault"/>
<vault-option name="SALT" value="12345678"/>
<vault-option name="ITERATION_COUNT" value="50"/>
<vault-option name="ENC_FILE_DIR" value="/opt/cgms/server/cgms/conf/"/>
</vault>
```

b) Edit `standalone.xml` and replace the existing section with the above.

c) Restart the CGMS service using `/etc/init.d/cgms/restart`.

d) Save and exit.

6. Perform the following steps to retain the certificates upon container reloads.

a) Copy the files from `opt/cgms/server/cgms/conf/` to `/opt/fnd/data`

> **Note**
>
> If you see any of the following files in `/opt/fnd/data`, delete the files and then copy the files from the container using the following commands:

Example

```
docker cp fnd-container: /opt/cgms/server/cgms/conf/jbossas.keystore
/opt/fnd/data
docker cp fnd-container:
/opt/cgms/server/cgms/conf/jbossas.keystore.password /opt/fnd/data
docker cp fnd-container: /opt/cgms/server/cgms/conf/vault.dat /opt/fnd/data
docker cp fnd-container: /opt/cgms/server/cgms/conf/vault.keystore
/opt/fnd/data
docker cp fnd-container: /opt/cgms/standalone/configuration/stanalone.xml
 /opt/fnd/data
```

7. Navigate to `/opt/fnd/scripts` and run the **`script upgrade.sh`** and enter option 2 **`custom web certificates`**
8. Choose option 2 **`continer reload`** to reload the container
9. Use your browser to connect to the NMS server.
   a) Accept and add the new certificates.
   b) Login to Cisco IoT FND.
   c) Backup the following files in your local
      - jbossas.keystore
      - jbossas.keystore.password
      - VAULT.dat
      - vault.keystore
      - standalone.xml

You've succesfully installed the custom certificates in the browser client for Cisco IoT FND 5.1 and later releases.

Retain custom certificates during Cisco IoT FND 5.1.0 upgrade

This section explains how to retain custom web certificates when upgrading Cisco IoT FND. It provides step-by-step guidance to ensure your existing certificates are properly transferred and recognized during the upgrade process.

This task helps you retain custom certificates from an older Cisco IoT FND version to the latest Cisco IoT FND 5.1.0 version.

The `upgrade.sh` script needs the custom web certificate keystore (`jbossas.keystore`) in `/opt/fnd/data`. You must copy the `jbossas.keystore` to `/opt/fnd/data` so the OVA upgrade can locate and import the custom certificate.

Before you begin

- Ensure that you have custom certificates installed in your Cisco IoT FND. For more information see, Install custom browser certificates.

- Ensure that you are following the steps provided in the Upgrade Cisco IoT FND on OVA to perform an upgrade of your Cisco IoT FND.
- Ensure to login as a root user.

Follow these steps to retain custom web certificates during Cisco IoT FND 5.1.0 upgrade.

Procedure

1. Get the upgrade package from Cisco and follow the steps in Upgrade Cisco IoT FND on OVA to upgrade your Cisco IoT FND.
2. Run the RPM file.

   Example

   ```
   http://10.78.81.200/fnd-postgres-ova/5.1.0/152/archive/upgrade-ova-5.1.0-152.rpm
   rpm -Uvh --force upgrade-ova-5.1.0-152.rpm
   ```

   The RPM verifies, prepares, and begins updating/installing.

3. You change to `/opt/fnd/scripts/` and execute **`./upgrade.sh`**

   Example

   ```
   Select the Type of Web certificate currently used
   Option 1: FND Self Signed Certificate
   Option 2: Custom Web Certificate
   ```

4. Select **`Option 2: Custom Web Certificate`**

   Example

   ```
   Enter option (1 or 2): 2
   You selected Option 2 (Custom Web Certificate).
   ```

5. The script reports a missing custom certificate keystore:

   Example

   ```
   jbossas.keystore was not found in /opt/fnd/data.
   Copy jbossas.keystore from /tmp to /opt/fnd/data and re-run the upgrade.
   ```

6. Copy the `jbossas.keystore` from `/tmp/jbossas.keystore` to `/opt/fnd/data`

   Example

   ```
   cp /tmp/jbossas.keystore /opt/fnd/data
   ```

7. Re-run the **`./upgrade.sh`** script.

You've copied the custom certificates from a temporary directory to `/opt/fnd/data` which results in retaining the custom certificates to the upgraded Cisco IoT FND version.

Retain custom web certificate during Cisco IoT FND 26.2.1 upgrade

Preserve the five custom web certificate files that Cisco IoT FND Release 26.2.1 requires when you upgrade a deployment with a custom web certificate from Release 5.0.0 or Release 5.1.0.

This task helps you retain custom web certificate when upgrading Cisco IoT FND from an earlier version to version 26.2.1.

To upgrade to Cisco IoT FND Release 26.2.1, copy these files to `/opt/fnd/data` before running the upgrade RPM.

- `jbossas.keystore`
- `jbossas.keystore.password`
- `VAULT.dat`
- `vault.keystore`
- `standalone.xml`
- `standalone-cluster.xml` (for cluster setup)

Before you begin

- Ensure that you have custom web certificate installed in your Cisco IoT FND.

  For more information see, Install custom browser certificates.
- Ensure that you perform the steps in the Upgrade Cisco IoT FND on OVA procedure to upgrade your Cisco IoT FND.
- Log in as the root user.

Follow these steps to retain custom web certificate during Cisco IoT FND 26.2.1 upgrade.

Procedure

1. Copy the custom web certificate files from `fnd-container` to `/opt/fnd/data/` on the OVA host.

   Copy jbossas.keystore, jbossas.keystore.password, VAULT.dat, vault.keystore, and standalone.xml before you run the upgrade RPM.

   Example

   ```
   docker cp fnd-container:/opt/cgms/server/cgms/conf/jbossas.keystore
   /opt/fnd/data/
   docker cp fnd-container:/opt/cgms/server/cgms/conf/jbossas.keystore.password
    /opt/fnd/data/
   docker cp fnd-container:/opt/cgms/server/cgms/conf/VAULT.dat /opt/fnd/data/
   docker cp fnd-container:/opt/cgms/server/cgms/conf/vault.keystore
   /opt/fnd/data/
   docker cp fnd-container:/opt/cgms/standalone/configuration/standalone.xml
   /opt/fnd/data/
   docker cp
   fnd-container:/opt/cgms/standalone/configuration/standalone-cluster.xml
   /opt/fnd/data/
   ```

2. Obtain the Cisco IoT FND upgrade package and follow the standard Upgrade Cisco IoT FND on OVA procedure.
3. Run the Cisco IoT FND Release 26.2.1 upgrade RPM.

Example

```
root@iot-fnd:/opt[root@iot-fnd opt]# rpm -Uvh upgrade-ova-<release>-<build
number>.rpm
```

The RPM verifies, prepares, and begins updating/installing.

4. Change to `/opt/fnd/scripts/` and execute **`./upgrade.sh`**.

Example

```
Select the Type of Web certificate currently used
Option 1: FND Self Signed Certificate
Option 2: Custom Web Certificate
```

5. Select **Option 2: Custom Web Certificate**.

Example

```
Enter option (1 or 2): 2
You selected Option 2 (Custom Web Certificate).
```

6. If the script reports any missing certificate files, make sure the files are present in /opt/fnd/data/. Repeat step 1 to copy files from the container to the host, and rerun /opt/fnd/scripts/upgrade.sh.

The five custom web certificate files are available in /opt/fnd/data/ so that the upgraded Cisco IoT FND deployment can retain the custom web certificate configuration.

> **Note**
>
> To install a new custom web certificate or update an existing certificate from the Cisco IoT FND user interface, refer to Upload a custom web certificate on page 37.
>
> After you upload the certificate and restart the Cisco IoT FND service, repeat Step 1 of this task to copy the five required files from `fnd-container` to `/opt/fnd/data/`. This retains the uploaded certificate configuration after a container restart or reload.

Upload a custom web certificate

Describes how to upload a custom web certificate to the system.

To use web certificates signed by your certificate authority, you must install or upload one. This procedure presents steps to upload a custom web certificate.

This section provides essential details to help you successfully complete the task of uploading a custom web certificate.

Before you begin

Ensure that the following conditions are met before you attempt to upload a certificate:

- You must have a valid X.509 web certificate in `.pfx` format with its password.
- The web certificate `jbossas.keystore` must be present.

- `/opt/cgms/server/cgms/conf/` directory must exist.
- `/opt/cgms/standalone/configuration/` directory must exist.
- `/opt/cgms/jre/bin/keytool` executable must exist.
- `/opt/cgms/bin/vault.sh` must exist.
- `standalone.xml` file must exist.
- For cluster setup, the `standalone-cluster.xml` file must exist.
- Cisco IoT FND must have Shell/RHEL root access to back up, modify, and run scripts.

Procedure

1. Navigate to **ADMIN** > **Certificates** > **Certificate Management**.
2. Under **Web Certificate Upload**, click **Upload Web Certificate**.
3. Click **Browse** and choose the `.pfx` certificate.
4. Enter the certificate's password and click **Upload**.
5. Click **Yes** in the confirmation popup.

   Confirm the upload of the certificate using the **Upload Status** message.

   > **Note**
   >
   > Restart the Cisco IoT FND service for the newly uploaded certificate to come into effect.

6. After the Cisco IoT FND service restarts, copy the following files from the fnd-container container to /opt/fnd/data/ on the OVA host to retain the uploaded custom web certificate.

   Example

```
docker cp fnd-container:/opt/cgms/server/cgms/conf/jbossas.keystore
/opt/fnd/data/
docker cp fnd-container:/opt/cgms/server/cgms/conf/jbossas.keystore.password
 /opt/fnd/data/
docker cp fnd-container:/opt/cgms/server/cgms/conf/VAULT.dat /opt/fnd/data/
docker cp fnd-container:/opt/cgms/server/cgms/conf/vault.keystore
/opt/fnd/data/
docker cp fnd-container:/opt/cgms/standalone/configuration/standalone.xml
/opt/fnd/data/
docker cp
fnd-container:/opt/cgms/standalone/configuration/standalone-cluster.xml
/opt/fnd/data/
```

7. Navigate to the **Trusted Certificates** tab and verify the uploaded certificate and addition information such as validity, status, domain name, and so on.

You have imported your pre-signed web certificate to Cisco IoT FND.

What to do next

Check the newly imported web certificate under the **Certificate for Web** tab. To see a list of all the trusted certificates, navigate to the **Trusted Certificates** tab.

Properties of Cisco IoT FND and TPS configuration

This topic explains the configurable properties for Cisco IoT FND and TPS, detailing example values and descriptions to help optimize, secure, and manage gateway and telemetry data.

The Cisco Gateway Management System (CGMS) and TPSPROXY properties in Cisco IoT FND and TPS are used to customize, optimize, and secure the management and transmission of gateway and telemetry data, ensuring effective network operation and performance. This section aims to provide all the CGMS properties, with example values and descriptions.

CGMS properties

Property name	Example value	Description
cgms-keystore-password-hidden=	< encrypted >	Encrypted password for the cgms keystore. Encrypt or decrypt with encryption_util.sh.
hsm-keystore-name=	testGroup1	HSM partition name.
hsm-keystore-password=	< encrypted >	Encrypted HSM partition password.
security-module=	ssm/hsm	Type of security module being used.
ssm-host=	<ipv4 address >	IP Address of SSM server.
ssm-port=	8445	Port of SSM server.
ssm-keystore-alias=	ssm_csmp	Alias name for SSM certificate in keystore.
ssm-keystore-password=	< encrypted >	Encrypted password for the SSM keystore.
ssm-key-password=	< encrypted >	Encrypted key for the SSM key.
multicast-interface-address=	< ipv6 address >	Cisco IoT FND IPv6 source address for multicast traffic.
dhcpV4ClientListenAddresses=	<ipv4 address >	IPv4 address on your Cisco IoT FND server used to exchange DHCPv4 messages.
dhcpV6ClientListenAddresses=	< ipv6 address >	IPv6 address on your Cisco IoT FND server used to exchange DHCPv6 messages.
optimizeTunnelProv=	true/false	Indicates whether or not to lock the HER during tunnel provisioning.
allowed-outage-skew=	1000	Allow outage skew in milliseconds, for outage or restoration events. The default is `1000 ms` (1 second).

Property name	Example value	Description
rf.validate-firmware-tlvs=	true/false	Skips CG-Mesh device firmware validation.
googleMapsClientId=	< Client ID >	Google maps client ID.
googleMapsApiKey=	< API key >	Google maps API key.
enable-bootstrap-service=	true/false	Used to enable PNP bootstrapping service.
scep-url=	http(s) :// < url of SCEP server >	URL of SCEP server.
ca-fingerprint=	< fingerprint of CA certificate >	Fingerprint of CA certificate.
proxy-bootstrap-ip=	<ipv4/v6 address or FQDN >	PNP server identity sent by Cisco IoT FND to the PNP agent.
bootstrap-fnd-alias=	subca	Alias name assigned to the CA certificate from the issuer in the Cisco IoT FND keystore.
pnp-server-port=	9125	PNP server port, default is 9125.
pnp-install-trustpool=	true/false	Send the CA bundle file which includes well known public CA certificates.
reload-during-bootstrap=	true/false	Indicates whether or not to reload a device after PNP boostrapping.
router-file-upload-retries	0	Number of retries for router file upload job
router-firmware-upload-retries	0	Number of retries for the firmware upload job.
router-firmware-install-retries	0	Number of retries for the firmware install job.
collect-cellular-link-metrics	true/false	Indicates whether or not to collect cellular metrics.
collect-cellular-link-metrics-interval	30	Interval for cellular metrics.
router-firmware-upload-timeout-minutes=	30	Firmware upload job timeout duration in minutes.
router-firmware-install-timeout-minutes=	60	Firmware install job timeout duration in minutes.
cgr-ha-fetch-mesh-key-attempts	3	Number of attempts to fetch the mesh keys.
cgr-ha-fetch-mesh-key-delay-mins	1	Number of minutes or interval between mesh-key-attempts.

TPSPROXY properties

This section provides a detailed reference for TPSPROXY properties, including their names, example values, and descriptions. It outlines the configuration settings for source IP addresses, certificate subjects, listening ports, destination addresses, and allowed outbound addresses, essential for managing TPSProxy operations in Cisco environments.

This topic describes the available TPSPROXY properties and their usage.

Table 6: TPSPROXY properties

Property name	Example value	Description
cgdm-tpsproxy-addr=	<ipv4/v6 address or FQDN >	Source IP address of messages coming from the TPSProxy.
cgdm-tpsproxy-subject=	CN="common_name", OU="organizational_unit", O="organization", L="location", ST="state", C="country"	The exact certificate subject contained in the TPSPROXY's certificate.
bootstrap-proxy-listen-port=	9125	Port on which TPS is listening for HTTP traffic.
inbound-bsproxy-destination=	<ipv4/v6 address or FQDN >	IP address and port to forward info received from the router over HTTP.
outbound-proxy-allowed-addresses=	<ipv4/v6 address or FQDN >	Comma separated list of FQDN/IP addresses, the proxy allows outbound messages to originate from it.

5 Configure Cisco IoT FND on OVA

Topics:

- Configure IPv6 tunnel provisioning and registration
- Start and stop Cisco IoT FND on OVA
- Start and stop Fog Director
- Obtain statuses of services running on the host
- Configure time and timezone using NTP service
- Configure NTP services

This chapter provides the essential procedures for initializing and managing the FND OVA environment. It covers the configuration of IPv6 tunnel provisioning for secure device communication, the management of FND and Fog Director service containers, and the synchronization of system time via NTP to ensure data integrity and accurate logging across the network.

Configure IPv6 tunnel provisioning and registration

This section describes how to configure Cisco IoT FND for IPv6 tunnel provisioning and device registration. It guides you through enabling IPv6 support and ensuring proper network settings for successful connectivity.

To enable secure device communication and management over an IPv6 network, follow these steps to configure tunnel provisioning and registration in Cisco IoT FND. Completing this setup ensures that devices can be registered and managed using IPv6 connectivity.

> **Note**
>
> Cisco IoT FND OVA supports only IPv4 tunnels and registration out of the box.

Use the following steps to to set up an IPv6 network for tunnel provisioning and registration:

Procedure

1. Ensure you have one interface with a valid IPv6 network which has a IPv6 prefix length less than 125. Here's an example from the ens224 interface:

   Example

   ```
   [root@iot-fnd ~]# ifconfig ens224
               ens224: flags=4163[UP,BROADCAST,RUNNING,MULTICAST] mtu 1500
               inet 2.2.56.117 netmask 255.255.0.0 broadcast 2.2.255.255
               inet6 fe80::54f0:5d24:d320:8e38 prefixlen 64 scopeid 0x20[ink]
               inet6 2001:420:7bf:5f::1522 prefixlen 64 scopeid 0x0[global]
               ether 00:0c:29:18:1b:3a txqueuelen 1000 (Ethernet)
               RX packets 97618 bytes 12391774 (11.8 MiB)
               RX errors 1001 dropped 1011 overruns 0 frame 0
               TX packets 3004 bytes 568097 (554.7 KiB)
               TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
   ```

2. Run the ./setup-IPv6-network.sh script in the /opt/fnd/scripts directory to obtain the FND IPv6 address on the router for tunnel provisioning and registration.

   Example

   ```
   [root@iot-fnd scripts]# ./setup-IPV6-network.sh
               Starting IPv6 network setup...
               Loading configuration file: /opt/fnd/config/network.conf
               Checking current network settings...
               Configuring network interface for IPv6...
               Assigning IPv6 address...
               IPv6 address successfully assigned:
   2001:0db8:85a3:0000:0000:8a2e:0370:7334
               Verifying network connectivity...
               IPv6 setup complete. Network is now configured for IPv6
   communication.
   ```

> **Note**
>
> While specifying the IPv6 address for the network-mgmt-bridge, provide an interface name and a valid IPv6 address (and IP address prefix length) that is in the subnet of the provided host interface. If IPv6 address is in a different subnet, the IPv6 tunnel provisioning and registration will not be successful.

You've configured tunnel provisioning and registered with Cisco IoT FND.

Start and stop Cisco IoT FND on OVA

Use this procedure to start, stop, restart, or check the status of Cisco IoT FND on an OVA. It provides instructions for using the fnd-container script to manage the application container through the command line.

Use the **fnd-container.sh {start|stop|status|restart}** script in the following directory to start, stop, obtain status, and restart Cisco IoT FND.

Procedure

1. In the SSH terminal, run the following command to change the directory to where the script is located:

   Example

   ```
   cd /opt/fnd/scripts/
   ```

2. Execute the following command to start the Cisco IoT FND container:

   Example

   ```
   ./fnd-container.sh start
   ```

   ```
   Starting FND container...
               FND container started successfully.
   ```

3. To stop the FND container, run:

   Example

   ```
   ./fnd-container.sh stop
   ```

   ```
   Stopping FND container...
               FND container stopped successfully.
   ```

4. To obtain the current status of the FND container, use:

Example

```
./fnd-container.sh status
```

```
Checking FND container status...
            FND container is currently running.
```

5. To restart the FND container, use:

Example

```
./fnd-container.sh restart
```

```
Restarting FND container...
            FND container restarted successfully.
```

Start and stop Fog Director

Use this procedure to start, stop, restart, or check the status of the Fog Director container. It explains how to navigate to the script directory and execute the fogd-container script commands within an SSH terminal to manage the container effectively.

Use the **fogd-container.sh {start|stop|status|restart}** script in the following directory to start, stop, obtain status, and restart Fog Director.

Procedure

1. In the SSH terminal, run the following command to change the directory to where the script is located:

Example

```
cd /opt/fnd/scripts/
```

2. Execute the following command to start the Fog Director:

Example

```
./fogd-container.sh start
```

```
Starting fogd container...
                        fogd container started successfully.
```

3. To stop the Fog Director container, run:

Example

```
./fogd-container.sh stop
```

```
Stopping fogd container...
                        fogd container stopped successfully.
```

4. To obtain the current status of the Fog Director container, use:

Example

```
./fogd-container.sh status
```

```
Checking fogd container status...
                        fogd container is currently running.
```

5. To restart the Fog Director container, use:

Example

```
./fogd-container.sh restart
```

```
Restarting fogd container...
                        Fog container restarted successfully.
```

Obtain statuses of services running on the host

Learn how to obtain the statuses of all services running on the host by using SSH to access the server and executing a status-check script.

To verify that essential applications and system components are functioning correctly, quickly identify any issues with stopped services, and maintain the overall health and reliability of your system, you might need the statuses of all the services.

Use the following instructions to obtain all the statuses of services running on the host:

Procedure

1. Using SSH terminal, run the following command to change the directory to where the scripts are located:

Example

```
cd /opt/scripts
```

2. Execute the script to check the status of all services:

Example

```
./status.sh
```

```
Service Status Report
            ---------------------
            - Apache: Running
            - MySQL: Running
            - Docker: Stopped
            - SSH: Running
            - Firewall: Running
            ---------------------
            Total services: 5
            Running: 4
            Stopped: 1
```

Configure time and timezone using NTP service

Learn how to configure time and timezone settings on a host VM using the timedatectl command and NTP service, ensuring your Docker containers inherit the correct system time configuration.

The `timedatectl` command is used to manage date and time settings. You must execute this command on the **host VM** where your docker containers are running. This is because docker containers typically rely on the host system time settings unless specifically configured otherwise.

Use the following steps to configure date and time settings:

Procedure

1. Use SSH or another remote access method to log into the host VM where Docker is installed.
2. Run the `timedatectl` command:

Example

```
timedatectl set-time 15:30:00
```

```
Local time: Fri 2023-10-06 15:30:00 UTC
            Universal time: Fri 2023-10-06 15:30:00 UTC
            RTC time: Fri 2023-10-06 15:30:00
            Time zone: UTC (UTC, +0000)
            System clock synchronized: yes
            NTP service: active
            RTC in local TZ: no
```

The example output confirms that the local time has been set to `15:30:00`. If you see this time reflected in the `Local time` field, the command was successful.

Configure NTP services

Learn how to configure NTP services to ensure accurate time synchronization across network devices, which helps improve security, data integrity, and overall network performance.

Configuring an NTP server ensures accurate time synchronization across network devices, enhancing security, data integrity, application performance, compliance, and troubleshooting efficiency.

To configure the NTP server, follow these steps:

Procedure

1. Open the terminal on Cisco IoT FND or the TPS server.
2. Run the following command to edit the chrony.conf file:

   Example

   ```
   nano /etc/chrony.conf
   ```

   ```
   This command opens the file in the nano text editor, displaying the current
   configuration details.
   ```

3. Add the NTP server IP address:

   Example

   ```
   server 209.165.200.225
   ```

   Replace `209.165.200.225` with the actual IP address of your desired NTP server.

4. Save and exit.
5. Restart the chrony service to apply the changes using the following command:

   Example

   ```
   systemctl restart chronyd
   ```

6. Verify the configured NTP server details using the following commend:

   Example

   ```
   chronyc sources
   ```

   ```
   MS Name/IP address         Stratum Poll Reach LastRx Last sample
   ===============================================================================
               ^* 209.165.200.225               2  10   377   915    +17us[
   +19us] +/-   21ms
   ```

6 Upgrade Cisco IoT FND on OVA

Topics:

- Device compatibility check
- Check the Postgres DB status
- Cisco IoT FND on OVA upgrade matrix
- Verify and reset the expired root password
- Upgrade the database and docker server image
- Upgrade Cisco IoT FND container images
- Check DB and status post upgrade
- Upgrade RHEL version

This chapter provides comprehensive instructions for upgrading the Cisco IoT Field Network Director on an Open Virtual Appliance (OVA) environment, including detailed procedures for updating the database, Docker server, and container images.

Device compatibility check

Use these guidelines when verifying device compatibility and upgrade requirements for Cisco IoT FND to ensure your devices and supporting modules are aligned with validated versions and system prerequisites.

Use the following to check for the validated device versions that are compatible with Cisco IoT FND Release 5.0. The release versions mentioned in this topic are only the list of validated versions by Cisco IoT FND and is not exhaustive. We recommend you to upgrade your devices to these validated versions, and it is not a mandate.

- FAR compatibility: See supported device types and versions on Cisco IoT FND Release Notes 5.0
- HER compatibility: See supported device types and versions on Cisco IoT FND Release Notes 5.0
- HSM compatibility: See upgrade hardware security module compatibility on Cisco IoT FND Release Notes 5.0
- TPS compatibility: The TPS server version must match the Cisco IoT FND release version. For example, If you're upgrading to Cisco IoT FND Release 5.0 from Cisco IoT FND Release 4.12.1, your TPS server should be upgraded to Cisco IoT FND Release 5.0.
- Cisco IoT FND compatibility: On initial installation, upgrade the OpenSSH package in the Cisco IoT FND server to RHEL version 8.8 and later versions.

Check the Postgres DB status

Use this procedure to check the status of your system and Postgres database before upgrading Cisco IoT FND on OVA. Verifying these statuses ensures that all necessary services are running before you proceed with the upgrade process.

Performing this task helps you verify the system and database status before upgrading Cisco IoT FND on OVA.

Before you begin

- Take a snapshot of the existing VM before you upgrade.
- Perform a backup of **cgms.properties** file and **cgms_keystore** file in the location, /opt/fnd/data/. You can either SCP these files to another server for backup or you can copy in the same or different folder. Here's an example:

```
root@iot-fnd:~[root@iot-fnd ~]#
              root@iot-fnd:~[root@iot-fnd ~]# cd /opt/fnd/data
              root@iot-fnd:/opt/fnd/data[root@iot-fnd data]#
              root@iot-fnd:/opt/fnd/data[root@iot-fnd data]#ls
              cgms_keystore  cgms.properties  cisco-sudi-ca.pem
userPropertyTypes.xml
              root@iot-fnd:/opt/fnd/data[root@iot-fnd data]#
              root@iot-fnd:/opt/fnd/data[root@iot-fnd data]# cp
cgms.properties cgms.properties_backup_09May2025
              [root@iot-fnd data]# keytool -importkeystore -srckeystore
cgms_keystore -destkeystore cgms_keystore_backup_9May2022 -deststoretype
PKCS12
              Importing keystore cgms_keystore to
cgms_keystore_backup_9May2022...
              Enter destination keystore password:
              Re-enter new password:
              Enter source keystore password:
              Entry for alias cgms successfully imported.
              Entry for alias cisco_sudi successfully imported.
              Entry for alias jmarconi successfully imported.
              Import command completed:  3 entries successfully imported, 0
 entries failed or cancelled
              [root@iot-fnd data]#
```

```
              [root@iot-fnd data]# ls
              cgms_keystore                  cgms_keystore.selfsigned
cgms.properties_backup_09May2022  fnd_psk.keystore
              cgms_keystore_backup_9May2025  cgms.properties
cisco-sudi-ca.pem                userPropertyTypes.xml
              [root@iot-fnd data]#
```

- Ensure that the Postgres services are running before perfomring an upgrade.

Note

For the Cisco IoT FND host in this deployment, use the `fnduser` account for SSH access after upgrade. Direct SSH login as `root` is disabled by default. When root privileges are required after that, log in as `fnduser` and run `sudo -i`.

Procedure

1. Using the SSH terminal and navigate to the root directory.
2. Run the following command:

Example

```
/opt/scripts/status.sh
```

```
[root@iot-fnd ~]# /opt/scripts/status.sh
            --------------------
            ● postgresql-12.service - PostgreSQL 12 database server
            Loaded: loaded (/usr/lib/systemd/system/postgresql-12.service;
enabled; vendor preset: disabled)
            Active: active (running) since Mon 2022-05-09 02:01:29 PDT; 2h
6min ago
            Docs: https://www.postgresql.org/docs/12/static/
            Main PID: 27638 (postmaster)
            Tasks: 26
            Memory: 250.5M
            CGroup: /system.slice/postgresql-12.service
            --------------------
            ● influxdb.service - InfluxDB is an open-source, distributed,
time series database
          Loaded: loaded (/usr/lib/systemd/system/influxdb.service; enabled;
 vendor preset: disabled)
            Active: active (running) since Mon 2022-05-09 02:02:39 PDT; 2h
5min ago
            Docs: https://docs.influxdata.com/influxdb/
            Main PID: 27892 (influxd)
            Tasks: 21
            Memory: 219.0M
            --------------------
            ● kapacitor.service - Time series data processing engine.
          Loaded: loaded (/usr/lib/systemd/system/kapacitor.service; enabled;
 vendor preset: disabled)
            Active: active (running) since Mon 2022-05-09 02:02:06 PDT; 2h
5min ago
```

```
            Docs: https://github.com/influxdb/kapacitor
            Main PID: 27805 (kapacitord)
            Tasks: 14
            Memory: 21.0M
            --------------------
            fnd-container is running, pid=61255
            CONTAINER ID        NAME                CPU %               MEM
USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O
  PIDS
            a02e6388607d        fnd-container       6.44%
2.612GiB / 23.38GiB   11.17%              17MB / 13.7MB       20.3MB / 2.64MB
     580
            --------------------
            fogd-container is running, pid=63469
            CONTAINER ID        NAME                CPU %               MEM
USAGE / LIMIT    MEM %               NET I/O             BLOCK I/O
 PIDS
          a40aa29e2392       fogd-container     6.38%               2.18GiB
 / 23.38GiB   9.32%               434kB / 135kB       8.19kB / 145kB      99

            --------------------
            [root@iot-fnd ~]#
            docker version
            [root@iot-fnd ~]# docker version
            Client: Docker Engine - Community
            Version:           19.03.15
            API version:       1.40
            Go version:        go1.13.15
            Git commit:        99e3ed8919
            Built:             Sat Jan 30 03:17:57 2021
            OS/Arch:           linux/amd64
            Experimental:      false
            Server: Docker Engine - Community
            Engine:
            Version:          19.03.15
            API version:      1.40 (minimum version 1.12)
            Go version:       go1.13.15
            Git commit:       99e3ed8919
            Built:            Sat Jan 30 03:16:33 2021
            OS/Arch:          linux/amd64
            Experimental:     false
            containerd:
            Version:          1.4.4
            GitCommit:        05f951a3781f4f2c1911b05e61c160e9c30eaa8e
            runc:
            Version:          1.0.0-rc93
            GitCommit:        12644e614e25b05da6fd08a38ffa0cfe1903fdec
            docker-init:
            Version:          0.18.0
            GitCommit:        fec3683
            You have new mail in /var/spool/mail/root
            [root@iot-fnd ~]#
            /opt/fnd/scripts/fnd-container.sh status
            [root@iot-fnd ~]# /opt/fnd/scripts/fnd-container.sh status
            fnd-container is running, pid=61255
            CONTAINER ID        NAME                CPU %               MEM
USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O
  PIDS
            a02e6388607d        fnd-container       6.47%
2.613GiB / 23.38GiB   11.18%              17MB / 13.8MB       20.3MB / 2.64MB
     592
            [root@iot-fnd ~]#
```

```
            You have new mail in /var/spool/mail/root
            [root@iot-fnd ~]#
            docker exec -it fnd-container /etc/init.d/cgms status
            [root@iot-fnd ~]# docker exec -it fnd-container /etc/init.d/cgms
 status
            IoT-FND Version 4.7.2-8
          05-09-2022 04:09:46 PDT: INFO: IoT-FND database server: 192.68.5.1

            05-09-2022 04:09:47 PDT: INFO: IoT-FND database connection
verified.
          05-09-2022 04:09:47 PDT: INFO: IoT FND timeseries database server:
 192.68.5.1
          05-09-2022 04:09:47 PDT: INFO: IoT FND kapacitor server: 192.68.5.1

            05-09-2022 04:09:47 PDT: INFO: IoT-FND timeseries
database/kapacitor connection verified.
            05-09-2022 04:09:49 PDT: INFO: IoT-FND application server is up
and running.
            05-09-2022 04:09:50 PDT: INFO: IoT-FND is up and running.
            [root@iot-fnd ~]#
            rpm -qa | grep -i postgres
            root@iot-fnd:/opt/fnd/data[root@iot-fnd data]# rpm -qa | grep -i
 postgres
            postgresql96-devel-9.6.15-1PGDG.rhel7.x86_64
            postgresql96-libs-9.6.15-1PGDG.rhel7.x86_64
            postgresql96-server-9.6.15-1PGDG.rhel7.x86_64
            postgresql96-9.6.15-1PGDG.rhel7.x86_64
            cgms-postgres-4.5.1-11.x86_64
            postgresql96-contrib-9.6.15-1PGDG.rhel7.x86_64
            root@iot-fnd:/opt/fnd/data[root@iot-fnd data]#
```

You can view all the statuses of your Cisco IoT FND on OVA before proceeding with an upgrade.

Cisco IoT FND on OVA upgrade matrix

This topic explains the upgrade matrix for Cisco IoT Field Network Director (FND) on OVA. It provides a detailed overview of the specific versions for Postgres, Docker server, and RHEL OS bundled with each FND release to ensure software compatibility and successful deployments.

The following table provides a detailed overview of the versions of different software components bundled with each Cisco IoT FND release.

Cisco IoT FND version	Postgres version	Docker server version	RHEL OS version
26.2.1	16.7	19.03.15	9.8
5.1.0	16.7	19.03.15	9.6
5.0.0	12.12	19.03.15	8.10
4.12.0	12.12	19.03.15	8.8
4.11.0	12.12	19.03.15	8.8
4.10.0	12.12	19.03.15	8.7

Cisco IoT FND version	Postgres version	Docker server version	RHEL OS version
4.9.1	12.12	19.03.15	8.6
4.9.0	12.9	19.03.15	8.6

Verify and reset the expired root password

Use this procedure to verify the root password status before proceeding with the Cisco IoT FND upgrade. If the root password has expired or is not set, use the instructions provided to reset it.

The root password must be valid and active before running the Cisco IoT FND upgrade. During the upgrade, scripts invoke sudo to execute multiple commands. If the root password has expired or is not set, sudo authentication fails. As a result, the upgrade stops..

Before you begin

Use these steps to verify the status of the root password and reset it if it has expired:

Procedure

1. Enter the ESXi server's Internet Protocol (IP) address in a web browser and log in with your ESXi credentials.
2. From the left pane, select the Cisco IoT FND VM.
3. Click **Console** and select **Open Console in new tab**.

   The RHEL server launches.
4. Log in with your Cisco IoT FND server user credentials.
5. Access the Cisco IoT FND Shell terminal.
6. Enter the following command to verify the root password status:

   Example

   ```
   [fnduser@iot-fnd ~]$ sudo -i
   [root@iot-fnd ~]# sudo test
   ```

   Table 7: Root password status

If the root password is	Then
active `[root@iot-fnd ~]#`	you can run the upgrade script. For upgrade instructions, refer to the next topics in this chapter.

expired and the sudo authentication fails with the following error:
`sudo: Account or password is expired, reset your password and try again`
`Current password:`
`sudo: unable to change expired password: Authentication token manipulation error`
	you must reset the root password. For instructions, go to step 7

7. Run the following command to reset the root password:

Example

```
[root@iot-fnd ~]# passwd
Changing password for user root.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

8. Run the following command to verify the sudo command.

Example

```
[root@iot-fnd ~]# sudo test
[root@iot-fnd ~]#
```

After the root password is active, sudo commands will run successfully without authentication errors during the upgrade.

Upgrade the database and docker server image

Use this procedure to upgrade the database and docker server image for Cisco IoT FND, ensuring compatibility and security by running upgrade scripts and integrating the database with Cisco IoT FND.

To ensure compatibility and security, upgrade the database and docker server image for Cisco IoT FND.

To upgrade the database and the docker server image, run the rpm scripts and auto-integrate the DB with Cisco IoT FND scripts.

Before you begin

Cisco IoT FND OVA upgrade does not upgrade the RHEL OS version. The RHEL version may vary depending on the Cisco IoT FND release. After upgrading the OVA, upgrade the OS as soon as possible. Cisco IoT FND is a secure application, but the OS security and patches must be regularly updated based on Cisco guidance.

Use this procedure to upgrade Cisco IoT FND Release 4.9.x, 4.10.x, 4.11.x, and 4.12.x to Release 5.0.x or higher.

Procedure

1. Obtain the Cisco IoT FND upgrade scripts from Cisco.
2. Check the RHEL OS version before upgrading Cisco IoT FND OVA to Cisco IoT FND Release 5.0 or higher.

Example

```
[root@fnd451testupgrade ~]# hostnamectl
            Static hostname: fnd451testupgrade
            Icon name: computer-vm
            Chassis: vm
            Machine ID: 58eb8d728d834d28ad426eca3c9b9c4e
            Boot ID: 40511dab9f4b4beaa8de82fb105423c9
            Virtualization: vmware
            Operating System: Red Hat Enterprise Linux
            CPE OS Name: cpe:/o:redhat:enterprise_linux:7.5:GA:server
            Kernel: Linux 3.10.0-862.el7.x86_64
```

```
            Architecture: x86-64
            [root@fnd451testupgrade ~]#r
```

3. Extract the cgms rpms files to the Cisco IoT FND server.

   Example

   If you are upgrading the DB and the docker server image for Cisco IoT FND release 5.0 or higher:

   a. Download the upgrade script from Cisco.

      Example:

      CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip

   b. Extract the file to get the rpm:

      Example:

      upgrade-ova-<release>-<build number>.rpm

   c. Transfer the extracted rpm file to the Cisco IoT FND server.

4. Navigate to the directory where the rpm file is located.

   Example

   cd /opt or any directory where the *upgrade-ova-<release>-<build number>.rpm file* is copied.

5. Run the upgrade script:

   Example

```
rpm -Uvh upgrade-ova-<release>-<build number>.rpm
```

   The sample logs:

   - For upgrade to Release 5.0.0 or 5.1.0
   - For upgrade to Release 26.2.1

   Sample log: For upgrade to Release 5.0.0 or 5.1.0, you may see output such as:

```
root@iot-fnd:/opt[root@iot-fnd opt]# rpm -Uvh upgrade-ova-5-0-2-8.rpm
            Preparing...
            (1%)##############(100%)

            Updating / installing...
            1:upgrade-ova-5-0-2-8
            (1%)##############(100%)
            Started installer in background. Please check ~/rpm.log in few
minutes for details.
            root@iot-fnd:/optYou have new mail in /var/spool/mail/root
            [root@iot-fnd opt]#
            Mon May  9 01:59:29 PDT 2022 Background installer started
            Mon May  9 01:59:29 PDT 2022 Please wait until the 'RPM
installation completed' message is logged
            Mon May  9 01:59:29 PDT 2022 Upgrading
cgms-postgres-5.0.2-8.x86_64.rpm
            Preparing...
########################################
            Updating / installing...
            cgms-postgres-5.0.2-8
```

```
########################################
            Cleaning up / removing...
            cgms-postgres-5.0-101
########################################
            Mon May  9 01:59:47 PDT 2022 Upgrading
cgms-influx-4.7.2-8.x86_64.rpm
            Preparing...
########################################
            Updating / installing...
            cgms-influx-5.0.2-8
########################################
            Cleaning up / removing...
            cgms-influx-5.0-101
########################################
            Mon May  9 02:00:04 PDT 2022 Upgrading
monit-5.25.3-1.el7.x86_64.rpm
            warning: monit-5.25.3-1.el7.x86_64.rpm: Header V4 RSA/SHA1
Signature, key ID 222b0e83: NOKEY
            Preparing...
########################################
            package monit-5.25.3-1.el7.x86_64 is already installed
            Mon May  9 02:00:18 PDT 2022 Stopping services
            Mon May  9 02:00:58 PDT 2022 Upgrading Postgresql to 12.5
            Preparing...
########################################
            Updating / installing...
            postgresql12-libs-12.5-1PGDG.rhel7
########################################
            postgresql12-12.5-1PGDG.rhel7
########################################
            postgresql12-server-12.5-1PGDG.rhel7
########################################
            postgresql12-contrib-12.5-1PGDG.rhel7
########################################
            Cleaning up / removing...
            postgresql12-contrib-12.4-1PGDG.rhel7
########################################
            postgresql12-server-12.4-1PGDG.rhel7
########################################
            postgresql12-12.4-1PGDG.rhel7
########################################
            postgresql12-libs-12.4-1PGDG.rhel7
########################################
            Mon May  9 02:01:27 PDT 2022 Restarting Postgresql
            Mon May  9 02:01:40 PDT 2022 Stopping InfluxDB and Kapacitor
           Mon May  9 02:01:50 PDT 2022 Upgrading influxdb-1.8.3.x86_64.rpm

            Preparing...
########################################
            Updating / installing...
            influxdb-1.8.3-1                      warning:
/etc/influxdb/influxdb.conf created as /etc/influxdb/influxdb.conf.rpmnew
            ########################################
            Cleaning up / removing...
            influxdb-1.5.3-1
########################################
          Mon May  9 02:02:02 PDT 2022 Upgrading kapacitor-1.5.7-1.x86_64.rpm

            Preparing...
########################################
            Updating / installing...
            kapacitor-1.5.7-1                     warning:
```

```
/etc/kapacitor/kapacitor.conf created as /etc/kapacitor/kapacitor.conf.rpmnew

            ########################################
            Cleaning up / removing...
            kapacitor-1.5.0-1
########################################
            Mon May  9 02:02:06 PDT 2022 Restarting InfluxDB and Kapacitor
            Mon May  9 02:02:20 PDT 2022 Stopping Docker
            Mon May  9 02:02:26 PDT 2022 Upgrading Docker to 19.03.15
            warning: container-selinux-2.119.2-1.911c772.el7_8.noarch.rpm:
Header V3 RSA/SHA256 Signature, key ID f4a80eb5: NOKEY
            Preparing...
            (1%)##############(100%)

            Updating / installing...
            1:container-selinux-2:2.119.2-1.911
            (1%)##############(100%)
            Cleaning up / removing...
            2:container-selinux-2:2.42-1.gitad8
            (1%)##############(100%)
            Preparing...
            (1%)##############(100%)

            Updating / installing...
            1:docker-ce-cli-1:19.03.15-3.el7
            (1%)##############(100%)
            2:containerd.io-1.4.4-3.1.el7
            (1%)##############(100%)
            3:docker-ce-3:19.03.15-3.el7
            (1%)##############(100%)
            /usr/bin/dockerd has not been configured as an alternative for
dockerd
            Cleaning up / removing...
            4:docker-ce-3:18.09.6-3.el7
            (1%)##############(100%)
            5:containerd.io-1.2.5-3.1.el7
            (1%)##############(100%)
            6:docker-ce-cli-1:18.09.6-3.el7
            (1%)##############(100%)
            Mon May  9 02:04:11 PDT 2022 Restarting Docker
            Mon May  9 02:04:29 PDT 2022 Restarting services
            Mon May  9 02:04:59 PDT 2022 RPM installation completed
```

Sample log: For upgrade to Release 26.2.1, sample log entries may include:

```
[root@iot-fnd opt]# rpm -Uvh --force upgrade-ova-26.2.1-120.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:upgrade-ova-26.2.1-120           ################################# [100%]
Deleting
/var/lib/docker/volumes/fogd_data/_data/dist/appmgr/extensions/fnd/dependencies...
Deletion complete.

Started installer in background. Please check ~/rpm.log in few minutes for
details.
IP address migration will run automatically after all installations complete.
[root@iot-fnd opt]#
Sat Sep 19 02:28:00 PM IST 2026 Background installer started
Sat Sep 19 02:28:00 PM IST 2026 Please wait until the 'RPM installation
completed' message is logged
```

```
Sat Sep 19 02:28:00 PM IST 2026 Upgrading
cgms-postgres-26.2.1-120.pg.x86_64.rpm
Verifying...                          ##########################################
Preparing...                          ##########################################
Updating / installing...
cgms-postgres-26.2.1-120.pg            ##########################################
Cleaning up / removing...
cgms-postgres-5.1.0-155                ##########################################

Sat Sep 19 02:28:05 PM IST 2026 Upgrading cgms-influx-26.2.1-120.x86_64.rpm
Verifying...                          ##########################################
Preparing...                          ##########################################
Updating / installing...
cgms-influx-26.2.1-120                ##########################################
Cleaning up / removing...
cgms-influx-5.1.0-155                 ##########################################

Sat Sep 19 02:28:09 PM IST 2026 monit-5.33.0-1.el9 already installed on RHEL
 9
Sat Sep 19 02:28:09 PM IST 2026 Already running Postgresql 16.7
Sat Sep 19 02:28:09 PM IST 2026 Using PostgreSQL version 16 for configuration
 path.
ALTER ROLE
ALTER ROLE
ALTER ROLE
ALTER ROLE
ALTER ROLE
ALTER ROLE
setting idle_in_transaction_session_timeout to 3h
set idle_in_transaction_session_timeout to 3h

Sat Sep 19 02:28:12 PM IST 2026 Already running influxdb 1.8.3

Sat Sep 19 02:28:12 PM IST 2026 Already running docker-ce 24.0.5-1.el9
Sat Sep 19 02:28:12 PM IST 2026 Checking for IP address migration
Sat Sep 19 02:28:12 PM IST 2026 Running IP address migration script...
Sat Sep 19 02:31:46 PM IST 2026 IP migration completed. Check
/var/log/ip-migration.log for details
Sat Sep 19 02:31:46 PM IST 2026 RPM installation completed
```

The upgrade of Cisco IoT FND on OVA is complete.

Note

When upgrading to Cisco IoT FND Release 26.2.1, Fog Director configurations may take up to 10 additional minutes to complete. This can increase the total upgrade time by up to 10 minutes.

What to do next

For the Cisco IoT FND host in this deployment, use the `fnduser` account for SSH access after upgrade. Direct SSH login as `root` is disabled by default. When root privileges are required after that, log in as `fnduser` and run `sudo -i`.

Upgrade Cisco IoT FND container images

Use this procedure to upgrade Cisco IoT FND container images by downloading the latest versions and reloading the containers using the provided upgrade script on your IoT server. This ensures your deployment stays current and secure.

Procedure

1. Open the SSH terminal and log in to Cisco IoT server as a root user.
2. Run the following script:

Example

```
/opt/fnd/scripts/upgrade.sh
```

```
[root@iot-fnd ~]# /opt/fnd/scripts/upgrade.sh
            This script must be run with root privileges.
            Usage: Load container images: No resource required
            For container reload: No resource required
            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 1
            Do you want to download docker image from registry (y/n)?y
            Enter docker registry [devhub-docker.cisco.com]:
dockerhub.cisco.com
            Enter docker image tag: 26.2.1-116
            Downloading FND docker image...
            26.2.1-116: Pull complete
            ea3c714182eb: Pull complete
            177abefb5b93: Pull complete
            e696bdc28724: Pull complete
            89dd87262f50: Pull complete
            ff6164c0609f: Pull complete
            89a0b2205b62: Pull complete
            4dbd23bb6e45: Pull complete
            Digest:
sha256:2ae8a3cba38ea28156a2c3db55cd8cea0448888a7704479cac33b665d8b2a132
            Status: Downloaded newer image for
dockerhub.cisco.com/field-network-director-dev-docker/fnd-image:26.2.1-116

dockerhub.cisco.com/field-network-director-dev-docker/fnd-image:26.2.1-116
            Downloading Fog Director docker image...
            26.2.1-116: Pulling from fog-director-dev-docker/fogd-image
            5e9a6732a7a3: Pull complete
            55a104320bff: Pull complete
            506e5a93cf62: Pull complete
            9b2523a38071: Pull complete
            8e8389537d47: Pull complete
            e6fcef979884: Pull complete
            e2e278b80221: Pull complete
            63bc79650477: Pull complete
            Digest:
sha256:16f3227fbac74804f1e2a77aa57ebeeb5b9f05eb4efb0ddccf242865fe673634
            Status: Downloaded newer image for
dockerhub.cisco.com/fog-director-dev-docker/fogd-image:26.2.1-116
          dockerhub.cisco.com/fog-director-dev-docker/fogd-image:26.2.1-116
```

```
            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 2
            Stopping FND container...
            fnd-container
            Remove FND container...
            fnd-container
            Prune Docker container...
            Starting FND container...
           a02e6388607d79504f082dccf179514e5dc2d6bcd34021beac21baf1a555c266

            Stopping Fog Director container...
            fogd-container
            Remove Fog Director container...
            fogd-container
            Prune Docker container...
            Starting Fog Director container...
           a40aa29e2392e1e99a5f024d3d5838712d66ef638f0c6b0bf209b1932076611c

            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 3
            You have new mail in /var/spool/mail/root
            [root@iot-fnd ~]#
```

3. Enter **1** to load the container images.

Example

```
[root@iot-fnd ~]# /opt/fnd/scripts/upgrade.sh
            This script must be run with root privileges.
            Usage: Load container images: No resource required
            For container reload: No resource required
            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 1
            Do you want to download docker image from registry (y/n)? y
            Enter docker registry [devhub-docker.cisco.com]:
dockerhub.example.com
            Enter docker image tag: 26.2.1-116
            Downloading FND docker image...
            26.2.1-116: Pulling from example-docker-repo/fnd-image
            a3ed95caeb02: Pull complete
            4e9f1a5e87b7: Pull complete
            bcd8f8dc5c34: Pull complete
            c5e155d5a1d1: Pull complete
            5b8c6e5e9b45: Pull complete
            b5e3b5c3f5b2: Pull complete
            f5b3c27c7f9f: Pull complete
            Digest:
sha256:8d3d7b5e8d3f1c2b1e8b2f1b7c9a1234567890abcdef1234567890abcdef1234
            Status: Downloaded newer image for
dockerhub.example.com/example-docker-repo/fnd-image:26.2.1-116
            dockerhub.example.com/example-docker-repo/fnd-image:26.2.1-116
            Downloading Fog Director docker image...
            26.2.1-116: Pulling from example-docker-repo/fogd-image
            9f9a1c3f5f12: Pull complete
            6c5a9f7e9b34: Pull complete
```

```
            8d5e3b3c5f6b: Pull complete
            b5f1c9e7d1b2: Pull complete
            a5c2b6e4f8e7: Pull complete
            d1e3f5b4c6d3: Pull complete
            Digest:
sha256:5e8d3f1a2b3c4d5e6f7a8b9c0d1e2f3a4b5c6d7e8f9a0b1c2d3e4f5g6h7i8j9k
            Status: Downloaded newer image for
dockerhub.example.com/example-docker-repo/fogd-image:26.2.1-116
            dockerhub.example.com/example-docker-repo/fogd-image:26.2.1-116
            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 3
            [root@iot-fnd ~]#
```

4. Download the latest container image for Cisco IoT FND from devhub-docker.cisco.com.
5. After the images are downloaded successfully, enter **2** to reload container.

Example

```
[root@iot-fnd ~]# /opt/fnd/scripts/upgrade.sh
            This script must be run with root privileges.
            Usage: Load container images: No resource required
            For container reload: No resource required
            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 2
            Stopping FND container...
            fnd-container
            Remove FND container...
            fnd-container
            Prune Docker containers...
            Deleted Containers:
            fnd-container
            Starting FND container...
           c8e1f9a8d7c2b3a4e5f6d7c8b9a0b1c2d3e4f5g6h7i8j9k0l1m2n3o4p5q6r7s8

            Stopping Fog Director container...
            fogd-container
            Remove Fog Director container...
            fogd-container
            Prune Docker containers...
            Deleted Containers:
            fogd-container
            Starting Fog Director container...
           b7c6d5e4f3g2h1i0j9k8l7m6n5o4p3q2r1s0t9u8v7w6x5y4z3a2b1c0d9e8f7g6

            1) Load container images
            2) Container reload
            3) Quit
            Enter your choice: 3
            [root@iot-fnd ~]#
```

6. Enter **3** to quit the menu.

You've upgraded the Cisco IoT FND containers.

Check DB and status post upgrade

Use this procedure to check the database and Cisco IoT FND status after an upgrade to ensure the process was successful. You will learn how to verify system services, Docker versions, and container health to confirm that all critical components are operational.

Use these steps to check the DB and Cisco IoT FND status to ensure that your upgrade is successful

Note

For Cisco IoT FND 26.2.1 upgrades performed using /opt/fnd/scripts/upgrade.sh, complete the Fog Director listener check first, then verify device IP information when the solution uses IOX or Edge Compute.

Before you begin

Use these credentials for SSH access after upgrading OVA:

- Username: fnduser
- Password: C!sco123

See Guidelines for resetting password.

Note

For the Cisco IoT FND host in this deployment, use the `fnduser` account for SSH access after upgrade. Direct SSH login as `root` is disabled by default. When root privileges are required after that, log in as `fnduser` and run `sudo -i`.

Procedure

1. Log in to the server or machine using SSH.
2. Execute the `/opt/scripts/status.sh` command:

Example

```
/opt/scripts/status.sh
```

```
Checking system status...
            Service: Database
            Status: Running
            Uptime: 24 hours
            Service: IoT FND
            Status: Running
            Uptime: 12 hours
            Service: Web Server
            Status: Stopped
            Action Required: Restart service
            Disk Usage:
```

```
            Filesystem      Size  Used Avail Use% Mounted on
            /dev/sda1        50G   30G   18G  63% /
            Memory Usage:
            Total: 8GB
            Used:  4GB
            Free:  4GB
            Network Status:
            eth0: Connected
            IP Address: 192.168.1.10
            All critical services are operational except the Web Server.
Please address the issues noted above.
```

3. Execute the `docker version` command:

   Example

```
docker version
```

```
Client: Docker Engine - Community
            Version:           20.10.14
            API version:       1.41
            Go version:        go1.16.15
            Git commit:        a224086
            Built:             Thu Mar 24 01:50:09 2022
            OS/Arch:           linux/amd64
            Context:           default
            Experimental:      true
            Server: Docker Engine - Community
            Engine:
            Version:          20.10.14
            API version:      1.41 (minimum version 1.12)
            Go version:       go1.16.15
            Git commit:       87a90dc
            Built:            Thu Mar 24 01:48:43 2022
            OS/Arch:          linux/amd64
            Experimental:     false
            containerd:
            Version:          1.5.11
            GitCommit:        3df54a852345ae127d1fa3092b95168e4a88e2f8
            runc:
            Version:          1.0.3
            GitCommit:        f46b6ba2c9314cfc8caae24a32ec5fe9ef1059fe
            docker-init:
            Version:          0.19.0
            GitCommit:        de40ad0
```

4. Execute the `/opt/fnd/scripts/fnd-container.sh status` command to check the Cisco IoT FND container status.

   Example

```
/opt/fnd/scripts/fnd-container.sh status
```

```
Checking IoT FND container status...
            Container Name: fnd-container
            Status: Running
            Uptime: 36 hours
            CPU Usage: 15%
```

```
            Memory Usage: 512MB / 2GB
            Network: Active
            IP Address: 172.17.0.2
            Port Mappings: 8080:80, 8443:443
            Last Health Check: Passed
            Health Check Interval: 5 minutes
            All monitored services within the container are operational.
```

5. Execute the `docker exec -it fnd-container /etc/init.d/cgms status` command to check the status of the cgms service inside the Cisco IoT FND container.

   Example

   ```
   docker exec -it fnd-container /etc/init.d/cgms status
   ```

   ```
   Checking status of cgms service...
               cgms service is running
               PID: 1234
               Uptime: 24 hours
               Listening on Port: 9090
               No issues detected with the cgms service.
   ```

6. Check whether the Fog Director listener is Up inside fogd-container.

   ```
   docker exec fogd-container sh -c "grep -i ':6FE2' /proc/net/tcp /proc/net/tcp6"
   ```

   Expected output:

   ```
   /proc/net/tcp6:   4: 00000000000000000000000000000000:6FE2
   00000000000000000000000000000000:0000 0A 00000000:00000000 00:00000000 00000000
        0        0 441522 1 0000000000000000 100 0 0 10 0 0
   ```

   The command displays a listener entry containing :6FE2. It usually takes less than five minutes for the expected output to appear.

   If the Fog Director listener check does not display any output, perform these actions:

   a) If the listener check still shows no output after five minutes, restart the Fog Director container.

      ```
      /opt/fogd/scripts/fogd-container.sh restart
      ```

   b) Wait a few minutes, then run the Fog Director listener check again.

      ```
      docker exec fogd-container sh -c "grep -i ':6FE2' /proc/net/tcp
      /proc/net/tcp6"
      ```

      The listener check displays the :6FE2 listener entry.

You have checked the Cisco IoT FND status. This confirms that the upgrade has succeeded.

What to do next

Log into Cisco IoT FND to confirm that services are operational. For example, refresh metrics for several devices or add and delete devices using CSV.

Upgrade RHEL version

This topic explains how to perform an in-place Red Hat Enterprise Linux upgrade using the Leapp utility, providing reference to official Red Hat documentation for detailed steps.

This section provides references to read about how to perform an in-place Red Hat Enterprise Linux upgrade using the Leapp utility.

For Cisco IoT FND Release 5.0.0 and earlier, follow the instructions provided in the Red Hat documentation about Upgrading from RHEL 7 to RHEL 8 .

For Cisco IoT FND Release 5.1.0 and later, follow the instructions provided in the Red Hat documentation about Upgrading from RHEL 8 to RHEL 9.

7 HA and DR

Topics:

- HA and DR for Cisco IoT FND with Postgres
- Prerequisites
- Restrictions
- HA and DR architecture
- Configure HA and DR
- Database and Cisco IoT FND Migration
- Monitor servers
- Configure self-signed certificates
- Example Influx DB to Postgres migration log
- Troubleshoot stale server entries
- Troubleshoot CPU and memory issues

This topic describes high availability (HA) and disaster recovery (DR) options for Cisco IoT Field Network Director (FND) when using a Postgres database.

HA and DR for Cisco IoT FND with Postgres

Use this topic to understand high availability (HA) and disaster recovery (DR) options for Cisco IoT Field Network Director (FND) when using a Postgres database.

High Availability (HA) and Disaster Recovery (DR) for Cisco IoT FND with PostgreSQL are designed to ensure that the Cisco IoT FND system remains continuously operational and resilient to failures, such as server crashes, database outages, or even complete data center failures.

- Replaces a single-node database deployment with an enterprise-grade, distributed PostgreSQL architecture
- introduces redundancy at every critical layer database, application, and network access.

Table 8: Feature history

Feature name	Release	Description
HA and DR for Cisco IoT FND with Postgres	Cisco IoT FND Release 5.1.1	HA and DR for Cisco IoT FND with PostgreSQL ensure the system remains operational and resilient against failures, including server crashes, database outages, and data center disruptions.

Benefits

- Keeps Cisco IoT FND running even during server, database, or infrastructure failures, ensuring uninterrupted network management.
- Database and application components automatically switch to healthy instances without manual intervention, reducing recovery time.
- Distributed PostgreSQL replication safeguards critical data and minimizes the risk of data loss during failures or outages.
- Supports disaster recovery across multiple sites, allowing operations to continue even if an entire data center becomes unavailable.
- Designed for large-scale IoT deployments, providing a robust and scalable foundation that supports future growth and mission-critical operations.

Aspect	Standalone Setup	HA Setup
Deployment Model	Single Cisco IoT FND with PostgreSQL OVA	Multiple servers deployed across multiple data centers
Database	Single-node PostgreSQL (no HA)	Enterprise-grade distributed PostgreSQL
Failover Capability	No automatic failover	Automatic failover across components
Application Servers	Single FND application server	Redundant, active FND application servers
Traffic Routing	Direct access to a single instance	DNS-based traffic routing to active instance

Aspect	Standalone Setup	HA Setup
Failure Handling	Manual recovery required	Automatic takeover by healthy components

Prerequisites

This topic outlines the prerequisites for implementing high availability (HA) and disaster recovery (DR) for Cisco IoT Field Network Director (FND) with a Postgres database.

Here are the prerequisities to use HA and DR on Cisco IoT FND with Postgres:

- Deploy EnterpriseDB PostgreSQL Standard v16 with Postgres Distributed (PGD), and manage database upgrades, backup, and restore in coordination with the EDB vendor.
- Provision separate Bare Metal servers or VMs for Cisco IoT FND application servers, the EDB Witness node, and PostgreSQL data nodes, with the PGD Proxy co-located on the data nodes; do not use the standalone OVA for HA deployments.
- Deploy the solution across multiple data centers with reliable, low-latency network connectivity, including multiple PostgreSQL data nodes and a witness node distributed across sites.
- Configure a PGD Proxy for each PostgreSQL data node and ensure reachability from all Cisco IoT FND application servers.
- Deploy two active Cisco IoT FND application servers in different data centers and enable secure REST-based communication between them.
- Deploy a high-availability DNS solution such as F5 GSLB with the required licenses, and configure Cisco IoT FND access using fully qualified domain names.
- Install custom JBoss web certificates on Cisco IoT FND servers to enable secure inter-application communication and accept the defined security behavior when using self-signed certificates.
- Run the setupCgms.sh script with high-availability options, including configuration of PGD proxy addresses and database node details.
- Acknowledge architectural constraints and requirements, including unsupported FogD HA, InfluxDB HA, and TPS HA, adequate infrastructure capacity for large-scale deployments, and compliance with Cisco security standards.

Restrictions

This topic describes the limitations and restrictions that apply when implementing high availability (HA) and disaster recovery (DR) for Cisco IoT Field Network Director (FND) with a Postgres database.

Here are the restrictions to use HA and DR on Cisco IoT FND with Postgres:

- HA is supported for both Oracle and PostgreSQL deployments; Oracle deployments utilize Real Application Clusters (RAC), while PostgreSQL deployments utilize the EnterpriseDB PGD solution.
- In Cisco IoT FND Release 5.1.1, the HA solution supports a maximum of 10,000 devices. In Cisco IoT FND Release 26.2.1, the HA solution supports a maximum of 50,000 devices. Additionally, you must configure firmware operations to use a maximum of 5 threads and restrict PnP image upgrades to a maximum of 5 devices concurrently.
- Fog Director HA and disaster recovery are not supported as part of this solution.
- InfluxDB is not part of this solution; all time-series data is stored in the EnterpriseDB PostgreSQL database.

- TPS Proxy HA is not supported; if the TPS node becomes unavailable, it must be manually brought up at another site.
- Database HA requires the EnterpriseDB Standard Postgres with Postgres Distributed (PGD) solution; the community version of PostgreSQL does not support this architecture.
- Cisco IoT FND application servers run in a multi-active, standalone mode; JBoss clustering across multiple subnets or data centers is not supported.

HA and DR architecture

This topic explains the architecture for Cisco IoT FND that employs a distributed multi-site deployment using EnterpriseDB Standard Postgres with the Postgres Distributed (PGD) solution. This design ensures continuous operation by utilizing two active FND application servers across separate data centers, automated database failover via co-located PGD Proxies, and DNS-based traffic routing, providing enterprise-grade resilience and data protection.

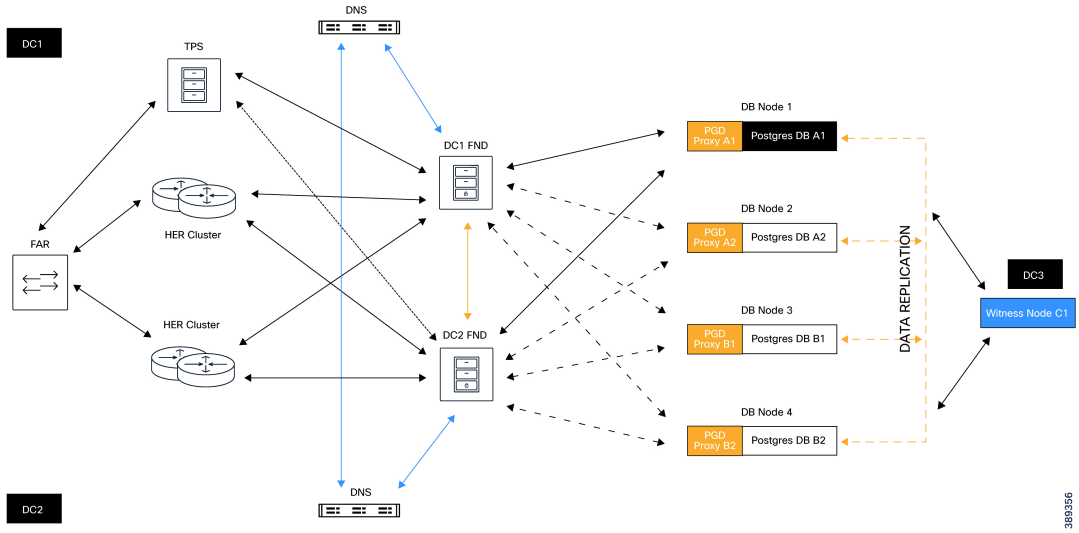
This architecture is intended for deployments that require continuous operation and resilience against server, database, and data center failures, and it is illustrated in the HA architecture diagram provided in the specification.

Summary

This process describes how the HA and DR architecture for Cisco IoT FND with PostgreSQL operates to maintain service availability through distributed database replication, active application servers, and DNS-based traffic redirection.

Workflow

Figure 1: HA and DR architecture



1. Cisco IoT FND application servers and PostgreSQL database nodes are deployed across multiple data centers to eliminate single points of failure.
2. EnterpriseDB PostgreSQL with Postgres Distributed replicates data across all database nodes, designating one node as the primary and the others as standby nodes.
3. PGD Proxy instances act as the access layer between FND and PostgreSQL, dynamically routing all database connections to the current primary node.
4. Two FND application servers run in active mode and connect to the database through the PGD proxies, ensuring uninterrupted application access during database failover.

5. The FND application servers synchronize cache and configuration updates using secure REST-based communication to maintain consistency across sites.
6. Routers using a fully qualified domain name that is resolved by a global server load balancing DNS service.
7. When a failure occurs at the database, application, or site level, the system automatically promotes healthy database nodes and redirects application traffic to the active site.

Result

The HA and DR architecture ensures continuous availability of Cisco IoT FND by automatically handling failures without manual intervention. This approach protects data integrity, minimizes service disruption, and supports reliable operation of large-scale IoT deployments across multiple data centers.

Configure HA and DR

Use this topic to configure and implement high availability for Cisco IoT FND using PostgreSQL, ensuring continuous service and minimal downtime.

Use these instructions to configure HA and DR on on Cisco IoT FND with Postgres.

Before you begin

- Ensure to have an EDB cluster up and running. For more information, see creating a cluster. Here's the config.yaml file used to provision and deploy the cluster.

```
[XXXXXXX@pgdah pgd_cluster]$ cat config.yml
---
architecture: PGD-Always-ON
cluster_name: pgd_cluster
cluster_tags: {}

keyring_backend: legacy
vault_name: xxxx-xxxxx-xxxxx-xxxxx-xxxxxxxx

cluster_vars:
  postgres_users:
  - username: cgms_dev
    generate_password: false
    role_attrs:
    - LOGIN
    - SUPERUSER
    - CREATEROLE
    - CREATEDB
    - REPLICATION
  apt_repository_list: []
  bdr_database: cgms
  bdr_node_group: pgd_cluster
  bdr_node_groups:
  - name: pgd_cluster
    options:
      enable_proxy_routing: true
  - name: dc1_subgroup
    options:
      location: dc1
    parent_group_name: pgd_cluster
  - name: dc2_subgroup
    options:
      location: dc2
    parent_group_name: pgd_cluster
  - name: dc3_subgroup
```

```
    options:
      location: dc3
    parent_group_name: pgd_cluster
  bdr_version: '5'
  bdr_package_version: '5.9.*'
  pgd_proxy_package_version: '5.9*'
  pgdcli_package_version: '5.9*'
  default_pgd_proxy_options:
    listen_port: 6432
    read_listen_port: 6433
  edb_repositories:
  - standard
  - postgres_distributed
  failover_manager: pgd
  postgres_coredump_filter: '0xff'
  postgres_flavour: edbpge
  postgres_version: '16'
  postgres_package_version: '16.9*'
  postgres_port: 5432
  postgres_data_dir: '/pgdata/data'
  postgres_wal_dir: '/pgdata/pgwal'
  postgres_tablespaces:
    ts_tblspc1:
       location: '/pgdata/tblspc'
  preferred_python_version: python3
  use_local_repo_only: true
  use_volatile_subscriptions: false
  yum_repository_list:
  - EPEL
  log_line_prefix: '%t [%p]: [%l-1] user=%u db=%d app=%a client=%h '
  postgres_log_file: '/pgdata/log/postgres/postgres-%Y-%m-%d.log'
  postgres_conf_settings:
    work_mem: "'32MB'"
    log_min_duration_statement: "'500ms'"
    log_checkpoints: "'on'"
    log_disconnections: "'on'"
    log_connections: "'on'"
    log_lock_waits: "'on'"
    log_temp_files: "'0'"
    log_rotation_age: "'1d'"
    log_statement: "'ddl'"
    log_rotation_size: "'1GB'"
    bdr.default_sequence_kind: 'local'
    wal_sender_timeout: "'30min'"
    wal_receiver_timeout: "'30min'"
    max_replication_slots: '26'
  log_destination: 'stderr'
  postgres_hba_settings:
  - host all all <Masked_IP_Range>/0 scram-sha-256
  - host all all <Masked_IPv6_Address_1>/128 scram-sha-256
  - host all all <Masked_IPv6_Address_2>/128 scram-sha-256

locations:
- Name: dc1
- Name: dc2
- Name: dc3


instance_defaults:
  platform: bare
  vars:
    ansible_user: pgdinstall
```

```
instances:
- Name: pgda1
  public_ip: <Masked_IP_1>
  location: dc1
  node: 1
  role:
  - bdr
  - pgd-proxy
  vars:
    bdr_child_group: dc1_subgroup
    bdr_node_options:
      route_priority: 100
- Name: pgda2
  public_ip: <Masked_IP_2>
  location: dc1
  node: 2
  role:
  - bdr
  - pgd-proxy
  vars:
    bdr_child_group: dc1_subgroup
    bdr_node_options:
      route_priority: 100
- Name: pgdb1
  public_ip: <Masked_IP_3>
  location: dc2
  node: 3
  role:
  - bdr
  - pgd-proxy
  vars:
    bdr_child_group: dc2_subgroup
    bdr_node_options:
      route_priority: 100
- Name: pgdb2
  public_ip: <Masked_IP_4>
  location: dc2
  node: 4
  role:
  - bdr
  - pgd-proxy
  vars:
    bdr_child_group: dc2_subgroup
    bdr_node_options:
      route_priority: 100
- Name: pgdw
  public_ip: <Masked_IP_5>
  location: dc3
  node: 5
  role:
  - bdr
  - witness
  vars:
    bdr_child_group: dc3_subgroup
```

- Ensure that your DC1 FND and DC2 FND are deployed using Bare Metal servers.
- Ensure to install and verify the RPM. Here are the RPM details:

RPM type	RPM ID
FND RPM	cgms-5.1.1-16.pg.x86_64.rpm

RPM type	RPM ID
Postgres RPM	cgms-postgres-5.1.1-16.x86_64.rpm
Tools RPM	cgms-tools-5.1.1-16.x86_64.rpm

- We recommend that you perform the following logging configuration when installing EDB, in coordination with the EDB vendor:

```
log_line_prefix: '%t [%p]: [%l-1] user=%u db=%d app=%a client=%h '
  postgres_log_file: '/pgdata/log/postgres/postgres-%Y-%m-%d.log'
  postgres_conf_settings:
    work_mem: "'32MB'"
    log_min_duration_statement: "'500ms'"
    log_checkpoints: "'on'"
    log_disconnections: "'on'"
    log_connections: "'on'"
    log_lock_waits: "'on'"
    log_temp_files: "'0'"
    log_rotation_age: "'1d'"
    log_statement: "'ddl'"
    log_rotation_size: "'1GB'"
    bdr.default_sequence_kind: 'local'
    wal_sender_timeout: "'30min'"
    wal_receiver_timeout: "'30min'"
    max_replication_slots: '26'
  log_destination: 'stderr'
```

Follow the two-step task to configure HA and DR for Cisco IoT FND with Postgres:

Procedure

1. Configure HA and DR on DC1 FND
2. Configure HA and DR on DC2 FND

What to do next

Configure HA and DR on DC1 FND

Configure HA and DR on DC1 FND

Configure high availability (HA) and disaster recovery (DR) for Cisco IoT FND on DC1 using a Postgres database.

This task helps you configure HA and DR on DC1 FND.

Procedure

1. Install the Cisco IoT FND RPM. For more information, see Install Cisco IoT FND RPM.
   You've installed the RPM.
2. Navigate to `cd/opt/cgms/bin` using the terminal or command prompt.

   The command changes the current directory to `/opt/cgms/bin`.
3. Execute the script **`setupCgms.sh`**.

Example

```
[root@dc1fnd bin]# ./setupCgms.sh
01-20-2026 12:44:08 IST: INFO: ========== IoT-FND Setup Started -
2026-01-20-12-44-08 ==========
01-20-2026 12:44:08 IST: INFO: Log file:
/opt/cgms/bin/../server/cgms/log/cgms_setup.log

Are you sure you want to setup IoT-FND (y/n)? y

01-20-2026 12:44:10 IST: INFO: User response: y

Do you want to change the database settings (y/n)? y

01-20-2026 12:44:12 IST: INFO: User response: y

Do you want to configure Postgres with High Availability (y/n)? y

01-20-2026 12:44:18 IST: INFO: User response: y

Enter comma separated proxy hostnames or IPs with port in the format <Proxy
IP>:6432
[[<proxy_ip_1>]:6432,[<proxy_ip_2>]:6432,[<proxy_ip_3>]:6432,[<proxy_ip_4>]:6432]:
01-20-2026 12:44:55 IST: INFO: Postgres Proxy server & Port:
[<proxy_ip_1>]:6432,[<proxy_ip_2>]:6432,[<proxy_ip_3>]:6432,[<proxy_ip_4>]:6432

Enter comma separated DB node hostnames or IPs with port in the format <DB
Node IP>:5432
[[<db_node_ip_1>]:5432,[<db_node_ip_2>]:5432,[<db_node_ip_3>]:5432,[<db_node_ip_4>]:5432]:
01-20-2026 12:45:04 IST: INFO: Postgres HA DB Node Ip & Port:
[<db_node_ip_1>]:5432,[<db_node_ip_2>]:5432,[<db_node_ip_3>]:5432,[<db_node_ip_4>]:5432

Enter database SID [cgms]:
01-20-2026 12:45:24 IST: INFO: Database SID: cgms
01-20-2026 12:45:24 IST: INFO: Configuring database settings. This may take
a while. Please wait ...
01-20-2026 12:45:24 IST: INFO: Database settings configured.
01-20-2026 12:45:24 IST: INFO: Configuring BDR sequence settings for High
Availability...

Enter database password:
Re-enter database password:

01-20-2026 12:45:39 IST: INFO: Database connection details - Proxy IP:
<proxy_ip_1>, Port: 6432, SID: cgms, User: cgms_dev
01-20-2026 12:45:39 IST: INFO: Checking sequences galloc status on
<proxy_ip_1>:6432 for user cgms_dev in database cgms
01-20-2026 12:45:39 IST: INFO: Sequences galloc check result: '0' non-galloc
 sequences found
01-20-2026 12:45:39 IST: INFO: BDR sequences are already configured for HA
(all sequences are galloc and default sequence kind is galloc). Skipping
conversion.

Do you want to change the database password (y/n)? n

01-20-2026 12:46:05 IST: INFO: User response: n

Do you want to change the keystore password (y/n)? n

01-20-2026 12:46:07 IST: INFO: User response: n

Do you want to change the web application 'root' user password (y/n)? n
```

```
01-20-2026 12:46:09 IST: INFO: User response: n

Do you want to change IPAM and PSK Settings (y/n)? n

01-20-2026 12:46:10 IST: INFO: User response: n

Do you want to change the FTP settings (y/n)? n

01-20-2026 12:46:12 IST: INFO: User response: n

Do you want to change router CGDM protocol settings (y/n)? n

01-20-2026 12:46:14 IST: INFO: User response: n

Do you want to change router management mode [Demo, Bandwidth Optimized,
Default] (y/n)? n

01-20-2026 12:46:15 IST: INFO: User response: n

Do you want to configure timeseries database (y/n)? n

01-20-2026 12:46:17 IST: INFO: User response: n
01-20-2026 12:46:17 IST: INFO: Configuring timeseries flag  none in system
properties. This may take a while. Please wait...
01-20-2026 12:46:17 IST: INFO: timeseries flag none

Do you want to change log file settings? (y/n)? n

01-20-2026 12:46:19 IST: INFO: User response: n
01-20-2026 12:46:19 IST: INFO: ========== IoT-FND Setup Completed Successfully
 ==========
```

You've setup the cgms service for DC1 FND.

4. Execute **`db-init.sh`** to initiate the database.

Example

Note

- Initiate the database only if you're installing Cisco IoT FND for the first time.
- You must install psql v16 client on Cisco IoT FND VM/Bare Metal before running db-init command for setting up the database. For more information, see PostgreSQL documentation.

Example

```
./db-init <db host> <db port> <db service> <username> <password>
```

```
[root@dc1fnd bin]# ./db-init [xxxx:xxxx:xxxx:4::212:20] 6432 cgms cgms_dev
cgms123
07-16-2025 19:46:20 IST: INFO: Log file:
/opt/cgms/bin/../server/cgms/log/cgms_init.log
07-16-2025 19:46:20 IST: INFO: Performing init.
07-16-2025 19:46:21 IST: INFO: Init completed.
```

```
07-16-2025 19:46:21 IST: INFO: Creating Schema.
07-16-2025 19:46:21 IST: INFO: Schema creation completed.
```

The database has been initialized successfully and the schema creation is complete.

5. Execute **db-migrate** to migrate the database.

Example

```
[root@dc1fnd bin]# ./db-migrate
Enter database password: <cgms123>
02-11-2022 10:48:11 EST: INFO: Log file:
/opt/cgms/bin/../server/cgms/log/cgms_setup.log
02-11-2022 10:48:11 EST: INFO: Performing pre migration. This may take a
while. Please wait ...
02-11-2022 10:48:12 EST: INFO: Pre migration completed.
02-11-2022 10:48:12 EST: INFO: Performing migration. This may take a while.
Please wait ...
02-11-2022 10:48:58 EST: INFO: Migration completed.
02-11-2022 10:48:58 EST: INFO: Performing post migration. This may take a
while. Please wait ...
02-11-2022 10:51:57 EST: INFO: Post migration completed.
[root@fnd-rhel8-33 bin]#
```

The database migration has completed successfully, including pre-migration and post-migration steps.

6. Start the cgms service using the `service cgms start` command and verify the status using the `service cgms status` command.

Example

```
IoT-FND Version 5.1.1-7
01-20-2026 12:53:47 IST: INFO: IoT-FND database server:
[xxxx:xxxx:xxxx:xx]:6432,[xxxx:xxxx:xxxx:xx]:6432,[xxxx:xxxx:xxxx:xx]:6432,[xxxx:xxxx:xxxx:xx]:6432
01-20-2026 12:53:47 IST: INFO: IoT-FND database connection verified.
01-20-2026 12:53:49 IST: INFO: IoT-FND application server is up and running.
01-20-2026 12:53:52 IST: INFO: IoT-FND is up and running.
```

You've configured HA and DR on DC1 FND.

What to do next

Configure HA and DR on DC2 FND.

Configure HA and DR on DC2 FND

Set up high availability and disaster recovery for Cisco IoT FND with PostgreSQL on DC2.

This task helps you configure HA and DR on DC2 FND.

Procedure

---

1. Install the Cisco IoT FND RPM. For more information, see Install Cisco IoT FND RPM.
   You've installed the RPM.
2. Navigate to `cd/opt/cgms/bin` using the terminal or command prompt.

   The command changes the current directory to `/opt/cgms/bin`.

3. Execute the script **`setupcgms.sh`**.

Example

```
[root@dc1fnd bin]# ./setupCgms.sh
01-20-2026 12:44:08 IST: INFO: ========== IoT-FND Setup Started -
2026-01-20-12-44-08 ==========
01-20-2026 12:44:08 IST: INFO: Log file:
/opt/cgms/bin/../server/cgms/log/cgms_setup.log

Are you sure you want to setup IoT-FND (y/n)? y

01-20-2026 12:44:10 IST: INFO: User response: y

Do you want to change the database settings (y/n)? y

01-20-2026 12:44:12 IST: INFO: User response: y

Do you want to configure Postgres with High Availability (y/n)? y

01-20-2026 12:44:18 IST: INFO: User response: y

Enter comma separated proxy hostnames or IPs with port in the format <Proxy
IP>:6432
[[2001:db8::212:20]:6432,[2001:db8::212:21]:6432,[2001:db8::213:20]:6432,[2001:db8::213:21]:6432]:
01-20-2026 12:44:55 IST: INFO: Postgres Proxy server & Port:
[2001:db8::212:20]:6432,[2001:db8::212:21]:6432,[2001:db8::213:20]:6432,[2001:db8::213:21]:6432

Enter comma separated DB node hostnames or IPs with port in the format <DB
Node IP>:5432
[[2001:db8::212:20]:5432,[2001:db8::212:21]:5432,[2001:db8::213:20]:5432,[2001:db8::213:21]:5432]:
01-20-2026 12:45:04 IST: INFO: Postgres HA DB Node Ip & Port:
[2001:db8::212:20]:5432,[2001:db8::212:21]:5432,[2001:db8::213:20]:5432,[2001:db8::213:21]:5432

Enter database SID [cgms]:
01-20-2026 12:45:24 IST: INFO: Database SID: cgms
01-20-2026 12:45:24 IST: INFO: Configuring database settings. This may take
a while. Please wait ...
01-20-2026 12:45:24 IST: INFO: Database settings configured.
01-20-2026 12:45:24 IST: INFO: Configuring BDR sequence settings for High
Availability...

Enter database password:
Re-enter database password:

01-20-2026 12:45:39 IST: INFO: Database connection details - Proxy IP:
2001:db8::212:20, Port: 6432, SID: cgms, User: cgms_dev
01-20-2026 12:45:39 IST: INFO: Checking sequences galloc status on
2001:db8::212:20:6432 for user cgms_dev in database cgms
01-20-2026 12:45:39 IST: INFO: Sequences galloc check result: '0' non-galloc
 sequences found
01-20-2026 12:45:39 IST: INFO: BDR sequences are already configured for HA
(all sequences are galloc and default sequence kind is galloc). Skipping
conversion.

Do you want to change the database password (y/n)? n

01-20-2026 12:46:05 IST: INFO: User response: n

Do you want to change the keystore password (y/n)? n

01-20-2026 12:46:07 IST: INFO: User response: n
```

```
Do you want to change the web application 'root' user password (y/n)? n

01-20-2026 12:46:09 IST: INFO: User response: n

Do you want to change IPAM and PSK Settings (y/n)? n

01-20-2026 12:46:10 IST: INFO: User response: n

Do you want to change the FTP settings (y/n)? n

01-20-2026 12:46:12 IST: INFO: User response: n

Do you want to change router CGDM protocol settings (y/n)? n

01-20-2026 12:46:14 IST: INFO: User response: n

Do you want to change router management mode [Demo, Bandwidth Optimized,
Default] (y/n)? n

01-20-2026 12:46:15 IST: INFO: User response: n

Do you want to configure timeseries database (y/n)? n

01-20-2026 12:46:17 IST: INFO: User response: n
01-20-2026 12:46:17 IST: INFO: Configuring timeseries flag  none in system
properties. This may take a while. Please wait...
01-20-2026 12:46:17 IST: INFO: timeseries flag none

Do you want to change log file settings? (y/n)? n

01-20-2026 12:46:19 IST: INFO: User response: n
01-20-2026 12:46:19 IST: INFO: ========== IoT-FND Setup Completed Successfully
 ==========
```

You've setup the cgms service for DC2 FND.

4. Start the cgms service using the `systemctl start cgms` command and verify the status using the `systemctl status cgms` command.

Example

```
IoT-FND Version 5.1.1-7
01-20-2026 12:53:47 IST: INFO: IoT-FND database server:
[xxxx:xxxx:xxxx:xx]:6432,[xxxx:xxxx:xxxx:xx]:6432,[xxxx:xxxx:xxxx:xx]:6432,[xxxx:xxxx:xxxx:xx]:6432
01-20-2026 12:53:47 IST: INFO: IoT-FND database connection verified.
01-20-2026 12:53:49 IST: INFO: IoT-FND application server is up and running.
01-20-2026 12:53:52 IST: INFO: IoT-FND is up and running.
```

You've configured HA and DR on DC2 FND.

Database and Cisco IoT FND Migration

This topic contains all the migration related information to achieve HA and DR on Cisco IoT FND with Postgres.

Migrate Influx DB to Postgres

Use this task to migrate time-series data from InfluxDB to Postgres.

This document provides a step-by-step guide for migrating time-series data (events and metrics) from InfluxDB to Postgres within the Cisco IoT FND environment. The current version of the FND is `5.1.0-155` is used for the Influx DB to Postgres migration.

Historically, Cisco IoT FND used Influx DB as a specialized time-series database to handle high volumes of events and performance metrics. Starting Cisco IoT FND Release `5.1.1`, Cisco IoT FND consolidates this data into PostgreSQL to simplify the database architecture and management.

Before you begin

- Ensure the following CPU and Memory configuration for Cisco IoT FND. Here's an example configuration for Cisco IoT FND running `50,000` routers:

Configuration	Value
CPU Cores	24
Sockets	6
Core per Socket	4
IOMMU	Disabled
Memory	96 GB

- In the Data Retention page, configure the following values:

  Based on the image provided, here are the Data Retention settings in a tabular format:

Field	Value (days)
Keep Event data for	7
Keep Firmware Operation data for	7
Keep Historical Dashboard data for	7
Keep Dashboard data for	7
Keep Closed Issues data for	7
Keep JobEngine data for	7
Keep Historical Metrics Statistics for	30
Keep Device Network Statistics for	7
Keep Open Issues data for	7
Keep Service Provider down routers data for	7

- Disable the data retention in Influx DB using the `influxdb.conf` script. Here's an example:

```
$ vim /etc/influxdb/influxdb.conf

[retention]
  # Determines whether retention policy enforcement enabled.
  enabled = false

$ Restart the influxdb service - systemctl restart influxdb.service
```

- Run `du -sh /var/lib/influxdb/data/fnd/` to confirm the total volume of InfluxDB data (e.g., 25GB) that will be moved to the PostgresDB.

- The Influx DB to Postgres migration process filters events based on current device inventory. Events linked to deleted or decommissioned devices are excluded from the migration scope. For further technical details, refer to CSCwp87487.
- The following table outlines the specific measurements being migrated from the InfluxDB time-series database to the PostgreSQL relational database:

InfluxDB Table (Source)	PostgresDB Table (Destination)
`events`	`events`
`historicalnameddata`	`historical_named_data`
`historicalnameddata_da`	`historical_named_data_da`
`historicalnameddata_1_hr`	`historical_named_data_hr`
`metricshistory`	`metrics_history`
`mhistory_1_hr`	`metrics_history_hr`

The migration process is resource-intensive. It is mandatory to monitor system performance from the start of the migration until completion. If you encounter timeout or memory exhaustion issues, you must adjust the thread parameters in the `influx2postgres.properties` file.

Use these instructions to modify the `influx2postgres.properties` file:

Procedure

1. Stop the Cisco IoT FND CGMS service or the Cisco IoT FND container:

   Example

```
[root@iot-fnd ~]# docker exec -it fnd-container /etc/init.d/cgms status
IoT-FND Version 5.1.0-154
02-04-2026 05:08:58 EST: INFO: IoT-FND database server: 192.68.5.1:5432
02-04-2026 05:08:59 EST: INFO: IoT-FND database connection verified.
02-04-2026 05:08:59 EST: INFO: IoT FND timeseries database server: 192.68.5.1
02-04-2026 05:08:59 EST: INFO: IoT FND kapacitor server: 192.68.5.1
02-04-2026 05:09:00 EST: INFO: IoT-FND timeseries database/kapacitor connection
 verified.
02-04-2026 05:09:02 EST: INFO: IoT-FND application server is up and running.
02-04-2026 05:09:05 EST: INFO: IoT-FND is up and running.
```

2. Disable the Kapacitor tasks or stop the Kapacitor service:

   Example

```
[root@iot-fnd ~]# vim /etc/kapacitor/kapacitor.conf
[http]
bind-address = ":9092"

[root@iot-fnd ~]# systemctl restart kapacitor.service
```

```
[root@iot-fnd ~]# kapacitor list tasks
ID                    Type      Status    Executing Databases and Retention
 Policies
```

```
epmhistory_plc_cnt_1_da  batch    disabled  false    ["fnd"."epmhistoryrp"]
epmhistory_plc_cnt_1_hr  batch    disabled  false    ["fnd"."epmhistoryrp"]
epmhistory_rf_cnt_1_da   batch    disabled  false    ["fnd"."epmhistoryrp"]
epmhistory_rf_cnt_1_hr   batch    disabled  false    ["fnd"."epmhistoryrp"]
evallfiltercnt           batch     enabled   true      ["fnd"."eventsrp"]
evissuefiltercnt         batch     enabled   true      ["fnd"."eventsrp"]
evtypecnt                batch     enabled   true      ["fnd"."eventsrp"]
historicalnameddata_1_hr batch     enabled   true      ["fnd"."histdatarp"]
historicalnameddata_da   batch     enabled   true      ["fnd"."histdatarp"]
mhistory_1_hr            batch     enabled   true      ["fnd"."mhistoryrp"]
mhistory_da              batch     disabled  false     ["fnd"."mhistoryrp"]
[root@iot-fnd ~]#
[root@iot-fnd ~]# kapacitor disable <tasks_ID>
[root@iot-fnd ~]# systemctl restart kapacitor.service

OR

[root@iot-fnd ~]# systemctl status kapacitor.service
● kapacitor.service - Time series data processing engine.
   Loaded: loaded (/usr/lib/systemd/system/kapacitor.service; enabled; vendor
 preset: disabled)
   Active: active (running) since Fri 2025-06-20 15:27:18 CEST; 14h ago
     Docs: https://github.com/influxdb/kapacitor
 Main PID: 1397 (kapacitord)
    Tasks: 43 (limit: 614881)
   Memory: 2.0G
   CGroup: /system.slice/kapacitor.service
           └─1397 /usr/bin/kapacitord -config /etc/kapacitor/kapacitor.conf
-pidfile=/etc/kapacitor/kapacitord.pid
[root@iot-fnd ~]# systemctl mask kapacitor.service
Created symlink /etc/systemd/system/kapacitor.service → /dev/null.

[root@iot-fnd ~]# systemctl stop kapacitor.service

[root@iot-fnd ~]# systemctl status kapacitor.service
● kapacitor.service
   Loaded: masked (Reason: Unit kapacitor.service is masked.)
   Active: inactive (dead) since Sat 2025-06-21 05:39:07 CEST; 3s ago
 Main PID: 408336 (code=exited, status=0/SUCCESS)
[root@iot-fnd ~]#
```

3. Install or upgrade the `cgms-5.1.1-16x86_64.rpm` on Cisco IoT FND.

   Example

```
[root@iot-fnd ~]# rpm -qa | grep cgms-tools
[root@iot-fnd ~]# rpm -ivh cgms-tools-5.1.1-16.x86_64.rpm --force
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:cgms-tools-5.1.1-16              ################################# [100%]
[root@iot-fnd ~]# rpm -qa | grep cgms-tools
cgms-tools-5.1.1-16.x86_64
[root@iot-fnd ~]#
```

4. Navigate to the `/opt/cgms-tools/conf` path and modify the `influx2postgres.properties` file:

Example

```
[root@iot-fnd ~ conf]# cat influx2postgres.properties
# InfluxDB connection
influx-host=192.68.5.1
influx-port=8086
influx-database=fnd
influx-username=<user-name>
influx-password=<password>
influx-connect-timeout=60
influx-read-timeout=120

# PostgreSQL connection
postgres-host=192.68.5.1
postgres-port=5432
postgres-database=cgms
postgres-username=<user-name>
postgres-password=<password>

# Migration options
parallelize-measurements=true

# should be multiple of 4
fields_per_batch=12

logging-interval=10
read-chunk-size=25000
write-chunk-size=20000
read-min-threads=10
read-max-threads=20
read-queue-size=100
# Adjust write threads based on the server capacity
write-min-threads=16
write-max-threads=32
write-queue-size=100
await-termination-time=30
measurements=events,historicalnameddata,historicalnameddata_1_hr,historicalnameddata_da,metricshistory,mhistory_1_hr
[root@iot-fnd ~ conf]#
```

5. Restart the Influx DB service. If any service is consuming the memory, restarting frees up the memory.

Example

```
[root@iot-fnd ~]#systemctl restart influxdb.service

Verify the free memory

[root@iot-fnd ~]# free -m
```

6. Navigate to the script path and execute the `influx2postgres.sh` script on OVA.

Example

```
[root@iot-fnd ~]# cd /opt/cgms-tools/bin
[root@<ip-address> bin]#./influx2postgres.sh

Note: During the migration, CPU slowly increase to 90% to 100% and immediately
```

```
 come down to 90% to 70% and vice versa.
Memory also will slowly increas to 85% to it remains to 85% till it completed.
```

You've successfully migrated time-series data from InfluxDB to Postgres.

What to do next
Example log is provided in the **Example Influx DB to Postgres migration log** topic.

Verify the data in the Postgres DB

Use this task to verify the data in the Postgres DB after the Influx DB to Postgres migration.

Procedure

1. Optional: Verify the data in the Postgres DB using a standalone Cisco IoT FND using these steps:
   a) Install the RHEL 8.10 OS. The standalone VMs can RHEL 8.10 and later OS.
   b) Install the Cisco IoT FND RPM `cgms-5.1.0-154.pg.x86_64.rpm`

      Example

      ```
      [root@iot-fnd home]# rpm -ivh cgms-5.1.0-154.pg.x86_64.rpm --force
      Verifying...                          ################################
      [100%]
      Preparing...                          ################################
      [100%]
      Updating / installing...
         1:cgms-5.1.0-74              ################################ [100%]
      [root@<ip-address> home]# rpm -qa | grep cgms-tools
      cgms-5.1.0-74.pg.x86_64.rpm
      [root@<ip-address> home]#
      ```

   c) Copy the Cisco IoT FND certificates to the standalone Cisco IoT FND:

      Example

      ```
      [root@iot-fnd ~]# cd /opt/cgms/server/cgms/conf/
      [root@<ip-address> conf]# ll
      cgms.properties
      cgms_keystore
      ```

   d) Execute the `setupCgms.sh` script. When prompted, configure the PostgreSQL database to point to the existing FND server and set the time-series database option to **No**.

      Example

      ```
      [root@iot-fnd ~]# cd /opt/cgms/bin/
      [root@iot-fnd ~]# ./setupCgms.sh
      06-20-2025 14:32:05 EDT: INFO: ========== IoT-FND Setup Started -
      2025-06-20-14-32-05 ==========
      06-20-2025 14:32:05 EDT: INFO: Log file:
      /opt/cgms/bin/../server/cgms/log/cgms_setup.log

      Are you sure you want to setup IoT-FND (y/n)? y
      ```

```
06-20-2025 14:32:50 EDT: INFO: User response: y

Do you want to change the database settings (y/n)? y

06-20-2025 14:32:52 EDT: INFO: User response: y

Do you want to configure Postgres with High Availability (y/n)? n

06-20-2025 14:32:54 EDT: INFO: User response: n

Do you want to use custom database connection string (y/n)? n

06-20-2025 14:32:57 EDT: INFO: User response: n

Enter database server hostname or IP [<ip-address>]: <remote-ip-address>
06-20-2025 14:33:01 EDT: INFO: Database server: <remote-ip-address>

Enter database server port [5432]:
06-20-2025 14:33:03 EDT: INFO: Database server port: 5432

Enter database SID [cgms]:
06-20-2025 14:33:04 EDT: INFO: Database SID: cgms

Do you wish to configure another database server for this IoT-FND (y/n)?
n

06-20-2025 14:33:06 EDT: INFO: User response: n
06-20-2025 14:33:06 EDT: INFO: Configuring database settings. This may take
 a while. Please wait ...
06-20-2025 14:33:07 EDT: INFO: Database settings configured.

Do you want to change the database password (y/n)? n

06-20-2025 14:33:08 EDT: INFO: User response: n

Do you want to change the keystore password (y/n)? n

06-20-2025 14:33:09 EDT: INFO: User response: n

Do you want to change the web application 'root' user password (y/n)? n

06-20-2025 14:33:09 EDT: INFO: User response: n

Do you want to change IPAM and PSK Settings (y/n)? n

06-20-2025 14:33:10 EDT: INFO: User response: n

Do you want to change the FTP settings (y/n)? n

06-20-2025 14:33:11 EDT: INFO: User response: n

Do you want to change router CGDM protocol settings (y/n)? n

06-20-2025 14:33:12 EDT: INFO: User response: n

Do you want to change router management mode [Demo, Bandwidth Optimized,
Default] (y/n)? n

06-20-2025 14:33:13 EDT: INFO: User response: n

Do you want to configure timeseries database (y/n)? n
```

```
06-20-2025 14:33:13 EDT: INFO: User response: n
06-20-2025 14:33:13 EDT: INFO: Configuring timeseries flag  none in system
 properties. This may take a while. Please wait...
06-20-2025 14:33:13 EDT: INFO: timeseries flag none

Do you want to change log file settings? (y/n)? n

06-20-2025 14:33:14 EDT: INFO: User response: n
06-20-2025 14:33:14 EDT: INFO: ========== IoT-FND Setup Completed
Successfully ==========
[root@iot-fnd ~]#
```

2. Perform the database migration using `db-migrate`:

   Example

   ```
   [root@scale-pg-fnd ~]# /opt/cgms/bin/db-migrate
   Enter database password: <cgms123>
   ```

3. Start the cgms service:

   Example

   ```
   [root@iot-fnd ~]# service cgms start
   ```

4. Log in to Cisco IoT FND and verify the events, dashboard charts, and graphs.

You've successfully verified the data in the Postgres DB after the Influx DB to Postgres migration.

Migrate data from standalone to EDB Postgres

Use this procedure to migrate Cisco IoT FND data from a standalone database to an external EDB Postgres environment. This process involves configuring the database connection, executing the migration script, and performing a backup and restore sequence to ensure data integrity and operational readiness.

This procedure configures the Cisco IoT FND to connect to an external EDB Postgres database and executes a migration script to transfer the necessary data to the new environment. It subsequently ensures data integrity and operational readiness by performing a complete backup and restore sequence on the database server.

Procedure

---

1. Configuration and Migration on DC2 FND:
   a) Navigate to the binary directory and execute the setup script to configure the database connection.

      Example

      ```
      [root@dc2fnd opt]# cd /opt/cgms/bin/
      [root@dc2fnd bin]# ./setupCgms.sh

      #02-02-2026 20:05:03 IST: INFO: ========== IoT-FND Setup Started -
      2026-02-02-20-05-03 ==========
      02-02-2026 20:05:03 IST: INFO: Log file:
      /opt/cgms/bin/../server/cgms/log/cgms_setup.log
      ```

```
Are you sure you want to setup IoT-FND (y/n)? y

02-02-2026 20:05:05 IST: INFO: User response: y

Do you want to change the database settings (y/n)? y

02-02-2026 20:05:07 IST: INFO: User response: y

Do you want to configure Postgres with High Availability (y/n)? n

02-02-2026 20:05:08 IST: INFO: User response: n

Do you want to use custom database connection string (y/n)? n

02-02-2026 20:05:12 IST: INFO: User response: n

Enter database server hostname or IP : [<IP-address>]
02-02-2026 20:05:24 IST: INFO: Database server: [<IP-address>]

Enter database server port [420]: 5432
02-02-2026 20:05:31 IST: INFO: Database server port: 5432

Enter database SID [cgms]:
02-02-2026 20:05:32 IST: INFO: Database SID: cgms

Do you wish to configure another database server for this IoT-FND (y/n)?
n

02-02-2026 20:05:35 IST: INFO: User response: n
02-02-2026 20:05:35 IST: INFO: Configuring database settings. This may take
 a while. Please wait ...
02-02-2026 20:05:36 IST: INFO: Database settings configured.

Do you want to change the database password (y/n)? n

02-02-2026 20:05:39 IST: INFO: User response: n

Do you want to change the keystore password (y/n)? n

02-02-2026 20:05:41 IST: INFO: User response: n

Do you want to change the web application 'root' user password (y/n)? n

02-02-2026 20:05:42 IST: INFO: User response: n

Do you want to change IPAM and PSK Settings (y/n)? n

02-02-2026 20:05:43 IST: INFO: User response: n

Do you want to change the FTP settings (y/n)? n

02-02-2026 20:05:44 IST: INFO: User response: n

Do you want to change router CGDM protocol settings (y/n)? n

02-02-2026 20:05:45 IST: INFO: User response: n

Do you want to change router management mode [Demo, Bandwidth Optimized,
Default] (y/n)? n

02-02-2026 20:05:45 IST: INFO: User response: n
```

```
Do you want to configure timeseries database (y/n)? n

02-02-2026 20:05:46 IST: INFO: User response: n
02-02-2026 20:05:46 IST: INFO: Configuring timeseries flag  none in system
 properties. This may take a while. Please wait...
02-02-2026 20:05:47 IST: INFO: timeseries flag none

Do you want to change log file settings? (y/n)? n

02-02-2026 20:05:47 IST: INFO: User response: n
02-02-2026 20:05:47 IST: INFO: ========== IoT-FND Setup Completed
Successfully ==========
```

b) Execute the migration script to migrate the schema and data.

Example

```
[root@dc2fnd bin]# ./db-migrate
Enter database password:
# Wait for "Migration completed" and "Post migration completed" messages.
```

2. Run the backup script pointing to the specific database server IP.

Example

```
[root@pgda1 ~]# cd /opt/cgms-postgres/scripts/
[root@pgda1 scripts]# ./backupdb.sh cgms cgms_dev /opt/5.1.1_backup_file
<Database_Server_IP> 5432

# Enter password when prompted.
# Wait for "Backup completed successfully!"
```

3. Restore the database.

Example

```
[root@pgda1 scripts]# ./restoredb.sh cgms cgms_dev /opt/5.1.1_backup_file
localhost 5432

# Enter password for user cgms_dev when prompted.
# Wait for "Restore completed successfully!" and "Firmware references cleaned
 successfully!"
```

Example

Note

Restore the data in the EDB write leader.

You've migrated the data from standalone to EDB Postgres.

Monitor servers

Learn how to monitor HA databases and nodes using Cisco IoT FND. This procedure explains how to access the server inventory and interpret key status fields such as connectivity, IP addresses, and open issues to ensure the health of your network infrastructure.

The task aims to help you monitor the HA databases and the nodes using Cisco IoT FND.

Procedure

1. From the Cisco IoT FND menubar, choose **Devices** > **Servers**.
2. A detailed inventory of all the servers is displayed. Here are the fields and their descriptions in collapsed view:

Field	Field Description
Name	Displays the name assigned to the server
EID	Displays the EID associated with the server.
Status	Displays the status of the server if it is up or down.
Last Heard	Displays the time at which Cisco IoT FND communicated with the server for the last time.
IP	Displays the IP addresses associated with the server.
Open Issues	Displays open issues that needs to be resolved in the server.
Labels	Displays the lables added to the server, if any.

3. Click the + icon adjacent to the **Name** column to expand a server's details. Here are the fields and their descriptions in expanded view:

Field	Field Description
NAME	Displays the name assigned to the server
EID	Displays the EID associated with the server.
STATUS	Displays the status of the server if it is up or down.
IP	Displays the IP addresses associated with the server.
TYPE	Displays if the server is either a node or a PGD proxy.

You've successfully learnt how to monitor the servers once HA is configured.

Configure self-signed certificates

Use this procedure to configure self-signed certificates for high availability AppServer communications in Cisco IoT FND. This setting allows the application to use self-signed certificates instead of default custom certificates to facilitate secure communication between HA server nodes.

This task guides you to configure self-signed certificates for HA AppServer communications instead of the default custom certificates.

Procedure

1. From the Cisco IoT FND menubar, choose **ADMIN** > **Certificates**.
2. Click the **Server Settings** tab.
3. Check the **Allow self-signed certificate for High Availability AppServer Communications** check box to allow self-signed certificates for HA AppServer communications.

Example Influx DB to Postgres migration log

This topic contains an example logs from Influx DB to Postgres migration process.

```
[root@iot-fnd log]# /opt/cgms-tools/log/influx2postgres.log

====== sample log =========
2025-08-05 17:09:17 INFO  Influx2PostgresMigrator:102 - ##### Migration of
records from Influx to Postgres process: Started 2025-08-05T15:09:17.144Z #####

2025-08-05 17:09:27 INFO  Influx2PostgresMigrator:119 - parallelize : true
2025-08-05 17:09:27 INFO  Influx2PostgresMigrator:131 - Total number of
measurements to migrate: 6
2025-08-05 17:09:27 INFO  Influx2PostgresMigrator:132 - Measurements : [events,
 historicalnameddata, historicalnameddata_1_hr, historicalnameddata_da,
metricshistory, mhistory_1_hr]

2025-08-05 17:09:39 INFO  Influx2PostgresMigrator:157 - Measurement: events
has a retention period of 90 days.
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:162 - Successfully updated
retention period for postgres table: events
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:157 - Measurement:
historicalnameddata has a retention period of 7 days.
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:162 - Successfully updated
retention period for postgres table: historical_named_data
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:157 - Measurement:
historicalnameddata_1_hr has a retention period of 90 days.
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:162 - Successfully updated
retention period for postgres table: historical_named_data_hr
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:157 - Measurement:
historicalnameddata_da has a retention period of 90 days.
2025-08-05 17:09:40 INFO  Influx2PostgresMigrator:162 - Successfully updated
retention period for postgres table: historical_named_data_da
2025-08-05 17:09:52 INFO  Influx2PostgresMigrator:157 - Measurement:
metricshistory has a retention period of 30 days.
2025-08-05 17:09:52 INFO  Influx2PostgresMigrator:162 - Successfully updated
retention period for postgres table: metrics_history
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:157 - Measurement:
mhistory_1_hr has a retention period of 30 days.
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:162 - Successfully updated
retention period for postgres table: metrics_history_hr
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:178 - Parallel migration
enabled at measurements level. Using maximum of 20 read threads and 32 write
threads.
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:246 - Starting migration for
 measurement: events
```

```
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:246 - Starting migration for
 measurement: historicalnameddata
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:246 - Starting migration for
 measurement: historicalnameddata_1_hr
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:246 - Starting migration for
 measurement: mhistory_1_hr
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:246 - Starting migration for
 measurement: metricshistory
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:246 - Starting migration for
 measurement: historicalnameddata_da
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:249 - Migration for
measurement: mhistory_1_hr started at 2025-08-05T15:10:00.811Z
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:249 - Migration for
measurement: historicalnameddata_1_hr started at 2025-08-05T15:10:00.812Z
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:249 - Migration for
measurement: metricshistory started at 2025-08-05T15:10:00.816Z
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:249 - Migration for
measurement: historicalnameddata started at 2025-08-05T15:10:00.820Z
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:249 - Migration for
measurement: events started at 2025-08-05T15:10:00.824Z
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:249 - Migration for
measurement: historicalnameddata_da started at 2025-08-05T15:10:00.830Z
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres
 table metrics_history_hr before migration: 64
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres
 table events before migration: 149
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres
 table historical_named_data_da before migration: 146748
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres
 table historical_named_data_hr before migration: 3471494
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres
 table historical_named_data before migration: 14524653
2025-08-05 17:10:00 INFO  Influx2PostgresMigrator:263 - Max ID value in Postgres
 table metrics_history before migration: 431

2025-06-21 15:48:36 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:48:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:48:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:48:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 4000/132270 records copied (3.02%)
2025-06-21 15:48:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:48:36 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:48:46 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:48:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:48:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:48:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 10000/132270 records copied (7.56%)
2025-06-21 15:48:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:48:46 INFO  Influx2PostgresStatsLogger:132 -
```

```
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:48:56 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:48:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:48:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:48:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 20000/132270 records copied (15.12%)
2025-06-21 15:48:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:48:56 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:49:06 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:49:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:49:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:49:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 28000/132270 records copied (21.17%)
2025-06-21 15:49:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:49:06 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:49:16 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:49:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:49:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:49:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 34000/132270 records copied (25.70%)
2025-06-21 15:49:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:49:16 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:49:26 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:49:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:49:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:49:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 40000/132270 records copied (30.24%)
2025-06-21 15:49:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:49:26 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################
```

```
2025-06-21 15:49:36 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:49:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:49:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:49:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 50000/132270 records copied (37.80%)
2025-06-21 15:49:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:49:36 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:49:46 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:49:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:49:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:49:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 54000/132270 records copied (40.83%)
2025-06-21 15:49:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:49:46 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:49:56 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:49:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:49:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:49:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 64000/132270 records copied (48.39%)
2025-06-21 15:49:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:49:56 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:50:06 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:50:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:50:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:50:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 70000/132270 records copied (52.92%)
2025-06-21 15:50:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:50:06 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:50:16 INFO  Influx2PostgresStatsLogger:124 -
```

```
############################################ INFLUX TO POSTGRES
############################################
2025-06-21 15:50:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:50:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:50:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 80000/132270 records copied (60.48%)
2025-06-21 15:50:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:50:16 INFO  Influx2PostgresStatsLogger:132 -
############################################ INFLUX TO POSTGRES
############################################

2025-06-21 15:50:26 INFO  Influx2PostgresStatsLogger:124 -
############################################ INFLUX TO POSTGRES
############################################
2025-06-21 15:50:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:50:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:50:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 84000/132270 records copied (63.51%)
2025-06-21 15:50:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:50:26 INFO  Influx2PostgresStatsLogger:132 -
############################################ INFLUX TO POSTGRES
############################################

2025-06-21 15:50:36 INFO  Influx2PostgresStatsLogger:124 -
############################################ INFLUX TO POSTGRES
############################################
2025-06-21 15:50:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:50:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:50:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 90000/132270 records copied (68.04%)
2025-06-21 15:50:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:50:36 INFO  Influx2PostgresStatsLogger:132 -
############################################ INFLUX TO POSTGRES
############################################

2025-06-21 15:50:46 INFO  Influx2PostgresStatsLogger:124 -
############################################ INFLUX TO POSTGRES
############################################
2025-06-21 15:50:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:50:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:50:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 100000/132270 records copied (75.60%)
2025-06-21 15:50:46 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:50:46 INFO  Influx2PostgresStatsLogger:132 -
############################################ INFLUX TO POSTGRES
############################################

2025-06-21 15:50:56 INFO  Influx2PostgresStatsLogger:124 -
############################################ INFLUX TO POSTGRES
############################################
```

```
2025-06-21 15:50:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:50:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:50:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 108000/132270 records copied (81.65%)
2025-06-21 15:50:56 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:50:56 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:51:06 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:51:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:51:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:51:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 114000/132270 records copied (86.19%)
2025-06-21 15:51:06 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:51:06 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:51:16 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:51:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:51:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:51:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 120000/132270 records copied (90.72%)
2025-06-21 15:51:16 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:51:16 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:51:26 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
##############################################
2025-06-21 15:51:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:51:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:51:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 130000/132270 records copied (98.28%)
2025-06-21 15:51:26 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:51:26 INFO  Influx2PostgresStatsLogger:132 -
############################################## INFLUX TO POSTGRES
##############################################

2025-06-21 15:51:29 INFO  Influx2PostgresMigrator:331 - Migration completed
for measurement: events in 183003 ms

2025-06-21 15:51:36 INFO  Influx2PostgresStatsLogger:124 -
############################################## INFLUX TO POSTGRES
```

```
##########################################
2025-06-21 15:51:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data: 5060/5060 records copied (100.00%)
2025-06-21 15:51:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_hr: 9960/9960 records copied (100.00%)
2025-06-21 15:51:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION] events:
 132270/132270 records copied (100.00%)
2025-06-21 15:51:36 INFO  Influx2PostgresStatsLogger:129 - [MIGRATION]
historical_named_data_da: 380/380 records copied (100.00%)
2025-06-21 15:51:36 INFO  Influx2PostgresStatsLogger:132 -
########################################## INFLUX TO POSTGRES
##########################################

2025-06-21 15:51:39 INFO  Influx2PostgresMigrator:182 - ##### Migration of
records from Influx to Postgres process: Completed 2025-06-21T13:51:39.535Z
Total time taken: 00:03:14 #####
```

Troubleshoot stale server entries

Use this procedure to troubleshoot stale server entries in Cisco IoT FND following a migration from InfluxDB to Postgres. Learn how to identify and remove outdated server records displayed with a "DOWN" status to ensure your device inventory remains accurate and up to date.

This topic helps you troubleshoot an issue that occurs post performing a migration from InfluxDB to Postgres that points the VM PG database to a standalone Cisco IoT FND.

You may notice that the **Servers** page is displaying the old servers with a status of **DOWN**.

You may observe the following specific server/device entries listed as DOWN:

- `IoT-FND-Kapacitor+192.68.5.1`
- `IoT-FND-TSDB+192.68.5.1`

Procedure

1. From the Cisco IoT FND menubar, choose **Devices** > **Servers**.
2. Select a device with the status **DOWN**.
3. Click the **More Actions** drop down list and click **Remove Devices**. Click **Yes** when asked to confirm.

You'll no longer view the servers that are down.

Troubleshoot CPU and memory issues

Use this topic to troubleshoot CPU and Memory issues while setting up HA on Cisco IoT FND with Postgres

This task helps you troubleshoot high CPU usage and memory issues while configuring HA and DR on Cisco IoT FND.

If you encounter timeout or memory issues, adjust the parameters in the `influx2postgres.properties` file.

Procedure

1. Navigate to `cd /opt/cgms-tools/conf` path.

2. Execute the `cat influx2postgres.properties` command.
3. Edit the following parameters and change them from their default values:

   Example

```
# should be multiple of 4
fields_per_batch=12
```

```
# Migration options
parallelize-measurements=false
```

```
# Adjust write threads based on the server capacity
write-min-threads=16
write-max-threads=32
write-queue-size=100
measurements=events,historicalnameddata,historicalnameddata_1_hr,historicalnameddata_da,metricshistory,mhistory_1_hr
```

4. If errors regarding measurements or partially migrated data occur, wait for the script to complete, delete the affected data in the PostgreSQL database, and re-run the `influx2postgres.sh` script for that specific measurement.
5. Ensure the PostgreSQL database contains the relevant historical data (named data, daily, or hourly) before deleting any data. Here's an example showing how to delete data in Postgres:

   Example

```
[root@iot-fnd ~]# psql -U <user-name> cgms

Password for user <user-name>:
psql (16.7)
Type "help" for help.

cgms=# delete from events;

cgms=# delete from metrics_history;

cgms=# delete from metrics_history_hr;
```

CPU and Memory issues are resolved.