Cisco IoT FND Postgres and Influx DB Deployment with Integrated Application Management on OVA, Release 4.3.1 and Later

First Published: 2018-08-31

Last Modified: 2024-05-15

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

© 2024 Cisco Systems, Inc. All rights reserved.

CONTENTS

CHAPTER 1 **Overview** 1

CHAPTER 2 **OVA Images and Upgrade Scripts Verification** 3

- Introduction 3
- Verifying the OVA Signature 4
- Verifying the Upgrade-Scripts RPM Signature 5
- Verifying the CGMS Tools RPM for Postgres Signature 6

CHAPTER 3 **Installing the OVA** 9

CHAPTER 4 **Installing Custom CA Certificates and Importing SUDI Certificate** 17

- Changing Password 20

CHAPTER 5 **Configuring IoT FND for IPv6 Tunnel Provisioning and Registration** 23

CHAPTER 6 **Starting and Stopping FND** 25

CHAPTER 7 **Starting and Stopping Fog Director** 27

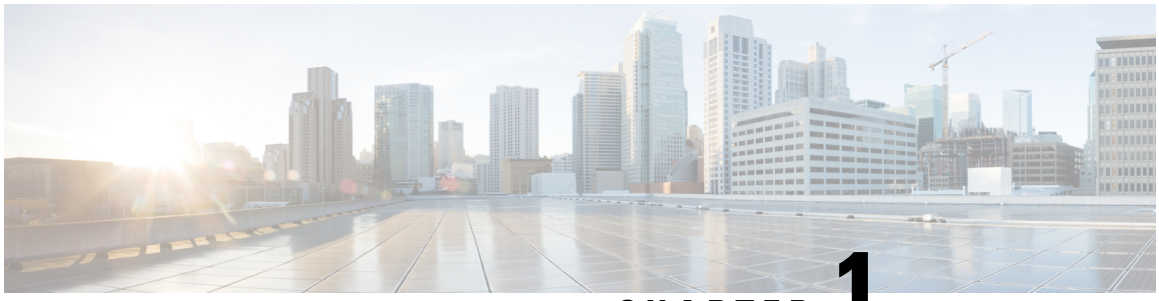
CHAPTER 8 **Upgrading IoT FND OVA** 29

- Pre-Upgrade Checklist 29
- Upgrading the Database and Docker Server Image 33
- Upgrading IoT FND and FD Container Images 45
- Post-Upgrade Checklist 48
- Upgrading IoT FND from 4.5.1 to later releases and Updating RHEL OS 49

CHAPTER 9 Obtaining Status of All Services Running on the Host 53

CHAPTER 10 Backup and Restore 55

CHAPTER 11 Setting the Time and Timezone Using NTP Service 57



CHAPTER 1

Overview

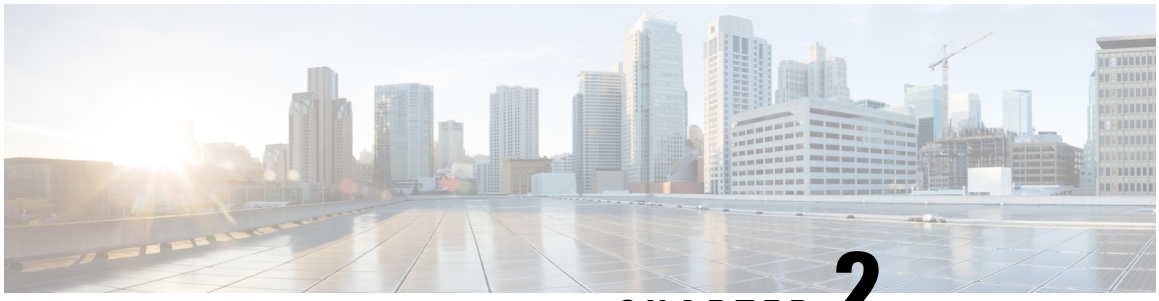
This document provides the steps required to install the Cisco IoT Field Network Director (Cisco IoT FND) Release 4.3.1 and Later application with Integrated Application Management (Fog Director) on an Open Virtual Appliance (OVA), VMware ESXi 5.5 or 6.0. You use the same instructions to install both VMware versions.

Note For information about installing Cisco IoT FND and Oracle on an OVA for Release 4.3 and Later, refer to the following guides:

- Cisco IoT FND Deployment on an Open Virtual Appliance, VMware ESXi 5.5/6.0
- Cisco IoT Field Network Director Installation Guide-Oracle Deployment, Release 4.3.x and Later

For an overview of the features and functionality of the IoT FND application and details on how to configure features and manage Cisco IoT FND after its installation, refer to the Cisco IoT Field Network Director User Guide.

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



CHAPTER 2

OVA Images and Upgrade Scripts Verification

- Introduction, on page 3
- Verifying the OVA Signature, on page 4
- Verifying the Upgrade-Scripts RPM Signature, on page 5
- Verifying the CGMS Tools RPM for Postgres Signature, on page 6

Introduction

Starting from Cisco IoT FND 4.9.0, you can verify the integrity of the OVA images and upgrade scripts before the installation or upgrade of IoT FND.

For more information, refer to:

- Verifying the OVA Signature, on page 4
- Verifying the Upgrade-Scripts RPM Signature, on page 5
- Verifying the CGMS Tools RPM for Postgres Signature, on page 6

Note From FND release 4.12 onwards, the Secure Hash Algorithm is SHA256 and the earlier FND releases use SHA1.

Table 1: OVA Images and Upgrade Scripts Zip File Contents

Zip File Contents	Description
CISCO-IOTFND-V-K9-<release>-<build number>.zip	Includes Oracle for Mesh management (CGR, IR5xx) use case.
1. iot-fnd-oracle-<release>-<build number>_SHA1_signed.ova **2.** iot-tps-<release>-<build number>_SHA1_signed.ova	
CISCO-IOTFND-VPI-K9-<release>-<build number>.zip	Includes Postgres / Influx for gateway management (IR8xx, IR1101, IC3K) use case.

Zip File Contents	Description
1. iot-fnd-<release>-<build number>_SHA256_signed.ova 2. iot-tps-<release>-<build number>_SHA256_signed.ova	
CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip **Attention** The CGMS tools file is bundled with `CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip`.	Includes cgms tools rpm for Postgres deployments.
1. cgms-tools-<release>-<build number>.x86_64.rpm 2. 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/. 3. cisco_openpgp_verify_release.py — Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate. 4. cisco_openpgp_verify_release.py.signature — Signature generated for the script cisco_openpgp_verify_release.py. 5. FND-rel-binary.gpg — Open-pgp public key is used for verification of signed RPM. 6. 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.
1. upgrade-ova-<release>-<build number>.rpm — Signature embedded RPM image. 2. 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/. 3. cisco_openpgp_verify_release.py — Signature verification program for verifying the Open-pgp Complaint Public Key against x.509 end-entity certificate. 4. cisco_openpgp_verify_release.py.signature — Signature generated for the script cisco_openpgp_verify_release.py. 5. FND-rel-binary.gpg — Open-pgp public key is used for verification of signed RPM. 6. FND-rel-ascii.gpg — Open-pgp public key is used for verification of signed RPM.	

Verifying the OVA Signature

To verify the OVA signature:

Step 1 Install the `ovftool`.

Step 2 Run the command to verify the signed `ova` file.

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

Verifying the Upgrade-Scripts RPM Signature

Prerequisites:

- Python 2.7.x
- OpenSSL
- Verification scripts running on customer-premises need internet connection to reach Cisco to download root and sub-CA certs

To verify the upgrade-scripts RPM signature:

Step 1 Unzip the file `CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>.zip`.

Step 2 Change directory (cd) to `CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-<release>-<build number>` folder.

Step 3 Extract the public key from the public cert:

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

Expected Result:

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

Step 4 Verify the verification script using the public key and the signature files.

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

Expected Result:

```
Verified OK
```

Step 5 Verify if the delivered binary and ASCII keys have matching fingerprints.

a) `gpg FND-rel-binary.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name  (FND.rel) identity-name@cisco.com
```

b) `gpg FND-rel-ascii.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name (FND.rel) identity-name@cisco.com
```

Step 6 Verify the binary GPG key against EE cert.

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

Expected Result:

```
 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.
```

Step 7 Verify the RPM Signature using the GPG ASCII key.

```
sudo rpm --import FND-rel-ascii.gpg
rpm -K upgrade-ova-<release>-<build number>.rpm
```

Expected Result:

```
upgrade-ova-<release>-<build number>.rpm: rsa sha1 (md5) pgp md5 OK
```

Step 8 Once the RPM is verified, you can upgrade OVA using the RPM.

Verifying the CGMS Tools RPM for Postgres Signature

Prerequisites:

- Python 2.7.x
- OpenSSL
- Verification scripts running on customer-premises need an internet connection to reach Cisco to download root and sub-CA certs

To verify the cgms tools rpm for Postgres signature:

Step 1 Unzip the file `CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip .`

Step 2 Change directory (cd) to `CISCO-IOTFND-VPI-K9-CGMS-TOOLS-<release>-<build number>.zip` folder.

Step 3 Extract the public key from the public cert:

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

Expected Result:

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

Step 4 Verify the verification script using the public key and the signature files.

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

Expected Result:

```
Verified OK
```

Step 5 Verify if the delivered binary and ASCII keys have matching fingerprints.

a) `gpg FND-rel-binary.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name  (FND.rel) identity-name@cisco.com
```

b) `gpg FND-rel-ascii.gpg`

Expected Result:

```
pub 2048R/F7D5ED29 2017-01-01 identity-name (FND.rel) identity-name@cisco.com
```

Step 6 Verify the binary GPG key against EE cert.

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

Expected Result:

```
 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.
```

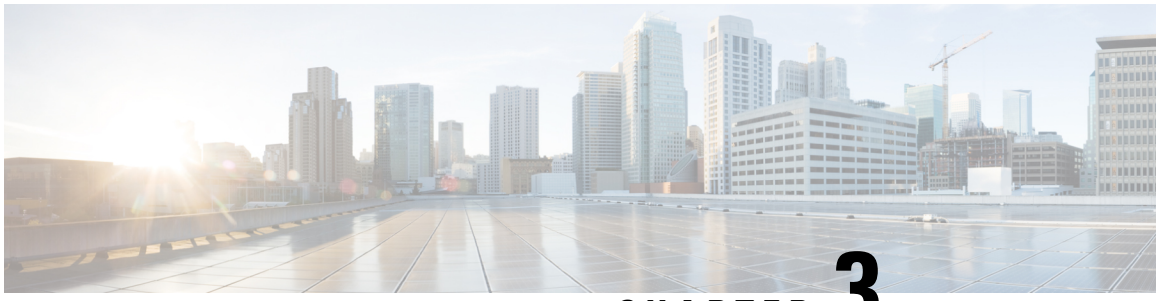
Step 7 Verify the RPM Signature using the GPG ASCII key.

```
sudo rpm --import FND-rel-ascii.gpg
rpm -K cgms-tools-<release>-<build number>.x86_64.rpm
```

Expected Result:

```
upgrade-cgms-tools-<release>-<build number>.x86_64.rpm: rsa sha1 (md5) pgp md5 OK
```

Step 8 Once the RPM is verified, you can upgrade cgms-tools using the RPM.



CHAPTER 3

Installing the OVA

Prerequisites

- Log in to the IP address of a VMware ESXi server running 6.5 and above via a web browser with your user credentials (username and password).
- Ensure that you meet the VMware server machine (VM CPU and memory) requirements as listed below.
  - 24 GB memory
  - 4 vCPUs
  - Hard disk: 450 GB

To install the OVA:

Attention From IoT FND 4.12 onwards, use the following credentials for SSH access after installing OVA. The existing credentials username/password (root/cisco123) is disabled for 4.12 and later releases:

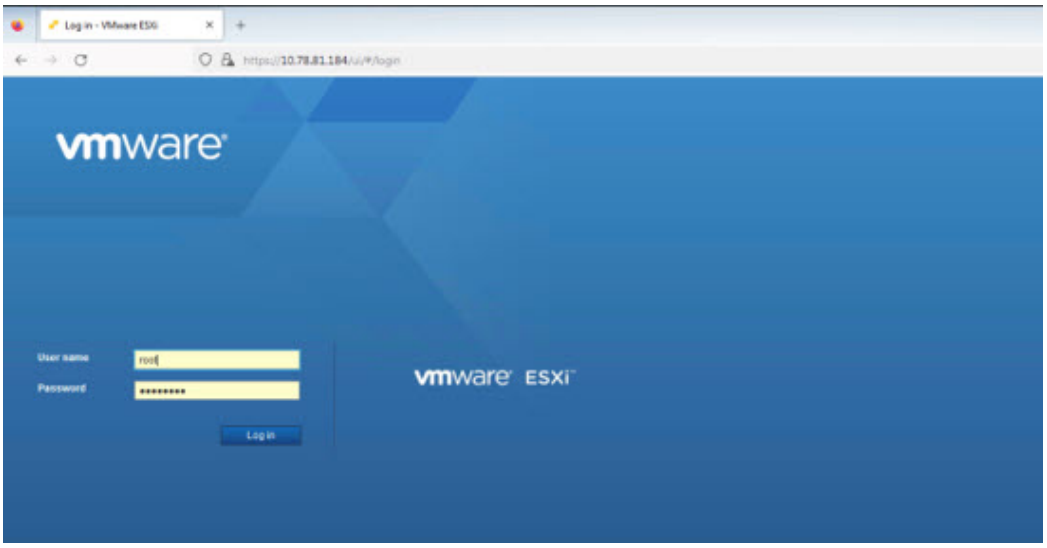
- Username: fnduser
- Password: C!sco123

See Step 10 for guidelines to reset the default password.

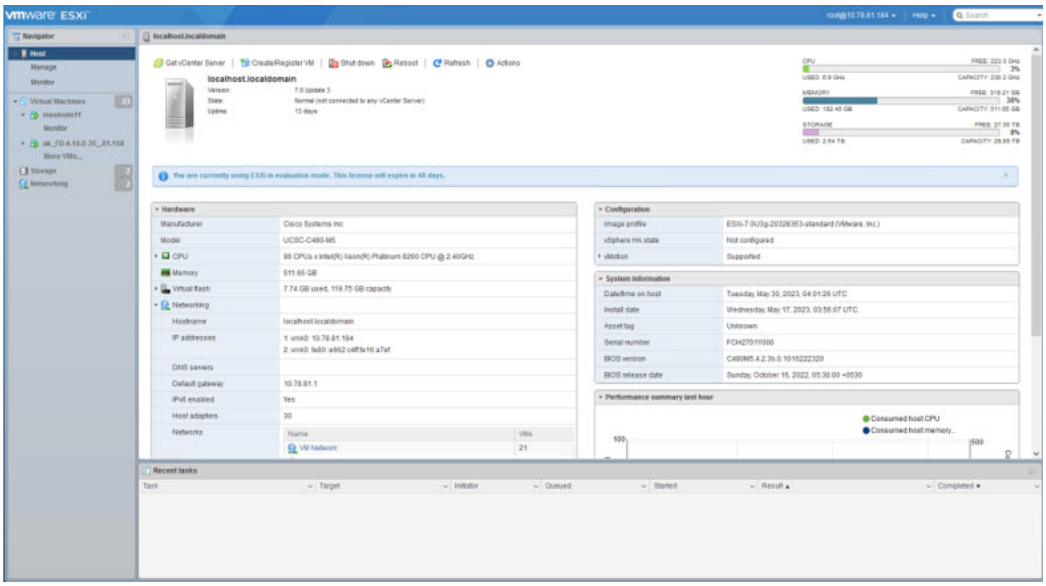
Step 1 Log in to the IP address of a VMware ESXi server running 6.5 and above via a web browser with your user credentials (username and password).

a) Enter the ESXi IP address in the URL.

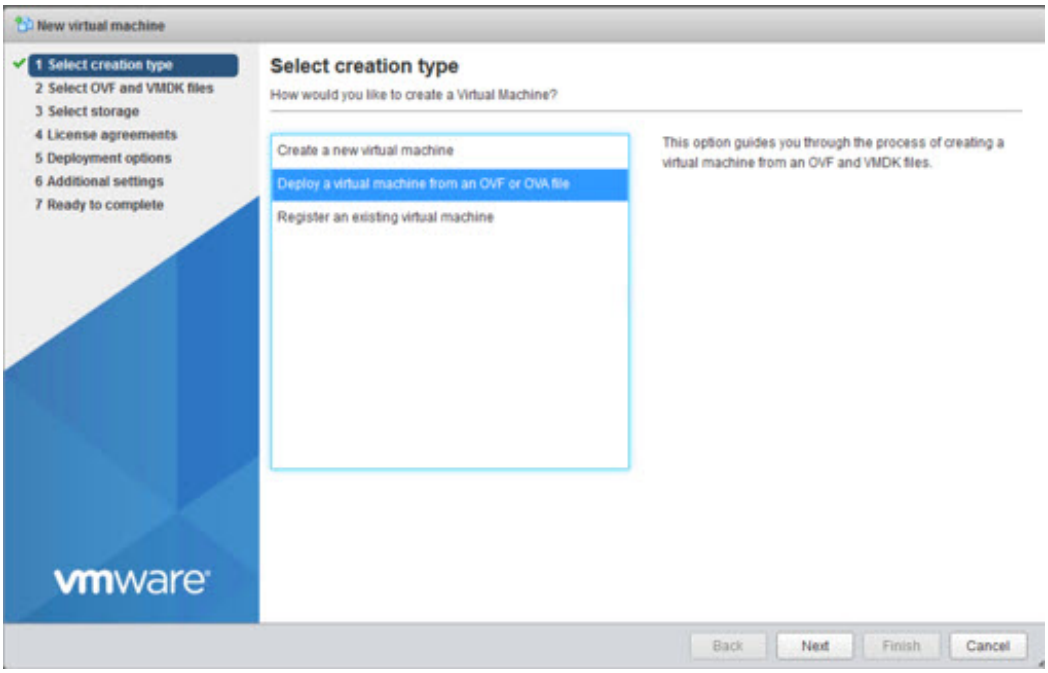
b) Provide the ESXi root login credentials and click **Log In**.



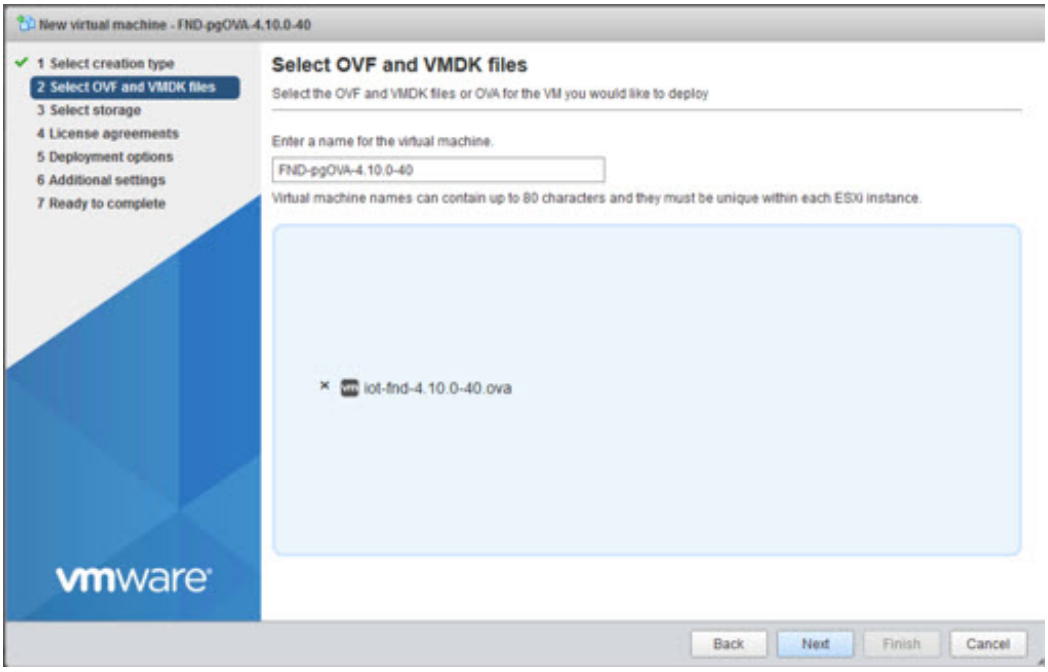
Step 2 In the Host page, select **Create/ Register VM**.



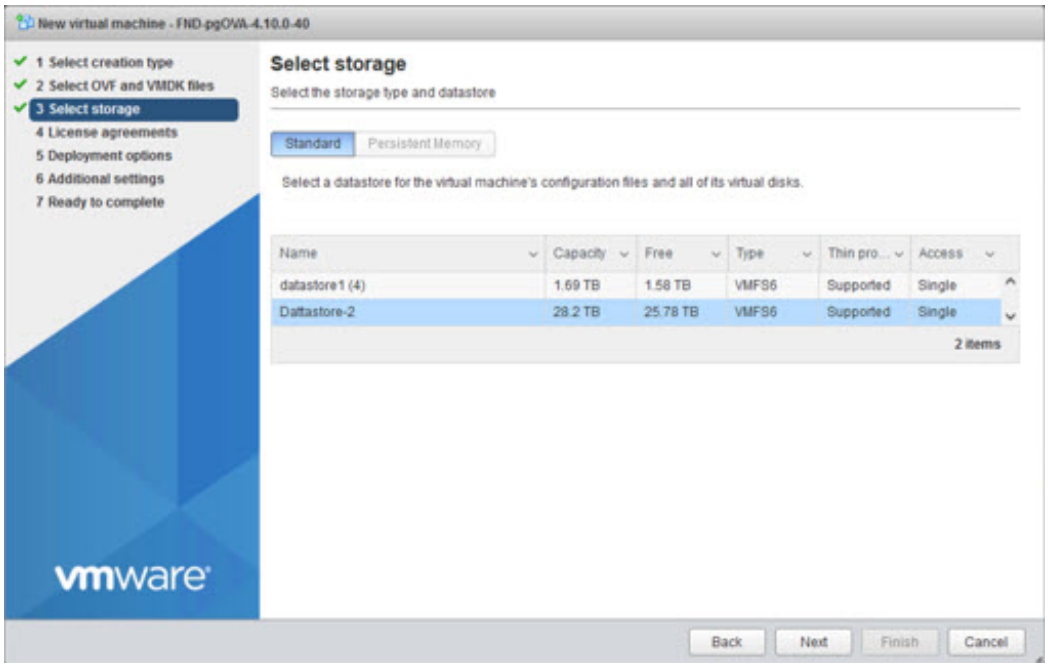
Step 3 In the New virtual machine window, select **Deploy a virtual machine from an OVF or OVA file** in Select creation type tab and click **Next**.



Step 4 In the Select OVF and VMDK files tab, provide a name for the virtual machine and browse to an OVF package from the internet or a file accessible from your computer (for example, iot-fnd-4.10.0-40.ova). Click **Next**.



Step 5 Select a storage location for the virtual machines from the listed options (example - Datastore-2).

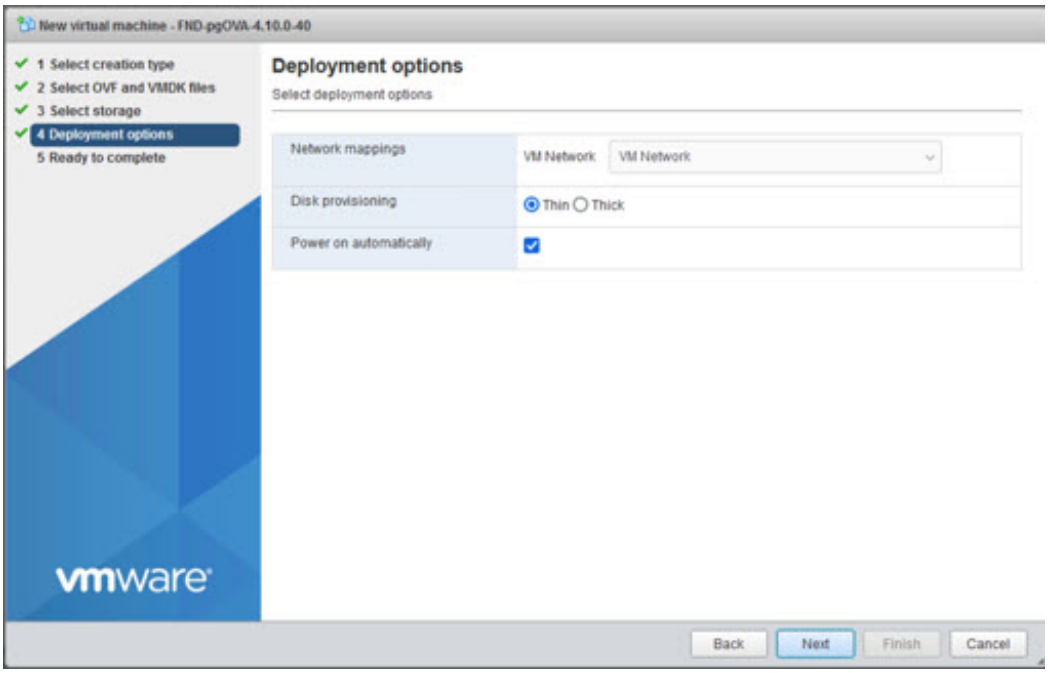


Step 6 After selecting the data store, select the provisioning type and enable the **Power on automatically** option. This ensures to power on the virtual machine once the deployment process is complete. Click **Next**.

Note

- Thick Provisioning — Absolute reservation on the disk space. For the IoT FND OVA deployment, the disk space required is 600 GB on the ESXi server.
- Thin Provisioning — The disk space grows on demand. For the 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.

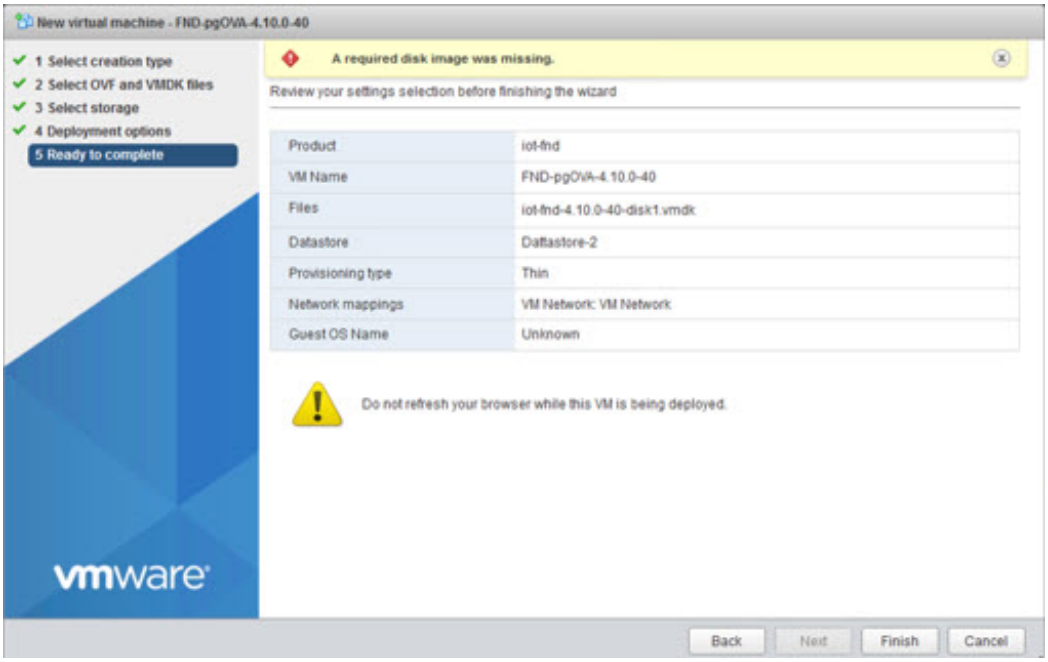
Note If the selected storage location does not have sufficient storage for the largest file installation option, a message displays noting insufficient storage. If the warning message appears, select another storage resource with greater capacity and click **Next**.



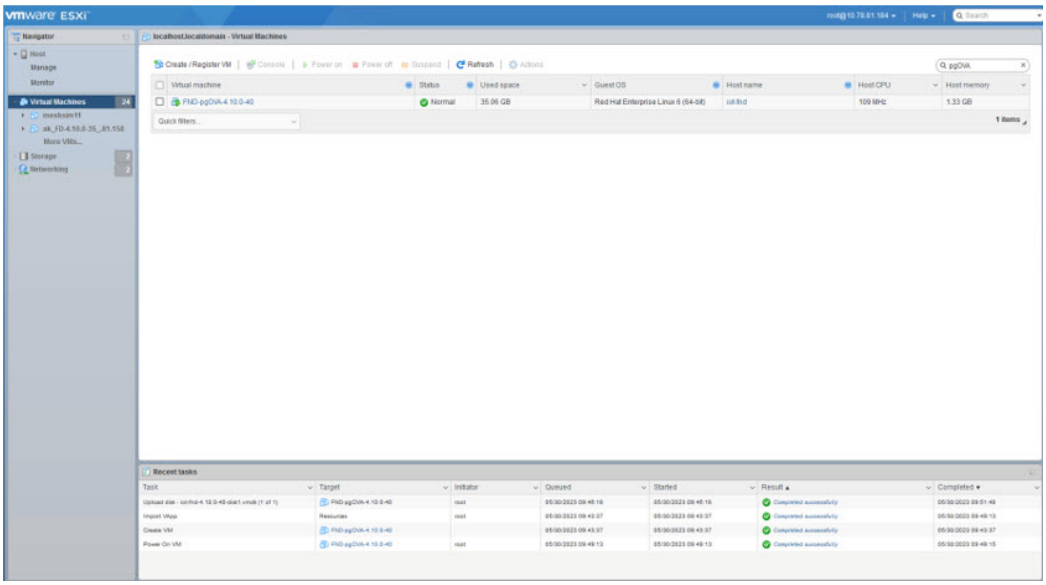
Step 7 Do a final review of the **Ready to Complete** window. If you do not want to change any settings, click **Finish**.

Note If you see the following warning message while deployment, then cancel the upload, disconnect the Esxi from vCenter Server (**Actions** > **Disconnect from vCenter Server**) and then re-upload OVA. The upload will be successful.

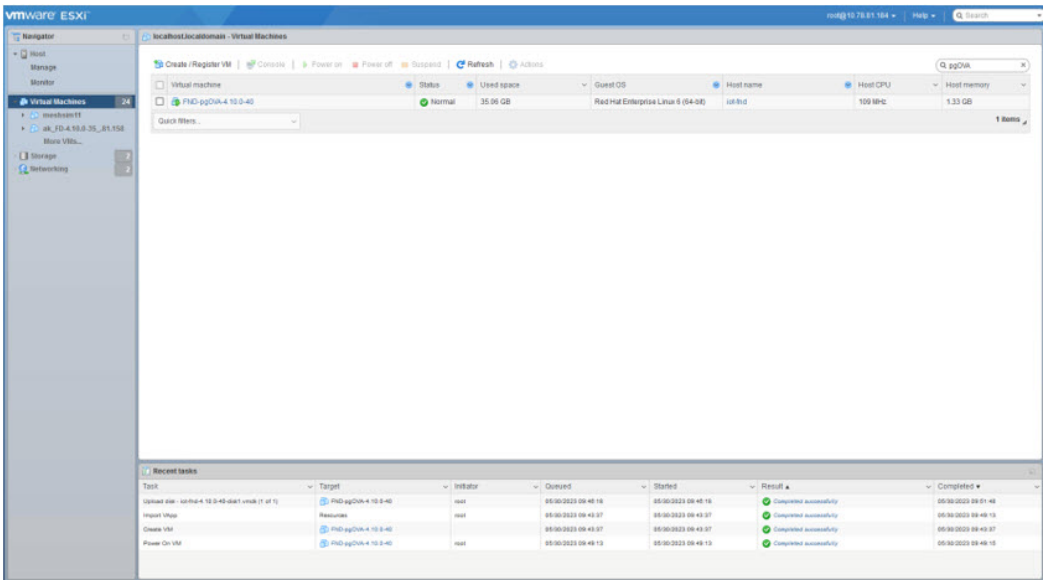
"Failed: Access to resource settings on the host is restricted to the server that is managing it 'vCenter Server IP'"



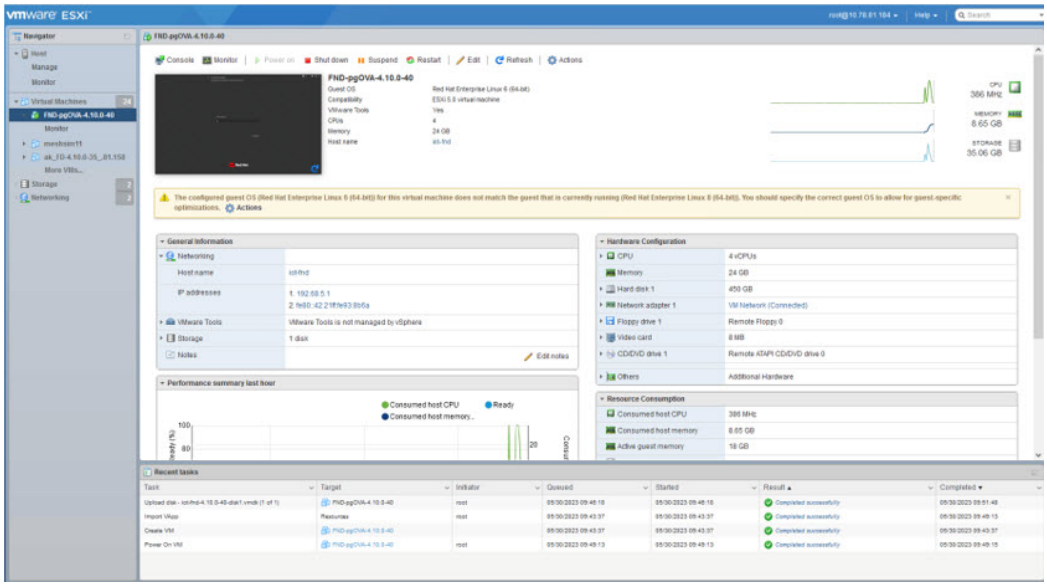
The virtual machine deployment is initiated. After completion of the install, the "Completed successfully" message appears in the Recent tasks pane at the bottom of the install window.



Step 8 Click Virtual Machines in the left pane and select the newly deployed VM.



Step 9 The deployed VM gets listed in the left pane. Select the IoT FND machine name.



Step 10 Click **Console** and login with root/cisco123 once the OS is up. Once you enter the default password, you are prompted to reset your password.

Important From IoT FND release 4.12 onwards:

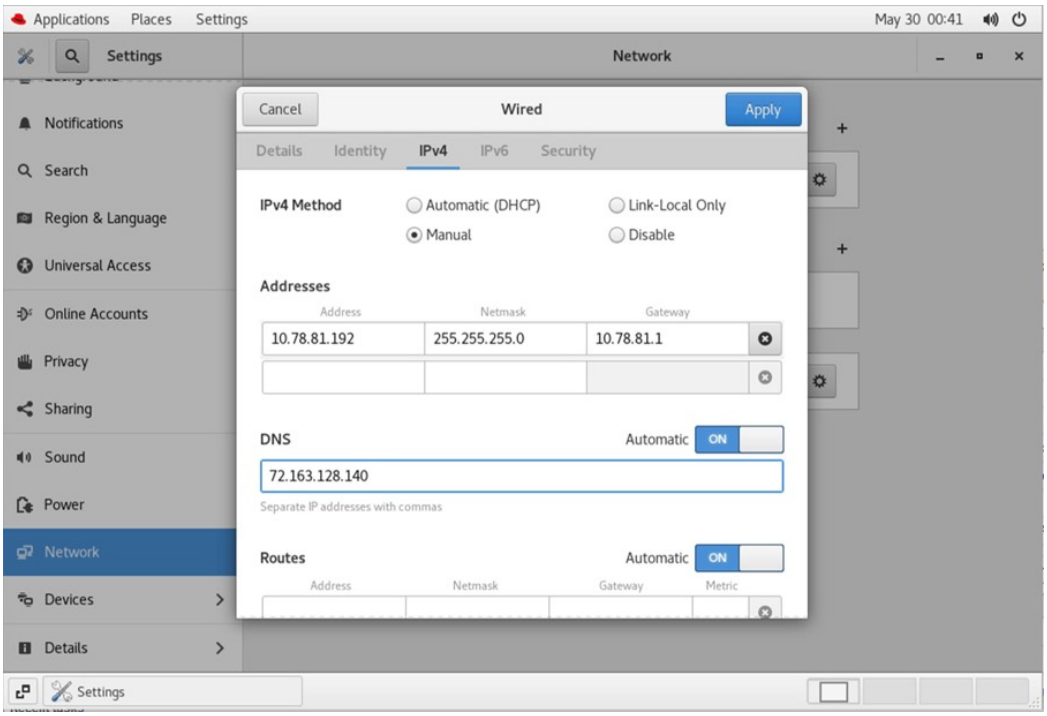
- The default root user password is C!sco123.
- The following conditions are applicable to reset the default password.
  - The password must be at least 8 characters in length
  - The password must have at least 1 uppercase character
  - The password must have at least 1 lowercase character
  - The password 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

Step 11 Reset the default root password. After you complete the password reset, IoT FND is fully deployed. From FND 4.12 release, the conditions mentioned in Step 10 are applicable to reset the default password.

Step 12 Once logged in, navigate to **Applications > System Tools > Settings > Network**.

Step 13 Click the cog icon under Wired, navigate to IPv4 tab to assign a static IP address or set up a DHCP server in the network.

- Under IPv4 tab, select the method as Manual and provide the IPv4 address as below and click **Apply**.
- Set up a valid, reachable working DNS server on the Host VM. (mandatory) Use this IP address to access the FND GUI.



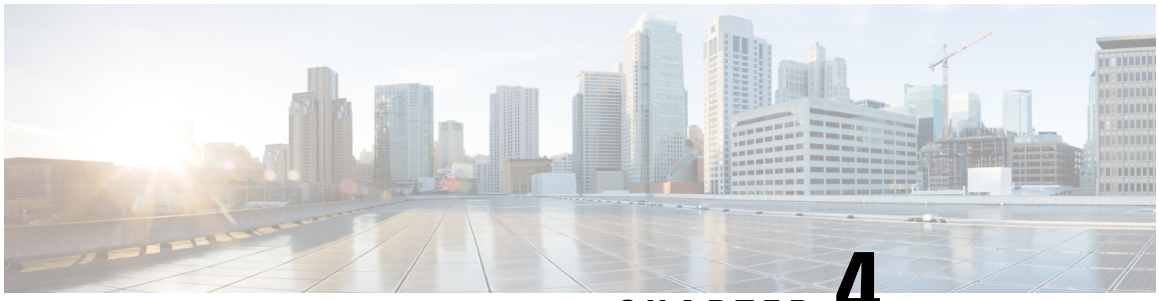
Important Follow the same steps for TPS OVA installation as well. In order to upgrade the TPS OVA, delete the existing TPS and reinstall the TPS OVA `iot-tps-version_number.ova` with the updated version number.

Step 14 Open a terminal window, and set up Health Monitoring for the Fog Director Container from FND.

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

```
[root@iot-fnd monitor]# ./setup.sh
Setup health metrics monitor for App Management Servers
Enter FND Username: root
Enter FND Password:
Successfully configured health metrics monitor for App Management Servers
```

After completing these steps, IoT FND starts monitoring the Fog Director container on the **ADMIN > SERVERS** page.



CHAPTER 4

Installing Custom CA Certificates and Importing SUDI Certificate

By default, the IoT FND OVA comes bundled with keys and certificates which is stored in a keystore. The default values are:

- On IoT FND OVA Linux Host:

  Keystore Location: /opt/fnd/data/

  Keystore Name: cgms_keystore.selfsigned

- On IoT FND container:

  Keystore Location: /opt/cgms/server/cgms/conf/

  Keystore Name: cgms_keystore

- **Default Password: Public123!**

Important

- This is the default password for both the files mentioned above.
- Both these files have the same content.
- When IoT FND container is restarted, the values of /opt/cgms/server/cgms/conf/cgms_keystore file in IoT FND container is overwritten by /opt/fnd/data/cgms_keystore file. If /opt/fnd/data/cgms_keystore file is not present in host, then /opt/fnd/data/cgms_keystore.selfsigned file is used.

When IoT FND OVA is a new installation, each certificate/key entry is referenced by an alias name in the keystore. The default alias are:

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

Note This keystore is specific for certificates used for IoT FND communication with its managed devices. There is a different keystore for web certificate.

Custom cgms_keystore

The cgms certificate in /opt/cgms/server/cgms/conf/cgms_keystore file in IoT FND container and /opt/fnd/data/cgms_keystore.selfsigned file of the linux host has by default self signed certificate of IoT FND. There are two options to build a custom cgms_keystore in /opt/fnd/data location on linux host, where the IoT FND certificate of the customer organisation can be imported and stored.

We can either copy the existing /opt/fnd/data/cgms_keystore.selfsigned file on the Linux host or build it from scratch. After the cgms_keystore file is present on the linux host, if both /opt/fnd/data/cgms_keystore.selfsigned and /opt/fnd/data/cgms_keystore files are present, then /opt/fnd/data/cgms_keystore takes precedence.

Note NTP is a mandatory requirement for Public Key Infrastructure. Hence NTP should be in sync between the issuing Certificate Authority (CA) server, IoT FND, TPS, and FAR/HER. If hostname or IP address has to be changed for the IoT FND host, it has to be done before certificate for IoT FND is issued and hence it should be done before starting to build cgms_keystore.

The SAN field in IoT FND certificate is a mandatory requirement and contains the hostname of the IoT FND server. Any change in hostname or IP address is listed in SAN field (if IP address is also present in the SAN field), then the certificate should be reissued. Depending on the PnP type used, the SAN field contains the hostname of the IoT FND or the IP address or both.

The cgms_keystore should contain the below mandatory certificates/keys:

- Issuing CA certificate of the organisation – This is the certificate of the issuing CA server of the organisation. The issuing CA server can be a root CA server or intermediate CA server. If it is an intermediate CA, it is recommended to import root CA and also intermediate CA certificates into the keystore.
- IoT FND device certificate is issued for IoT FND by issuing CA server.
- Cisco SUDI with 2029 expiry date – This is the cisco manufacturer certificate for IoT FND issued by Cisco with expiry date 2029.
- Cisco SUDI with 2099 expiry date – This is the cisco manufacturer certificate for IoT FND issued by Cisco with expiry date 2099.

The below option shows how to build cgms_keystore file from scratch that contains the required certificates and keys.

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

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

Importing Root/Issuing CA Certificate

Step 2 Importing any certificate using keytool command creates the keystore file, if keystore file does not exist. Note that the name of the file has to be cgms_keystore as IoT FND refers the file with this name. Copy the issuing CA certificate of your organisation to any location (using scp or any other file transfer method). In this illustration, it is copied to

/root/rootca.pem. The certificate can be of the format .cer or .crt or .pem. In this illustration, the issuing CA is the root CA and hence the alias name root is used.

```
# 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
```

Importing IoT FND Certificate

Step 3 Import the IoT FND certificate.

Note IoT FND certificate has to be imported ONLY with alias name of cgms.

The below steps tells how to generate a key pair .csr file that can be presented to the issuing CA server for a certificate to be granted for IoT FND server. The .csr file is required by few issuing CA servers of few PKI vendors. If the .csr certificate is given to the issuing CA server, then a certificate is generated based on the contents of the .csr file. If a certificate of IoT FND has already been issued, use the following steps, if the IoT FND certificate issued has .pem or .cer or .crt extension. If the IoT FND certificate has .pfx extension, follow step 4.

a) Generate a key pair and .csr file .

```
# 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 in this example is 2048, but 4096 can also be used.

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

Note This .csr file is then presented to the issuing CA server and a certificate is obtained for IoT FND server.

b) Copy the issued certificate to FND server in any location. In this sample, it is copied to /opt/fnd/data as labfnd.pem file. Import the certificate using below command.

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

Step 4 If the FND issued certificate issued has .pfx format, then we will have a .pfx file instead of the .pem file. If it is a .pfx file, check the alias name of the .pfx file and then import it using the alias cgms in the cgms_keystore.

- Find the alias name of the pfx file. In this case, the nms.pfx is copied to the current location.

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

- Import the pfx into the cgms_keystore with alias cgms. In this sample, "le-IoT FND-8f0908aa-dc8d-4101-a526-93b4eaad9481" is the alias present in the .pfx file.

```
# 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
```

Importing SUDI with 2029 Expiry

Step 5 The SUDI certificate with 2029 expiry is present in /opt/fnd/data directory as cisco-sudi-ca.pem. Import this file to cgms_keystore.

```
# keytool -import -trustcacerts -alias cisco_sudi -file
/opt/fnd/data/cisco-sudi-ca.pem -keystore /opt/fnd/data/cgms_keystore
```

Importing SUDI with 2099 Expiry

Step 6 The updated SUDI certificate with 2099 expiry is present in IoT FND container as cisco-ca.pem file, copy this to /opt/fnd/data in linux host.

```
docker cp fnd-container:/opt/cgms/server/cgms/conf/ciscosudi/cisco-ca.pem
/opt/fnd/data/
```

Import the SUDI with 2099 expiry, that is, cisco-ca.pem to cgms_keystore.

```
# keytool -import -trustcacerts -keystore /opt/fnd/data/cgms_keystore
-alias sudi1 -file /opt/fnd/data/cisco-ca.pem
```

Step 7 Restart the container.

```
docker stop fnd-container
docker start fnd-container
```

Important It is not advised to use the restart command. It is best practice to stop the container and then start the container so the services can stop gracefully. Sometimes restart will not be graceful and can lead to operational issues.

Step 8 After restart, verify that the contents of the cgms_keystore in the IoT FND container has same contents as that of the cgms_keystore in /opt/fnd/data of linux host using the below command.

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

```
# docker exec -it fnd-container keytool -list -v -keystore /opt/cgms/server/cgms/conf/cgms_keystore
```

Step 9 To configure or change the cgms_keystore password, see Changing Password for more information.

• Changing Password, on page 20

Changing Password

The cgms.properties file should contain the password for cgms_keystore, so that IoT FND application can access the cgms_keystore. Hence the first time cgms_keystore is created, encrypt the password of cgms_keystore and provide this encrypted password in cgms.properties file.

If at any time the password for cgms_keystore is changed, then the changed password has to be encrypted again and updated in the cgms.properties file.

Step 1 Run the following command to encrypt the password for the new **cgms_keystore**. The sample is provided below.

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

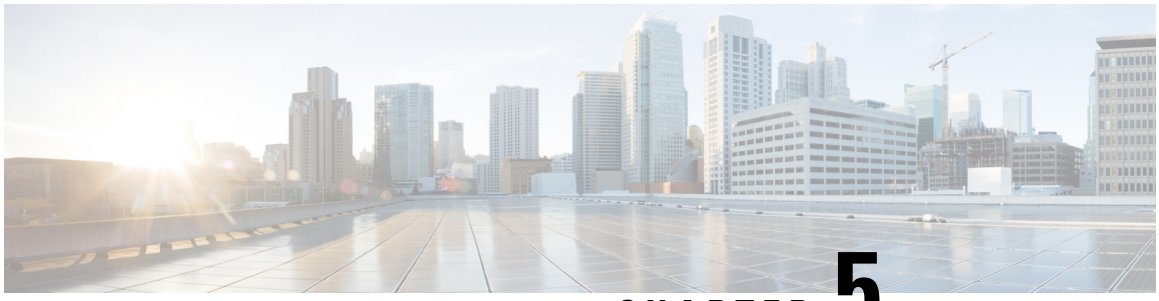
```
# docker exec -it fnd-container /opt/cgms/bin/encryption_util.sh
encrypt cisco123
#2bVvZsq+vsq94YxuAKdaag--
```

Step 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:

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

Note: With OVA 4.3.x and above, you can leave the `cgms_keystore.selfsigned` default bundled keystore untouched.

If both the files (**cgms_keystore** and **cgms_keystore.selfsigned**) are present, the **cgms_keystore** will be used by the container.



CHAPTER 5

Configuring IoT FND for IPv6 Tunnel Provisioning and Registration

IoT FND OVA supports only IPv4 tunnels and Registration out of the box.

To setup an IPv6 network for tunnel provisioning and registration, follow these steps:

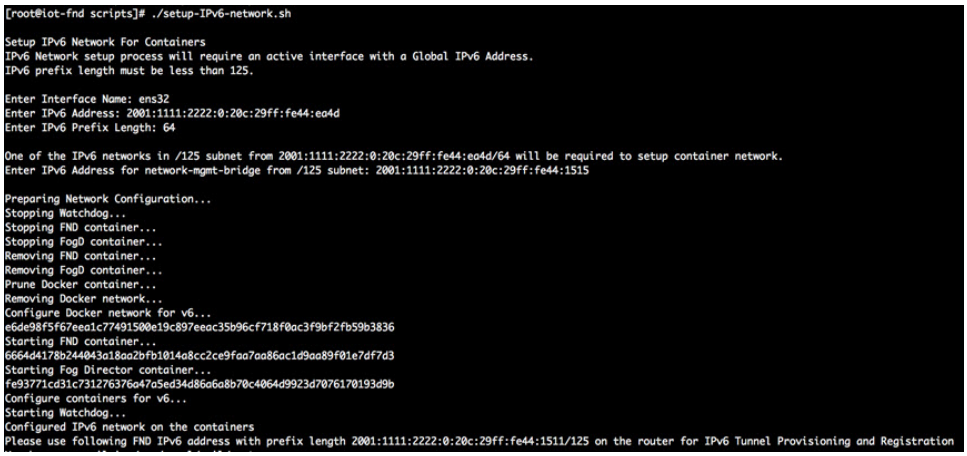
Step 1 Ensure you have one interface with a valid IPv6 network which has a IPv6 prefix length less than125.

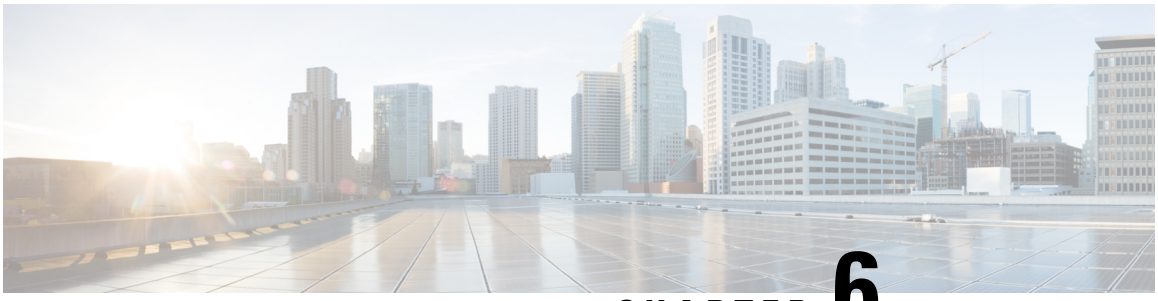
See the following example of the ens224 interface:

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

Step 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.

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.


[root@iot-fnd scripts]# ./setup-IPv6-network.sh
Setup IPv6 Network For Containers
IPv6 Network setup process will require an active interface with a Global IPv6 Address.
IPv6 prefix length must be less than 125.
Enter Interface Name: ens32
Enter IPv6 Address: 2001:1111:2222:0:20c:29ff:fe44:ea4d
Enter IPv6 Prefix Length: 64
One of the IPv6 networks in /125 subnet from 2001:1111:2222:0:20c:29ff:fe44:ea4d/64 will be required to setup container network.
Enter IPv6 Address for network-mgmt-bridge from /125 subnet: 2001:1111:2222:0:20c:29ff:fe44:1515
Preparing Network Configuration...
Stopping Watchdog...
Stopping FND container...
Stopping FogD container...
Removing FND container...
Removing FogD container...
Prune Docker container...
Removing Docker network...
Configure Docker network for v6...
e6de98f5f67eea1c77491500e19c897eeac35b96cf718f0ac3f9bf2fb59b3836
Starting FND container...
6664d4178b244043a18aa2bfb1014a8cc2ce9faa7aa86ac1d9aa89f01e7df7d3
Starting Fog Director container...
fe93771cd31c731276376a47a5ed34d86a6a8b70c4064d9923d7076170193d9b
Configure containers for v6...
Starting Watchdog...
Configured IPv6 network on the containers
Please use following FND IPv6 address with prefix length 2001:1111:2222:0:20c:29ff:fe44:1511/125 on the router for IPv6 Tunnel Provisioning and Registration



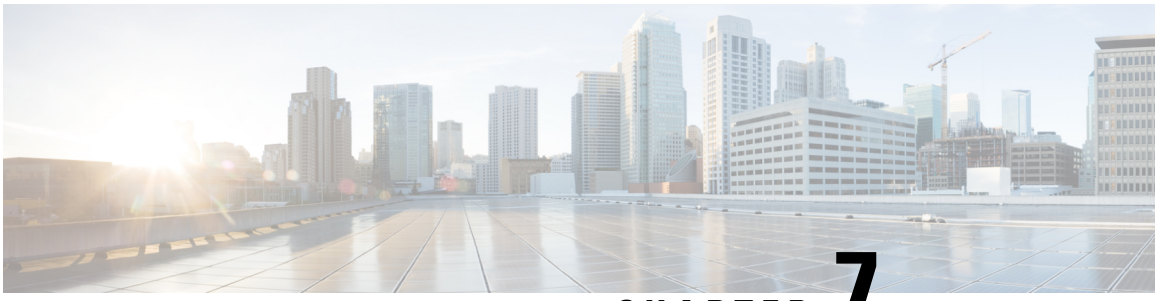
CHAPTER 6

Starting and Stopping FND

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

cd /opt/fnd/scripts/

```
[root@iot-fnd scripts]# ./fnd-container.sh status
fnd-container is running, pid=22745
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O           PIDS
4bc00c18b2c8        fnd-container       1.99%               1.064GiB / 23.38GiB   4.55%               8.63MB / 8.07MB     0B / 1.76MB         272
[root@iot-fnd scripts]# ./fnd-container.sh stop
Stopping FND container...
fnd-container
[root@iot-fnd scripts]# ./fnd-container.sh start
[root@iot-fnd scripts]# Starting FND container...
fnd-container

[root@iot-fnd scripts]# ./fnd-container.sh restart
Stopping FND container...
fnd-container
[root@iot-fnd scripts]# Starting FND container...
fnd-container
```



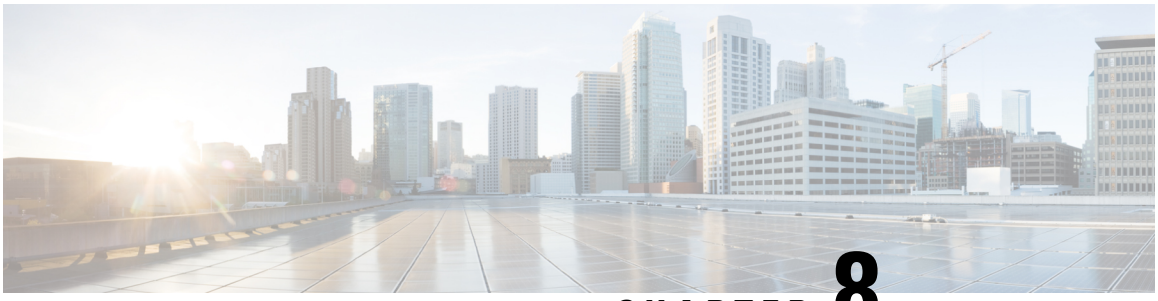
CHAPTER 7

Starting and Stopping Fog Director

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

```
cd /opt/fogd/scripts
```

```
[root@iot-fnd scripts]# ./fogd-container.sh stop
Stopping Fog Director container...
fogd-container
[root@iot-fnd scripts]# ./fogd-container.sh start
[root@iot-fnd scripts]# Starting Fog Director container...
fogd-container

[root@iot-fnd scripts]# ./fogd-container.sh status
fogd-container is running, pid=10759
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O           PIDS
f2bc75fa77c2        fogd-container      2.00%               764.6MiB / 23.38GiB   3.19%               849kB / 1.5MB       0B / 41kB           119
[root@iot-fnd scripts]# ./fogd-container.sh restart
Stopping Fog Director container...
fogd-container
[root@iot-fnd scripts]# Starting Fog Director container...
fogd-container

[root@iot-fnd scripts]#
```



CHAPTER 8

Upgrading IoT FND OVA

Note Ensure to upgrade the DB and the docker server image first before upgrading the IoT FND and FD container images.

To upgrade the IoT FND OVA, follow the upgrade sequence given below:

1. Upgrade the DB and the docker server image using rpm scripts.

   For more information, refer to Upgrading the Database and Docker Server Image, on page 33.

2. Upgrade the IoT FND and FD container images.

   For more information, refer to Upgrading IoT FND and FD Container Images, on page 45.

3. Restart Postgres service if the current IoT FND release is prior to 4.9.1 and the target IoT FND release is 4.9.1 or above.

Note

- Postgres service restart is not required if the target IoT FND release is greater than 4.9.1. In this case, we assume that during the upgrade to IoT FND 4.9.1, the postgres service is already restarted.
- Postgres service restart is a must if you are directly upgrading to 4.10 from a release prior to 4.9.1.

- Pre-Upgrade Checklist, on page 29
- Upgrading the Database and Docker Server Image, on page 33
- Upgrading IoT FND and FD Container Images, on page 45
- Post-Upgrade Checklist, on page 48
- Upgrading IoT FND from 4.5.1 to later releases and Updating RHEL OS, on page 49

Pre-Upgrade Checklist

The section identifies the tasks that you must perform before you begin the upgrade to ensure successful upgrade and limited downtime.

Step 1 Take a snapshot of the existing VM before you upgrade.

This helps in restoring if there is an upgrade failure.

Step 2 Take a backup of the PostgreSQL DB.

Note For any clarification on backup procedure, contact your DB administrator.

Step 3 Take 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.

```
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_09May2022
[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_9May2022  cgms.properties           cisco-sudi-ca.pem
userPropertyTypes.xml
[root@iot-fnd data]#
```

a) During the IoT FND container upgrade, the following files get overwritten in the directories mentioned below:

- Directory — `/opt/cgms/server/cgms/conf/`:
  - jbossas.keystore.password
  - jbossas.keystore
  - VAULT.dat
  - vault.keystore
- Directory — `/opt/cgms/server/cgms/deploy/`:
  - security-service.xml file

Backup can be done in the same directory using different name or backup in a different directory or backup and store the files in the SCP server.

For example, taking backup in the same directory:

```
Login to the FND container
[root@iot-fnd ~]# docker exec -it fnd-container /bin/bash
[root@fnd-server /]#
```

```
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/jbossas.keystore.password
/opt/cgms/server/cgms/conf/jbossas.keystore.password.bkp1
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/jbossas.keystore
/opt/cgms/server/cgms/conf/jbossas.keystore.bkp1
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/vault.keystore
/opt/cgms/server/cgms/conf/vault.keystore.bkp1
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/VAULT.dat
/opt/cgms/server/cgms/conf/VAULT.dat.bkp1
[root@fnd-server /]# cp /opt/cgms/server/cgms/deploy/security-service.xml
/opt/cgms/server/cgms/deploy/security-service.xml.bkp1
[root@fnd-server /]#
```

b) If you are using *userpropertyTypes.xml* to define custom properties for backup, then follow the steps that are mentioned in the workaround of the bug ID: CSCwc12435. This will be fixed in IoT FND release 4.9 or later.

Step 4 Run the following commands and check the output before you start the upgrade process.

- `/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 04:09:48 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]#
```

Upgrading the Database and Docker Server Image

This section provides steps for upgrading the database and the docker server image by running the `rpm` upgrade scripts for releases 4.7.0 to later versions and 4.5.1 to later versions. By running the rpm scripts, you automatically integrate the DB with IoT FND scripts, upgrade the DB, and upgrade the docker server (Community Edition) image.

Note IoT FND version 4.5.1 provides the option to manually upgrade the DB and docker server image instead of running the Cisco rpm scripts. For more information, refer to Manual Upgrade Option in FND 4.5.1.

Note IoT FND OVA upgrade will NOT upgrade the RHEL OS version. The RHEL version differs for different versions of IoT FND as in the table below. After upgrading the OVA, it is recommended to upgrade the OS sooner than later. Although IoT FND is a secure application, OS security and patches must be regularly updated with Cisco's guidance.

Table 2: List of IoT FND and the bundled Postgres, Docker, and RHEL OS versions:

IoT FND Version	Postgres Version	Docker Server Version	RHEL OS Version
4.11.0	12.12	19.03.15	8.8
4.10.0	12.12	19.03.15	8.7
4.9.1	12.12	19.03.15	8.6
4.9.0	12.9	19.03.15	8.6
4.8.1	12.9	19.03.15	8.5
4.8.0	12.5	19.03.15	7.7
4.7.2	12.5	19.03.15	7.7
4.7.1	12.5	19.03.15	7.7
4.7.0	12.4	18.09.6	7.7
4.5.1	9.6	18.09.6	7.5

Note Starting from FND 4.8.1 release, all python scripts are compatible only for Python 3 which comes as default python interpreter in RHEL 8.x. It is recommended to install Python 3.6 manually if IoT FND OVA is upgraded to 4.8.1 or higher without base OS upgrade.

Step 1 Obtain the IoT FND upgrade scripts from Cisco.

Step 2 Check the RHEL OS version before upgrading IoT FND OVA to 4.7.1 or higher.

```
[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
```

- If the RHEL version on the Linux server is lesser than 7.7, then use the following steps to upgrade. You can either do an automatic or manual upgrade.
- If the RHEL version on the Linux server is 7.7 or above, then you can skip the steps below.

a) **Method 1 — Automatic Upgrade**: For this method, you require subscription to RHEL subscription-manager and active internet connection.

Run the following command to upgrade the container-selinux package.

```
subscription-manager repos --enable=rhel-7-server-extras-rpms
yum update container-selinux
```

Example

```
[root@fnd451testupgrade ~]# subscription-manager repos --enable=rhel-7-server-extras-rpms
Repository 'rhel-7-server-extras-rpms' is enabled for this system.
[root@fnd451testupgrade ~]# yum update container-selinux
Loaded plugins: langpacks, product-id, search-disabled-repos, subscription-manager
https://download.postgresql.org/pub/repos/yum/9.4/redhat/rhel-7Server-x86_64/repodata/repomd.xml:
 [Errno 14] HTTPS Error 404 - Not Found
Trying other mirror.
To address this issue please refer to the below knowledge base article

https://access.redhat.com/articles/1320623

If above article doesn't help to resolve this issue please open a ticket with Red Hat Support.

Resolving Dependencies
--> Running transaction check
---> Package container-selinux.noarch 2:2.42-1.gitad8f0f7.el7 will be updated
---> Package container-selinux.noarch 2:2.119.2-1.911c772.el7_8 will be an update
--> Processing Dependency: selinux-policy >= 3.13.1-216.el7 for package:
2:container-selinux-2.119.2-1.911c772.el7_8.noarch
--> Processing Dependency: selinux-policy-base >= 3.13.1-216.el7 for package:
2:container-selinux-2.119.2-1.911c772.el7_8.noarch
--> Processing Dependency: selinux-policy-targeted >= 3.13.1-216.el7 for package:
2:container-selinux-2.119.2-1.911c772.el7_8.noarch
```

```
--> Running transaction check
---> Package selinux-policy.noarch 0:3.13.1-192.el7 will be updated
---> Package selinux-policy.noarch 0:3.13.1-268.el7_9.2 will be an update
--> Processing Dependency: libsemanage >= 2.5-13 for package:
selinux-policy-3.13.1-268.el7_9.2.noarch
--> Processing Dependency: policycoreutils >= 2.5-24 for package:
selinux-policy-3.13.1-268.el7_9.2.noarch
---> Package selinux-policy-targeted.noarch 0:3.13.1-192.el7 will be updated
---> Package selinux-policy-targeted.noarch 0:3.13.1-268.el7_9.2 will be an update
--> Running transaction check
---> Package libsemanage.x86_64 0:2.5-11.el7 will be updated
--> Processing Dependency: libsemanage = 2.5-11.el7 for package:
libsemanage-python-2.5-11.el7.x86_64
---> Package libsemanage.x86_64 0:2.5-14.el7 will be an update
--> Processing Dependency: libselinux >= 2.5-14 for package: libsemanage-2.5-14.el7.x86_64
--> Processing Dependency: libsepol >= 2.5-10 for package: libsemanage-2.5-14.el7.x86_64
---> Package policycoreutils.x86_64 0:2.5-22.el7 will be updated
--> Processing Dependency: policycoreutils = 2.5-22.el7 for package:
policycoreutils-python-2.5-22.el7.x86_64
---> Package policycoreutils.x86_64 0:2.5-34.el7 will be an update
--> Processing Dependency: libselinux-utils >= 2.5-14 for package: policycoreutils-2.5-34.el7.x86_64
--> Running transaction check
---> Package libselinux.x86_64 0:2.5-12.el7 will be updated
--> Processing Dependency: libselinux(x86-64) = 2.5-12.el7 for package:
libselinux-python-2.5-12.el7.x86_64
---> Package libselinux.x86_64 0:2.5-15.el7 will be an update
---> Package libselinux-utils.x86_64 0:2.5-12.el7 will be updated
---> Package libselinux-utils.x86_64 0:2.5-15.el7 will be an update
---> Package libsemanage-python.x86_64 0:2.5-11.el7 will be updated
---> Package libsemanage-python.x86_64 0:2.5-14.el7 will be an update
---> Package libsepol.x86_64 0:2.5-8.1.el7 will be updated
---> Package libsepol.x86_64 0:2.5-10.el7 will be an update
---> Package policycoreutils-python.x86_64 0:2.5-22.el7 will be updated
---> Package policycoreutils-python.x86_64 0:2.5-34.el7 will be an update
--> Processing Dependency: setools-libs >= 3.3.8-4 for package:
policycoreutils-python-2.5-34.el7.x86_64
--> Running transaction check
---> Package libselinux-python.x86_64 0:2.5-12.el7 will be updated
---> Package libselinux-python.x86_64 0:2.5-15.el7 will be an update
---> Package setools-libs.x86_64 0:3.3.8-2.el7 will be updated
---> Package setools-libs.x86_64 0:3.3.8-4.el7 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
Package                                Arch                           Version
                                Repository                                Size
================================================================================
Updating:
container-selinux                      noarch
2:2.119.2-1.911c772.el7_8                   rhel-7-server-extras-rpms
        40 k
Updating for dependencies:
libselinux                             x86_64                         2.5-15.el7
                                rhel-7-server-rpms                        162 k
libselinux-python                      x86_64                         2.5-15.el7
                                rhel-7-server-rpms                        236 k
libselinux-utils                       x86_64                         2.5-15.el7
                                rhel-7-server-rpms                        151 k
libsemanage                            x86_64                         2.5-14.el7
                                rhel-7-server-rpms                        151 k
libsemanage-python                     x86_64                         2.5-14.el7
                                rhel-7-server-rpms                        113 k
```

```
libsepol                                    x86_64                      2.5-10.el7
                                  rhel-7-server-rpms                       297 k
policycoreutils                             x86_64                      2.5-34.el7
                                  rhel-7-server-rpms                       917 k
policycoreutils-python                      x86_64                      2.5-34.el7
                                  rhel-7-server-rpms                       457 k
selinux-policy                              noarch                    3.13.1-268.el7_9.2
                                 rhel-7-server-rpms                        498 k
selinux-policy-targeted                     noarch                    3.13.1-268.el7_9.2
                                 rhel-7-server-rpms                        7.0 M
setools-libs                                x86_64                      3.3.8-4.el7
                                  rhel-7-server-rpms                       620 k

Transaction Summary
================================================================================
Upgrade  1 Package (+11 Dependent packages)

Total download size: 11 M
Is this ok [y/d/N]: y
Downloading packages:
No Presto metadata available for rhel-7-server-rpms
No Presto metadata available for rhel-7-server-extras-rpms
(1/12): container-selinux-2.119.2-1.911c772.el7_8.noarch.rpm
                                                          |  40 kB  00:00:01
(2/12): libselinux-2.5-15.el7.x86_64.rpm
                                                          | 162 kB  00:00:01
(3/12): libselinux-python-2.5-15.el7.x86_64.rpm
                                                          | 236 kB  00:00:01
(4/12): libselinux-utils-2.5-15.el7.x86_64.rpm
                                                          | 151 kB  00:00:01
(5/12): libsemanage-2.5-14.el7.x86_64.rpm
                                                          | 151 kB  00:00:01
(6/12): libsemanage-python-2.5-14.el7.x86_64.rpm
                                                          | 113 kB  00:00:01
(7/12): libsepol-2.5-10.el7.x86_64.rpm
                                                          | 297 kB  00:00:01
(8/12): policycoreutils-python-2.5-34.el7.x86_64.rpm
                                                          | 457 kB  00:00:01
(9/12): policycoreutils-2.5-34.el7.x86_64.rpm
                                                          | 917 kB  00:00:02
(10/12): selinux-policy-3.13.1-268.el7_9.2.noarch.rpm
                                                          | 498 kB  00:00:02
(11/12): setools-libs-3.3.8-4.el7.x86_64.rpm
                                                          | 620 kB  00:00:02
(12/12): selinux-policy-targeted-3.13.1-268.el7_9.2.noarch.rpm
                                                          | 7.0 MB  00:00:08
--------------------------------------------------------------------------------
Total
                                                 679 kB/s |  11 MB  00:00:15
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating   : libsepol-2.5-10.el7.x86_64
                                                                        1/24
  Updating   : libselinux-2.5-15.el7.x86_64
                                                                        2/24
  Updating   : libsemanage-2.5-14.el7.x86_64
                                                                        3/24
  Updating   : libselinux-utils-2.5-15.el7.x86_64
                                                                        4/24
  Updating   : policycoreutils-2.5-34.el7.x86_64
                                                                        5/24
  Updating   : selinux-policy-3.13.1-268.el7_9.2.noarch
```

```
                                                                                                                6/24
  Updating   : selinux-policy-targeted-3.13.1-268.el7_9.2.noarch
                                                                                                                7/24
  Updating   : libsemanage-python-2.5-14.el7.x86_64
                                                                                                                8/24
  Updating   : libselinux-python-2.5-15.el7.x86_64
                                                                                                                9/24
  Updating   : setools-libs-3.3.8-4.el7.x86_64
                                                                                                               10/24
  Updating   : policycoreutils-python-2.5-34.el7.x86_64
                                                                                                               11/24
  Updating   : 2:container-selinux-2.119.2-1.911c772.el7_8.noarch
                                                                                                               12/24
  Cleanup    : 2:container-selinux-2.42-1.gitad8f0f7.el7.noarch
                                                                                                               13/24
  Cleanup    : selinux-policy-targeted-3.13.1-192.el7.noarch
                                                                                                               14/24
  Cleanup    : policycoreutils-python-2.5-22.el7.x86_64
                                                                                                               15/24
  Cleanup    : selinux-policy-3.13.1-192.el7.noarch
                                                                                                               16/24
  Cleanup    : policycoreutils-2.5-22.el7.x86_64
                                                                                                               17/24
  Cleanup    : libselinux-utils-2.5-12.el7.x86_64
                                                                                                               18/24
  Cleanup    : setools-libs-3.3.8-2.el7.x86_64
                                                                                                               19/24
  Cleanup    : libselinux-python-2.5-12.el7.x86_64
                                                                                                               20/24
  Cleanup    : libsemanage-python-2.5-11.el7.x86_64
                                                                                                               21/24
  Cleanup    : libsemanage-2.5-11.el7.x86_64
                                                                                                               22/24
  Cleanup    : libselinux-2.5-12.el7.x86_64
                                                                                                               23/24
  Cleanup    : libsepol-2.5-8.1.el7.x86_64
                                                                                                               24/24
rhel-7-server-rpms/7Server/x86_64/productid
                                                                                          | 2.1 kB  00:00:00
  Verifying  : libselinux-2.5-15.el7.x86_64
                                                                                                                1/24
  Verifying  : 2:container-selinux-2.119.2-1.911c772.el7_8.noarch
                                                                                                                2/24
  Verifying  : selinux-policy-3.13.1-268.el7_9.2.noarch
                                                                                                                3/24
  Verifying  : selinux-policy-targeted-3.13.1-268.el7_9.2.noarch
                                                                                                                4/24
  Verifying  : policycoreutils-2.5-34.el7.x86_64
                                                                                                                5/24
  Verifying  : libselinux-utils-2.5-15.el7.x86_64
                                                                                                                6/24
  Verifying  : policycoreutils-python-2.5-34.el7.x86_64
                                                                                                                7/24
  Verifying  : libsemanage-python-2.5-14.el7.x86_64
                                                                                                                8/24
  Verifying  : libsemanage-2.5-14.el7.x86_64
                                                                                                                9/24
  Verifying  : libselinux-python-2.5-15.el7.x86_64
                                                                                                               10/24
  Verifying  : libsepol-2.5-10.el7.x86_64
                                                                                                               11/24
  Verifying  : setools-libs-3.3.8-4.el7.x86_64
                                                                                                               12/24
  Verifying  : libsemanage-python-2.5-11.el7.x86_64
```

```
                                                                                13/24
  Verifying  : libsemanage-2.5-11.el7.x86_64
                                                                                14/24
  Verifying  : libselinux-python-2.5-12.el7.x86_64
                                                                                15/24
  Verifying  : setools-libs-3.3.8-2.el7.x86_64
                                                                                16/24
  Verifying  : policycoreutils-2.5-22.el7.x86_64
                                                                                17/24
  Verifying  : 2:container-selinux-2.42-1.gitad8f0f7.el7.noarch
                                                                                18/24
  Verifying  : policycoreutils-python-2.5-22.el7.x86_64
                                                                                19/24
  Verifying  : selinux-policy-targeted-3.13.1-192.el7.noarch
                                                                                20/24
  Verifying  : libsepol-2.5-8.1.el7.x86_64
                                                                                21/24
  Verifying  : selinux-policy-3.13.1-192.el7.noarch
                                                                                22/24
  Verifying  : libselinux-2.5-12.el7.x86_64
                                                                                23/24
  Verifying  : libselinux-utils-2.5-12.el7.x86_64
                                                                                24/24

Updated:
  container-selinux.noarch 2:2.119.2-1.911c772.el7_8

Dependency Updated:
  libselinux.x86_64 0:2.5-15.el7                 libselinux-python.x86_64 0:2.5-15.el7
       libselinux-utils.x86_64 0:2.5-15.el7     libsemanage.x86_64 0:2.5-14.el7
  libsemanage-python.x86_64 0:2.5-14.el7         libsepol.x86_64 0:2.5-10.el7
       policycoreutils.x86_64 0:2.5-34.el7      policycoreutils-python.x86_64 0:2.5-34.el7
  selinux-policy.noarch 0:3.13.1-268.el7_9.2    selinux-policy-targeted.noarch 0:3.13.1-268.el7_9.2
    setools-libs.x86_64 0:3.3.8-4.el7

Complete!
[root@fnd451testupgrade ~]#
```

Enabling Selinux with Enforce Mode:

From IoT FND 5.0 release onwards, the Mandatory Access Controls (MAC) system such as selinux should be pre-installed, if an operating system is capable of using a MAC.

1. Check the selinux status by using the command **sestatus**.
2. Install selinux using the necessary packages, if selinux is not installed already.

   For CentOS/RHEL OS version:

   ```
   sudo yum install selinux-policy selinux-policy-targeted
   ```

3. Edit to set the selinux configuration file to enforcing mode.

   ```
   sed -i 's/^SELINUX=.*$/SELINUX=enforcing/' /etc/selinux/config
   ```

4. Reboot the virtual machine to apply the changes.

   ```
   sudo reboot
   ```

5. Ensure the selinux is enabled and in enforcing mode after rebooting the virtual machine by using the command **sestatus**.

b) **Method 2 — Manual Upgrade**: If the IoT FND server is offline, that has no internet connection because of security reasons, then you have to upgrade the container-selinux and the dependent packages manually by downloading them from the CentOS Mirror website. Download the 11 dependent packages and install them.

Run the following command to install the dependent packages in the same sequence listed in the Table 3: The dependent packages below apply only for container-selinux-2.107-3.el7.noarch.rpm..

```
rpm -Uvh package-name
```

Note Minimum required version of the container-selinux package is container-selinux-2.107-3.el7.noarch.rpm.

Note If the version of the container-selinux is higher, then the dependent rpm packages that are required is also higher. Refer to the CentOS Mirror website on the version requirements of the dependent packages.

Table 3: The dependent packages below apply only for container-selinux-2.107-3.el7.noarch.rpm.

Container-Selinux — Dependent Packages
libsepol-2.5-10.el7.x86_64.rpm
libselinux-2.5-15.el7.x86_64.rpm
libsemanage-2.5-14.el7.x86_64.rpm
libselinux-utils-2.5-15.el7.x86_64.rpm
policycoreutils-2.5-34.el7.x86_64.rpm
selinux-policy-3.13.1-268.el7_9.2.noarch.rpm
selinux-policy-targeted-3.13.1-268.el7_9.2.noarch.rpm
libsemanage-python-2.5-14.el7.x86_64.rpm
libselinux-python-2.5-15.el7.x86_64.rpm
setools-libs-3.3.8-4.el7.x86_64.rpm
policycoreutils-python-2.5-34.el7.x86_64.rpm

Step 3 Extract the `cgms rpms` files to the IoT FND server.

Based on the OS that you are using, you can extract the scripts (in ZIP format) as follows:

- For Windows—Extract the upgrade scripts on PC and then transfer to the IoT FND server.
- For extracting the upgrade scripts directly on IoT FND server or Linux—Run the following commands:

  - ```
    [root@iot-fnd opt]# ls
    cgms-influx  cgms-postgres  CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-4.7.0-101.zip  containerd
    fnd  fogd  monitor  rh  scripts
    [root@iot-fnd opt]#
    [root@iot-fnd opt]# rpm -qa | grep unzip
    unzip-6.0-20.el7.x86_64
    [root@iot-fnd opt]#
    ```

  - ```
    [root@iot-fnd opt]# unzip CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-4.7.0-101.zip
    Archive:  CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-4.7.0-101.zip
      inflating: upgrade-ova-4.7.0-101.rpm
    [root@iot-fnd opt]#
    ```

```
[root@iot-fnd opt]# ls
cgms-influx  cgms-postgres  CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-4.7.0-101.zip  containerd
fnd  fogd  monitor  rh  scripts  upgrade-ova-4.7.0-101.rpm
[root@iot-fnd opt]#
```

For example, if you are upgrading the DB and the docker server image for IoT FND release 4.7.0.

a) Download the following upgrade script from Cisco.

CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-4.7.0-101.zip

b) Extract the file to get the rpm:

upgrade-ova-4.7.0-101.rpm

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

You can copy the rpm file to any directory. In this example, the file is copied to `/opt`.

Step 4 Go to the directory where you have copied the rpm file.

For example, `cd /opt` or any directory where the *upgrade-ova-4.7.0-101.rpm file* is copied.

Step 5 Run the the following upgrade script.

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

For example, `rpm -Uvh upgrade-ova-4.7.2-8.rpm`.

The upgrade script automatically integrates the DB with IoT FND scripts (Postgres with Influx DB) and upgrades the docker server image.

Note You can find the install log information in `/root/rpm.log`.

Sample log information for the rpm upgrade script:

```
root@iot-fnd:/opt[root@iot-fnd opt]# rpm -Uvh upgrade-ova-4.7.2-8.rpm
Preparing...
(1%)###############(100%)

Updating / installing...
   1:upgrade-ova-4.7.2-8
   (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-4.7.2-8.x86_64.rpm
Preparing...                          ########################################
Updating / installing...
cgms-postgres-4.7.2-8                 ########################################
Cleaning up / removing...
cgms-postgres-4.7.0-101               ########################################

Mon May  9 01:59:47 PDT 2022 Upgrading cgms-influx-4.7.2-8.x86_64.rpm
Preparing...                          ########################################
Updating / installing...
cgms-influx-4.7.2-8                   ########################################
Cleaning up / removing...
cgms-influx-4.7.0-101                 ########################################
```

```
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
```

Example

Manual Upgrade of IoT FND 4.5.1 to Later Versions—Use this upgrade procedure ONLY if you want to upgrade on your own without using Cisco rpm (*upgrade-ova-4.7.0-101.rpm*) that is provided to you:

1. Extract the rpm scripts by running the following command:

```
rpm2cpio upgrade-ova-4.7.0-101.rpm | cpio -idmv
```

```
[root@iot-fnd opt]# rpm2cpio upgrade-ova-4.7.0-101.rpm | cpio -idmv
./upgrade-ova-4.7.0-101
./upgrade-ova-4.7.0-101/Application-Watchdog
./upgrade-ova-4.7.0-101/Application-Watchdog/README.md
./upgrade-ova-4.7.0-101/Application-Watchdog/monitor-args.ini
./upgrade-ova-4.7.0-101/Application-Watchdog/monitor.sh
./upgrade-ova-4.7.0-101/Application-Watchdog/monitor_app_health.py
./upgrade-ova-4.7.0-101/Application-Watchdog/plugin_categories.py
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_registration.py
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_registration.yapsy-plugin
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_stats_collection.py
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_stats_collection.yapsy-plugin
./upgrade-ova-4.7.0-101/Application-Watchdog/postgres-vacuum.sh
./upgrade-ova-4.7.0-101/Application-Watchdog/setup.sh
./upgrade-ova-4.7.0-101/Continuous-Integration
./upgrade-ova-4.7.0-101/Continuous-Integration/README.md
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/conf
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/conf/fnd-env.list
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/data
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/data/cgms_keystore.selfsigned
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/data/cisco-sudi-ca.pem
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/data/userPropertyTypes.xml
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/logs
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/scripts
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/scripts/fnd-container.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/scripts/fnd-task
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/scripts/setup-IPv6-network.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/fnd/scripts/upgrade.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/conf
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/conf/fogd-env.list
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/scripts
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/scripts/fogd-container.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/scripts/fogd-info.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/scripts/fogd-stats.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/fogd/scripts/fogd-task
./upgrade-ova-4.7.0-101/Continuous-Integration/scripts
./upgrade-ova-4.7.0-101/Continuous-Integration/scripts/status.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/upgrade-ova.spec
```

```
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/field-network-director.conf
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/field-network-director.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/fog-director.conf
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/fog-director.sh
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/influxdb.conf
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/kapacitor.conf
./upgrade-ova-4.7.0-101/Continuous-Integration/watchdog/postgresql.conf
./upgrade-ova-4.7.0-101/rpms
./upgrade-ova-4.7.0-101/rpms/cgms-influx-4.7.0-101.x86_64.rpm
./upgrade-ova-4.7.0-101/rpms/cgms-postgres-4.7.0-101.x86_64.rpm
./upgrade-ova-4.7.0-101/rpms/delay-installer.sh
./upgrade-ova-4.7.0-101/rpms/migrate-postgres.sh
./upgrade-ova-4.7.0-101/rpms/monit-5.25.3-1.el7.x86_64.rpm
./upgrade-ova-4.7.0-101/rpms/postgresql12-12.4-1PGDG.rhel7.x86_64.rpm
./upgrade-ova-4.7.0-101/rpms/postgresql12-contrib-12.4-1PGDG.rhel7.x86_64.rpm
./upgrade-ova-4.7.0-101/rpms/postgresql12-libs-12.4-1PGDG.rhel7.x86_64.rpm
./upgrade-ova-4.7.0-101/rpms/postgresql12-server-12.4-1PGDG.rhel7.x86_64.rpm
./upgrade-ova-4.7.0-101/Application-Watchdog/monitor_app_health.pyc
cpio: ./upgrade-ova-4.7.0-101/Application-Watchdog/monitor_app_health.pyo linked to
./upgrade-ova-4.7.0-101/Application-Watchdog/monitor_app_health.pyc
./upgrade-ova-4.7.0-101/Application-Watchdog/monitor_app_health.pyo
./upgrade-ova-4.7.0-101/Application-Watchdog/plugin_categories.pyc
cpio: ./upgrade-ova-4.7.0-101/Application-Watchdog/plugin_categories.pyo linked to
./upgrade-ova-4.7.0-101/Application-Watchdog/plugin_categories.pyc
./upgrade-ova-4.7.0-101/Application-Watchdog/plugin_categories.pyo
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_registration.pyc
cpio: ./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_registration.pyo
linked to ./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_registration.pyc
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_registration.pyo
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_stats_collection.pyc
cpio: ./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_stats_collection.pyo
 linked to
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_stats_collection.pyc
./upgrade-ova-4.7.0-101/Application-Watchdog/plugins/container_stats_collection.pyo
189297 blocks
[root@iot-fnd opt]#
[root@iot-fnd opt]#
[root@iot-fnd opt]# ls
cgms-influx  cgms-postgres  containerd  fnd  fogd  monitor  rh  scripts
upgrade-ova-4.7.0-101  upgrade-ova-4.7.0-101.rpm
[root@iot-fnd opt]#
[root@iot-fnd opt]#
[root@iot-fnd opt]# cd upgrade-ova-4.7.0-101
[root@iot-fnd upgrade-ova-4.7.0-101]# ls
Application-Watchdog  Continuous-Integration  rpms
[root@iot-fnd upgrade-ova-4.7.0-101]#
[root@iot-fnd upgrade-ova-4.7.0-101]#
[root@iot-fnd upgrade-ova-4.7.0-101]# cd rpms
[root@iot-fnd rpms]#
[root@iot-fnd rpms]# ls
cgms-influx-4.7.0-101.x86_64.rpm    migrate-postgres.sh
postgresql12-contrib-12.4-1PGDG.rhel7.x86_64.rpm
cgms-postgres-4.7.0-101.x86_64.rpm  monit-5.25.3-1.el7.x86_64.rpm
postgresql12-libs-12.4-1PGDG.rhel7.x86_64.rpm
delay-installer.sh                  postgresql12-12.4-1PGDG.rhel7.x86_64.rpm
postgresql12-server-12.4-1PGDG.rhel7.x86_64.rpm
[root@iot-fnd rpms]#
```

2. Run the following script.

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

3. Select options 3 and 4 in a sequence to integrate the DB with IoT FND scripts (Postgres and Influx) as shown in the log information:

```
[root@iot-fnd rpms]# /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
       For FND Postgres RPM upgrade: Requires <path to cgms-postgres.rpm>
       FND Influx RPM upgrade: Requires <path to cgms-influx.rpm>

1) Load container images     4) FND Influx RPM upgrade
2) Container reload          5) Quit
3) FND Postgres RPM upgrade
Enter your choice: 3
Enter cgms-postgres rpm file path: cgms-postgres-4.7.0-101.x86_64.rpm
Stopping FND container...
fnd-container
Preparing...                          ################################# [100%]
Updating / installing...
   1:cgms-postgres-4.7.0-101          ################################# [ 50%]
Cleaning up / removing...
   2:cgms-postgres-4.5.1-11           ################################# [100%]
Starting FND container...
Enter your choice: fnd-container
^C
[root@iot-fnd rpms]# pwd
/opt/upgrade-ova-4.7.0-101/rpms
[root@iot-fnd rpms]# /opt/fnd/scripts/fnd-container.sh status
fnd-container is running, pid=37806
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %
               NET I/O             BLOCK I/O           PIDS
61921642276c        fnd-container       2.41%               2.764GiB / 23.38GiB   11.82%
              11.3MB / 9.84MB     0B / 2.33MB         315
[root@iot-fnd rpms]#
[root@iot-fnd rpms]# /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
       For FND Postgres RPM upgrade: Requires <path to cgms-postgres.rpm>
       FND Influx RPM upgrade: Requires <path to cgms-influx.rpm>

1) Load container images     4) FND Influx RPM upgrade
2) Container reload          5) Quit
3) FND Postgres RPM upgrade
Enter your choice: 4
Enter cgms-influx rpm file path: cgms-influx-4.7.0-101.x86_64.rpm
Stopping FND container...
fnd-container
Preparing...                          ################################# [100%]
Updating / installing...
   1:cgms-influx-4.7.0-101            ################################# [ 50%]
Cleaning up / removing...
   2:cgms-influx-4.5.1-11             ################################# [100%]
Starting FND container...
Enter your choice: fnd-container
^C
[root@iot-fnd rpms]#
[root@iot-fnd rpms]# /opt/fnd/scripts/fnd-container.sh status
fnd-container is running, pid=45404
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %
               NET I/O             BLOCK I/O           PIDS
61921642276c        fnd-container       2.44%               2.095GiB / 23.38GiB   8.96%
               11.3MB / 9.84MB     0B / 2.45MB         315
[root@iot-fnd rpms]#
```

Note The options 3 and 4 present in the script, **./upgrade.sh**, ONLY install the database integration scripts and they do not upgrade the entire DB.

4. To upgrade the entire DB, contact your DB Administrator or visit https://www.postgresql.org/docs/current/upgrading.html to upgrade the Postgres.
5. Install the docker server image from https://docs.docker.com/engine/install/rhel/.

What to do next

Upgrading IoT FND and FD Container Images, on page 45

Upgrading IoT FND and FD Container Images

Before you begin

- Pre-Upgrade Checklist, on page 29
- Upgrading the Database and Docker Server Image, on page 33

Step 1 Run the following script:

```
/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: 4.7.2-8
Downloading FND docker image...
4.7.2-8: Pulling from field-network-director-dev-docker/fnd-image
42ae914c6f41: 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:4.7.2-8
dockerhub.cisco.com/field-network-director-dev-docker/fnd-image:4.7.2-8
Downloading Fog Director docker image...
4.7.2-8: 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:4.7.2-8
dockerhub.cisco.com/fog-director-dev-docker/fogd-image:4.7.2-8

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 ~]#
```

Step 2 Enter **1** to load container images.

```
[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
```

Step 3 Download the container image for IoT FND from devhub-docker.cisco.com.

Note You need valid CCO credentials to log into Cisco external docker registry.

Step 4 After the images are downloaded successfully, enter **2** to reload container.

IoT FND upgrade is complete.

```
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...
3da4837b448548c06e0ee2eac75696231462a2bba480bfa6a75358812095da60
Stopping Fog Director container...
fogd-container
Remove Fog Director container...
fogd-container
Remove FND container...
fnd-container
Prune Docker container...
Starting FND container...
3da4837b448548c06e0ee2eac75696231462a2bba480bfa6a75358812095da60
Stopping Fog Director container...
fogd-container
Remove Fog Director container...
fogd-container
Prune Docker container...
Starting Fog Director container...
6b6fdbb4810bb8cb471e16717a9a3adbc4b3a9f666e5a423e62c7d57014c8c5c

1) Load container images
2) Container reload
3) Quit
Enter your choice: 3
You have new mail in /var/spool/mail/root
```

Enter **3** to Quit the menu.

What to do next

Post-Upgrade Checklist, on page 48

Post-Upgrade Checklist

Attention From IoT FND 4.12 onwards, use the following credentials for SSH access after upgrading OVA. The existing credentials username/password (root/cisco123) is disabled for 4.12 and later releases:

- Username: fnduser
- Password: C!sco123

See List item. for resetting password.

Step 1 Restart Postgres service if the current IoT FND release is prior to 4.9.1 and the target IoT FND release is 4.9.1 or above.

Step 2 Check the DB and IoT FND status by running the following commands:

- `/opt/scripts/status.sh`
- `docker version`
- `/opt/fnd/scripts/fnd-container.sh status`
- `docker exec -it fnd-container /etc/init.d/cgms status`

Note On completion of the upgrade process, restart the IoT FND container after replacing the files from backup to their original location.

```
Login to the FND container
[root@iot-fnd ~]# docker exec -it fnd-container /bin/bash
[root@fnd-server /]#
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/jbossas.keystore.password.bkp1
/opt/cgms/server/cgms/conf/jbossas.keystore.password
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/jbossas.keystore.bkp1
/opt/cgms/server/cgms/conf/jbossas.keystore
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/vault.keystore.bkp1
/opt/cgms/server/cgms/conf/vault.keystore
[root@fnd-server /]# cp /opt/cgms/server/cgms/conf/VAULT.dat.bkp1
/opt/cgms/server/cgms/conf/VAULT.dat
[root@fnd-server /]# cp /opt/cgms/server/cgms/deploy/security-service.xml.bkp1
/opt/cgms/server/cgms/deploy/security-service.xml
[root@fnd-server /]#exit
[root@fnd ~]# /opt/fnd/scripts/fnd-container.sh stop
[root@fnd ~]# /opt/fnd/scripts/fnd-container.sh start
```

Step 3 Log into IoT FND to check if the services are working fine.

For example, you can refresh the metrics for a couple of devices or add/delete devices using CSV.

Upgrading IoT FND from 4.5.1 to later releases and Updating RHEL OS

Note This procedure is applicable only when you want to upgrade IOT FND version from FND 4.5.1 to FND 4.9.x along with RHEL base OS upgrade.

Step 1 Download the latest 4.5.1-11 upgrade zip from Cisco Download page.

CISCO-IOTFND-VPI-K9-UPGRADE-SCRIPTS-4.5.1-11.zip

Step 2 Extract the file to get the rpm.

Step 3 Install the upgrade rpm using the following command.

```
rpm -ivh upgrade-ova-4.5.1-11.rpm
```

Step 4 Run the `./upgrade.sh` script in /opt/fnd/scripts directory.

Note You can skip the FND postgres rpm and FND influx upgrade rpm.

Step 5 To upgrade IoT FND from 4.5.1-11 to 4.7.2-8, download the latest 4.7.2-8 upgrade rpm from the Cisco Download page.

Step 6 Upgrade the upgrade-ova-4.7.2- 8.rpm using the following command.

```
rpm -Uvh upgrade-ova-4.7.2-8.rpm
```

Step 7 Run the `./upgrade.sh` script in /opt/fnd/scripts directory.

Note IoT FND OVA upgrade will NOT upgrade the RHEL OS version. After upgrading the OVA, it is recommended to upgrade the OS as well.

Step 8 Upgrade base OS from RHEL 7.5 to 7.9.

Step 9 To upgrade from IoT FND 4.7.2-8 to 4.9.x, download the latest 4.9.x upgrade rpm from Cisco Download page.

Step 10 Upgrade the upgrade-ova-4.9. x.rpm using the following command.

```
rpm -Uvh upgrade-ova-4.9.x.rpm
```

Step 11 Run the `./upgrade.sh` script in /opt/fnd/scripts directory.

Step 12 Upgrade base OS from RHEL 7.9 to 8.6.

Step 13 IoT FND 4.9.0 OVA is bundled with Postgres 12.9 rpms of rhel7. In order to upgrade Postgres 12.9 rpms of base OS rhel8 manually:

Note Starting from IoT FND 4.9.1 release, the postgres rpm upgrade is automated.

a) Run the following commands to uninstall the old Postgres (**rhel7**) rpms.

```
rpm -qa | grep postgres
```

```
rpm -e <postgresql12.9xxxx.rhel7.x86_64.rpm>
```

Note Keep the cgms-postgres rpm.

b) Download all the four Postgres dependent packages from the YUM link and place the packages in /opt/ directory.

postgresql12-libs-12.9-1PGDG.rhel8.x86_64.rpm

postgresql12-12.9-1PGDG.rhel8.x86_64.rpm
postgresql12-server-12.9-1PGDG.rhel8.x86_64.rpm
postgresql12-contrib-12.9-1PGDG.rhel8.x86_64.rpm

c) Install all the above rpms in the same sequential order with the following command.

```
rpm -ivh <12.9.1PGDG.rhel8.rpm>
```

d) Make symlink with below command.

```
chkconfig postgresql-12 on
```

e) Start the postgres service:

```
service postgresql-12.service start
```

f) Check if the postgres status is Active (running):

```
service postgresql-12.service status
```

g) Reload all the required container with FND upgrade script by using 'Option 2) Container Reload'.

- Run the `./upgrade.sh` script in /opt/fnd/scripts/ directory.
- Enter 2 to reload container.

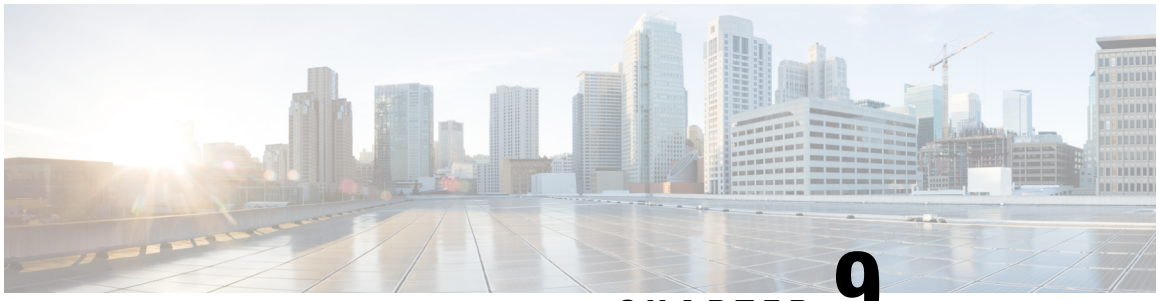
```
[[root@bgl12-iot-fnd scripts]# cd /opt/fnd/scripts/
[[root@bgl12-iot-fnd scripts]# ./upgrade.sh

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
```

- Enter 3 to quit menu.

h) Run `./status.sh` script in /opt/scripts/ directory to get the running status of all the required services.

```
[root@bgl12-iot-fnd ~]# cd /opt/scripts/
[root@bgl12-iot-fnd 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 2022-10-31 13:27:20 IST; 4 days ago
     Docs: https://www.postgresql.org/docs/12/static/
  Process: 271967 ExecStartPre=/usr/pgsql-12/bin/postgresql-12-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
 Main PID: 271973 (postmaster)
    Tasks: 27 (limit: 152444)
   Memory: 1.2G
--------------------
● 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 Thu 2022-10-27 12:59:32 IST; 1 weeks 1 days ago
     Docs: https://docs.influxdata.com/influxdb/
 Main PID: 1520 (influxd)
    Tasks: 21 (limit: 152444)
   Memory: 611.5M
--------------------
● kapacitor.service - Time series data processing engine.
   Loaded: loaded (/usr/lib/systemd/system/kapacitor.service; enabled; vendor preset: disabled)
   Active: active (running) since Thu 2022-10-27 12:59:32 IST; 1 weeks 1 days ago
     Docs: https://github.com/influxdb/kapacitor
 Main PID: 1519 (kapacitord)
    Tasks: 14 (limit: 152444)
   Memory: 59.7M
--------------------
fnd-container is running, pid=272372
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
CONTAINER ID   NAME           CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
9b27aeac63fe   fnd-container   1.81%     1.738GiB / 23.32GiB   7.45%     963MB / 943MB   8.19kB / 3.2MB   628
--------------------
fogd-container is running, pid=274778
*** WARNING : deprecated key derivation used.
Using -iter or -pbkdf2 would be better.
CONTAINER ID   NAME            CPU %     MEM USAGE / LIMIT     MEM %     NET I/O         BLOCK I/O        PIDS
b3d97b27913e   fogd-container   0.51%     804.6MiB / 23.32GiB   3.37%     665MB / 1.23GB   713kB / 8.19kB   91
--------------------
```

i) Log into IoT FND UI to check if the services are working fine. For example, you can refresh the metrics for a couple of devices or add/delete devices using CSV.



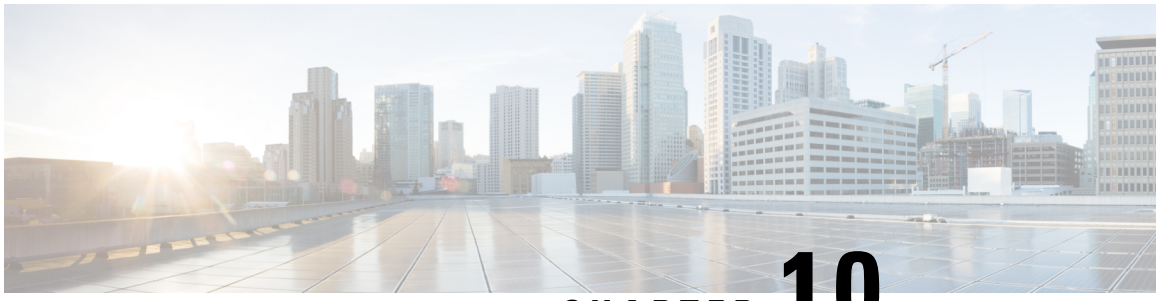
CHAPTER 9

Obtaining Status of All Services Running on the Host

Use the **status.sh** script in the following directory to show the status of all services running on the host.

```
cd /opt/scripts
```

```
[root@iot-fnd ~]# cd /opt/scripts/
[root@iot-fnd scripts]# ./status.sh
--------------------
● postgresql-9.6.service - PostgreSQL 9.6 database server
   Loaded: loaded (/usr/lib/systemd/system/postgresql-9.6.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2018-06-15 17:02:07 PDT; 13min ago
     Docs: https://www.postgresql.org/docs/9.6/static/
  Process: 1016 ExecStartPre=/usr/pgsql-9.6/bin/postgresql96-check-db-dir ${PGDATA} (code=exited, status=0/SUCCESS)
 Main PID: 1070 (postmaster)
    Tasks: 24
   Memory: 166.2M
--------------------
● 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 Fri 2018-06-15 17:02:03 PDT; 13min ago
     Docs: https://docs.influxdata.com/influxdb/
 Main PID: 1024 (influxd)
    Tasks: 11
   Memory: 47.4M
--------------------
fnd-container is running, pid=2064
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O           PIDS
a67827470562        fnd-container       1.04%               1.064GiB / 23.38GiB   4.55%               6.69MB / 8.19MB     581MB / 2.22MB      275
--------------------
fogd-container is running, pid=5192
CONTAINER ID        NAME                CPU %               MEM USAGE / LIMIT     MEM %               NET I/O             BLOCK I/O           PIDS
f6c0c5c313cb        fogd-container      1.64%               762.3MiB / 23.38GiB   3.18%               1.84MB / 3.45MB     106kB / 184kB       117
--------------------
[root@iot-fnd scripts]#
```



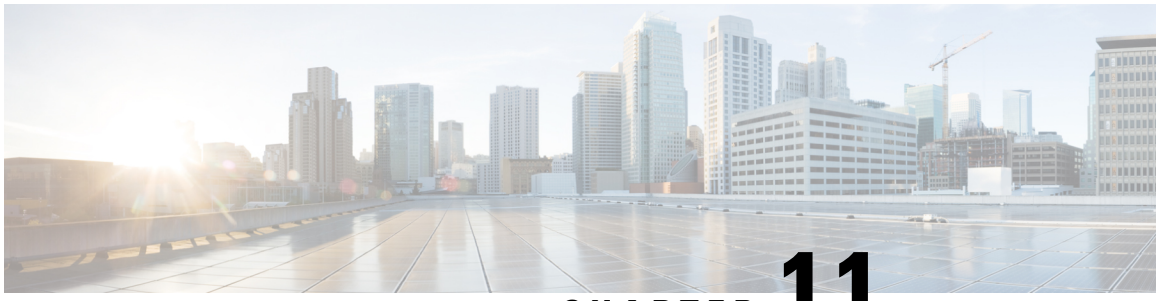
CHAPTER 10

Backup and Restore

You can export the entire OVA image file as backup, port it to different deployment or restore from an older image file.

Step 1 Power down the OVA in vSphere Client.

Step 2 Select the **OVA**, and then select **File** > **Export** > **Export OVF Template**.



CHAPTER 11

Setting the Time and Timezone Using NTP Service

Use the **timedatectl** command on the Host VM to perform following operations to sync the time between the host and the docker:

- Displaying the Current Date and Time: **timedatectl**
- Changing the Current Time: **timedatectl set-time** HH:MM:SS
- Changing the Current Date: **timedatectl set-time** YYYY-MM-DD
- Listing the Time Zone: **timedatectl list** - timezones
- Changing the Time Zone: **timedatectl set-timezone** time_zone
- Enabling NTP Service: **timedatectl set-ntp** yes

```
[root@iot-fnd ~]# timedatectl
      Local time: Tue 2018-08-28 07:18:37 PDT
  Universal time: Tue 2018-08-28 14:18:37 UTC
        RTC time: Tue 2018-08-28 14:18:37
       Time zone: America/Los_Angeles (PDT, -0700)
     NTP enabled: yes
NTP synchronized: yes
 RTC in local TZ: no
      DST active: yes
 Last DST change: DST began at
                  Sun 2018-03-11 01:59:59 PST
                  Sun 2018-03-11 03:00:00 PDT
 Next DST change: DST ends (the clock jumps one hour backwards) at
                  Sun 2018-11-04 01:59:59 PDT
                  Sun 2018-11-04 01:00:00 PST
[root@iot-fnd ~]#
```