

Bare Metal Server Handbook for Cisco IoT FND

First Published: 2025-10-03

Last Modified: 2025-11-14

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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

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

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

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

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

© 2025 Cisco Systems, Inc. All rights reserved.



CONTENTS

CHAPTER 1 **Introduction** 1

Cisco IoT FND On Bare Metal Servers 1

Components of Cisco IoT FND 2

Managed Devices 3

Field Area Routers 3

Headend Routers 4

Endpoints 4

Gateways 4

Cisco IoT FND Managed Devices 5

Security Components in a FAN 7

Public Key Infrastructure 7

Registration Authority 7

Hardware Security Module 7

Software Security Module 7

Authenticate, Authorize, and Account 7

DHCP Sever 8

CHAPTER 2 **Install** 9

System Requirements 9

Obtain Cisco IoT FND and Cisco Network Register Licenses 12

Install Linux Packages 12

Obtain Cisco IoT FND RPM Packages 12

Configure NTP Services 13

Use the Map View 14

Signature Verification 14

Verify the signature 14

Cisco Provided ISO and RPM Images 16
Certificates 17
Cisco IoT FND certificates 17
Types of certificates 17
Role of certificates 17
Keystore 18
Generate and export certificates 19
Create certificate template using Microsoft CA 19
Enable a certificate template 19
Generate a certificate for TPS Proxy 19
Export the certificates 20
Install certificates on Cisco IoT FND and TPS proxy servers 20
Create the cgms_keystore file using Keytool 20
Copy the cgms_keystore file to Cisco IoT FND 22
Extract the cgms.pem and cgms.key files 22
Copy the cgms.pem and cgms.key files 22
Import the CA certificate 22
Import the SUDI certificate 24
Configure Cisco IoT FND to access the keystore 24
Configure the TPS Proxy to access the keystore 25
Install Supporting Applications and Hardware 26
Install Cisco IoT FND For the First Time 26
Install and Setup Cisco IoT FND Database Overview 26
Setup Cisco IoT FND Database 28
Define Oracle Database Environment Variables 28
Install Cisco IoT FND Oracle Database Scripts 29
Create Cisco IoT FND Oracle Database 30
Start the Cisco IoT FND Oracle Database 31
Upgrade Cisco IoT FND Database 31

CHAPTER 3 **Deploy** 33
Deploy Cisco IoT FND Using Bare Metal Servers 33
Recommended Computing Resources 34
Recommended Resources For Managing Mesh Endpoints 34

Recommended Resources For Managing Routers 35

CHAPTER 4 Upgrade 37

Upgrade Cisco IoT FND 37

Upgrade Cisco IoT FND TPS Proxy 38

CHAPTER 5 Troubleshoot 41

Determine the RHEL Version 56

Debug Using Cisco IoT FND 57

Encrypt and Decrypt Passwords Using Cisco IoT FND 57

Find the Cisco IoT FND Version 58

SSL Debug 58

Troubleshoot Incorrect CGNA Profile URL 59

Troubleshoot CGMS Errors 59

Troubleshoot TPS Proxy Errors 60

Troubleshoot HER FlexVPN 61

Troubleshoot SSM Errors 61

Troubleshoot RA Errors 62

Troubleshoot Profile Environment Variable 62

Troubleshoot High CPU Issues 63

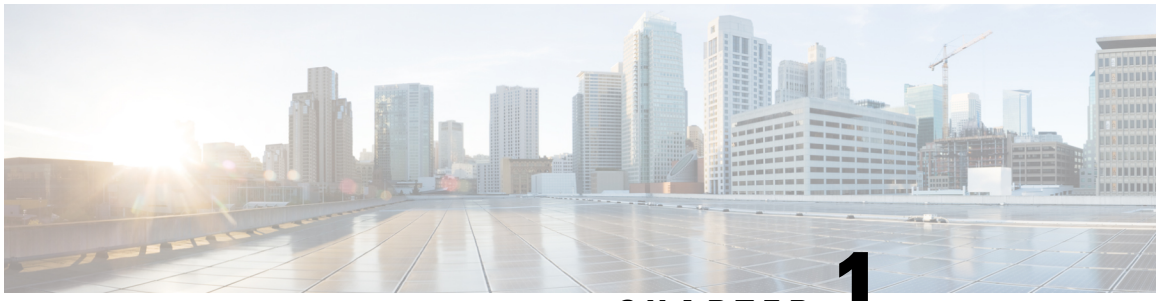
Troubleshoot Tunnel Provisioning Issues 66

Troubleshoot Tunnel Provisioning DHCP Configuration Issue 67

Troubleshoot Profile Environment Variable 68

Verify Tunnel Configuration Information 69

Troubleshoot Archive Directory Creation During ZTD 69



CHAPTER 1

Introduction

- Cisco IoT FND On Bare Metal Servers, on page 1
- Components of Cisco IoT FND, on page 2

Cisco IoT FND On Bare Metal Servers

Cisco IoT Field Area Network Director (FND) is a network management system for the Cisco Field Area Network (FAN).

The FAN is a communication network overlay that provides an end-to-end, IP-based reference architecture from data centers responsible for grid operations to homes and industries. It seamlessly connects Cisco IoT devices and services in a power grid infrastructure and provides multiservice enablement, security, fog computing, and more. It also serves as a communication platform for various advanced applications, including Distribution Automation (DA), Advanced Metering Infrastructure (AMI), and renewable energy.

Cisco IoT FND is a core component of a FAN. Through an intuitive user interface, it provides features for managing, monitoring, and troubleshooting supported devices and network operations in FANs of almost any size. The supported devices include various models of Field Area Routers (FARs), Headend Routers (HERs), gateways, and endpoints. Cisco IoT FND ensures smooth device setup through Zero-touch Deployment (ZTD), secure communication, seamless scalability, and increased network efficiency. It provides real-time visibility of devices and assets, proactive fault detection, and many other important features for managing the FAN.

Cisco IoT FND uses digital certificates that undergo a stringent authentication process based on 802.1x standards for data protection and security. Cisco IoT FND uses multiple IP encryptions and a standard FlexVPN solution to establish a secure data transmission environment that implements hop-by-hop encryption to safeguard data integrity and confidentiality.

Here are some of the key features of Cisco IoT FND:

- **Lifecycle management**:
  - On-premises network management system for FAN devices
  - Secure zero-touch provisioning for enrollment and deployment of devices
  - Device inventory
  - Secure tunnel provisioning
  - Rich APIs for third-party application integration

- **Network optimization**:
  - Configuration and firmware management
  - Network management for constrained bandwidth
  - Multitenancy and RBAC support
  - Network troubleshooting
- **Real-time monitoring**:
  - Enterprise-class visibility for routers and endpoints
  - Device status, health, and performance metrics
  - Alerts for critical events
  - Location tracking of all network assets and geofencing
  - Dynamic and customizable dashboard

Components of Cisco IoT FND

Cisco IoT FND lets you manage other components in a FAN, and perform various monitoring and administrative operations. The key FAN components include:

- Cisco IoT FND
- Routers
- Endpoints
- Gateways
- Hardware Security Module (HSM)
- Software Security Module (SSM)

Here is an overview of the roles and interactions of these key FAN components.

Cisco IoT Application Server

The Cisco IoT FND application server resides in a data center and performs the functions to monitor and manage the FAN and devices. The application server hosts the Cisco IoT FND application and interacts with many of the components in the headend, including the Cisco IoT FND database server, DHCP server, headend routers (HERs), and tunnel provisioning server (TPS). This server hosts the Cisco IoT FND user interface using which you perform Cisco IoT FND operations and management procedures. You can view information about the network, devices, and related items. The application server also stores the Cisco IoT FND log files.

The Cisco IoT FND application server runs on the Red Hat Enterprise Linux (RHEL) operating system and you can install the application server on a bare metal server.

Cisco IoT Database Server

The Cisco IoT FND database server resides in the headend and is the storage repository for the data that Cisco IoT FND generates or collects. This data includes metrics, device properties such as firmware images, configuration templates, and event notifications.

The Cisco IoT FND database server runs on the Red Hat Enterprise Linux (RHEL) operating system and can run the Oracle database. When running Oracle, you can install the Cisco IoT FND database server on a bare metal server.

The Cisco IoT FND application server is the only component that interacts directly with the Cisco IoT FND database server.

Tunnel Provisioning Server

The tunnel provisioning server (TPS) resides in the DMZ and is a proxy server for Cisco IoT FND. The TPS provides a bridge for the communication between Cisco IoT FND and FARs. It relays tunnel requests from FARs to Cisco IoT FND and provides FARs with the configuration for the tunnel to the headend.

When they first start up, routers communicate with Cisco IoT FND through the TPS. After Cisco IoT FND provisions tunnels, routers communicate with Cisco IoT FND directly.

Load Balancer

An optional load balancer provides Cisco IoT FND server high availability. You can connect multiple Cisco IoT FND servers to a load balancer.

Load balancing is configured using a third-party device and is supported only in an Cisco IoT FND bare metal server deployment.

Managed Devices

Cisco IoT FND lets you manage and monitor the following types of devices in a FAN:

- Field Area Routers (FARs)
- Headend Routers (HERs)
- Endpoints
- Gateways

Field Area Routers

Field Area Routers (FARs) are integral components of a Field Area Network (FAN). They establish communication with headend routers via tunnels, which are provisioned and managed by the Tunnel Provisioning System (TPS).

Use Cisco IoT FND to manage the lifecycle of FARs and to monitor and troubleshoot their performance.

Here's a list of Cisco IoT FND supported Cisco FARs:

- Cisco Catalyst IR1800 Rugged Series Router
- Cisco Catalyst IR8100 Heavy-Duty Series Router
- Cisco Catalyst IR1100 Rugged Series Router

- Cisco 1000 Series Connected Grid Router
- Cisco 800 Series Industrial Integrated Services Router
- Cisco 800 Series Router

Headend Routers

The Headend routers (HERs) reside in the Demilitarized Zone (DMZ) and connect FARs to other headend components in the DMZ and data center. These components include Cisco IoT FND, the Authentication, Authorization, and Accounting (AAA) server, the Dynamic Host Configuration Protocol (DHCP) server, and utility application servers like Supervisory Control and Data Acquisition (SCADA) and Meter Data Management System (MDMS) servers.

IoT FND does not provide options for managing lifecycle activities of headend routers, but it does provide options for monitoring their status and configuration. It also provides options for provisioning tunnels between HERs and FARs and for viewing the status of tunnels.

You can monitor the status of HERs and provision tunnels between HERs and FARs using Cisco IoT FND. You can view the status of these tunnels as well. You can't manage the lifecycle of HERs using Cisco IoT FND.

Here's a list of Cisco IoT supported Cisco HERs:

- Cisco 8000 Series Router
- Cisco ASR 1000 Series Aggregation Services Routers (ASR 1001 or 1002)
- Cisco 4000 Series Integrated Services Router (ISR)
- Cisco Cloud Services Router 1000V Series (CSR)

Endpoints

Endpoints include Cisco and third-party devices, such as meters from Itron and Landis+Gyr, range extenders, cameras, battery endpoints, and cellular endpoints.

Use Cisco IoT FND to manage the lifecycle of endpoints, monitor them, and troubleshoot any issues.

Here's a list of Cisco IoT FND supported endpoints:

- Cisco 500 Series Wireless Personal Area Network (WPAN) Industrial Routers (IR500)
  - Cisco IR510 (DA Gateway)
  - Cisco IR530 Series Resilient Mesh Range Extenders
- Mesh endpoints (from Itron and Landis+Gyr)

Gateways

Gateways connect the FAN to other networks or the internet. Cisco IoT FND offers options to manage the lifecycle of gateways, as well as to monitor and troubleshoot them.

Here's a list of Cisco IoT FND supported gateways:

- Cisco IC3000 Industrial Compute Gateway

- Long Range Wide Area Network (LoRaWAN) Interface Module for Cisco 800 Series Industrial Integrated Services Routers (IR800)
- Landis+Gyr N2450 Network Gateway

Cisco IoT FND Managed Devices

Cisco IoT FND manages the lifecycle of Cisco and third-party devices, including automated deployment, firmware updates, and real-time visualization of device operations.

Cisco Devices

Device Type	Device
FARs	• Cisco Catalyst IR1800 Rugged Series Routers • Cisco Catalyst IR8100 Heavy-Duty Series Routers (IR8140) • Cisco 1101 Series Industrial Integrated Services Routers (IR1101) • Cisco 1000 Series Connected Grid Routers (CGR1120 and CGR1240) • Cisco 800 Series Industrial Integrated Services Routers (IR800) • Cisco 800 Series Access Points (AP800) when integrated with C800 and IR829 devices
HERs	• Cisco 8000 Series Routers • Cisco ASR 1000 Series Aggregation Services Routers (ASR 1001 or 1002) serving as a head-end router • Cisco 4000 Series Integrated Services Routers (ISR) • Cisco Cloud Services Router 1000V Series (CSR)

Device Type	Device

Endpoints	• Cisco 500 Series Wireless Personal Area Network (WPAN) Industrial Routers (IR500)
	  • Cisco IR510 (DA Gateway)
	  • Cisco IR530 Series Resilient Mesh Range Extenders
	• Mesh endpoints
	Note
	Cisco IoT FND manages third-party endpoints only if the endpoints are running Cisco Cisco Smart Message Protocol (CSMP) agent.

Gateways	• Cisco IC3000 Industrial Compute Gateway • Long Range Wide Area Network (LoRaWAN) Interface Module for Cisco 800 Series Industrial Integrated Services Routers (IR800)

Third-Party Devices

Vendors	Products
ITRON	• OpenWay Riva ACT • OpenWay Riva ACT Gateway • OpenWay Riva ACT Extender • OpenWay Riva BACT • OpenWay Riva BACT Controller • OpenWay Riva CAM
Landis+Gyr (L+G)	• Landis+Gyr Series 6 N2450 Network Gateway N2450 Gateway • Revelo E360/E660 • M125 RF Residential Gas Communications Module • M225 RF Commercial & Industrial Gas and Communications Module • Landis+Gyr R651 Network Router

Vendors	Products
MMB Networks	• Nansen/Sanxing • Hexing • Grupo Union (AMS) • Seltic (Donsun)

Security Components in a FAN

Here's a list of key security componets in a FAN:

Public Key Infrastructure

A Public Key Infrastructure (PKI) includes hardware and software to manage and control digital certificates and public-key encryption. In a FAN solution, it issues digital certificates to Cisco IoT FND and FARs, enabling them to authenticate and securely communicate with each other and other FAN components.

The PKI includes a Certificate Authority (CA) server that issues RSA certificates for routers, Elliptic Curve Cryptography (ECC) certificates for endpoints, and web certificates for Cisco IoT FND. Routers use the Simple Certificate Enrollment Protocol (SCEP) to request and receive certificates, while endpoints use Enrollment over Secure Transport (EST) for the same purpose.

Registration Authority

A Registration Authority (RA) acts as a proxy for FARs when they request digital certificates. Since a CA server cannot reside in a public network, FARs use the RA to communicate with the CA and obtain digital certificates for secure communication. The RA facilitates requests between public and private networks.

Hardware Security Module

A Hardware Security Module (HSM) is a third-party appliance used to sign CSMP messages that Cisco IoT FND sends to meters and Cisco IR500 endpoints. An HSM scales to support large FANs and ensures High Availability (HA). We recommend using an HSM instead of the SSM in production environments.

Software Security Module

The software security module (SSM) is an optional component of Cisco IoT FND. It is used to verify the CSMP messages that Cisco IoT FND sends to meters and to Cisco IR500 endpoints. The SSM is bundled with the Cisco IoT FND image.

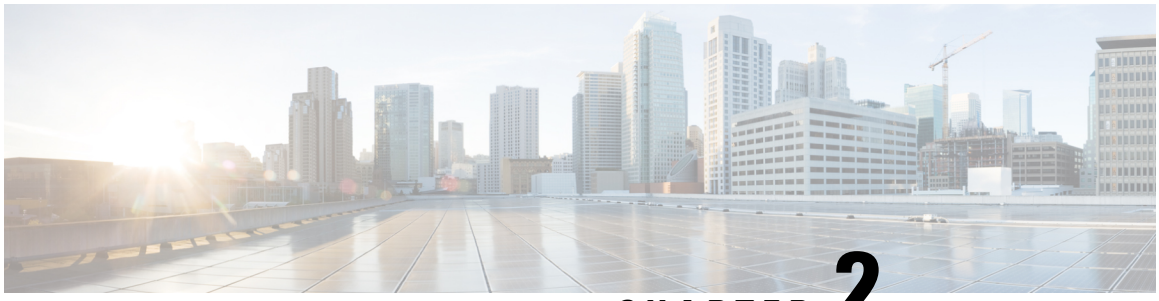
The SSM has a limited scaling capability and does not support high availability. We recommend that a hardware security module be used instead of the SSM in production environments.

Authenticate, Authorize, and Account

AAA is a security framework that authenticates FARs and endpoints before they join the FAN, authorizes these devices to join the FAN, and tracks their activities. The RADIUS protocol is used for authorization.

DHCP Sever

The Dynamic Host Configuration Protocol (DHCP) server allocates IPv4 and IPv6 IP addresses to FARs. During tunnel provisioning, Cisco IoT FND sends requests to the DHCP server for IP addresses on behalf of FARs, compiles the received addresses as part of the tunnel provisioning configuration, and pushes the addresses to FARs through the TPS.



CHAPTER 2

Install

- System Requirements, on page 9
- Obtain Cisco IoT FND and Cisco Network Register Licenses, on page 12
- Install Linux Packages, on page 12
- Obtain Cisco IoT FND RPM Packages, on page 12
- Configure NTP Services, on page 13
- Use the Map View, on page 14
- Signature Verification, on page 14
- Certificates, on page 17
- Generate and export certificates, on page 19
- Install certificates on Cisco IoT FND and TPS proxy servers, on page 20
- Install Supporting Applications and Hardware, on page 26
- Setup Cisco IoT FND Database, on page 28

System Requirements

Table 1: Minimum Hardware and Software Requirements

Component	Minimum Hardware Requirements	Minimum Software Requirements

Cisco IoT FND application server (or comparable system that meets the hardware and software requirements)	• Processor:
	  • Intel Xeon x5680 2.27 GHz (64-bit)
	  • 4 CPUs
	• RAM: 16 GB
	• Disk space: 100 GB
		• Red Hat Enterprise Linux (RHEL) 7.5 and above, 64-bit with all packages installed (software development and web server)
		• Internet connection
		When you access Cisco IoT FND from a client browser, the browser connects to the Internet to download the necessary data files from the GIS maps provider.
		• A license to use SafeNet for mesh endpoint security
		Note
		Cisco IoT FND software bundle includes required Java version.

Component	Minimum Hardware Requirements	Minimum Software Requirements

Cisco IoT FND TPS Proxy	• Processor:
	  • Intel Xeon x5680 2.27 GHz (64-bit)
	  • 2 CPUs (virtual cores)
	• RAM: 4 GB
	• Hard Disk space: 100 GB
		• RHEL 7.5 and later with all packages installed (software development and web server)
		• Internet connection
		Note
		Cisco IoT FND software bundle includes required Java version.

Database server for Cisco IoT FND Scalable to 5,000/5,000,000 endpoints with minimum hardware requirements.	• Processor: Intel Xeon x5680 3.33 GHz (64-bit) • 4 CPUs • RAM: 32 GB • Disk space: 150 GB	Install both Linux and Oracle software on the database server. Install Linux packages on the database server before you install Oracle: • RHEL 7.5 and above, 64-bit with all packages installed (software development and web server) **Note** RHEL 8.5 is supported in Cisco IoT FND 4.8.1 release. • Oracle Database 19c Enterprise Edition for Cisco IoT FND Release 4.6.1 and later releases. • Oracle Database 18c Enterprise Edition (formerly identified as 12.2c) for Cisco IoT FND Release 4.5.x., 4.6.x and later releases. • Oracle Database 12c2 Enterprise Edition Release for IoT FND 4.3.x, 4.4.x, and 4.5.x. • Oracle database 12cR2 Enterprise Edition Release for Cisco IoT FND 4.6.x
Hardware Security Module (HSM)	Luna SA appliance, with client software installed on the Cisco IoT FND application servers	Luna SA appliance: • Release 6.10.2 firmware **Note** Contact SafeNet to determine if you can run a higher version. • Release 5.4.7-1 software, plus security patches. Luna SA client software: • Release 5.4.7-1 software.

Component	Minimum Hardware Requirements	Minimum Software Requirements
Software Security Module (SSM)	• RAM: 8 GB • Processor: 2 GHz • 2 CPUs	• RHEL 7.5, 64-bit with all packages installed (software development and web server).

Table 2: Hardware Requirements For Oracle Database Server

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
25/10,000	2	16	100
50/50,000	4	16	200
500/500,000	8	32	500
1,000/1,000,000	12	48	1000
2,000/2,000,000	16	64	1000
5,000/5,000,000	20	96	1000
6,000/6,000,000	20	96	1000

Table 3: Hardware Requirements For Tunnel Provisioning Server (TPS)

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
25/10,000	2	4	50
50/50,000	2	4	100
500/500,000	2	4	100
1,000/1,000,000	2	4	100
2,000/2,000,000	2	4	100
5,000/5,000,000	2	4	100
6,000/6,000,000	2	4	100

Table 4: Hardware Requirements For Application Server

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
25/10,000	2	16	100
50/50,000	4	16	200
500/500,000	4	16	250
1,000/1,000,000	8	16	250

Nodes (Routers/Endpoints)	CPU (virtual cores)	Memory (RAM GB)	Disk Space (GB)
2,000/2,000,000[1]	8	16	500
5,000/5,000,000 [1]	8	32	500
6,000/6,000,000 [1]	8	32	500

Obtain Cisco IoT FND and Cisco Network Register Licenses

- Contact your Cisco partner to obtain the necessary licenses to use Cisco IoT FND and Cisco Network Register (CNR).
- Obtain a license to use SafeNet as your Hardware Security Module (HSM) for mesh endpoint security.

Install Linux Packages

If you're installing a new Cisco IoT FND that requires an Oracle database, install the following Linux packages on your Cisco IoT FND server, before you install the Oracle database. Follow the mentioned order:

1. libaio-devel-0.3.106-5.i386.rpm
2. libaio-devel-0.3.106-5.x86_64.rpm
3. sysstat-7.0.2-11.el5.x86_64.rpm
4. unixODBC-libs-2.2.11-10.el5.i386.rpm
5. unixODBC-libs-2.2.11-10.el5.x86_64.rpm
6. unixODBC-2.2.11-10.el5.i386.rpm
7. unixODBC-2.2.11-10.el5.x86_64.rpm
8. unixODBC-devel-2.2.11-10.el5.i386.rpm
9. unixODBC-devel-2.2.11-10.el5.x86_64.rpm

Obtain Cisco IoT FND RPM Packages

Install the following RPM packages on the Cisco IoT FND servers before setting up your Cisco IoT FND:

RPM Package	Description
cgms-*version_buildnumber* .x86_64.rpm	Contains the Cisco IoT FND installer. This is the main RPM that contains the Cisco IoT FND application server itself. Install this package on the Cisco IoT FND application servers.

RPM Package	Description
cgms-oracle-*version_number* .x86_64.rpm	Contains the scripts and tools to create the Cisco IoT FND Oracle database. This package contains the Oracle database template and management scripts. Install this package on the Cisco IoT FND database server system.
cgms-tools-*version_number* .x86_64.rpm	Contains a few optional command-line tools. If needed, install this package on the system running the Cisco IoT FND application server.
cgms-ssm-*version_number* .x86_64.rpm	Contains the Software Security Module (SSM). Install this package on the system running the Cisco IoT FND application server.
cgms-tpsproxy-*version_number* .x86_64.rpm	Contains the TPS proxy application. Install this package on the Cisco IoT FND TPS proxy system.

Configure NTP Services

Before you Begin

Configure all RHEL servers (including all servers that run Cisco IoT FND) in your Cisco IoT FND to enable NTP service. The configuration ensures that Cisco IoT FND and the rest of the RHEL servers run the same time servers.

Note Before you generate certificates, synchronize the clocks of all the components associated with Cisco IoT FND.

1. Configure the /etc/ntp.conf file.

   Here's an example:

   ```
   cat /etc/ntp.conf
   ...
   # Use the same NTP servers on all our Connected Grid systems.
   server 0.ntp.example.com
   server 1.ntp.example.com
   server 2.ntp.example.com
   ...
   ```

2. Restart the NTP daemon and ensure that it is set to run during the boot.

   Here's an example:

   ```
   service chrony restart
   chkconfig chrony on
   ```

3. Check the status of the NTP daemon to view the changes in the configuration.

   Here's an example that shows the system at 192.0.2.1 is configured to be a local NTP server. This server synchronizes its time using the NTP server at 10.0.0.0.

   ```
   # ntpq -p
        remote           refid      st t when poll reach   delay   offset  jitter
   ==============================================================================
   *192.0.2.1      198.51.100.1     3 u   309 1024  377    0.694    0.899  0.435
    LOCAL(0)        .LOCL.          10 l    36   64  377    0.000    0.000   0.001
   ```

Use the Map View

When you're done installing Cisco IoT FND, perform the following instructions to use the map view:

1. From the Cisco IoT FND menubar, choose **Devices** > **Field Devices /Hear-end Routers/Servers/Assets**.
2. In the **Browse Devices** tab, navigate to the **Map** pane.
3. Cisco IoT FND displays the map of the device locations using the Geographic Information System (GIS) to display the device locations.



Note
- Ensure to configure the firewall to enable the map view access for all Cisco IoT FND operator systems to Cisco-provided GIS map file servers.
- Note: Only the Cisco IoT FND operator system has access to the GIS map file servers.

4. Assign a fully qualified domain name (FQDN) for each of the Cisco IoT server installations and send an email to ask-fnd-pm-external@cisco.com with the following information:
   - The number of Cisco IoT FND installation environments (test and production)
   - The FQDN of the Cisco IoT FND server
   - For cluster deployments, the FQDN of any load balancer in the deployment



Note The FQDN is only used in provisioning and authorizing the licensed Cisco IoT FND installation and make API calls to Enterprise Google Maps to download the map files. No utility operational data or asset information is ever used (that is, sent over Internet) to retrieve Google maps files. The Map files are retrieved only using geographic location information.

Signature Verification

Verify the signature

Here's the instructions for running a signature verification program:

Before you begin

- Python
- OpenSSL
- Internet connectivity for downloading the root and sub-CA certificates

Procedure

Step 1 Unzip and navigate to the directory:

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

Step 2 Extract the public key from the public certificate:

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

Expected result: FND-EE-cert.pubkey is created in the same folder.

Step 3 Verify the verification scripts using the public key and signature files:

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

Expected result: Verified OK

Step 4 Verify the ISO file:

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

Expected results:

- Successfully retrieves and verifies `crcam2.cer` from Cisco.
- Successfully retrieves and verifies `innerspace.cer` from Cisco.
- Successfully verifies root, sub-CA, and end-entity certificate chain.
- Successfully verifies the signature of the ISO image.

Step 5 Install the ISO image file:

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

Step 6 Verify if the delivered binary and ASCII keys have matching fingerprints:

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

Expected result:

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

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

Expected result:

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

Step 7 Verify the binary GPG key against the end-entry cerfificate:

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

Expected result:

- Successfully downloads and verifies `crcam2.cer` and `innerspace.cer` from Cisco.
- Successfully verifies Cisco root, sub-CA, and end-entity certificate chain.
- Successfully authenticates `FND-rel-binary.gpg` key using Cisco X.509 certificate trust chain.

Step 8 Verify the RPM signature using the GPG ASCII key:

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

Expected result:

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

Step 9 Repeat Step 8 for all the RPMs.

What to do next

Once you are done verifying, install or upgrade the RPM.

Cisco Provided ISO and RPM Images

Starting from Cisco IoT FND Release 4.9.0 and later releases, you can verify ISO and RPM images before proceeding with the installation or upgrade of Cisco IoT FND. This verification process ensures the authenticity and integrity of the images provided by Cisco, giving you confidence that the images have not been tampered with.

Zip File Contents	Description
`iot-fnd-<release>-<build number>.iso`	Cisco provided image for which the signature is to be verified.
`iot-fnd-<release>-<build number>.iso.signature`	Signature generated for the image.
`FND_RPM_SIGN-CCO_RELEASE.pem`	Cisco-signed x.509 end-entity certificate containing the public key used to verify the signature.
`cisco_x509_verify_release.py`	Signature verification program. This script verifies the 3-tier x.509 certificate chain and the signature using the certificate chain from Cisco.
`cisco_x509_verify_release.py.signature`	Signature generated for the script `cisco_x509_verify_release.py`.
`cisco_openpgp_verify_release.py`	Signature verification program for verifying the OpenPGP compliant public key against the x.509 end-entity certificate.
`cisco_openpgp_verify_release.py.signature`	Signature generated for the script `cisco_openpgp_verify_release.py`.
`FND-rel-binary.gpg`	OpenPGP public key used for verification of the signed RPM.

Zip File Contents	Description
`FND-rel-ascii.gpg`	OpenPGP public key used for verification of the signed RPM.

Certificates

Cisco IoT FND certificates

Digital certificates are used by Cisco IoT FND to ensure secure communication and device authentication within the network.

- Certificates validate the identities of devices and users.
- They help maintain the integrity and confidentiality of data transmitted across the Cisco IoT infrastructure.
- Certificates ensure that only authorized devices and users can access network resources.
- Cisco IoT FND enhances the security of IoT deployments by using digital certificates to protect against unauthorized access and potential cyber threats.

Types of certificates

- Device certificates are digital certificates used to authenticate and secure devices within a network, serving as electronic credentials that verify device identity.
- Web certificates (SSL/TLS certificates) authenticate the identity of a website and enable secure, encrypted communication between web servers and browsers.
- CSMP certificates are used within Cisco Smart Licensing to ensure secure communication and authentication for managing Cisco software licenses.

There are three primary types of certificates relevant to network security and management.

- Device certificates: Authenticate and secure devices within a network.
- Web certificates: Authenticate websites and enable encrypted communication.
- CSMP certificates: Used for secure communication in Cisco Smart Licensing.

For example, a router may use a device certificate to securely join a network, a website uses a web certificate to enable HTTPS, and Cisco Smart Licensing uses CSMP certificates to validate license management communications.

Role of certificates

Certificates are digital credentials that enable mutual authentication between Cisco routers and the Cisco Connected IoT FND, ensuring secure communication.

- Both the device and Cisco IoT FND must possess certificates signed by the same Certificate Authority (CA).
- Mutual authentication is required for all communications between supported Cisco routers and the Cisco Connected IoT FND.
- Certificates can be signed by either a root CA or a subordinate CA (sub-CA).

Generating and managing certificates involves creating certificates for both the devices and the Cisco IoT FND, including the TPS Proxy component.

- Certificates must be imported into the keystore on both the TPS proxy and Cisco IoT FND after generation.

For more information on generating certificates for CGRs, refer to the Certificate Enrollment Guide for the Cisco 1000 Series Connected Grid Routers .

Keystore

The keystore is a secure storage repository that holds device certificates, private keys, aliases, and issuer information for specific systems such as Cisco IoT FND or TPS proxy.

- Stores device certificates and private keys.
- Contains aliases and issuer information.
- Used by systems like Cisco IoT FND and TPS proxy.

The following table lists the certificate types used by Cisco IoT FND and the keystore location of the certificates.

Table 5: Keystore Certificate Types and Locations

Cisco IoT FND Certificate Type	Keystore Location
Device Certificate	`cgms_keystore`
Web Certificate	`jbossas.keystore`
CSMP Certificate	HSM or SSM

The keystore must contain the following information for Cisco IoT FND to function:

- Cisco SUDI or the device manufacturing certificate.
- The root CA certificate of the issuing root CA or sub-CA server.
- The certificate that is generated for the Cisco IoT FND server or TPS server for TPS (private key for the system).

For example, the keystore for Cisco IoT FND may contain a device certificate stored in `cgms_keystore` , a web certificate in `jbossas.keystore` , and a CSMP certificate in HSM or SSM.

Generate and export certificates

Create certificate template using Microsoft CA

Here's the instructions to create a certificate template:

Before you begin

Create certificate templates using the Microsoft Certificate Authority (CA) to generate certificates for Cisco IoT FND and TPS proxy. For more information about certificate templates see, certificate templates .

Procedure

Step 1 Open Microsoft CA application.

Ensure that you are on a system operating with the enterprise edition of Microsoft CA.

Step 2 Follow the instructions provided in the Create a Certificate Template: Enterprise CAs to create a certificate template.

Enable a certificate template

Before you begin

You must create and configure a certificate template.

Procedure

Step 1 Open the CA application using the Windows server.

Step 2 Follow the instructions provided in Configure the Certificate Template to enable a certificate template.

Generate a certificate for TPS Proxy

Before you begin

You must create and enable a certificate template using Microsoft CA.

Procedure

Step 1 Log in to the target machine with administrator rights.

Step 2 Follow the instruction provided in the Request a Certificate Using a Request File to generate a certificate.

Export the certificates

Follow these instructions to export certificates from Microsoft CA or subordinate CA to Cisco IoT FND.

Procedure

Step 1 Open the certificate management console for the local computer by running the following command. Some methods of running this command are from the *Start menu* , the Windows *Run* prompt, PowerShell prompt, or command prompt.

Step 2 Follow the instructions provided in Export a Certificate .

Install certificates on Cisco IoT FND and TPS proxy servers

Create a **cgms_keystore** file on both the servers running Cisco IoT FND and Cisco IoT FND TPS proxy using tools like Keytool.

Here's a quick overview of how the cgms_keystore file is copied to Cisco IoT FND and Cisco IoT FND TPS proxy servers:

- On Cisco IoT FND server: When you create the cgms_keystore file, you import the Cisco IoT FND certificate, the private key, and the certificate chain. Once you create the cgms_keystore file, copy the file into a specific directory on the Cisco IoT FND server.
- Cisco IoT FND TPS proxy server: When you create the cgms_keystore file, you import the Cisco IoT FND TPS proxy certificate, the private key, and the certificate chain. Once you create the cgms_keystore file, copy the file into a specific directory on the Cisco IoT FND proxy server.

Procedure

Step 1 Create the cgms_keystore file using Keytool on both Cisco IoT FND and Cisco IoT FND TPS proxy servers.

Step 2 Copy cgms_keystore files to Cisco IoT FND.

Step 3 Extract cgms.pem and cgms.key files from .pfx file.

Step 4 Copy the cgms.pem and cgms.key files.

Step 5 Import the CA certificates to the cgms_keystore.

Step 6 Import the SUDI certificate.

Step 7 Configure Cisco IoT FND server to access the keystore.

Step 8 Configure the TPS proxy to access the cgms_keystore.

Create the cgms_keystore file using Keytool

Here are the instructions to create cgms_keystore files using Keytool:

Procedure

Step 1 Log in to the root directory of Cisco IoT FND using the SSH command.

Step 2 Enter the following command to view the content of the .pfx file:

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

Step 3 Enter the keystore password. The keystore password is the password created while creating the .pfx file.

Step 4 Here's an example of the output:

```
# keytool -list -v -keystore nms_cert.pfx -srcstoretype pkcs12
Enter keystore password: keystore
Keystore type: PKCS12
Keystone provider: SunJSSE
Your keystore contains 1 entry

   Alias name: le-cgnms-75edd1e3-7e65-41b4-97f1-a913ebf21c8b

Creation date: March 29.2018
Entry type: PrivateKey Entry
Certificate chain length: 3
Certificate[1]:
Owner: CN=nms.sgbu.cisco.com
Issuer: CN=cisco-RSA-SUBCA-CA, DC=cisco, DC=com
...
```

Note
Make a note of the **Alias name** that you require while importing the certificates to the cgms_keystore file.

Step 5 Enter the following command to import the certificates to the cgms_keystore file:

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

Step 6 Storing cgms_keystore indicates successful completion.

What to do next

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

Copy the cgms_keystore file to Cisco IoT FND

Procedure

Step 1 For Cisco IoT FND server, copy the cgms_keystore file that you just created to the following directory:

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

Step 2 For the Cisco IoT FND TPS proxy server, copy the cgms_keystore file that you just created to the following directory:

```
/opt/cgms-tpsproxy/conf/
```

Extract the cgms.pem and cgms.key files

Here are the instructions to extract the cgms.pem and cgms.key files from the .pfx file:

Procedure

Step 1 Enter the following command to extract the certificates into a file named as cgms.pem:

```
openssl pkcs12 -in cgms.pfx -nokeys -out cgms.pem
```

Step 2 Enter the following command to extract the certificates into a file named as cgms.key:

```
openssl pkcs12 -in cgms.pfx -nocerts -out cgms.key -nodes
```

Copy the cgms.pem and cgms.key files

Procedure

Copy the cgms.pem and cgms.key files to the following directory using SCP or SFTP:

Example:

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

Import the CA certificate

Here are the instructions to import the CA certificates to the cgms_keystore:

Procedure

Step 1 Log in to the root directory of Cisco IoT FND using the SSH command.

Step 2 Change directory to the folder in which you have the cgms_key file

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

Step 3 Import the CA certificate using the following command:

```
# keytool -import -trustcacerts -alias
                        root -keystore cgms_keystore -file ca_cert.cer
```

Step 4 Enter the keystore password when asked.

Step 5 Enter **yes** when prompted to trust the certificate.

Step 6 The certificate is added to the cgms_keystore.

Here's an example of the output:

```
# keytool -import -trustcacerts -alias root -keystore
                        cgms_keystore -file ca_cert.cer
                        Enter keystore password: keystore_password
                        Owner: CN=SGBUNMSCA-WIN-4BGS4M94L66-CA,DC=SGBUNMSCA,DC=lab,DC=co
                        Issuer: CN=SGBUNMSCA-WIN-4BGS4M94L66-CA,DC=SGBUNMSCA,DC=lab,DC=co
                        Serial number:50adbd57e6b136984f9c1512a0eb7174
                        Valid from: Wed Jan 11 10:58:09 PDT 2018 until: Wed Jan 11:08:59 PDT 2018
                        Certificate _fingerprints:
                        MD5: AE:5D:F4:0A:2B:E5:C8:D8:4A:F4:18:56:FD:A7:8D:7D
                        SHA1: 83:22:12:8C:6A:23:D3:08:2B:00:55:EF:BD:FF:BA:47:97:99:7E:41
                        Signature algorithm name: SHA1withRSA
                        Version:3
                        Extensions:
                        #1: ObjectId: 2.5.29.19 Criticality=true
                        BasicConstraints:[
                        CA:true
                        PathLen:2147483647
                        ]
                        #2:ObjectId: 2.5.29.15 Criticality=false
                        KeyUsage[
                        DigitalSignature
                        Key_CertSign
                        Crl_Sign
                        ]
                        #3:ObjectId: 2.5.29.14 Criticality=false
                        SubjectKeyIdentifier [
                        KeyIdentifier [
                        0000:C6 AB 38 CC EE 79 B0 51 3B 4D 13 c8 9A 56 F6 73..8..y.Q;M...V.s
                        0010:B9 19 FF 7B....
                        ]
                        ]
                        #4: ObjectId:1.3.6.1.4.1.311.21.1 Criticality=false
                        Trust this certificate [no] yes
                        Certificate was added to the keystore.
```

Note
Follow the same steps to import the CA certificate into cgms_keystore on the TPS proxy.

Import the SUDI certificate

A Secure Unique Device Identifier (SUDI) certificate is a cryptographic certificate that contains a unique identifier for the device, along with a public key and other relevant information. It is typically issued by a trusted Certificate Authority (CA) and is an essential component of secure device authentication and zero-touch provisioning.

Here are the instructions to import the SUDI certificate into the cgms_keystore:

Procedure

Step 1 Log in to the root directory of Cisco IoT FND using the SSH command.

Step 2 Change the directory to the folder where the cgms_keystore file is present:

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

Step 3 Enter the following command to import the SUDI certificate:

```
#keytool -import -trustcacerts -alias sudi -keystore cgms_keystore -file ciscosudi/cisco-sudi-ca.pem

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

Note
Import both the SUDI certificates with 2029 and 2099 expiry periods.

Step 4 Enter the keystore password when asked to.

Step 5 The certificate is added to the keystore.

Note
For the TPS proxy, copy the cgms_keystore file to this directory:

/opt/cgms-tpsproxy/conf/ciscosudi

Configure Cisco IoT FND to access the keystore

Here are the steps to configure Cisco IoT FND server to access the keystore:

Before you begin

Once you are done creating the cgms_keystore and importing the NMS and CA certificates, configure the Cisco IoT FND server to access the cgms_keystore file.

Procedure

Step 1 Stop the Cisco IoT FND.

Step 2 Run the following setup cgms.sh script:

```
pwd
                        /opt/cgms/bin ./setupCgms.sh
```

```
                        06-12-2024 10:21:39 PDT: INFO: ========== CG-NMS Setup Started -
2018-06-12-10-21-39 =====
                        06-12-2024 10:21:39 PDT: INFO: Log file:
/opt/cgms/bin/../server/cgms/log/cgms_setup.log
                        Are you sure you want to setup CG-NMS (y/n)? y
                        06-12-2024 10:21:39 PDT: INFO: User response: y
                        ...
                        Do you want to change the keystore password (y/n)? y
                        06-12-2024 10:21:52 PDT: INFO: User response: y
                        Enter keystore password: keystore_password
                        Re-enter keystore password: keystore_password
                        06-12-2024 10:21:59 PDT: INFO: Configuring keystore password. This may take
 a while.
                        Please wait...
                        06-12-2024 10:22:00 PDT: INFO: Keystore password configured.
```

The cgms.sh script saves the password set in the cgms.properties file.

Step 3 Restart Cisco IoT FND.

Note
Set the permissions to root read only and protect the cgms_keystore and cgms.properties files.

Configure the TPS Proxy to access the keystore

Here are the instructions to configure the TPS proxy to access the `cgms_keystore`:

Procedure

Step 1 Change the directory to the tpsproxy bin directory:

```
cd /opt/cgms-tpsproxy/bin
```

Step 2 Encrypt your password:

```
[root@fndnms bin]#./encryption_util.sh
                        Usage: EncryptionUtil <encrypt|decrypt> <data>
                        [root@fndnms bin]#./encryption_util.sh encrypt test
                        Kq+bA+oLBlo=
```

Step 3 Copy the encrypted password into the tpsproxy.properties file:

a) Open the file for editing:

```
cd /opt/cgms-tpsproxy/conf
                                    emacs tpsproxy.properties
```

b) Add the following line to the file:

```
cgms-keystore-password-hidden=keystore_password
```

Step 4 Restart the TPS proxy using the appropriate command for your RHEL version:

Example:

```
systemctl restart tpsproxy
service tpsproxy restart
```

Install Supporting Applications and Hardware

This section aims to provide guidance on installing supporting applications and database hardware servers to Cisco IoT FND. You'll find more specific information on installing the Redhat Packaging Manager (RPM) for large-scale Advanced Metering Infrastructure (AMI) use cases. The AMI deployments allow for an a la carte deployment of each head-end component with dedicated hardware. With AMI deployments, you're given the flexibility to horizontally scale Cisco IoT FND in supporting upto 11 million endpoints.

Install Cisco IoT FND For the First Time

Install and Setup Cisco IoT FND Database Overview

Perform the following tasks to install and setup Cisco IoT FND Database:

Single Server Database Deployment

Here's an overview to install and setup Cisco IoT FND database for a single-server database deployment:

1. Log in to the database server.
2. Download and unpack Oracle database.
3. Run the Oracle database installer.
4. Setup the Cisco IoT FND database.

High Availability Database Server Deployment

Here's an overview of installing and setting up Cisco IoT FND database for High Availablity (HA):

1. Log in to the primary Cisco IoT FND database server.
2. Download and unpack the Oracle database.
3. Run the Oracle database installer.
4. Log in to the standby database server.
5. Download and unpack the Oracle database.
6. Run the Oracle database installer.

Download and Unpack Oracle Database

Here are the instructions to download and unpack Oracle Database:

Procedure

Step 1 Log in to the root directory of Cisco IoT FND using the SSH command.

Step 2 Download the Oracle database software from the official Oracle website.

Step 3 Run the following command to avoid display-related errors while installing the Oracle database software:

```
# xhost + local:oracle
```

Step 4 Create the Oracle user and dba group using the following command:

```
# groupadd dba
# adduser -d /home/oracle -g dba -s /bin/bash oracle
```

Step 5 Run the following command to unpack the Oracle database zip archives:

```
p10404530_121020_Linux-x86-64_1of7.zip
p10404530_121020_Linux-x86-64_2of7.zip
p10404530_121020_Linux-x86-64_3of7.zip
p10404530_121020_Linux-x86-64_4of7.zip
p10404530_121020_Linux-x86-64_5of7.zip
p10404530_121020_Linux-x86-64_6of7.zip
p10404530_121020_Linux-x86-64_7of7.zip
```

You're done with unpacking the Oracle database.

Run the Oracle Database Installer

Here are the instructions to install the Oracle database:

Before you begin

Disable your firewall.

Procedure

Step 1 Log in to the root directory of Cisco IoT FND using the SSH command.

Step 2 Follow the instructions provided in the Running Oracle Database Setup Wizard to Install Oracle Database.

Apply Patches During Oracle Database Installation

Here are the instructions to apply patches during oracle database installation:

Procedure

Step 1 Stop Cisco IoT FND application.

Step 2 Step the Oracle database service.

Step 3 Follow the instructions provided in the Applying Patches During an Oracle Database Installation or Upgrade.

Setup Cisco IoT FND Database

Here's an overview of the steps involved in setting up Cisco IoT FND database:

Define Oracle Database Environment Variables

Before setting up the Cisco IoT FND database you should perform the following steps and define the oracle database environment variables:

1. Execute the following command on your command terminal, switch to the **oracle** user account:

   ```
   su - oracle
   ```

2. Enter the password for the **oracle** user account. Here's an example:

   ```
   $ su - oracle
   Password: [enter oracle user password]
   oracle@hostname:~$
   ```

3. In the **oracle** user account, define the following environment variables:

Variable	Description
ORACLE_BASE	Defines the path to the Oracle root directory on your system. For example: `$ export ORACLE_BASE=/home/oracle/app/oracle` If this variable is not set, the IoT FND setup script displays an error.
ORACLE_HOME	Defines the path to the Oracle home of the IoT FND database. For example: `$ export ORACLE_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome_1`
PATH	Defines the path to the Oracle binaries. For example: `$ export PATH=$PATH:$ORACLE_HOME/bin`
LD_LIBRARY_PATH	Defines the path to the libraries. For example: `$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH`

Variable	Description

ORACLE_SID	Defines the Oracle System ID (SID).
	If you are only using one database server or installing an HA deployment, set this variable on the *primary* database server to **cgms** :
	`$ export ORACLE_SID=cgms`
	If deploying a standby database server, set this variable on the *standby* database server to **cgms_s** :
	`$ export ORACLE_SID=cgms_s`
	If this variable is not set, the Cisoc IoT FND setup script displays an error.

4. You can define these variables manually, here's an example:

On a Single or Primary Database Server	On a Standby Database Server

`$ su - oracle`
`$ export ORACLE_BASE=/home/oracle/app/oracle`
`$ export ORACLE_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome_1`
`$ export PATH=$PATH:$ORACLE_HOME/bin`
`$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH`
`$ export ORACLE_SID=cgms`
	`$ su - oracle`
	`$ export ORACLE_BASE=/home/oracle/app/oracle`
	`$ export ORACLE_HOME=/home/oracle/app/oracle/product/12.1.0/dbhome_1`
	`$ export PATH=$PATH:$ORACLE_HOME/bin`
	`$ export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH`
	`$ export ORACLE_SID=cgms_s`

Install Cisco IoT FND Oracle Database Scripts

Here are the instructions to install the Oracle scripts on the Oracle server:

Procedure

Step 1 Login to the root directory of Cisco IoT FND using the SSH command.

Step 2 Securely copy the Cisco IoT FND Oracle script RPM on your Oracle server. Here's an example:

```
$ scp cgms-oracle-version_number
.x86_64.rpm root@oracle-machine:~
$ rpm –ivh cgms-oracle-version_number
.x86_64.rpm
```

Step 3 Run the following script from the path /opt/cgms-oracle/scripts/

```
installCgmsOracleScripts.sh
```

You're done installing Cisco IoT FND Oracle Database scripts.

Create Cisco IoT FND Oracle Database

Here are the instructions to create a Cisco IoT FND Oracle database:

1. Using the following command on your command terminal, switch to the **oracle** user account:

   ```
   su - oracle
   ```

2. Enter the password for the **oracle** user account. Here's an example:

   ```
   $ su - oracle
   Password: [enter oracle user password]
   oracle@hostname:~$
   ```

3. Run the following script:

   ```
   setupCgmsDb.sh
   ```

4. The script creates a user **cgms_dev** that is used by Cisco IoT FND to access the Oracle database. The default password for this user account is **cgms123**. The default password for sys DBA account is **cgmsDba123**.

   Here's an example of the script:

   ```
   $ su - oracle
   $ export DISPLAY=localhost:0
   $ cd $ORACLE_BASE/cgms/scripts
   $ ./setupCgmsDb.sh
   09-13-2018 10:38:07 PDT: INFO: ======== CGMS Database Setup Started ==========
   09-13-2018 10:38:07 PDT: INFO: Log file: /tmp/cgmsdb_setup.log

   Are you sure you want to setup CG-NMS database (y/n)? y

   09-13-2018 10:38:08 PDT: INFO: User response: y
   09-13-2018 10:38:08 PDT: INFO: CGMS database does not exist.
   Enter new password for SYS DBA:
   Re-enter new password for SYS DBA:
   09-13-2018 10:38:14 PDT: INFO: User entered SYS DBA password.

   Enter new password for CG-NMS database:
   Re-enter new password CG-NMS database:
   09-13-2018 10:38:18 PDT: INFO: User entered CG-NMS DB password.
   09-13-2018 10:38:18 PDT: INFO: Stopping listener...
   09-13-2018 10:38:18 PDT: INFO: Listener already stopped.
   09-13-2018 10:38:18 PDT: INFO: Deleting database files...
   09-13-2018 10:38:18 PDT: INFO: Creating listener...
   09-13-2018 10:38:19 PDT: INFO: Listener creation completed successfully.
   09-13-2018 10:38:19 PDT: INFO: Configuring listener...
   09-13-2018 10:38:19 PDT: INFO: Listener successfully configured.
   09-13-2018 10:38:19 PDT: INFO: Creating database. This may take a while. Please be
   patient...
   09-13-2018 10:42:55 PDT: INFO: Database creation completed successfully.
   09-13-2018 10:42:55 PDT: INFO: Updating /etc/oratab...
   09-13-2018 10:42:55 PDT: INFO: /etc/oratab updated.
   09-13-2018 10:42:55 PDT: INFO: Configuring database...
   09-13-2018 10:42:56 PDT: INFO: Starting listener...
   09-13-2018 10:42:56 PDT: INFO: Listener start completed successfully.
   09-13-2018 10:42:56 PDT: INFO: Starting database configuration...
   09-13-2018 10:43:17 PDT: INFO: Database configuration completed successfully.
   09-13-2018 10:43:17 PDT: INFO: Starting Oracle...
   09-13-2018 10:43:17 PDT: INFO: Starting Oracle in mount state...
   ORACLE instance started.

   Total System Global Area 1.6836E+10 bytes
   ```

```
Fixed Size 2220032 bytes
Variable Size 8589934592 bytes
Database Buffers 8187281408 bytes
Redo Buffers 56487936 bytes
Database mounted.
09-13-2018 10:43:26 PDT: INFO: Opening database for read/write...

Database altered.

09-13-2018 10:43:29
PDT: INFO: ========== CGMS Database Setup Completed Successfully ==========
```

Note

- We strongly recommend that you change all default passwords. Do not use special characters such as, @, #, !, or + when using the encryption_util.sh script. The script cannot encrypt special characters.
- This script might run for several minutes. To check the setup progress, run the command:*$ tail -f /tmp/cgmsdb_setup.log*

Start the Cisco IoT FND Oracle Database

Here are the instructions to start a Cisco IoT FND Oracle database for the first time:

Procedure

Step 1 Using the following command on your command terminal, switch to the **oracle** user account:

```
su - oracle
```

Step 2 Enter the password for the **oracle** user account. Here's an example:

```
$ su - oracle
Password: [enter oracle user password]
oracle@hostname:~$
```

Step 3 Run the following script:

```
$ cd $ORACLE_BASE/cgms/scripts
$ ./startOracle.sh
```

Step 4 Configure a cron job that starts the Cisco IoT FND database during the bootup by running the following script:

```
./installOracleJob.sh
```

Upgrade Cisco IoT FND Database

Here are the instructions to upgrade the Cisco IoT FND database:

Before you begin

Ensure to take a full backup of your database before performing any upgrade or significant changes.

Procedure

Step 1 Connect to your Oracle database as a DBA user using SQL*Plus.

```
sqlplus sys/password@cgms as sysdba
```

Step 2 Execture the following command to add the data files. You should run this command multiple times, once for each file from 2 to 15.

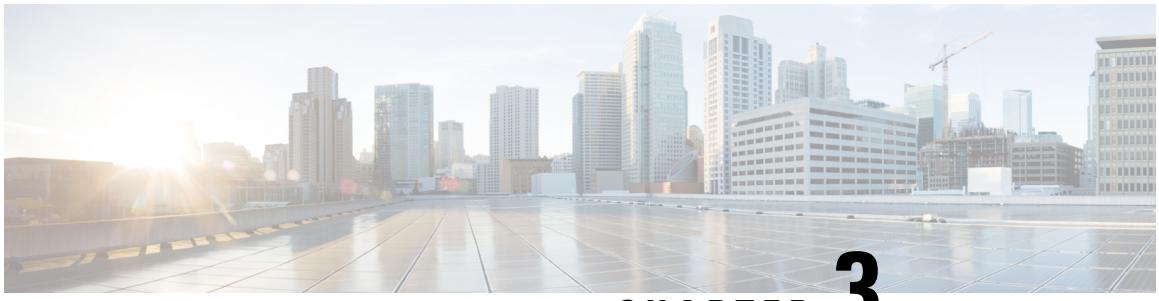
```
ALTER TABLESPACE USERS ADD DATAFILE
'&oracle_base/oradata/&sid_caps/users02.dbf' SIZE 5M AUTOEXTEND ON;
ALTER TABLESPACE USERS ADD DATAFILE
'&oracle_base/oradata/&sid_caps/users03.dbf' SIZE 5M AUTOEXTEND ON;
...
ALTER TABLESPACE USERS ADD DATAFILE
'&oracle_base/oradata/&sid_caps/users15.dbf' SIZE 5M AUTOEXTEND ON;
```

Step 3 Execute the following command to enable block-change tracking, which is necessary for incremental backups:

```
ALTER DATABASE ENABLE BLOCK CHANGE TRACKING USING FILE
'&oracle_base/oradata/&sid_caps/rman_change_track.f' REUSE;
```

Step 4 Execute the following command to disable parallel execution and set parallel_max_servers parameters to 0:

```
ALTER SYSTEM SET parallel_max_servers = 0 SCOPE=BOTH;
```



CHAPTER 3

Deploy

- Deploy Cisco IoT FND Using Bare Metal Servers, on page 33
- Recommended Computing Resources, on page 34

Deploy Cisco IoT FND Using Bare Metal Servers

Deploying Cisco IoT FND involves setting up and configuring the software to manage and monitor Cisco IoT devices and networks effectively. You can install Cisco IoT FND on a Bare Metal server and integrate Cisco IoT FND with your existing Oracle database. Once you download the Cisco IoT FND software package from cisco.com, you can manually configure the connection to your database, install database certificates, and configure related items. Once you deploy Cisco IoT FND on a Bare Metal server, you can manage all the supported device types and devices.

Here's a table that describes the key elements of deploying Cisco IoT FND on a Bare Metal server:

Element	Requirement/Process
Accessing and managing the database	Requires an existing Oracle database.
Installing and configuring	Install Cisco IoT FND and manually configure the connection to your database, install database certificates, and configure related items.
Upgrading	Install the ISO package released during each Cisco IoT FND release, to upgrade Cisco IoT FND.
Hosting	The Cisco IoT FND application and the Oracle database run on a dedicated server.
Distributing the images	Cisco distributes an ISO file that contains the RPM installation files. Provided as a ZIP file.
Naming the image	Here's an example filename: iot-fnd-*release_number-build_ number*-signed.zip
Managing devices	You can manage all the Cisco IoT FND supported devices using Cisco IoT FND.

Element	Requirement/Process
Configuring High Availability (HA)	High Availability (HA) and load balancing are supported for the Cisco IoT FND application server and Cisco IoT FND database server. Load-balancing is configured using a third-party load-balancing device.
Managing IOx apps	Not supported

Installing Cisco IoT FND on a Bare Metal server is an Advanced Metering Infrastructure (AMI) deployment that supports up to 8,000 routers and 8,000,000 endpoints. High availability (HA) refers to a system or the ability of a component to remain operational and accessible for a high percentage of time, often measured as uptime. The AMI deployment provides High Availability (HA) for both the Cisco IoT FND application server and the Cisco IoT database server in single or clustered Cisco IoT FND server deployments.

On the Cisco IoT FND application server, you achieve HA by connecting multiple Cisco IoT FND servers to a load balancer. The traffic from routers and endpoints goes to the load balancer, which uses a load-balancing algorithm to distribute the load among the Cisco IoT FND servers.

On the Cisco IoT FND database server, you achieve HA by configuring two Cisco IoT FND database servers: a primary database server and a standby (or secondary) database server. When the primary database server receives new data, it sends a copy to the standby database server. The observer, a program that monitors the Cisco IoT FND database servers, runs on a separate server or the standby server. If the primary database server fails, the observer configures the standby database server as the new primary database server.

Recommended Computing Resources

Recommended Resources For Managing Mesh Endpoints

For improved device scalability and device performance, we recommend that you cluster application servers as follows:

- For deploying 2,000 routers/2,000,000 endpoints and 6,000 routers/6,000,000 endpoints: Two application servers
- For deploying more than 6, 000 routers and 6,000,000 endpoints: Four application servers

Table 6: Recommended Resources For Application Server

Nodes Deployed	CPU	Memory(RAM GB)	Disk Space(GB)
Up to 25 routers and 10,000 endpoints	2	16	100
Up to 50 routers and 50,000 endpoints	4	16	200
Up to 500 routers and 500,000 endpoints	4	16	250
Up to 1,000 routers and 1,000,000 endpoints	8	16	250
Up to 2,000 routers and 2,000,000 endpoints	8	16	500

Nodes Deployed	CPU	Memory(RAM GB)	Disk Space(GB)
Up to 6,000 routers and 6,000,000 endpoints	8	16	500
Up to 8,000 routers and 8,000,000 endpoints	8	32	500

Table 7: Recommended Resources For Database Server

Nodes Deployed	CPU	Memory(RAM GB)	Disk Space(GB)
Up to 25 routers and 10,000 endpoints	2	16	100
Up to 50 routers and 50,000 endpoints	4	16	200
Up to 500 routers and 500,000 endpoints	8	32	500
Up to 1,000 routers and 1,000,000 endpoints	12	48	1000
Up to 2,000 routers and 2,000,000 endpoints	16	64	1000
Up to 6,000 routers and 6,000,000 endpoints	20	96	1000
Up to 8,000 routers and 8,000,000 endpoints	32	160	2000

Table 8: Recommended Resources For Tunnel Provisioning Server

Nodes Deployed	CPU	Memory(RAM GB)	Disk Space(GB)
Up to 25 routers and 10,000 endpoints	2	4	50
Up to 50 routers and 50,000 endpoints	2	4	100
Up to 500 routers and 500,000 endpoints	2	4	100
Up to 1,000 routers and 1,000,000 endpoints	2	4	100
Up to 2,000 routers and 2,000,000 endpoints	2	4	100
Up to 6,000 routers and 6,000,000 endpoints	2	4	100
Up to 8,000 routers and 8,000,000 endpoints	2	4	100

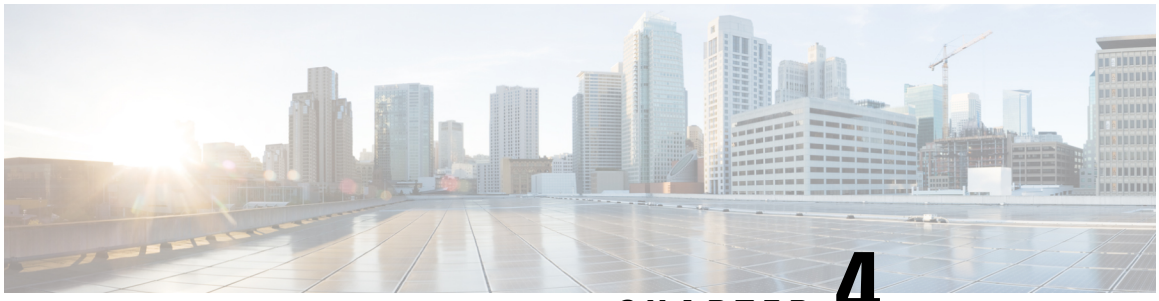
Recommended Resources For Managing Routers

Table 9: Recommended Resources For Application Server

Nodes Deployed	CPU(Virtual Cores)	Memory(RAM GB)	Disk Space(GB)
Up to 25,000 routers	24	32	500
Up to 10,000 routers	16	24	500

Table 10: Recommended Resources For Database Server

Nodes Deployed	CPU(Virtual Cores)	Memory(RAM GB)	Disk Space(GB)
Up to 25,000 routers	48	64	1000
Up to 10,000 routers	24	48	1000



CHAPTER 4

Upgrade

- Upgrade Cisco IoT FND, on page 37
- Upgrade Cisco IoT FND TPS Proxy, on page 38

Upgrade Cisco IoT FND

Here are the instructions to upgrade your Cisco IoT FND application:

Before you begin

- Perform a back up of the application directory. For example, if you want to upgrade **cgms RPM**, then you must back up the **/opt/cgms** folder.
- Perform a back up of the Cisco IoT FND database.
- The steps mentioned in this topic needs to be performed sequentially.

Procedure

Step 1 Obtain the latest verion of Cisco IoT FND ISO from Cisco.

Step 2 Extract **cgms rpms** into a directory from Cisco IoT FND release ISO file.

Step 3 Run the following command on your command terminal to get a list of rpms isntalled in the Cisco IoT FND application:

```
rpm -qa | grep cgms
```

Step 4 Run the following command to stop Cisco IoT FND.

```
systemctl stop cgms
```

Note
The application typically takes approximately 10 seconds to stop.

Step 5 Run the following command to verify if no Java processes are running.

```
ps | grep java
```

Step 6 Run the following command to make sure that the cgms service has stopped.

```
systemctl status cgms
```

Step 7 Run the following command to upgrade the Cisco IoT FND RPM:

```
rpm -Uvh <new_cgms_rpm_filename> --force
```

Here's an example:

```
rpm -Uvh cgms-4.11.0-46.x86_69.rpm --force
```

Note
We recommend you to upgrade all the installed **rpms** with the same Cisco IoT FND version that you will upgrade to. The new *rpm* files overwrite the existing files in **/opt/cgms**.

Step 8 Run the following command to upgrade the database:

```
./db-migrate in /opt/cgms/bin
```

Step 9 Enter the database password when prompted.

Note
The default password is **cgms123**.

Step 10 Run the following command to start the Cisco IoT FND application after an upgrade:

```
systemctl start cgms
```

What to do next

Upgrade Cisco IoT FND TPS Proxy

Here are the instructions to upgrade the TPS proxy server manually:

Before you begin

Based on the RHEL version, choose either systemctl or service commands.

RHEL Version	Command
8.x	`systemctl stop tpsproxy.service`
7.x	`service tpsproxy stop`

Procedure

Step 1 Perform a back up of the following files in the **/opt/cgms-tpsproxy/conf** directory.

- tpsproxy.properties
- cgms_keystore

Step 2 Using the following command, ensure that the following services are running:

```
systemctl status tpsproxy.service
```

Here's an example:

```
[root@iot-tps opt]# systemctl status tpsproxy.service
● tpsproxy.service - SYSV: CGMS Tunnel Provisioning proxy server
   Loaded: loaded (/etc/rc.d/init.d/tpsproxy; generated)
   Active: active (running) since Mon 2023-11-06 23:51:00 IST; 6 months 2 days ago
     Docs: man:systemd-sysv-generator(8)
  Process: 35730 ExecStart=/etc/rc.d/init.d/tpsproxy start (code=exited, status=0/SUCCESS)
    Tasks: 45 (limit: 152533)
   Memory: 204.1M
   CGroup: /system.slice/tpsproxy.service
           └─35748 java -server -Xms128m -Xmx2g -XX:MaxPermSize=256m -server
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/opt/cgms-tpsproxy/log -XX:-OmitStackTraceInFastThrow
 -XX:-Use>

Nov 06 23:50:58 iot-tps systemd[1]: Starting SYSV: CGMS Tunnel Provisioning proxy server...
Nov 06 23:50:59 iot-tps runuser[35746]: pam_unix(runuser:session): session opened for user root by
(uid=0)
Nov 06 23:50:59 iot-tps runuser[35746]: pam_unix(runuser:session): session closed for user root
Nov 06 23:51:00 iot-tps tpsproxy[35730]: [36B blob data]
Nov 06 23:51:00 iot-tps systemd[1]: Started SYSV: CGMS Tunnel Provisioning proxy server.
```

Step 3 Run the following command to stop the tpsproxy services:

```
systemctl stop tpsproxy
```

Step 4 Copy the cgms-tpsproxy.rpm downloaded from software.cisco.com to a directory in the TPS server.

Step 5 Run the following command to upgrade the tpsproxy server:

```
[root@iot-tps opt]# rpm -Uvh cgms-tpsproxy-<fnd-version>.rpm
Verifying...                          ################################# [100%]
Preparing...                          ################################# [100%]
Updating / installing...
   1:cgms-tpsproxy-4.12.0-56          ################################# [ 50%]
PSK key already present
05-09-2024 15:36:02 IST: INFO: Looking for /opt/cgms-tpsproxy/conf/tpsproxy.properties
05-09-2024 15:36:02 IST: INFO: cgms-keystore-password-hidden already exists. Doing nothing.
Cleaning up / removing...
   2:cgms-tpsproxy-4.10.0-45          ################################# [100%]
```

Step 6 Run the following command to start the tpsproxy services:

```
systemctl start tpsproxy.service
```

Here's an example:

```
[root@iot-tps opt]# systemctl start tpsproxy.service
[root@iot-tps opt]# systemctl status tpsproxy.service
● tpsproxy.service - SYSV: CGMS Tunnel Provisioning proxy server
   Loaded: loaded (/etc/rc.d/init.d/tpsproxy; generated)
   Active: active (running) since Thu 2024-05-09 15:36:45 IST; 4s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 209931 ExecStart=/etc/rc.d/init.d/tpsproxy start (code=exited, status=0/SUCCESS)
    Tasks: 39 (limit: 152533)
   Memory: 71.1M
   CGroup: /system.slice/tpsproxy.service
           └─209949 java -server -Xms128m -Xmx2g -XX:MaxPermSize=256m -server
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/opt/cgms-tpsproxy/log -XX:-OmitStackTraceInFastThrow
 -XX:-Us>

May 09 15:36:44 iot-tps systemd[1]: Starting SYSV: CGMS Tunnel Provisioning proxy server...
May 09 15:36:44 iot-tps runuser[209947]: pam_unix(runuser:session): session opened for user root by
 (uid=0)
```

```
May 09 15:36:44 iot-tps runuser[209947]: pam_unix(runuser:session): session closed for user root
May 09 15:36:45 iot-tps tpsproxy[209931]: [36B blob data]
May 09 15:36:45 iot-tps systemd[1]: Started SYSV: CGMS Tunnel Provisioning proxy server.
```

Step 7 Run the following command to check the log

```
tail -f /opt/cgms-tpsproxy/log/tpsproxy.log
```

Here's an example:

```
[root@iot-tps opt]# tail -f /opt/cgms-tpsproxy/log/tpsproxy.log
32: iot-tps: May 09 2024 06:06:45.108 -0400: %CGMS-6-UNSPECIFIED:
%[ch=BootstrapProxy][sev=INFO][tid=main]: bootstrap-proxy-listen-port=9125
33: iot-tps: May 09 2024 06:06:45.108 -0400: %CGMS-6-UNSPECIFIED:
%[ch=BootstrapProxy][sev=INFO][tid=main]: inbound-proxy-destination=http://<fnd hostname/ip
address>:9125
34: iot-tps: May 09 2024 06:06:45.108 -0400: %CGMS-6-UNSPECIFIED:
%[ch=BootstrapProxy][sev=INFO][tid=main]: inbound-proxy-timeout=300000
35: iot-tps: May 09 2024 06:06:45.108 -0400: %CGMS-6-UNSPECIFIED:
%[ch=BootstrapProxy][sev=INFO][tid=main]: inbound-proxy-idle-timeout=300000
36: iot-tps: May 09 2024 06:06:45.108 -0400: %CGMS-6-UNSPECIFIED:
%[ch=BootstrapProxy][sev=INFO][tid=main]: inbound-proxy-connections=30
37: iot-tps: May 09 2024 06:06:45.108 -0400: %CGMS-6-UNSPECIFIED:
%[ch=BootstrapProxy][sev=INFO][tid=main]: inbound-proxy-max-response-timeout=305000
38: iot-tps: May 09 2024 06:06:45.110 -0400: %CGMS-6-UNSPECIFIED: %[ch=Server][sev=INFO][tid=main]:
 jetty-9.2.19.v20160908
39: iot-tps: May 09 2024 06:06:45.111 -0400: %CGMS-6-UNSPECIFIED:
%[ch=ContextHandler][sev=INFO][tid=main]: Started
o.e.j.s.ServletContextHandler@4d41cee{/,null,AVAILABLE}
40: iot-tps: May 09 2024 06:06:45.112 -0400: %CGMS-6-UNSPECIFIED:
%[ch=ServerConnector][sev=INFO][tid=main]: Started ServerConnector@3712b94{HTTP/1.1}{0.0.0.0:9125}
41: iot-tps: May 09 2024 06:06:45.112 -0400: %CGMS-6-UNSPECIFIED: %[ch=Server][sev=INFO][tid=main]:
 Started @697ms
```

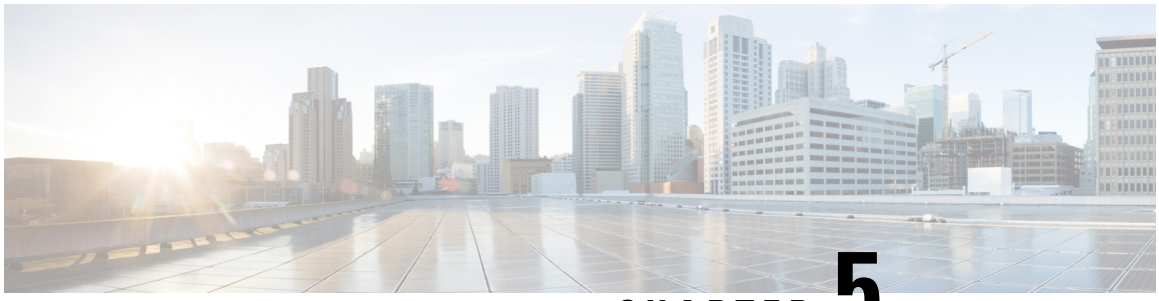
Step 8 Check the tpsproxy service status using the following command:

```
systemctl status tpsproxy.service
```

Here's an example:

```
[root@iot-tps opt]# systemctl status tpsproxy.service
● tpsproxy.service - SYSV: CGMS Tunnel Provisioning proxy server
   Loaded: loaded (/etc/rc.d/init.d/tpsproxy; generated)
   Active: active (running) since Thu 2024-05-09 15:36:45 IST; 1min 52s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 209931 ExecStart=/etc/rc.d/init.d/tpsproxy start (code=exited, status=0/SUCCESS)
    Tasks: 39 (limit: 152533)
   Memory: 71.1M
   CGroup: /system.slice/tpsproxy.service
           └─209949 java -server -Xms128m -Xmx2g -XX:MaxPermSize=256m -server
-XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/opt/cgms-tpsproxy/log -XX:-OmitStackTraceInFastThrow
 -XX:-Us>

May 09 15:36:44 iot-tps systemd[1]: Starting SYSV: CGMS Tunnel Provisioning proxy server...
May 09 15:36:44 iot-tps runuser[209947]: pam_unix(runuser:session): session opened for user root by
 (uid=0)
May 09 15:36:44 iot-tps runuser[209947]: pam_unix(runuser:session): session closed for user root
May 09 15:36:45 iot-tps tpsproxy[209931]: [36B blob data]
May 09 15:36:45 iot-tps systemd[1]: Started SYSV: CGMS Tunnel Provisioning proxy server.
```



CHAPTER 5

Troubleshoot

Table 11: Common Errors and Resolutions

Common Errors	Items to Check and/or Resolve Errors
Checkpoint Failed	Check the archive
CiscoIosFileUploadException: Full error: Error occurred while verifying file upload operation for net element CGR1120/K9+FOC21255MYX	Check provisioning URL (HTTP, HTTPS) Check WSMA with test script: user and port
org.apache.cxf.interceptor.Fault: Connection refused (Connection refused)	Check port used for HTTPS communication (varies by platform). For example: • FAR: ip http secure-port 8443 • IR1101: ip http secure-port 443

Common Errors	Items to Check and/or Resolve Errors
PnP Service Error 3341 Full error: Error while creating FND trustpoint on the device. errorCode: PnP Service Error 3341, errorMessage: SSL Server ID check failed after cert-install	Check SAN field in the FND certificate: • Certificate which FND offers for PNP: https://10.48.43.229:9120/pnp/HELLO • Trustpoint which FND offers for PNP: Click to view the truspoint. For additional information, click to view the document: Enter the keystore command to list SAN fields on the certificate in the keystore used for PNP. This verifies the accuracy of the SAN field(s). keytool -list -v -keystore cgms_keystore | grep SubjectAlt -A3 Enter keystore password: keystore SubjectAlternativeName [IPAddress: 10.48.43.229]

Common Errors	Items to Check and/or Resolve Errors
PnP Service Error 1702 Full error: Error while deploying odm/config file on the device. errorCode: PnP Service Error 1702, errorMessage: I/O error	If error is seen, enable debug in FND for bootstrapping, Ensure that FAR is able to reach TPS or FND using its hostname. For example, in the below debug logs for FND bootstrapping, FAR should be able to resolve and reach iot-tps.example.cisco.com on 9120 and viceversa. [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.4/16]: <fileTransfer> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.5/16]: <copy> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.6/16]: <source> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.7/16]: <location>https://iot-tps.example.cisco.com:9120/pnp/odm/IR829GW</location> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.8/16]: </source> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.9/16]: <destination> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.10/16]: <location>flash:/managed/odm/cg-nms.odm</location> [sev=DEBUG][tid=tunnelProvJetty-534][part=33728.11/16]: </destination>
java.lang.reflect. InnvocationTargetException. Full error description: PnP request for element ID [IR1101-K9+FCW223700AV] failed [java.lang.reflect.InvocationTargetException].	Check bootstrap configuration. If error is seen immediately after updating ODM: • Check provisioning settings in the user interface. • Check debug log for empty value for proxy-bootstrap-ip property field. • Must provide a valid IP address or hostname.
Could not generate DH keypair. Full error description: java.security.Invalid.AlgorithmParameterException: DH key size must be multiple of 64 and must be in the range of 512 to 2048 (inclusive). The specific key size 4096 is not supported.	Check: ip http secure-ciphersuite

Common Errors	Items to Check and/or Resolve Errors
Error: PKIX path building failed: sun.security.provider.certpath. SunCertPathBuilderException: unable to find valid certification path to requested target. Cause: Wrong certificate is offered through HTTPS-server on FAR.	Check the certificate for Web communication with IoT FND on the router (FAR): 1. Check the configuration of the secure-transport: • Router# sh run | i secure-trustpoint • ip http secure-trustpoint LDevID • ip http client secure-trustpoint LDevID 2. If the secure-transport configuration is correct, then restart https server on FAR: • router(config)# no ip http secure-server • router(config)# ip http secure-server

Common Errors	Items to Check and/or Resolve Errors
Error: PKIX path validation failed: java.security.cert.CertPathValidatorException: validity check failed. Cause: Wrong certificate is offered through HTTPS-server on FAR.	If this error is seen, then there is an issue with the certificate used for https communication between IoT FND and FAR. In certain situations, for example, if reload-during-bootstrap=true property is used in the cgms.properties file, then this error might be seen once, after which the tunnel formation is successful. This is because of the delay in obtaining the LDevID certificate after the router boots up. But the first tunnel formation request has already been sent before LDevID is obtained. So the first time failure of tunnel formation, this error message is seen. However, when the second tunnel formation request in sent, the LDevID has already been obtained by this time for the https communication and hence the tunnel formation is successful. Workaround: From IoT FND 4.6.x onwards, remove reload-during-bootstrap=true from the cgms.properties file, as this property was introduced as a workaround for CSCvk66991. **Note** CSCvk66991 is fixed now, hence this property is not mandatory from IoT FND 4.6.x onwards.

Common Errors	Items to Check and/or Resolve Errors
Error: sun.security.validator.ValidatorException: PKIX path building failed: sun.security.provider.certpath. SunCertPathBuilderException: unable to find valid certification path to requested target Cause: Issuing CA certificate is missing in keystore.	Install Issuing CA cert.
Error in running file check command Full error: Error in running file check command: dir flash:/managed/odm/cg-nms.odm., Reason: javax.xml.ws.soap.SOAPFaultException: Serve D-H key verification failed	Add the following command to the file check: • ip http secure-client-auth • Check username and password or http conf.
Error during registration process: javax.xml.ws.WebServiceException: Could not send Message	Check WSMA. On the router (FAR), run debug: Router# debug ip http all
HTTP response ‘502: Bad Gateway’ Full error: org.apache.cxf.transport.http.HTTPException: HTTP response ‘502:Bad Gateway’ when communicating with https://10.48.43.249.443/wsma/config Error is typically seen with NGINX on IR1101. **Note** NGINX is a software-based web server. **Note** In most cases, the ‘502: Bad Gateway’ error is related to http max-connections set in the command below. tunnel(config)# ip http max-connections 20 **Note** Should the value that you enter in the command (noted above) return an error, you can increase the value until the error goes away.	On the IR1101, check NGINX log by entering one of the commands: IR1101# show platform software trace message nginx RP active -or- You can find the latest nginx file in the directory: IR1101# dir bootflash/tracelogs/nginx* To copy the latest nginx file, use one of the following: Cisco IOS file operations such as SCP or TFTP.

Common Errors	Items to Check and/or Resolve Errors
Failed to load function ‘CA InitRolePIN’Issue with (outdated) HSM Java libraries Full error: Failed to load function ‘CA_InitSlotRolePIN’ Failed to load function ‘CA_...Failed to load function ‘CA_DescribeUtilizationCounterId’ Failed to load function ‘CA TestTrace’	Backup/copy new libs to cgms or cgms-tools libs folder: [root@FNDPRDAPP01 bin]# cp -r /opt/cgms-tools/jre/lib/ext/opt/cgms-tools/jre/lib/ext-bc/ root@FNDPRDAPP01 bin]# cp /usr/safenet/lunaclient/jsp/lib/*/opt/cgms-tools/jre/lib/ext/
Reverse DNS (1 of 2) Nothing in FND log when running CGNA on FAR tcpdump does not show incoming traffic to FND Debugging CGNA/HTTP on FAR shows: cgna_httpc_post: http_send_request rc= 0 tid=55 cgna_prf timer_start:cg-nms-register:timer started Thu Jul 18 14:10:55 2019 httpc_request:Do not have the credentials cgna_http_resp_data: Received for sid=5 tid=55 status= 7	Debugging CGNA/HTTP on FAR should be (rather than the display to the left): cgna_httpc_post: http_send_request rc= 0 tid=114 cgna_prf timer_start:cg-nms-periodic: timer started Thu Jul 18 16:37:38 2019 httpc_request: Dont have the credentials Jul 18 16:37:40.844 UTC: Thu, 18 Jul 2019 14:37:40 GMT 10.48.43.251 http:10.48.43.299/cgna/ios/metrics ok Protocol = HTTP/1.1 Jul 18 16:37:40.844 UTC: Date =Thu, 18 Jul 2019 14:40:27 GMT cgna_http_resp_data: Received for sid= 4 tid=114 status=8
Reverse DNS (2 of 2) Every time FAR tries (http client) to create a TLS connection with FND, Java does a reverse DNS lookup of the source IP of the device. This is by design in Java. Apparently, for preventing DDoS attacks.	Remove DNS server or set the following in the cgms.properties: enable-reverse-dns-lookup=false (Addressed in CSCvk59944)

Common Errors	Items to Check and/or Resolve Errors
FND will not start (1 of 2) Symptom: FND stops suddenly or is unable to start on an Oracle installation where the database is installed locally.	Check the hard disk space using the command 'df-h' on the linux shell. If the disk is showing as 'full', most likely the Oracle DB archive logs have filled up the disk space and needs cleaning. Another reason could be that the database password has expired. Run the command to confirm: /opt/cgms/server/cgms/log/cgms_db_connection_test.log To change the password, become the oracle user and use the script provided in the Oracle RPM: su - oracle $ORACLE_BASE/cgms/scripts/change_password.sh
FND will not start (2 of 2) Symptom: FND service is up but GUI will not load.	Issue is mostly likely due to Linux firewall getting enabled. Disable firewall using the Linux CLI command: systemctl firewalld stop

Common Errors	Items to Check and/or Resolve Errors
After FND is upgraded to FND 4.8, the HSM Client to FND Server communication does not work and displays the following error message: 'Could not get CsmpSignatureKeyStore instance. Please verify HSM connection. Exception: Object not found.' The error above is seen in FND Deployments with HSM that are running with or without High Availability (HA).	This is an HSM library issue. HSM client is not sending right slot ID to the FND server. Hence, the customer will have to follow up with HSM support. 'Could not get CsmpSignatureKeyStore instance. Please verify HSM connection. Exception: Object not found.' (CSCvz59702) Although, the HSM client resides on the same Linux server, where the FND Application Server is also installed. The HSM client is not provided by HSM and not by Cisco. Only HSM has the expertise and visibility to the HSM code and the HSM support team can help fix this issue. FND uses SSM or HSM to store encrypted information and keys. If there is an issue with SSM or HSM, then FND will not initialize. The IoT FND component remains in Down state even if the FND application server is in UP state. In this case, when the SSM is used, then you can contact Cisco Support. They have the expertise and visibility to the code to help you resolve this issue. However, if the HSM client to server connection has issues, then the Thales/HSM vendor has the visibility and expertise to help resolve the issue.

Common Errors	Items to Check and/or Resolve Errors
CSMP certificate not displayed in IoT FND GUI during fresh install.	

Common Errors	Items to Check and/or Resolve Errors
	For a fresh install of IoT FND and HSM integration, the CSMP certificate appears in the FND UI only when an endpoint/meter is added to FND, irrespective of whether th emeter/endpoint is registered to FND or not. You can also add a dummy entry for meter/endpoint. If there is no real endpoint or meter to add at the point of testing CSMP certificate display. Apart from the CSMP certificate displayed in the GUI, you can also use the following methods to verify if IoT FND can access and retrieve the CSMP certificate from HSM: • **Method 1** Run the following command: cat /opt/cgms/server/cgms/log/server.log | grep -i HSM If you get the below message, then IoT FND and HSM communication is successful, and FND can retrieve the public key. %IOTFND-6-UNSPECIFIED: %[ch=HSMKeyStore][sev=INFO] [tid=MSC service thread 1-3]: Retrieved public key: 3059301306072a8648ce3d020106082a864 8ce3d03010703 420004d914167514ec0a110 f3170eef742a000572cea6f0285a3074db 87e43da398 ab016e40ca4be5b888c26c4 fe91106cbf685a04b0f61d599826bdbcff 25cf065d24 • **Method 2** Run the following command. The cmu list command checks if FND can see

Common Errors	Items to Check and/or Resolve Errors
	two objects stored in HSM partition, namely private keys and CSMP certificate. [root@iot-fnd ~]# cd /usr/safenet/lunaclient/bin [root@iot-fnd bin]# ./cmu list Certificate Management Utility (64-bit) v7.3.0-165. Copyright (c) 2018 SafeNet. All rights reserved. Please enter password for token in slot 0 : ******* handle=2000001 label=NMS_SOUTHBOUND_KEY handle=2000002 label=NMS_SOUTHBOUND_KEY--cert0 You have new mail in /var/spool/mail/root
Error: Caused by FATAL: terminating connection due to idle-in-transaction timeout	**Note** This is applicable only to FND-Postgres ova deployments. Edit the idle_in_transaction_session_timeout property in postgresql.conf file. By default it is set to 3h. If any operation requires the transaction to be opened for more than 3h then on getting the above error, set the value for the idle_in_transaction_session_timeout property to more than 3h and restart Postgresql service for the property to take effect. **Note** • The postgresql.conf file is located in the path: /var/lib/pgsql/12/data. • The postgres version is 12. (replace this with the current version that you are using).

Common Errors	Items to Check and/or Resolve Errors
With IoT FND and HSM integration, the CSMP certificate will not load in IoT FND UI after the upgrade.	The inability of the certificate to load is mostly likely due to the upgrade process overwriting the old HSM client libraries (example: version 5.x) with the new client libraries (example: version 7.x or 10.x or higher) that are bundled with FND 4.4 and later releases. **Note** For more information on the HSM client version that is bundled with IoT FND, refer to the corresponding FND release notes. To restore the old libraries, perform the following on the Linux shell: cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms/jre/lib/ext/ cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms/jre/lib/ext/ cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms/safenet/ cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms/safenet/ To restore the tools package: cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms-tools/jre/lib/ext cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms-tools/jre/lib/ext cp /usr/safenet/lunaclient/jsp/lib/LunaProvider.jar /opt/cgms-tools/safenet/ cp /usr/safenet/lunaclient/jsp/lib/libLunaAPI.so /opt/cgms-tools/safenet/

Common Errors	Items to Check and/or Resolve Errors
ODM file will not update on the router Symptom: During Plug and Play (PnP) or ZTD, the ODM file on the router does not get updated, which results in failure to register the device.	Issue is most likely due to the following entry in the cgms.properties file: update-files-oncgr=false Either remove the entry above or change it to 'true' as shown below: update-files-oncgr=true
Any CGR running Cisco IOS 15.6.x will not register with FND 4.3 or newer release.	Problem occurs because the WPAN high-availability (HA) feature was introduced in FND 4.3. This feature requires a minimum Cisco IOS release of 15.7(M)4.

Common Errors	Items to Check and/or Resolve Errors
SSM certificate will not load.	After upgrading to FND 4.4 or newer versions, the SSM cert is no longer seen in the CSMP certificates page. This occurs because the web certificate is getting changed after every upgrade. The web cert is used for establishing secure communication with the SSM. This change was done as part of the security compliance in FND 4.4. and all subsequent releases of FND, which generates a unique web (browser) certificate upon install or upgrade. To fix, export the self-signed web certificate from FND GUI: **1.** Go to Admin > Certificates > web certificate tab. Use the base64 format. **2.** Transfer the file to the opt/cgms-ssm directory. **3.** Stop SSM service: service ssm stop. **4.** Enter cd /opt/cgms-ssm/bin. **5.** Execute: /ssm setup.sh. **6.** Select option 8 : Import a trusted certificate to SSM-Web keystore. **7.** Enter current ssm_web_keystore password: *ssmweb*. **8.** Enter the alias for import: *fnd*. **9.** Enter Certificate filename: /opt/cgms-ssm/certForWeb.pem. **10.** Start the SSM service: service ssm start.
Could not get CsmpSignatureKeyStore instance. Please verify HSM connection.	This is an HSM client library issue. The HSM client is not sending the correct slot ID to the FND server. Please follow up with HSM support.

Common Errors	Items to Check and/or Resolve Errors
fndserver1.test.com: %IOTFND-3-UNSPECIFIED: %[ch=CgmsAuthenticator][sev=ERROR] [tid=http-/0.0.0.0:443-4] [part=150156.1/55]: Exception when adding remote user to the db. fndserver1.test.com: %IOTFND-3-UNSPECIFIED: %[ch=CgmsAuthenticator][sev=ERROR] [tid=http-/0.0.0.0:443-4] [part=150156.2/55]: com.cisco.cgms.exceptions.AAAException: failed to decrypt stored shared secret	The IoT FND server certificate contents for HA setup is: • The Subject — Must have the FQDN of the VIP. Example: FNDSERVERVIP.TEST.COM • The Subject Alternative Name (SAN) — Added must include the FQDN of the VIP. Example: FNDSERVERVIP.TEST.COM (same as the subject) • The Subject Alternative Name — Must NOT have the individual server names. Example: It must not contain FNDSERVER1.TEST.COM, FNDSERVER2.TEST.COM

- Determine the RHEL Version, on page 56
- Debug Using Cisco IoT FND, on page 57
- Encrypt and Decrypt Passwords Using Cisco IoT FND, on page 57
- Find the Cisco IoT FND Version, on page 58
- SSL Debug, on page 58
- Troubleshoot Incorrect CGNA Profile URL, on page 59
- Troubleshoot CGMS Errors, on page 59
- Troubleshoot TPS Proxy Errors, on page 60
- Troubleshoot HER FlexVPN, on page 61
- Troubleshoot SSM Errors, on page 61
- Troubleshoot RA Errors, on page 62
- Troubleshoot Profile Environment Variable, on page 62
- Troubleshoot High CPU Issues, on page 63
- Troubleshoot Tunnel Provisioning Issues, on page 66
- Troubleshoot Tunnel Provisioning DHCP Configuration Issue, on page 67
- Troubleshoot Profile Environment Variable, on page 68
- Verify Tunnel Configuration Information, on page 69
- Troubleshoot Archive Directory Creation During ZTD, on page 69

Determine the RHEL Version

Problem

Unable to determine the RHEL version of Cisco IoT FND.

Solution

Here are the instructions to determine the RHEL version of Cisco IoT FND:

1. Log in to the root directory of Cisco IoT FND using the SSH command.
2. Run the following command:

```
cat /etc/redhat-release
```

3. You can see the RHEL version of the Cisco IoT FND. Here's an example:

```
[user@hostname ~]$ cat /etc/redhat-release
Red Hat Enterprise Linux release 8.3 (Ootpa)
```

Debug Using Cisco IoT FND

Problem

Unable to debug using Cisco IoT FND.

Solution

Here are the instructions to debug using Cisco IoT FND:

1. From the Cisco IoT FND menubar, choose **ADMIN** > **System Management** > **Logging**.
2. Navigate to the **Log Level Settings** tab.
3. In the **Change Log Level to** drop-down list, choose **Debug**.
4. Select a **Category** that you want to debug.



Note You can select multiple categories as well.

5. Click **Go**.
6. The EIDs selected for debugging appears in the **Eids for debugging** text box.
7. Click the **Save** icon to save.

Encrypt and Decrypt Passwords Using Cisco IoT FND

Problem

When you're prompted to enter the password for the cgms.properties file, enter an encrypted password. How to encrypt the password?

Solution

Here are the instructions to encrypt and decrypt your password:

1. To check if the .cgms properties file password is correct, decrypt the password.
2. Execute the following command to encrypt the password:

   ```
   /opt/cgms/bin/encryption_util.sh encrypt yourpassword
   ```

   The execution of the command before this, returns an encrypted version of your password.
3. Execute the following command to decrypt the password:

   ```
   /opt/cgms/bin/encryption_util.sh decrypt encryptedpassword
   ```

   The execution of the command before this, returns the original password without the encryption.

Example

Here's an example of the encryption and decryption of the password:

Encription:

```
/opt/cgms/bin/encryption_util.sh encrypt testpasswordencryption
```

The output would be something like the following:

```
imUwTCuURBU39x8oZWHxBMGWYf1+e0mN6Hwa1quKC1xZ8troq+3Wt7KQWgVmbOMd3jo=
```

Decryption:

```
/opt/cgms/bin/encryption_util.sh decrypt
imUwTCuURBU39x8oZWHxBMGWYf1+e0mN6Hwa1quKC1xZ8troq+3Wt7KQWgVmbOMd3jo=
```

The output would be something like the following:

```
testpasswordencryption
```

Find the Cisco IoT FND Version

Problem

Unable to find the Cisco IoT FND version.

Solution

You can find the version of the Cisco IoT FND at the bottom left-hand side of the Cisco IoT FND screen.

SSL Debug

Problem

Unable to debug the Secure Socket Layer (SSL) of Cisco IoT FND:

Solution

To debug the SSL of Cisco IoT FND, you should set `DEBUG_SSL` to `true` in the `/opt/bin/cgms/bin/cgms.conf` file and restart the `cgms` service.

Here are the instructions to set `DEBUG_SSL` to `True`:

1. Open the cgms.conf file using a text editor. You can use editors like nano, vi, or vim.

   Here's an example:

   ```
   s[root@fnd bin]# nano /opt/bin/cgms/bin/cgms.conf
   ```

2. Edit the configuration file. Locate the line that starts with `DEBUG_SSL` in the code. If it doesn't exist, you should add the line.

   Set the value of `DEBUG_SSL` to true.

   Here's an example:

   ```
   DEBUG_SSL=true
   ```

3. Save and exit the editor, then run the following command to restart the cgms services:

   ```
   [root@fnd bin]# service cgms restart
   ```

This process should successfully set `DEBUG_SSL` to `true` in the configuration file and restart the `cgms` service to apply the changes.

Troubleshoot Incorrect CGNA Profile URL

Problem

Seeing an incorrect CGNA profile configuration URL error.

Solution

Verify if you've entered the URL in the right format. The right format of the CGNA profile configuration URL is https://<FND IP>:9120/cgna/ios/tunnel

Troubleshoot CGMS Errors

Problem

Cisco IoT FND stopped working because of a CGMS error.

Solution

You can identify CGMS errors using the following instructions:

1. Log in to the root directory of Cisco IoT FND using the SSH command.
2. Run the following commands to view CGMS related information:

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> cgms`
7.x	`service cgms <status/start/restart/stop>`

Here's an example:

```
[root@iot-fnd-oracle ~]# systemctl status cgms
Warning: The unit file, source configuration file or drop-ins of cgms.service c>
● cgms.service - SYSV: cgms is a network management system for the Cisco Connec>
   Loaded: loaded (/etc/rc.d/init.d/cgms; generated)
   Active: active (running) since Wed 2024-11-06 12:49:43 UTC; 4 days ago
     Docs: man:systemd-sysv-generator(8)
  Process: 1915951 ExecStop=/etc/rc.d/init.d/cgms stop (code=exited, status=1/F>
  Process: 1916297 ExecStart=/etc/rc.d/init.d/cgms start (code=exited, status=0>
    Tasks: 600 (limit: 152533)
   Memory: 2.3G
   CGroup: /system.slice/cgms.service
           ├─1916367 runuser -c bin/cgms -s /bin/bash root
           ├─1916370 /bin/sh bin/cgms
           ├─1916550 /bin/sh bin/standalone.sh -Djboss.server.log.dir=bin/../se>
           └─1916678 java -D[Standalone] -verbose:gc -Xloggc:/opt/cgms/server/c>

Nov 06 12:49:37 iot-fnd-oracle systemd[1]: Starting SYSV: cgms is a network man>
Nov 06 12:49:43 iot-fnd-oracle runuser[1916367]: pam_limits(runuser:session): i>
Nov 06 12:49:43 iot-fnd-oracle runuser[1916367]: pam_unix(runuser:session): ses>
Nov 06 12:49:43 iot-fnd-oracle systemd[1]: Started SYSV: cgms is a network mana>
```

Troubleshoot TPS Proxy Errors

Problem

Cisco IoT FND stopped working due to some TPS proxy error.

Solution

Use the following instructions to troubleshoot TPS Proxy errors:

1. Log in to the root directory of Cisco IoT FND using the SSH command.
2. Run the following commands to view TPS Proxy related information:

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> tpsproxy`
7.x	`service tpsproxy <status/start/restart/stop>`

Here's an example:

```
[root@iot-fnd-oracle ~]# systemctl status tpsproxy
tpsproxy.service - TPS Proxy Service
   Loaded: loaded (/etc/systemd/system/tpsproxy.service; enabled; vendor preset: enabled)

   Active: active (running) since Fri 2023-10-06 10:15:42 UTC; 2h 34min ago
 Main PID: 1234 (tpsproxy)
    Tasks: 5 (limit: 4915)
   CGroup: /system.slice/tpsproxy.service
           └─1234 /usr/bin/tpsproxy --option
```

Troubleshoot HER FlexVPN

Problem

Unable to configure the FlexVPN client on a Cisco Connected Grid Router (CGR).

Solution

Here are the instructions to configure the FlexVPN on a Cisco Connected Grid Router:

1. Access the Cisco Connected Grid Router's configuration using the SSH command.
2. Run `show running-config`, `show crypto ikev2 sa`, and `show cypto ipsec sa` to review the FlexVPN configuration.
3. Use `show interfaces` to check the status of the dynamically created tunnel interface.
4. Verify that the tunnel interface associated with FlexVPN is up and running.
5. Run `show logging` and review the logs for any errors or warnings related to FlexVPN.
6. Look for messages indicating issues in tunnel establishment or connectivity.
7. Use `show template` to review the applied templates and confirm if they match the intended configuration for FlexVPN.

   Verify that the HER FlexVPN configuration on the Cisco IOS Connected Grid Router is set up and operational. This ensures that the CGR can dynamically create tunnel endpoints and maintain secure communication with the HER.

Troubleshoot SSM Errors

Problem

Cisco IoT FND stopped working due to some SSM error.

Solution

Here are the instructions to find more about SSM errors. Use the following instructions to troubleshoot SSM errors:

1. Log in to the root directory of Cisco IoT FND using the SSH command.
2. Run the following commands to view SSM related information:

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> ssm`
7.x	`service ssm <status/start/restart/stop>`

Here's an example:

```
[root@iot-fnd-oracle ~]# systemctl status ssm
ssm.service - System Service Manager
   Loaded: loaded (/etc/systemd/system/ssm.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-10-06 09:00:00 UTC; 3h 45min ago
 Main PID: 5678 (ssm)
    Tasks: 8 (limit: 4915)
   CGroup: /system.slice/ssm.service
           └─5678 /usr/bin/ssm --config /etc/ssm/config.cfg

Oct 06 09:00:00 server-name systemd[1]: Started System Service Manager.
Oct 06 09:00:01 server-name ssm[5678]: Initializing system services...
Oct 06 09:00:02 server-name ssm[5678]: System services initialized successfully.
```

Troubleshoot RA Errors

Problem

Cisco IoT FND stopped working due to some RA error.

Solution

Here are the instructions to troubleshoot RA errors.

1. Log in to the root directory of Cisco IoT FND using the SSH command.
2. Run the following commands to view RA related information:

RHEL Version	Command
8.x	`systemctl <status/start/restart/stop> fnd-ra`
7.x	`service fnd-ra <status/start/restart/stop>`

Here's an example:

```
[root@iot-fnd-oracle ~]# systemctl status fnd-ra
fnd-ra.service - FND Remote Access Service
   Loaded: loaded (/etc/systemd/system/fnd-ra.service; enabled; vendor preset: enabled)

   Active: active (running) since Fri 2023-10-06 08:45:00 UTC; 4h 15min ago
 Main PID: 4321 (fnd-ra)
    Tasks: 10 (limit: 4915)
   CGroup: /system.slice/fnd-ra.service
           └─4321 /usr/bin/fnd-ra --config /etc/fnd-ra/config.cfg

Oct 06 08:45:00 server-name systemd[1]: Started FND Remote Access Service.
Oct 06 08:45:01 server-name fnd-ra[4321]: Establishing remote connections...
Oct 06 08:45:05 server-name fnd-ra[4321]: Remote connections established successfully.
```

Troubleshoot Profile Environment Variable

Problem

Need to verify that the correct profile is configured for the `ZTD_SCEP_CGNA_Profile` environment variable.

Solution

You'll need to check the Cisco Global Network Architecture (CGNA) profile settings on your device:

1. Log in to the device using the SSH command.
2. Run `show cgna profile-all` and see all the configured CGNA profiles and check which one is active.
3. Look for `ZTD_SCEP_CGNA_Profile` environment variable ensuring it is set to the intended profile.
4. Confirm that the TCL script `tm_ztd_scep.tcl` has run without errors. This script is responsible for activating the correct CGNA profile.
5. Ensure that the LDevID certificate is correctly retrieved from the PKI, as this is necessary for the profile activation.
6. Check logs for any errors or warnings that might indicate problems with profile configuration or activation.

Troubleshoot High CPU Issues

Problem

Often facing a spike in the CPU usage.

Solution

Here are the instructions to analyze and troubleshoot some of the high CPU issues:

1. Analyze if the CPU spike is intermittent or constant. If intermittent, make a note of the duration.
2. From the Cisco IoT FND Menubar, choose **Devices** > **Servers**.
3. Choose the problematic Cisco IoT FND server which has CPU issues and analyze the charts to spot any CPU spike issues.
4. Run the following commands as root using the SSH command:

Command	Description
lscpu	Gathers CPU architecture information from sysfs and /proc/cpuinfo, information includes number of CPUs, threads, cores, sockets etc.
ps-aux	View statistics about system's running processes, like the percent of CPU and memory that the process is using etc
ps aux k-pcpu | head -6	This command sorts the list of running processes by CPU usage in descending order (using `k-pcpu`) and then displays the top 6 processes (`head -6`). It's useful for identifying which processes are consuming the most CPU.

Command	Description
ps aux | sort -nrk 3,3 | head -n 5	This command sorts the running processes by the third column (which is typically %CPU) in numerical reverse order (`-nrk 3,3`) and displays the top 5 entries (`head -n 5`). Similar to the previous command, it helps identify CPU-intensive processes.
top	View a real-time view of running processes in Linux and displays kernel-managed tasks, enables user to monitor CPU-intensive tasks
top -b -n 1 | head -n 12 | tail -n 5	This command runs `top` in batch mode (`-b`) for one iteration (`-n 1`). It then takes the first 12 lines of the output and displays the last 5 lines (`head -n 12 | tail -n 5`). This typically shows the header information and the most CPU-intensive processes at a given moment.
sar -t -r -f	The `sar` command is used for collecting, reporting, or saving system activity information. The `-r` flag is for memory usage, and `-t` specifies time format. The `-f` flag is used to read data from a file, typically a `sar` data file. This would provide historical data on memory usage over time.
sar -t -q -f	Similar to the previous `sar` command, but with the `-q` flag, which is used to report queue length and load average. This helps in understanding how the system load has been distributed over time.
iostat	Monitor CPU utilization, system input/output statistics for all the disks and partitions
free -m	Get detailed report on the system's memory usage
mpstat	View information about CPU utilization and performance
vmstat	View information about processes, memory, disk, and CPU activity

Here's a sample output:

```
ps -aux
USER PID %CPU %MEM VSZ RSS TTY STAT START TIME COMMAND
root 1 0.0 0.0 191428 4332 ? Ss Jul09 20:48 /usr/lib/systemd/systemd --switched-roo
root 2 0.0 0.0 0 0 ? S Jul09 0:00 [kthreadd]
root 4 0.0 0.0 0 0 ? S< Jul09 0:00 [kworker/0:0H]
root 6 0.0 0.0 0 0 ? S Jul09 3:22 [ksoftirqd/0]
root 7 0.0 0.0 0 0 ? S Jul09 0:06 [migration/0]
root 8 0.0 0.0 0 0 ? S Jul09 0:00 [rcu_bh]
root 9 0.0 0.0 0 0 ? R Jul09 59:17 [rcu_sched]
root 10 0.0 0.0 0 0 ? S< Jul09 0:00 [lru-add-drain]
root 11 0.0 0.0 0 0 ? S Jul09 0:49 [watchdog/0]
```

```
root 12 0.0 0.0 0 0 ? S Jul09 0:17 [watchdog/1]
root 52 0.0 0.0 0 0 ? S Jul09 0:20 [watchdog/9]
root 53 0.0 0.0 0 0 ? S Jul09 0:19 [migration/9]
root 54 0.0 0.0 0 0 ? S Jul09 0:28 [ksoftirqd/9]
root 56 0.0 0.0 0 0 ? S< Jul09 0:00 [kworker/9:0H]
```

Gather the Heap and Thread Dumps

When you notice slowness in your Cisco IoT FND, collect the JAVA heap and thread dumps. Use the following instructions to collect the heap and thread dumps:

1. Download the JDK package which is an OpenJDK distribution for Linux (64-bit) from https://corretto.aws/downloads/latest/amazon-corretto-8-x64-linux-jdk.tar.gz.
2. Extract the JDK tar.gz file.

   For example:

   ```
   tar -xzf amazon-corretto-8-x64-linux-jdk.tar.gz
   ```

3. Transfer the extracted JDK file to the Cisco IoT FND server using the scp command.

   For example:

   ```
   scp -r amazon-corretto-8-x64-linux-jdk user@fnd-server:/path/to/destination
   ```

4. Navigate to the `bin` directory of the JDK installation to access the `jmap` utility. Example:

   ```
   cd /path/to/destination/amazon-corretto-8-x64-linux-jdk/bin
   ```

5. Find the docker ID using the `docker ps` command. Here's an example:

   ```
   [root@iot-fnd ~]# docker ps
   CONTAINER ID        IMAGE               COMMAND                  CREATED
   STATUS              PORTS

                                                                          NAMES
   54e349b30f46        fogd-image:active   "/bin/sh -c /usr/loc…"   12 days ago
   Up 12 days          443/tcp

                                                                          fogd-container
   f67d89e4188b        fnd-image:active    "/bin/sh -c /opt/fnd…"   12 days ago
   Up 2 days           0.0.0.0:80->80/tcp, 0.0.0.0:162->162/udp, 0.0.0.0:443->443/tcp,
   0.0.0.0:9120-9121->9120-9121/tcp, 0.0.0.0:5683->5683/udp,
   0.0.0.0:61624-61626->61624-61626/udp, 0.0.0.0:9124-9125->9124-9125/tcp,
   0.0.0.0:61628->61628/udp   fnd-container
   ```

6. Copy the tar.gz file to the docker using the following command and paste the docker ID:

   ```
   docker cp /amazon-corretto-8.422.05.1-linux-x64.tar.gz f67d89e4188b
   ```

7. Log in to the docker using the following command:

   ```
   [root@iot-fnd ~]# docker exec -i -t fnd-container /bin/bash
   ```

8. Unzip the tar.gz file using the `tar-xvf<complete path>` command. Here's an example:

   ```
   [root@iot-fnd ~]# tar -xvf /CISCO/amazon-corretto-8.422.05.1-linux-x64.tar.gz]
   ```

9. Find the CGMS process ID using the `ps -eaf | grep cgms` command. Here's an example:

   ```
   [root@iot-fnd ~]# ps -eaf | grep cgms
   root      308261       1  0 Jul29 pts/0    00:00:00 runuser -c bin/cgms -s /bin/bash
   root
   root      308262  308261  0 Jul29 pts/0    00:00:00 /bin/sh bin/cgms
   root      308265  308262  0 Jul29 pts/0    00:00:00 /bin/sh bin/standalone.sh
   ```

```
-Djboss.server.log.dir=bin/../server/cgms/log --server-config=standalone.xml -b 0.0.0.0
root      308382  308265  2 Jul29 pts/0    05:55:42 java -D[Standalone] -verbose:gc
-Xloggc:/opt/cgms/server/cgms/log/gc.log -XX:+PrintGCDetails -XX:+PrintGCDateStamps
-XX:+UseGCLogFileRotation -XX:NumberOfGCLogFiles=5 -XX:GCLogFileSize=3M
-XX:-TraceClassUnloading -Xms1g -Xmx6g -XX:MaxPermSize=512m
-Dcom.cisco.cgms.ciscolog.host=fnd-server -server -XX:+HeapDumpOnOutOfMemoryError
-XX:HeapDumpPath=bin/../server/cgms/log -Djava.security.egd=file:/./dev/urandom
-XX:+UnlockDiagnosticVMOptions -XX:+LogVMOutput
-XX:LogFile=bin/../server/cgm/log/cgms_stacktrace.log -XX:-OmitStackTraceInFastThrow
-Dorg.terracotta.quartz.skipUpdateCheck=true -Dbase.dir=bin
-Dorg.jboss.boot.log.file=/opt/cgms/server/cgms/log/server.log
-Dlogging.configuration=file:/opt/cgms/standalone/configuration/logging.properties
-jar /opt/cgms/jboss-modules.jar -mp /opt/cgms/modules org.jboss.as.standalone
-Djboss.home.dir=/opt/cgms -Djboss.server.base.dir=/opt/cgms/standalone
-Djboss.server.log.dir=bin/../server/cgms/log --server-config=standalone.xml -b 0.0.0.0
root      430277  430190  0 17:34 pts/1    00:00:00 grep --color=auto cgms
```

10. Use the following command to generate the heap dump:

```
[root@iot-fnd ~]# /CISCO/amazon-corretto-8.422.05.1-linux-x64/bin/jmap
-dump:file=heap_dump_11082024.hprof 308382
Dumping heap to /ENEDIS/heap_dump_11082024.hprof ...
Heap dump file created
[root@fnd-server CISCO]#
[root@fnd-server CISCO]#
[root@fnd-server CISCO]# ls -l | grep -i hprof
-rw------- 1 root root 1148131128 Aug 11 11:34 heap_dump_11082024.hprof
```

11. Use the following command to generate the thread dump:

```
[root@iot-fnd ~]# kill -3<CGNMSpid>
```

12. Upload both the cgms_stacktrace.log to TAC SR for further analysis.

Troubleshoot Tunnel Provisioning Issues

Problem

Require guidance on troubleshooting issues related to tunnel provisioning failures by examining server logs and adjusting logging levels.

Solution

Use server.log Command

Here are the steps to troubleshoot tunnel provisioning issues:

1. Log in to the Cisco IoT FND server using the SSH command.
2. Navigate to the directory where your logs are stored using the cd command. For example: `/opt/fnd/logs` or `/opt/cgms/server/cgms/logs/`.
3. Run the `server.log` command to view the log files.

Use Cisco IoT FND GUI

1. Log in to Cisco IoT FND.

2. From the Cisco IoT FND menubar, choose **ADMIN** > **Logging** > **Log Level Settings**.
3. Ensure that **Tunnel Provisioning** option is checked.
4. Choose the **Log Level** as **Debug**.

Identify Issues In Logs

Look for specific error messages in the logs. For example, a common issue is the **Tunnel provisioning request queue is full**. This error indicates that the request queue has reached its capacity and new requests are being dropped.

Understand the Queue Limitation

Until Cisco IoT FND Release 4.12.1, the tunnel provisioning queue size is limited to 12 requests. If the queue is full, additional requests are not processed.

Resolution

- Consider upgrading to a Cisco IoT FND version later than Cisco IoT FND Release 4.12. The queue size is increased.
- Try restarting Cisco IoT FND which might temporarily resolve the issue by clearing the queue.

Troubleshoot Tunnel Provisioning DHCP Configuration Issue

Problem

Seeing a **Tunnel Provisioning Failure** event in the Cisco IoT FND logs.

Solution

You need to monitor the address allocation process using the following steps:

1. Check the Cisco IoT FND server.log file to determine if Cisco IoT FND is sending a DHCP request during tunnel provisioning.
2. Check the DHCP server.log file to determine if the DHCP request from Cisco IoT FND reached the DHCP server.
3. If the requests are not reaching the server. From the Cisco IoT FND menubar, choose **ADMIN** > **System Management** > **Provisioning Settings** and ensure that the DHCP **Server Address** is right.
4. Check for network problems between Cisco IoT FND server and the DHCP server.
5. Analyze the DHCP server log files to identify errors in the configuration and fix them.

Example

The Cisco Connected Grid Routers time out during Tunnel Provisioning and then rollback to before-tunnel-config repeatedly:

Error seen in server.log:

```
238471507: imcltcgnmsp01: Dec 13 2017 03:24:03.709 +0000: %CGMS-3-UNSPECIFIED:
%[ch=CiscoIosTunnelProvServlet$CiscoIosTunnelProvProcess][eid=CGR1240/K9+FTX2130G06C][ip=10.189.110.137][sev=ERROR][tid=IOS
 CGR Tunnel-892]: Tunnel provisioning request for element [CGR1240/K9+FTX2130G06C] failed
[com.cisco.cgms.dhcp.proxyclient.v4.TimeoutWaitingForOfferException: Timed out waiting for
 a DHCPOFFER message].
```

In the above error, you can see that the Cisco IoT FND server has sent a DHCPDISCOVER request and is waiting for a DHCPOFFER message from the DHCP server and it times out waiting.

Here are the DHCP logs:

```
Bluecat DHCP: Customer's DHCP
Dec 13 15:24:44 cltuofdhcp dhcpd: DHCPRELEASE of 10.190.140.110 from 00:00:00:00:00:00 via
 10.200.230.34 (found)
Dec 13 15:24:44 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34
Dec 13 15:24:45 cltuofdhcp dhcpd: DHCPOFFER on 10.190.140.110 to 00:00:00:00:00:00 via
10.200.230.34
Dec 13 15:24:45 cltuofdhcp dhcpd: DHCPREQUEST for 10.190.140.110 (10.200.230.33) from
00:00:00:00:00:00 via 10.200.230.34
Dec 13 15:24:45 cltuofdhcp dhcpd: DHCPACK on 10.190.140.110 to 00:00:00:00:00:00 via
10.200.230.34
Dec 13 15:24:46 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34:
network 10.190.192.0/18: no free leases
Dec 13 15:24:50 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34:
network 10.190.192.0/18: no free leases
Dec 13 15:24:58 cltuofdhcp dhcpd: DHCPDISCOVER from 00:00:00:00:00:00 via 10.200.230.34:
network 10.190.192.0/18: no free leases
```

From the above logs, you can see that there is an issue which prevents IPs in the 10.190.192.0/18 scope from being allocated. DHCP configuration needs to be fixed in order for tunnel provisioning to work in this case.

Troubleshoot Profile Environment Variable

Problem

Need to verify that the correct profile is configured for the `ZTD_SCEP_CGNA_Profile` environment variable.

Solution

You'll need to check the Cisco Global Network Architecture (CGNA) profile settings on your device:

1. Log in to the device using the SSH command.
2. Run `show cgna profile-all` and see all the configured CGNA profiles and check which one is active.
3. Look for `ZTD_SCEP_CGNA_Profile` environment variable ensuring it is set to the intended profile.
4. Confirm that the TCL script `tm_ztd_scep.tcl` has run without errors. This script is responsible for activating the correct CGNA profile.
5. Ensure that the LDevID certificate is correctly retrieved from the PKI, as this is necessary for the profile activation.
6. Check logs for any errors or warnings that might indicate problems with profile configuration or activation.

Verify Tunnel Configuration Information

Problem

Unable to verify the tunnel configuration information.

Solution

Here are the steps to verify the tunnel configuration information:

1. From the Cisco IoT FND menubar, choose **CONFIG** > **Tunnel Provisioning** > **Router Tunnel Addition**.
2. You can see the entire router tunnel addition information in the tab.

Troubleshoot Archive Directory Creation During ZTD

Problem

Cisco IoT FND fails to create the `bootflash:archive` directory during the bootstrap phase. This causes the ZTD (Zero Touch Deployment) onboarding to fail, resulting in configuration rollback error during tunnel provisioning.

Note If the directory already exists, the issue does not occur.

Solution

Here are the steps to address the directory creation issue:

1. Check the **cgms.properties** if the `reload-during-bootstrap` property is set to `true` or `false`.

Note By default, the `reload-during-bootstrap` property is set to `false`.

2. Take action based on the `reload-during-bootstrap` value:

   If `reload-during-bootstrap` is set to `true`.

   Create the archive directory using one of the following methods:

   - Manually create the directory `bootflash:archive` on the router before initiating the bootstrap process.
   - Automate the process using an EEM (Embedded Event Manager) script and add it to the bootstrap config. Here's an example:

```
event manager applet post_pnp
event timer watchdog time 30
action 10.0 cli command "enable"
action 15.0 cli command "file prompt quiet"
action 18.0 cli command "do mkdir bootflash:archive"
```

```
    action 20.0 cli command "archive"
    action 21.0 cli command "path bootflash:/archive/"
    action 22.0 cli command "maximum 8"
    action 24.0 cli command "no event manager applet post_pnp"
```

3. If `reload-during-bootstrap` is set to false, add `file prompt quiet` in the bootstrap template:

   Ensure the `file prompt quiet` command is included to suppress interactive confirmation prompts during directory creation. Here's an example:

```
!
<#-- Creating archive directory on bootflash -->
file prompt quiet
do mkdir bootflash:archive
!
archive
 path bootflash:/archive/
 maximum 8
!
```

4. Alternatively, you can use an alternative archive path:

   Specify a different directory path, such as `bootflash:`, to avoid the issue. Here's an example:

```
!
<#-- Configuring archive path to bootflash -->
archive
 path bootflash:
 maximum 8
!
```

5. Initiate the ZTD to confirm the issue is resolved after applying one of the solutions.