Cisco Spaces: IoT Service Configuration Guide (Wireless)

First Published: 2020-08-31

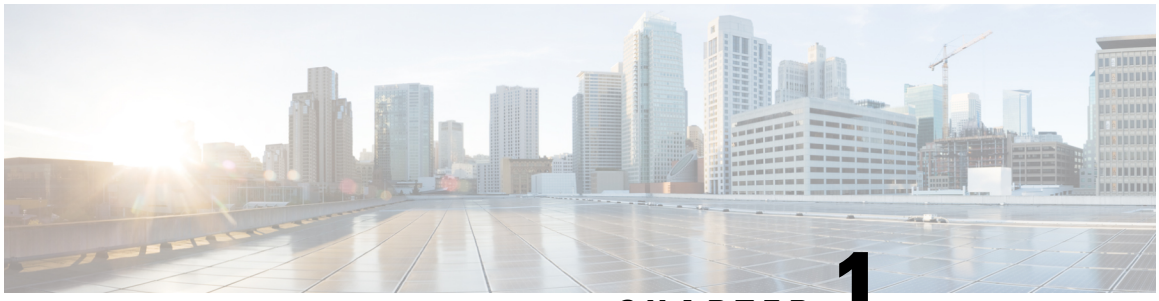
Last Modified: 2024-08-02

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

CONTENTS

PART I

Overview



CHAPTER 1

Overview

Note **Cisco DNA Spaces** is now **Cisco Spaces**. We are in the process of updating our documentation with the new name. This includes updating GUIs and the corresponding procedures, screenshots, and URLs. For the duration of this activity, you might see occurrences of both **Cisco DNA Spaces** and **Cisco Spaces**. We take this opportunity to thank you for your continued support.

Overview of Cisco Spaces: IoT Service (Wireless)

Cisco Spaces: IoT Service (Wireless) is a platform service within Cisco Spaces that enables you to claim, manage, and monitor IoT devices using Cisco's wireless infrastructure. IoT Service is designed to enable management of IoT devices across vendors, form factors, and technology protocols. Bluetooth Low Energy (BLE) is the first technology available for management using IoT services.

IoT service (wireless) encompasses hardware, software, and partner components to enable the management of devices that support critical business outcomes. IoT service (wireless) uses Cisco Catalyst 9800 Series Wireless Controllers, Cisco Spaces: Connector, Cisco Wi-Fi6 access points, and Cisco Spaces. IoT service (wireless) adopts a next-generation approach to manage complexity in an enterprise environment.

Using the IoT service (wireless), you can perform the following IoT management activities:

- Deploy BLE gateways on supported APs in your network.
- Claim the BLE beacons that you acquired from Cisco Spaces: IoT Device Marketplace.
- Configure APs and manage floor beacons.
- Monitor device attributes such as location, telemetry, battery status, and movement status.

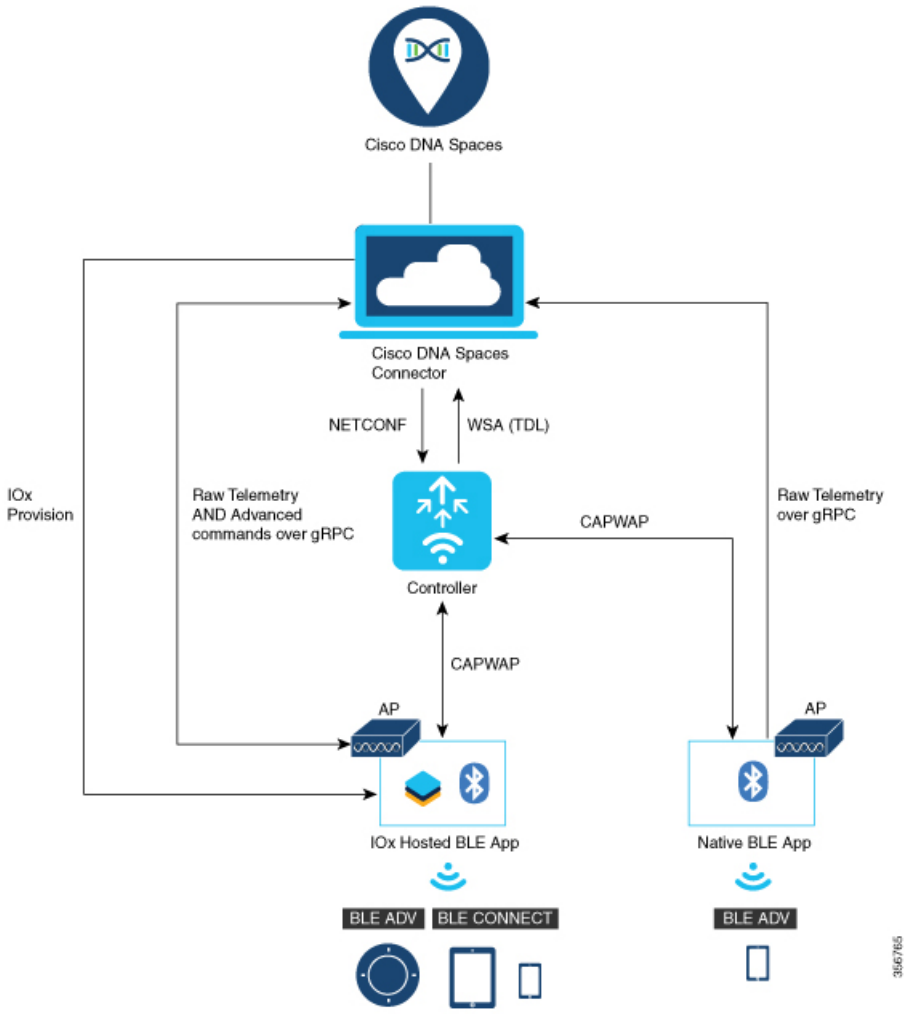
Components of Cisco Spaces: IoT Service

The section describes the various components that work to complete the Cisco Spaces: IoT Service solution.

The Cisco Catalyst 9100 Series Family of Access Points acts as a gateway of communication between Cisco Spaces and the IoT devices. Cisco Spaces: IoT Service can then use a range of common APIs to communicate with edge devices and apps. The Cisco Spaces: IoT Service collects data from devices and apps, and passes

it to Cisco-partnered websites that manage these devices far more extensively (referred to in this document as Device Manager websites). These Device Manager websites can use edge-device signals to enable outcomes specialized and targeted for each industry.

Figure 1: Components of IoT Service



Access Points

You can configure access points as gateways in Cisco Spaces. You can find the list of supported APs in the **Compatibility Matrix** section.

Depending on the type of Cisco APs, you can configure an AP as one of the following types of BLE gateways:

- **Base BLE Gateway**: This is a type of AP that you can configure in either the **Transmit** mode or the **Scan** mode.

  In the **Transmit** mode, the AP can broadcast iBeacon, Eddystone URL, and Eddystone UID profiles.

  In the **Scan** mode, the AP can scan the vicinity for other BLE devices. Using gRPC, an AP sends the scanned data to Cisco Spaces: Connector. The AP can also receive telemetry data from floor beacons. The Cisco Spaces: Connector dashboard decodes and displays this information.

- **Advanced BLE Gateway:** This gateway is an AP that is installed with the Cisco IOx App. Using the installed Cisco IOx App, you can configure floor beacons on the Cisco Spaces dashboard. You can also upgrade the floor beacon firmware from the Cisco Spaces dashboard.

  You can configure this AP in the **Scan** mode and the **Transmit** mode.

  In the **Transmit** mode, the AP can broadcast iBeacon, Eddystone URL, and Eddystone UID profiles.

  In the **Scan** mode, the AP can scan the vicinity for other BLE devices. Using gRPC, an AP sends the scanned data to Cisco Spaces: Connector. The AP can also receive telemetry data from floor beacons. The Cisco Spaces: Connector dashboard decodes and displays this information.

Cisco Catalyst 9800 Series Wireless Controllers

The Cisco Catalyst 9800 Series Wireless Controller (Catalyst 9800 controller) combines RF excellence with Cisco IOS-XE benefits, and comes in physical or virtual form factor. This wireless controller is reliable and highly secure. You can manage this Catalyst 9800 controller using CLI, GUI, NETCONF, Yang, or the Catalyst Center.

The Catalyst 9800 controller is the single point for configuring and managing a wireless network and access points. The Catalyst 9800 controller configures and manages APs using the CAPWAP protocol.

The Catalyst 9800 controller receives BLE configuration from Cisco Spaces over NETCONF and passes the configuration to AP over CAPWAP. The feedback path from the AP to the wireless controller is through CAPWAP, and from the Catalyst 9800 controller to Cisco Spaces through Telemetry data logger (TDL) telemetry streaming. The gRPC configuration from Cisco Spaces also goes through the Catalyst 9800 controller, and from there to the corresponding AP. The configuration sets up the gRPC channel between the AP and Cisco Spaces. The AP sends the gRPC channel statistics to the Catalyst 9800 controller, and you can view these statistics on the Catalyst 9800 controller.

Note

- You can have only one gRPC session between an AP and connector.
- Cisco Catalyst 9800 Series Wireless Controller running Cisco IOS XE Amsterdam 17.3.x supports only one of the following:
  - IoT service (wireless) with Cisco Spaces.
  - Network Assurance solution on Catalyst Center using Intelligent Capture (iCAP)

IoT service (wireless) and Intelligent Capture (iCAP) can co-exist from Cisco IOS XE Cupertino 17.7.x or higher.

Cisco Spaces: IoT Device Marketplace

Cisco Spaces: IoT Device Marketplace is a platform where you can discover, research, and purchase IoT devices. IoT Device Marketplace is a part of the Cisco Spaces full-stack partner ecosystem. Each device is preconfigured to give the customer an out-of-the-box experience with sensors, tags, wearables, and more. All the devices are compatible with the applications in the App Center. Current devices in the IoT Device Marketplace leverage BLE to transmit telemetry, with plans to add other technology in the future, such as Ultra Wide Band (UWB) and Zigbee.

Cisco Spaces: Connector

Cisco Spaces: Connector allows Cisco Spaces to communicate with more than one

- Cisco AireOS Wireless Controllers, and
- Cisco Catalyst 9800 Series Wireless Controllers

.

APs connect to connector using the gRPC framework.

The APs establish a connection to connector using the gRPC protocol. The gRPC protocol configures floor beacons and receives telemetry data from the floor beacons. gRPC is a bidirectional streaming service, and requires a certificate to validate the host connection and a token for authentication. Each AP creates a gRPC connection. Connector can thus support many simultaneous connections.

Compatibility Matrix for IoT Service (Wireless)

Application Name	Support for Cisco Spaces: IoT Service
Supported wireless controllers	• Supported on Cisco Catalyst 9800 Series Wireless Controllers, Release 17.3.1 and later • Not supported on Cisco AireOS Wireless Controller • Not supported on Cisco Embedded Wireless Controller on Cisco Catalyst Access Points (Cisco EWC-AP) • Supported on Catalyst 9800 Controller running on Catalyst Switches in SD-Access mode (ECA)
Cisco Spaces: Connector Docker	2.0.455 and later
Cisco Spaces: Connector OVA	2.3 and later
Cisco Prime Infrastructure	Cisco Prime Infrastructure Release 3.8 MR1 and later
Catalyst Center (for map import)	Catalyst Center Release 2.1.1 and later

Application Name	Support for Cisco Spaces: IoT Service
Access Points for advanced BLE gateway (Wi-Fi 6)	• Cisco Catalyst 9105 Series Access Points • Cisco Catalyst 9115 Series Access Points • Cisco Catalyst 9117 Series Access Points • Cisco Catalyst 9120 Series Access Points • Cisco Catalyst 9130 Series Access Points • Cisco Catalyst 9136 Series Access Points • Cisco Catalyst 9162 Series Access Points • Cisco Catalyst 9164 Series Access Points • Cisco Catalyst 9166 Series Access Points • Cisco Aironet 4800 Series Access Points
Access points for basic BLE gateway	• Cisco Aironet 1815 Series Access Points • Cisco Aironet 2800 Series Access Points (USB dongle needed. No in-built USB radio) • Cisco Aironet 3800 Series Access Points (USB dongle needed. No in-built USB radio)
Cisco IOx App Version	1.0.46 and later **Note** For Cisco Catalyst 9800 Series Wireless Controllers Cisco IOS XE Cupertino 17.7.x, ensure that the IoX Application version is upgraded to Version 1.3.x

IoT Service is not supported on the following:

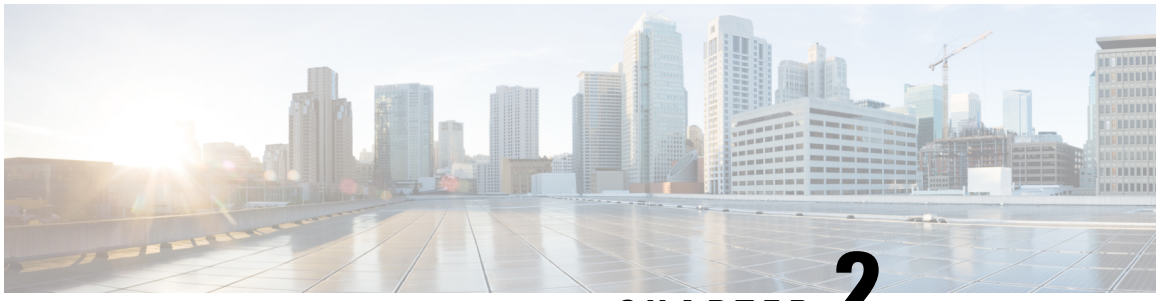
- Directly connected and CMX Tethering connectors.

The following table lists the compatibility of the Advanced BLE Gateway for BLE and the Base BLE Gateway App with various AP modes. This table is not applicable to Cisco Embedded Wireless Controller on Cisco Catalyst Access Points (Cisco EWC-AP).

Table 1: AP Modes and App Support

AP Mode	Advanced BLE Gateway App	Base BLE Gateway App
PI: Local	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported
P1: Flex	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported

AP Mode	Advanced BLE Gateway App	Base BLE Gateway App
P2: Fabric	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported
P3: Mesh	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported



CHAPTER 2

Prerequisites

Prerequisites of IoT Service (Wireless)

Cisco Spaces: Connector Scale and Size Guidance for IoT Service

This section guides you on choosing a size for the Connector based on your scale of your deployment, such as

- number of APs in your network
- the messages that the Connector may have to send, and
- and the number of devices handled.

Note

- The table below is an approximation and assumes that only two services, namely Service manager service and IoT service (wireless), are in use. Also, every deployment is different and multiple factors impact the load on the Connector.
- Ensure that you have upgraded to the latest versions of these services to achieve the numbers mentioned in the table below.

Table 2: Cisco Spaces: Connector Scale and Size Guidance for IoT Service

Connector Size	Scale
Standard Connector (2vCPU, 4 GB RAM)	The **Standard** Connector can • Support up to 500 APs. • Send up to 25,000 outbound messages per second. • Process up to 1000 BLE tags or devices.

Connector Size	Scale
Advanced1 Connector (4vCPU, 8 GB RAM)	**Advanced1** Connector can • Support up to 2500 APs • Send up to 120,000 outbound messages per second. • Process up to 10,000 BLE tags or devices.

Prerequisites

The following prerequisites can get you started with Cisco Spaces: IoT Service.

- Install Cisco Spaces: Connector in your network.
- Install a Cisco Catalyst 9800 Series Wireless Controller with a Cisco IOS XE Amsterdam 17.3.x image.
- Deploy supported APs in your network (see the Compatibility Matrix for IoT Service (Wireless) , on page 4).
- Ensure that Cisco Spaces is configured with maps either from Cisco Prime Infrastructure or Catalyst Center.
- If the Cisco Spaces: Connector is an Amazon Elastic Compute Cloud (EC2) Instance from Amazon Machine Images (AMI), ensure that the wireless controller and connector are in the same virtual private cloud (VPC). Ensure that the wireless controller has a private IP address so that the security group of connector does not block the traffic, allowing enabled IOT streams to function.
- Permit all the TCP traffic at the Virtual private clouds (VPC) level so that the Telemetry Data Logger (TDL) is established without any issues.
- Before adding a Cisco Catalyst 9800 Series Wireless Controller to a connector, run the following commands on the Catalyst 9800 controller in a sequence:
  - **aaa new-model**
  - **aaa authentication login default local**
  - **aaa authorization exec default local**

  These commands disable the connection services to Cisco Spaces.
- Cisco Spaces: IoT Service and Intelligent Capture (iCAP) feature can now co-exist on Cisco Catalyst 9800 Series Wireless Controller Cisco IOS XE Cupertino 17.7.x release and later. For releases earlier than Cisco IOS XE Cupertino 17.7.x, disable iCAP, if already enabled on the controller.
- Perform NTP synchronization over wireless controllers, a connector, and APs in the network.
- If a USB BLE module is inserted in an AP, reboot the AP.
- NETCONF must be enabled in Cisco Catalyst 9800 Series Wireless Controller in port 830, along with permission to use NETCONF.

Caution The application (app) installed and running over the AP uses the default 17.17.0.0/16 subnet. So, using this subnet for other purposes might create network issues.

- IPv6 is not supported on Cisco Spaces: Connector.
- If your require two connectors installed with 3.x to work with IoT service (wireless) and function as a high-availability pair, you must configure the connectors as Virtual IP (VIP) pair.

Access Points that support IoT Service (Wireless) are as follows:

- Cisco Catalyst 9105 Series Access Points
- Cisco Catalyst 9115 Series Access Points
- Cisco Catalyst 9117 Series Access Points
- Cisco Catalyst 9120 Series Access Points
- Cisco Catalyst 9130 Series Access Points
- Cisco Catalyst 9136 Series Access Points
- Cisco Catalyst 9162 Series Access Points
- Cisco Catalyst 9164 Series Access Points
- Cisco Catalyst 9166 Series Access Points
- Cisco Aironet 4800 Series Access Points

Compatibility Matrix for IoT Service (Wireless)

Application Name	Support for Cisco Spaces: IoT Service
Supported wireless controllers	• Supported on Cisco Catalyst 9800 Series Wireless Controllers, Release 17.3.1 and later • Not supported on Cisco AireOS Wireless Controller • Not supported on Cisco Embedded Wireless Controller on Cisco Catalyst Access Points (Cisco EWC-AP) • Supported on Catalyst 9800 Controller running on Catalyst Switches in SD-Access mode (ECA)
Cisco Spaces: Connector Docker	2.0.455 and later
Cisco Spaces: Connector OVA	2.3 and later
Cisco Prime Infrastructure	Cisco Prime Infrastructure Release 3.8 MR1 and later

Application Name	Support for Cisco Spaces: IoT Service
Catalyst Center (for map import)	Catalyst Center Release 2.1.1 and later
Access Points for advanced BLE gateway (Wi-Fi 6)	• Cisco Catalyst 9105 Series Access Points • Cisco Catalyst 9115 Series Access Points • Cisco Catalyst 9117 Series Access Points • Cisco Catalyst 9120 Series Access Points • Cisco Catalyst 9130 Series Access Points • Cisco Catalyst 9136 Series Access Points • Cisco Catalyst 9162 Series Access Points • Cisco Catalyst 9164 Series Access Points • Cisco Catalyst 9166 Series Access Points • Cisco Aironet 4800 Series Access Points
Access points for basic BLE gateway	• Cisco Aironet 1815 Series Access Points • Cisco Aironet 2800 Series Access Points (USB dongle needed. No in-built USB radio) • Cisco Aironet 3800 Series Access Points (USB dongle needed. No in-built USB radio)
Cisco IOx App Version	1.0.46 and later **Note** For Cisco Catalyst 9800 Series Wireless Controllers Cisco IOS XE Cupertino 17.7.x, ensure that the IoX Application version is upgraded to Version 1.3.x

IoT Service is not supported on the following:

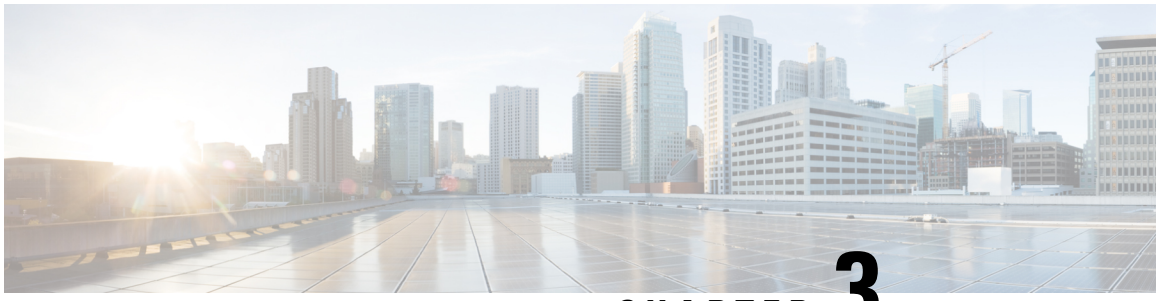
- Directly connected and CMX Tethering connectors.

The following table lists the compatibility of the Advanced BLE Gateway for BLE and the Base BLE Gateway App with various AP modes. This table is not applicable to Cisco Embedded Wireless Controller on Cisco Catalyst Access Points (Cisco EWC-AP).

Table 3: AP Modes and App Support

AP Mode	Advanced BLE Gateway App	Base BLE Gateway App
PI: Local	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported

AP Mode	Advanced BLE Gateway App	Base BLE Gateway App
P1: Flex	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported
P2: Fabric	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported
P3: Mesh	• 11-AX: Supported • Wave2: Not supported	• 11-AX: Supported • Wave2: Supported



CHAPTER 3

Open Ports

Information About Open Ports (Wireless)

This chapter lists the connector ports that need to be open for the proper functioning of various services or protocols.

The following ports need to be opened to allow for the basic functionality of Cisco Spaces.

Figure 2: Basic Functionality

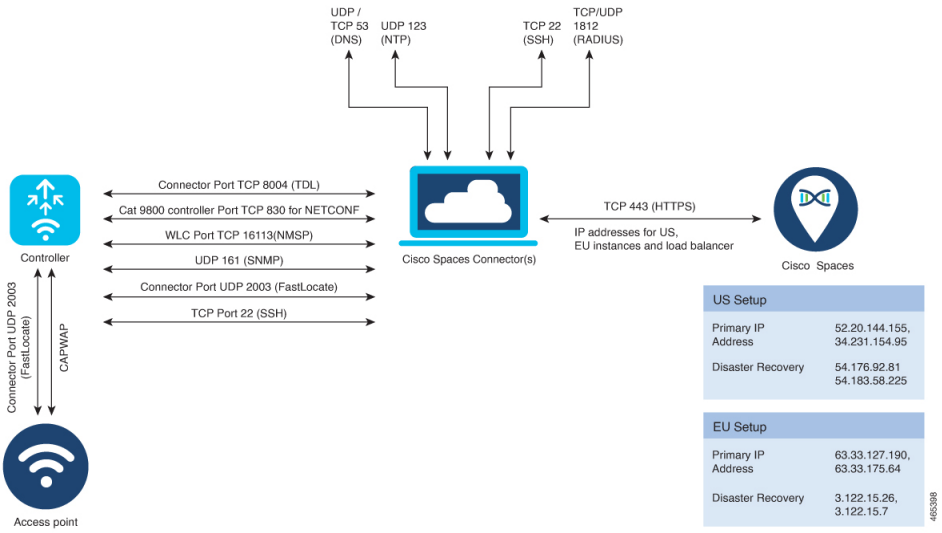


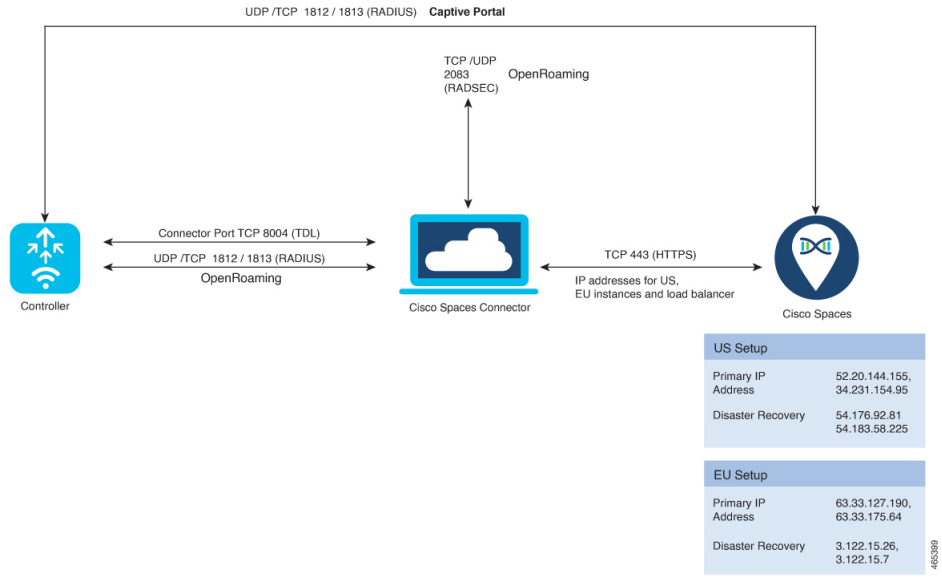
Table 4: Setups

Setup Type	Primary IP Address	Disaster Recovery
US Setup	52.20.144.155 34.231.154.95	54.176.92.81 54.183.58.225

Setup Type	Primary IP Address	Disaster Recovery
EU Setup	63.33.127.190 63.33.175.64	3.122.15.26 3.122.15.7
Singapore Setup (SG)	13.228.159.49 54.179.105.241	13.214.251.223 54.255.57.46

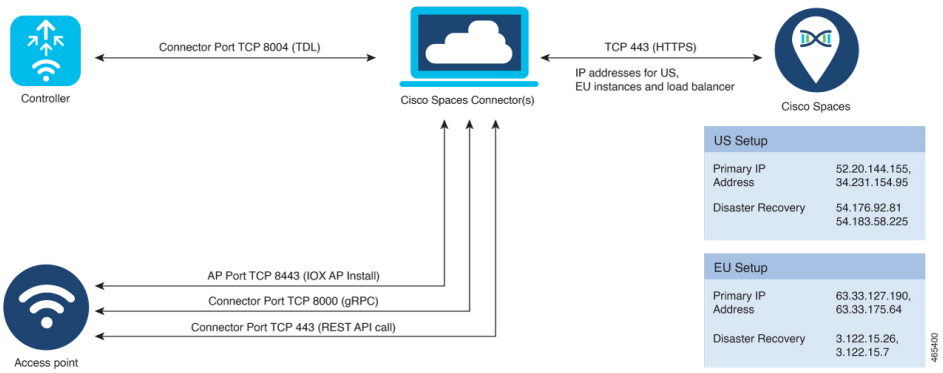
In addition to basic functionality, additional ports need to be opened for other additional functionality like guest onboarding and IoT Services.

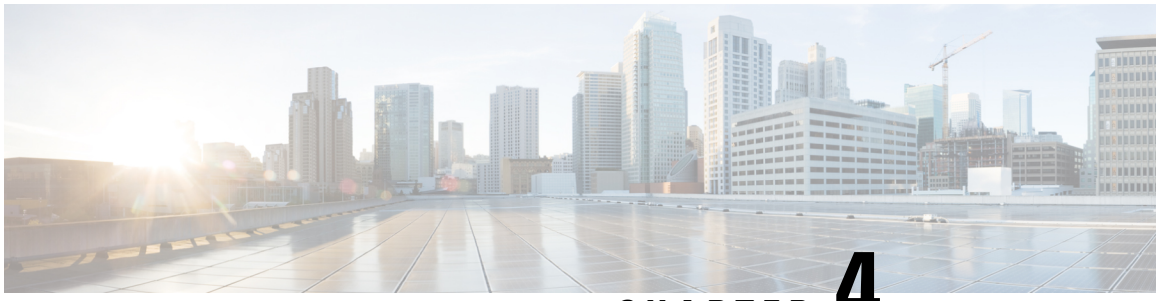
Figure 3: Guest Onboarding



The following ports need to be opened for configuring IoT Services (wireless). To configure IoT Services (wired), see Open Ports (Wired)

Figure 4: IoT Services





CHAPTER 4

Getting Started

Activate IoT Service (Wireless)

This task shows you how to activate IoT service (wireless) on some or all your devices, from the Cisco Spaces dashboard.

Before you begin

To activate IoT service (wireless), your network must meet the below prerequisites :

- Cisco Spaces: Connector
- Cisco Catalyst 9800 Series Wireless Controllers, installed with version 17.3.1 or higher
- Supported access points. See Prerequisites of IoT Service (Wireless)

Note

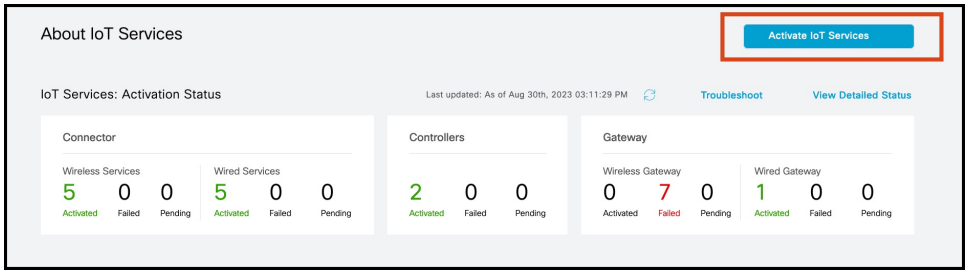
- This workflow is applicable only for Connector Release 3. We recommend that you upgrade from Connector 2.x for smooth functioning of your services. If it is absolutely essential to enable IoT service (wireless) on Connector 2.x, open a support case.
- The workflow initiated by this procedure automatically checks for prerequisites necessary to complete this task.

Step 1 Log in to Cisco Spaces.

Step 2 From the left navigation pane, click **IoT Services > About IoT Services**.

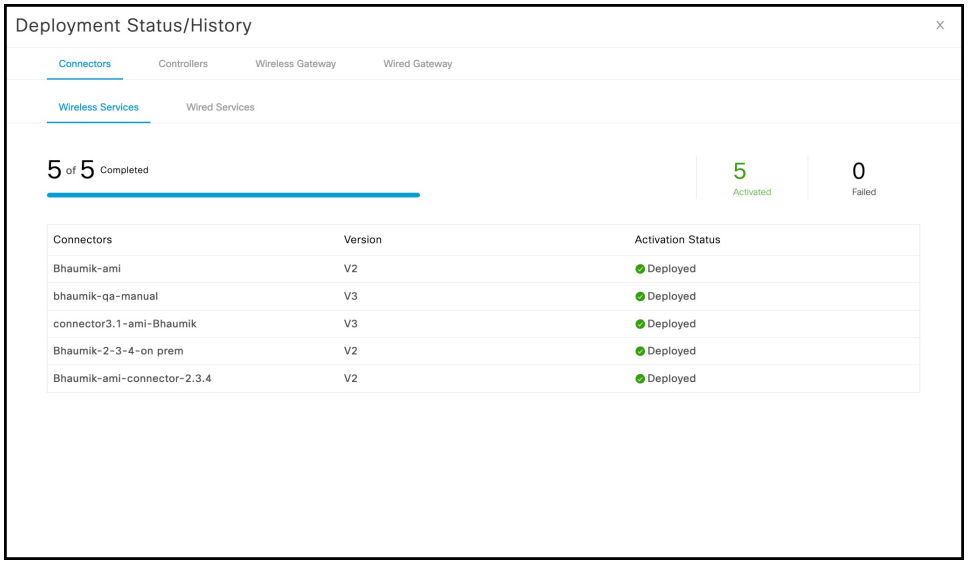
You can see the number of connectors activated with the IoT service (wireless) service. You can also see the number of APs deployed as an IoT service (wireless) gateway.

Figure 5: About IoT Services



Click **View Detailed Status** to see the breakdown of the activation status of various individual devices.

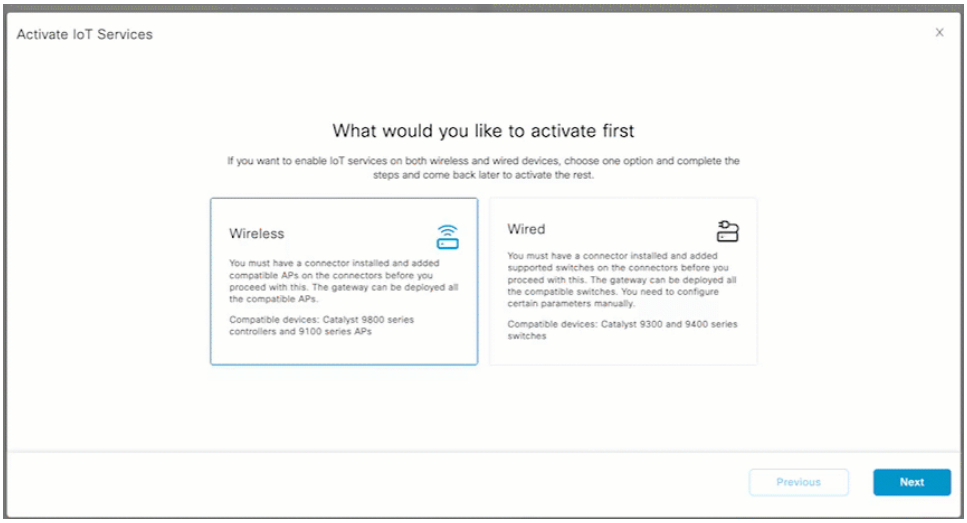
Figure 6: Detailed Status of Devices Activated With IoT Service (Wireless)



Step 3 In the **About IoT Services** window top-right corner, click **Activate IoT Services**.

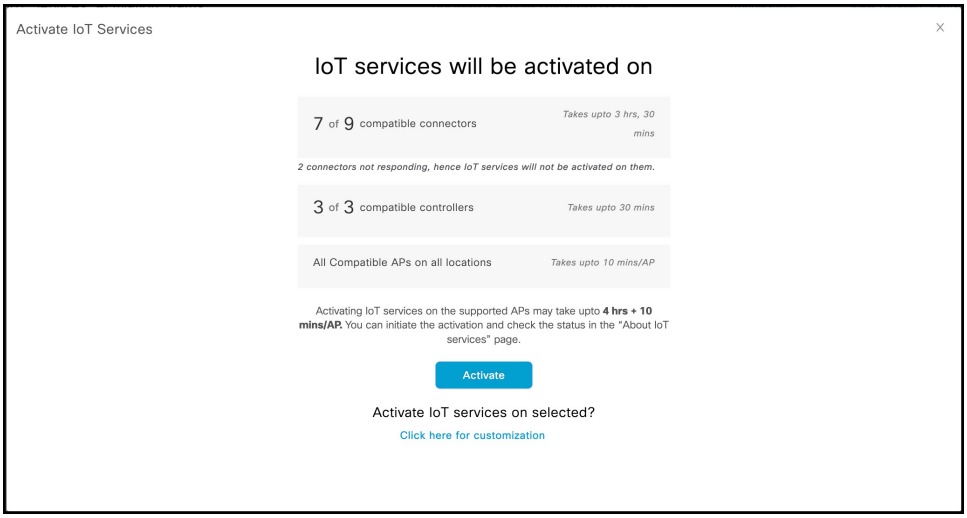
Step 4 In the **Activate IoT Services** window that is displayed, choose **Wireless**.

Figure 7: Activate IoT Service (Wireless)



You can see the list of all devices on which IoT service (wireless) can be activated, along with the activation time.

Figure 8: List of Supported Devices

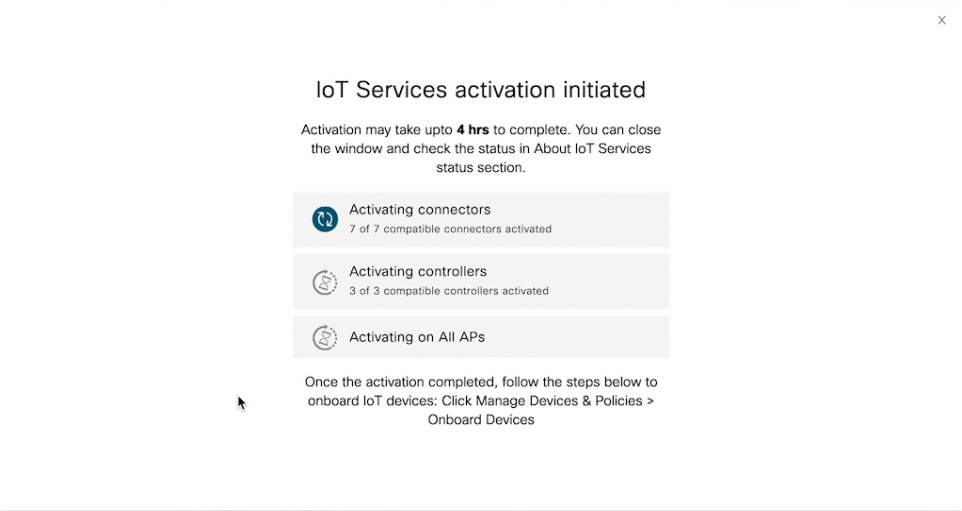


Step 5 To activate IoT service (wireless) on all devices on your network, in the **IoT services will be activated on** window, click **Activate**.

This activation of IoT service (wireless) automates the following tasks:

- Enables IoT streams on the connector
- Enables the wireless controller stream
- Configures APs as a Bluetooth Low Energy (BLE) gateway (this includes turning on the BLE radio, BLE scanning, and deploying the BLE gateway app)

Figure 9: Activate IoT Service (Wireless) on all devices



Step 6 To activate IoT service (wireless) only on specific devices of your network, do the following:

a) Choose one or more connectors to activate IoT service (wireless).

b) To activate the wireless gateway, click **Activate Wireless**.

c) In the **Deploy Wireless Gateway** window, select the APs on which you want to activate IoT service (wireless).

Figure 10: Activate IoT Service (Wireless) on Preferred Devices

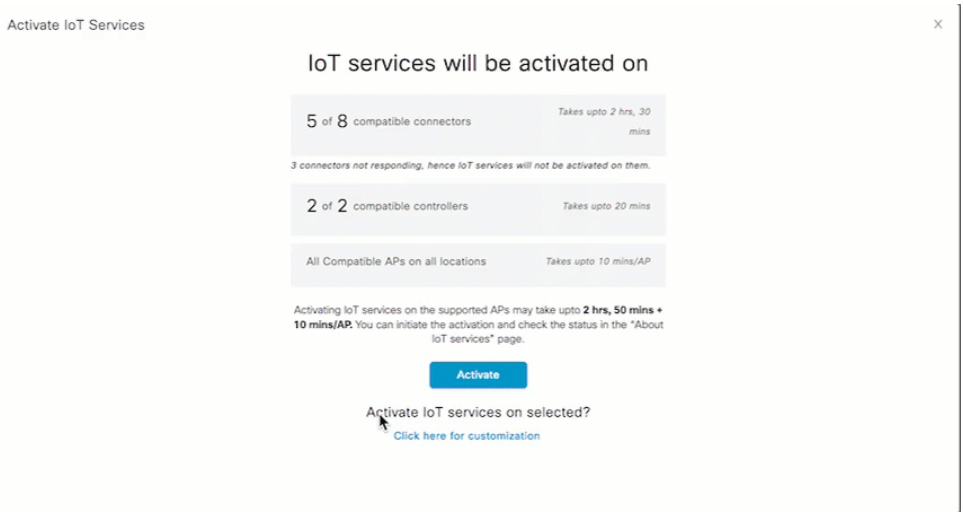
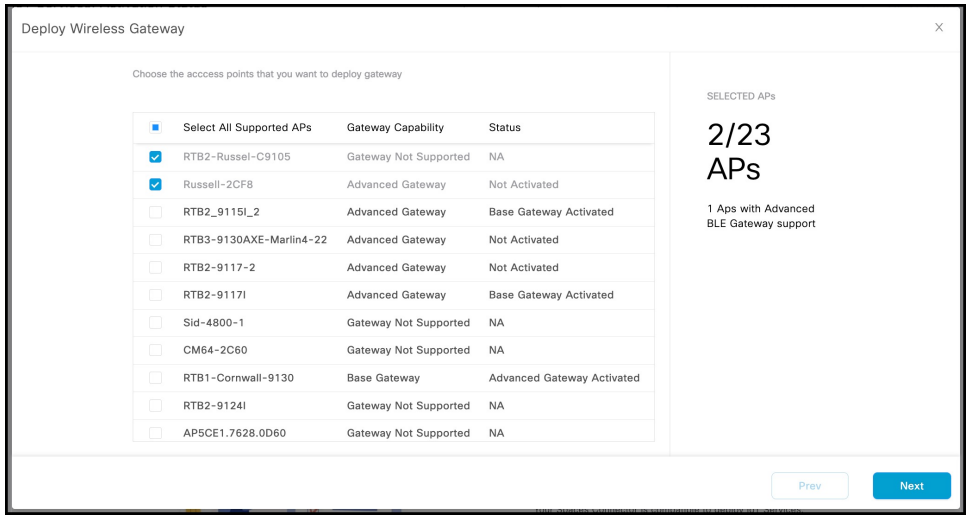


Figure 11: Activate IoT Service (Wireless) on Preferred Devices

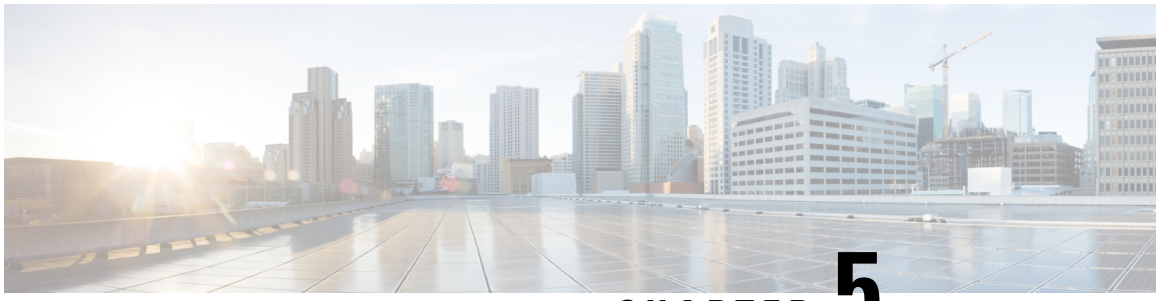


What to do next

Once the activation completed, you can onboard the IoT Service (Wireless) devices. Click **Manage Devices & Policies > Onboard Devices**.

PART II

Configuration



CHAPTER 5

AP as a Beacon

AP as a Beacon

You can configure your access point (AP) to act as a beacon (AP beacons) by enabling BLE on it.

IoT Service categorizes APs according to their configurations as the following:

- **Disabled:** APs with BLE disabled. These APs are not scanning or transmitting.
- **Scan Mode:** AP beacons that are only scanning.
- **Transmit Mode:** AP beacons configured in one of the beacon transmit profiles. You can configure up to five iBeacons in this mode.
  - The MAC address advertised in the iBeacon payload is derived from the radio MAC address of the AP. (iBeacon MAC address).
  - The MAC address advertised in the Eddystone payload is the default MAC address of the AP's BLE chip, which is preset by the chip vendor.
- **Dual Mode:** AP beacons that are transmitting and scanning. You can configure only one iBeacon in this mode.
  - The MAC address advertised in this mode is the default MAC address of the AP's BLE chip, which is preset by the chip vendor (For both Eddystone and iBeacon single advertisement profiles)
- **Needs Config Change:** AP's that have an error in configuration. You can configure these APs in Scan Mode, Dual Mode, or the Transmit Mode.

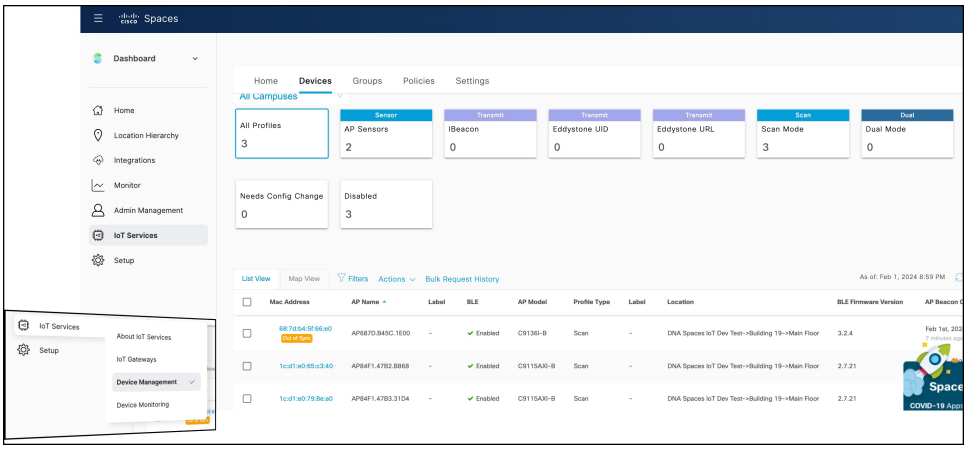
You can configure an AP Beacon in one of the following transmit modes.

- iBeacon
- Eddystone UID

- Eddystone URL

You can also see all the APs irrespective of their configurations under **All Profiles**.

Figure 12: Various Profiles of AP Beacons



You can also enable telemetry on the AP beacon and collect sensor information.

iBeacon Transmit Mode

A single AP can support up to five iBeacons in the transmit mode. Each iBeacon has a unique address derived from the base radio MAC address of the AP.

Use Cisco Spaces to configure an iBeacon's payload.

Following are some terms related to iBeacons:

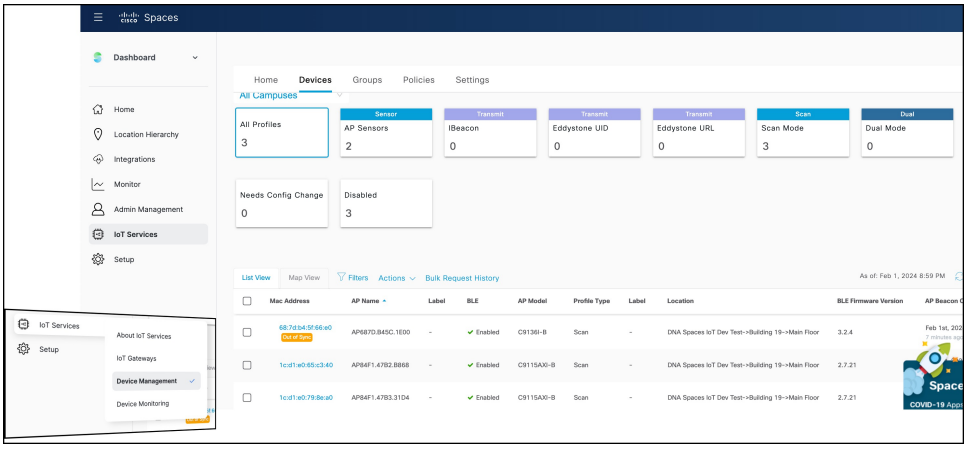
- **Transmit mode**: Mode that allows nearby devices to pick up an iBeacon's broadcasting (or 'advertising') signals.
- **Advertisement payload**: Data broadcast by an iBeacon. The advertisement payload contains information relevant to the iBeacon's purpose, such as the iBeacon's location. Use Cisco Spaces to configure this payload.
- **iBeacon MAC address**: Unique identifier of an iBeacon on the network that helps other devices recognize and differentiate one iBeacon from another. This address is part of the iBeacons' advertisement payload. The AP uses the AP's own base radio MAC address to derive this unique address. The address is derived by adding a predefined address block value to the last byte of the base radio MAC address and decrementing this value by the beacon ID.

Configure AP as a Beacon in Scan Mode

You can configure an AP as a beacon in the scan mode.

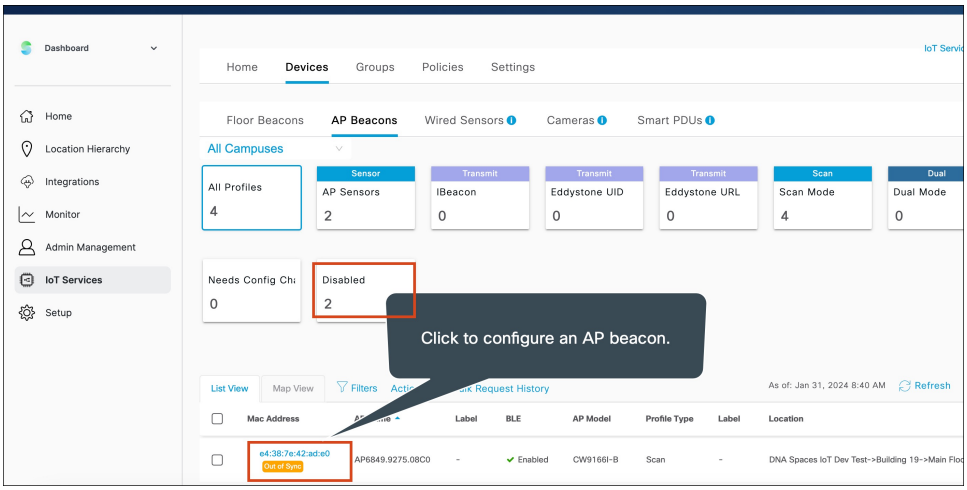
Step 1 In the Cisco Spaces dashboard left-navigation pane, click **IoT Service > Device Management > Devices**, and then click **AP Beacons**.

Figure 13: List of AP Beacons



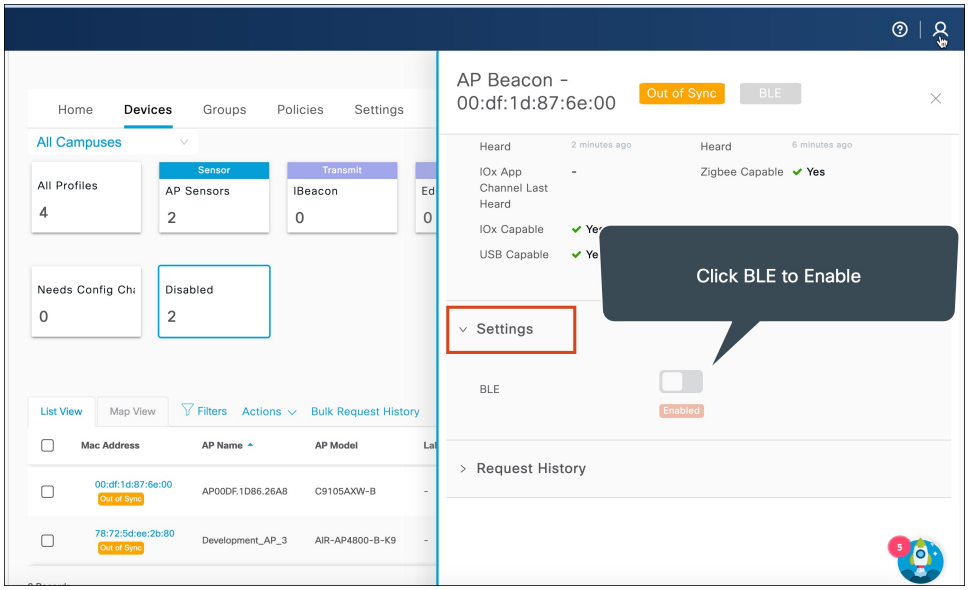
Step 2 Click the **Disabled** tab, if the count is greater than zero. Click the MAC address of one of the listed APs to open a detailed view.

Figure 14: Select an AP to Configure



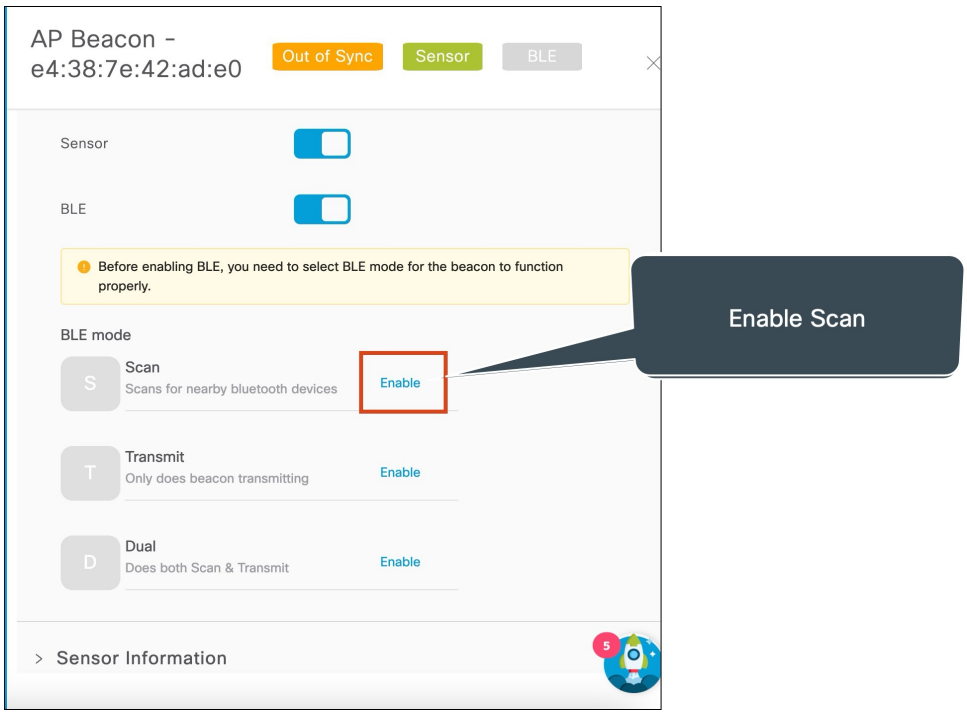
Step 3 In the **Settings** area, click **BLE**.

Figure 15: Enable BLE



Step 4 In the **BLE mode** area for the **Scan** option, click **Enable**.

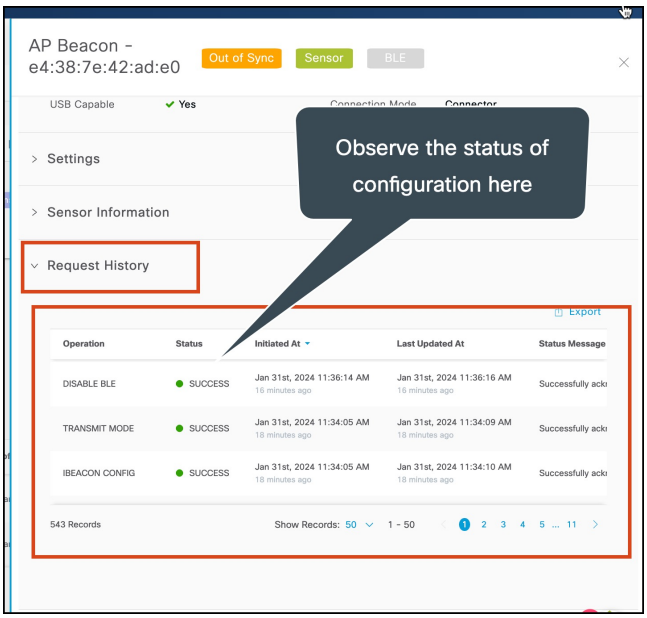
Figure 16: Enable Scan Mode



AP is enabled as a beacon in **Scan** mode. You can observe the AP under the **Scan** tab.

Step 5 From the **Request History** area, observe the status of the configuration change you requested. On the **AP Beacons** page, notice that the AP now has an **Out of Sync** message beside it. This message disappears once the configuration requested is complete.

Figure 17: Configuration Status

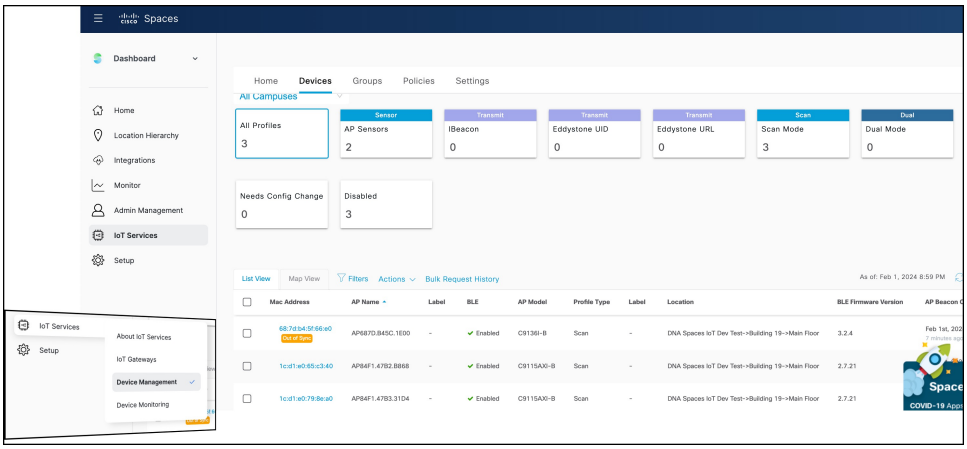


Configure AP as a Beacon in Transmit Mode

You can configure an AP as a beacon in transmit mode.

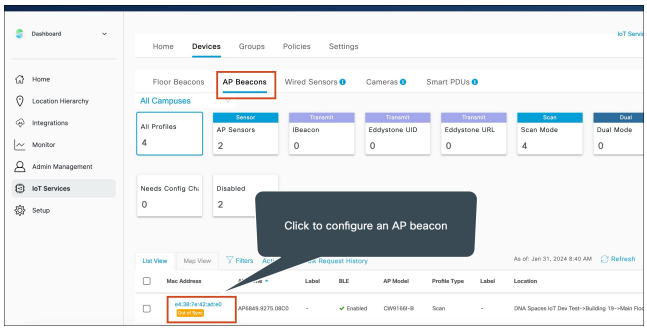
Step 1 In the Cisco Spaces dashboard left-navigation pane, click **IoT Service > Device Management > Devices**, and then click **AP Beacons**.

Figure 18: List of AP Beacons



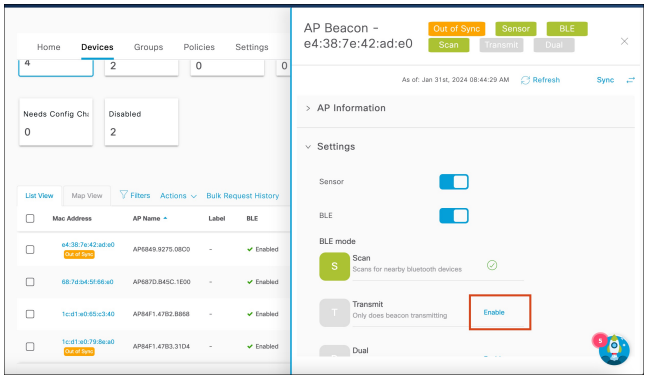
Step 2 Click the **AP Beacons** tab. Click the MAC address of one of the listed APs to open a detailed view.

Figure 19: Select an AP to Configure



Step 3 In the **BLE mode** area for the **Transmit** option, click **Enable**.

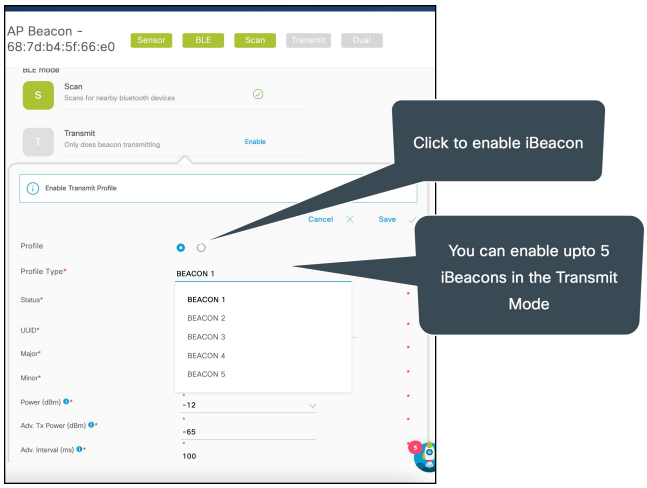
Figure 20: Enable BLE



Step 4 In the **Enable Transmit Profile** area, you can configure this beacon in two modes. Do one of the following:

- Check the first checkbox to enable iBeacon. From the **Profile Type** drop-down, choose one of the beacons. Configure the remaining values for the iBeacon's payload.

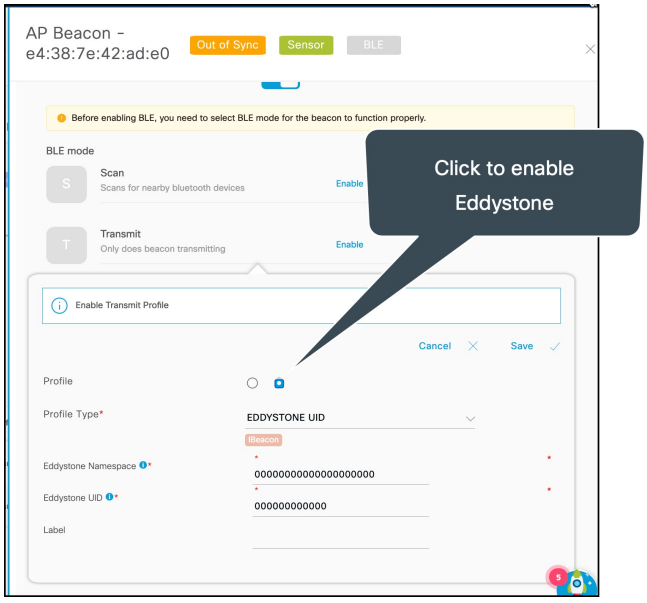
Figure 21: Configuring an AP as an iBeacon



Note APs can support up to five iBeacons in the **Transmit** mode. For more information, see iBeacon Transmit Mode, on page 24

- Select the second checkbox to enable Eddystone. Configure the values for the Eddystone payload.

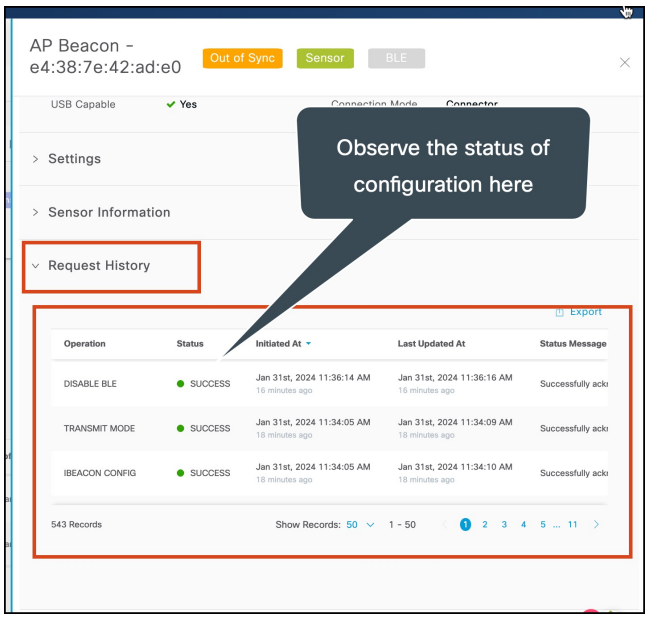
Figure 22: Configure an AP Beacon as an Eddystone



AP is enabled as a beacon in **Transmit** mode. You can observe the AP under the **Transmit** tab.

Step 5 From the **Request History** area, observe the status of the configuration change you requested. On the **AP Beacons** page, notice that the AP now has an **Out of Sync** message beside it. This message disappears once the configuration requested is complete.

Figure 23: Configuration Status

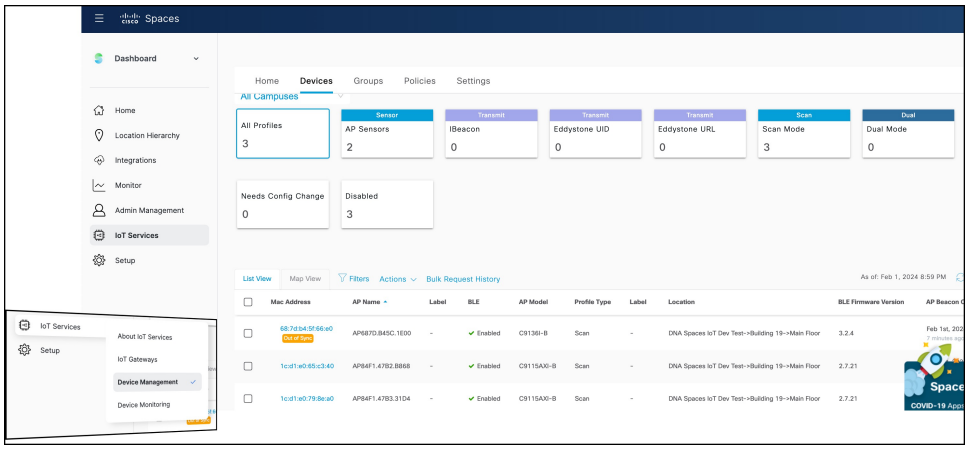


Configure AP as a Beacon in Dual Mode

You can configure an AP as a beacon in dual mode.

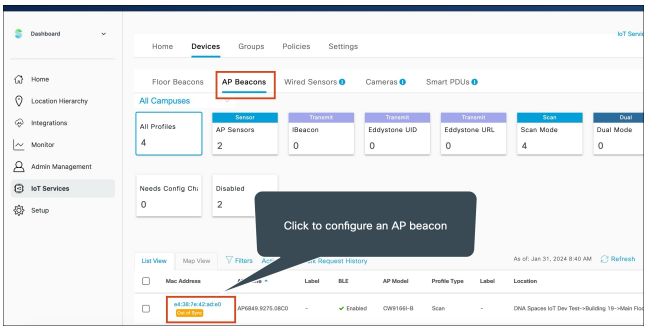
Step 1 In the Cisco Spaces dashboard left-navigation pane, click **IoT Service > Device Management > Devices**, and then click **AP Beacons**.

Figure 24: List of AP Beacons



Step 2 Click the **AP Beacons** tab. Click the MAC address of one of the listed APs to open a detailed view.

Figure 25: Select an AP to Configure

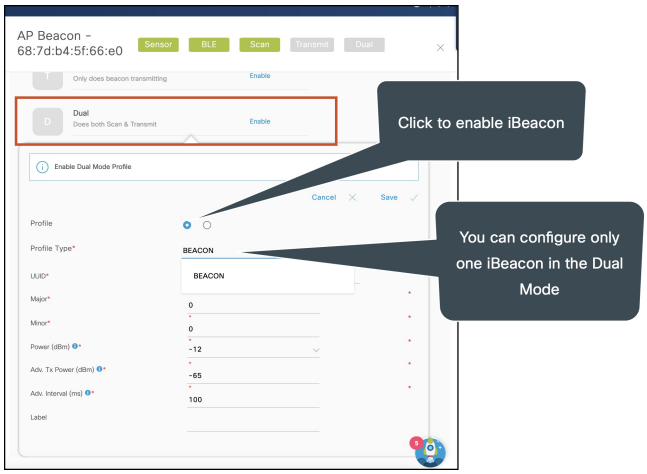


Step 3 In the **BLE mode** area for the **Dual** option, click **Enable**.

Step 4 In the **Enable Transmit Profile** area, you can configure this beacon in two modes. Do one of the following:

- Check the first checkbox to enable iBeacon. Configure the remaining values for the iBeacon's payload.

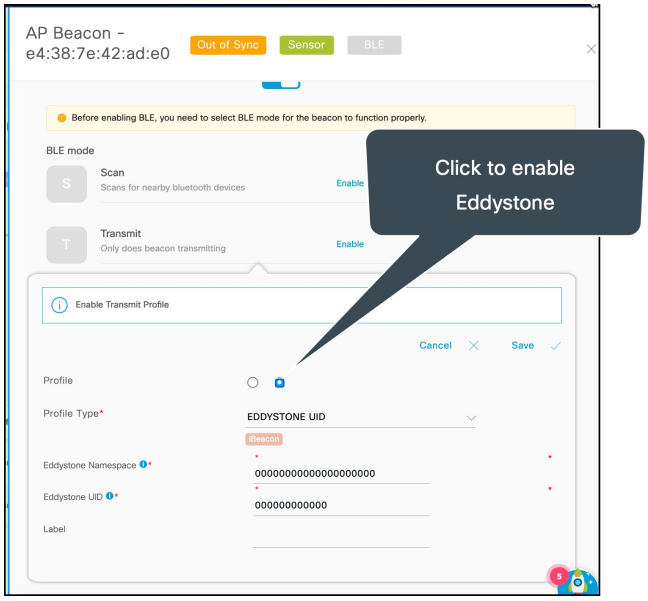
Figure 26: Configuring an AP as an iBeacon



Note APs can support only one iBeacon in the **Dual** mode. For more information, see iBeacon Transmit Mode, on page 24

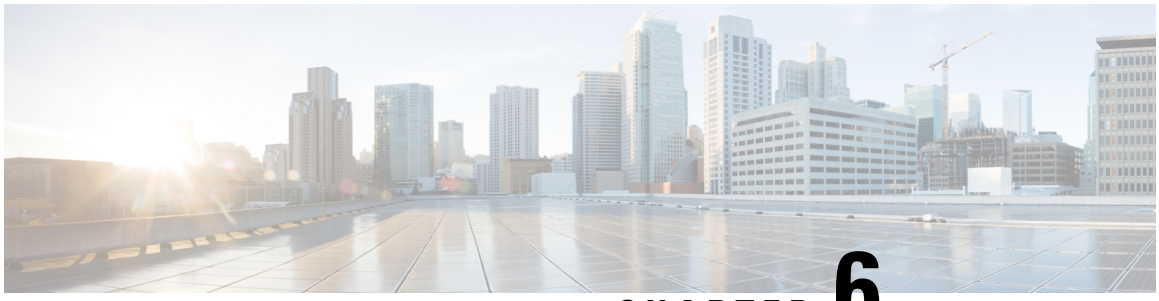
- Select the second checkbox to enable Eddystone. Configure the values for the Eddystone payload.

Figure 27: Configure an AP Beacon as an Eddystone



AP is enabled as a beacon in **Dual** mode. You can observe the AP under the **Dual** tab.

Step 5 From the **Request History** area, observe the status of the configuration change you requested. On the **AP Beacons** page, notice that the AP now has an **Out of Sync** message beside it. This message disappears once the configuration requested is complete.



CHAPTER 6

AP as a Gateway

Access Point as a BLE Gateway

Depending on the type of Cisco access points (AP), you can configure an AP as one of the following types of Bluetooth Low Energy (BLE) gateways:

- **Base BLE Gateway**: The Base BLE gateway is a type of AP that you can configure in different modes (Transmit, Scan, or Dual).
- **Advanced BLE Gateway:** The advanced BLE gateway is an AP that is installed with an IoX Application. Using the installed IoX Application, you can configure floor beacons on the Cisco-partnered Device Manager website.

You can configure this AP (which is now a base or advanced gateway) in **Scan** mode, **Transmit** mode, or **Dual** mode. In the **Transmit** mode or **Dual**, mode the AP can broadcast iBeacon, Eddystone URL, and Eddystone UID profiles.

In the **Scan** mode, the AP can scan the vicinity for other BLE devices. Using gRPC on the AP, the AP sends the scanned data to Cisco Spaces: Connector. The AP can also receive telemetry data from floor beacons. The IoT Service dashboard decodes and displays this information.

Configure an AP as a Bluetooth Low Energy (BLE) Gateway

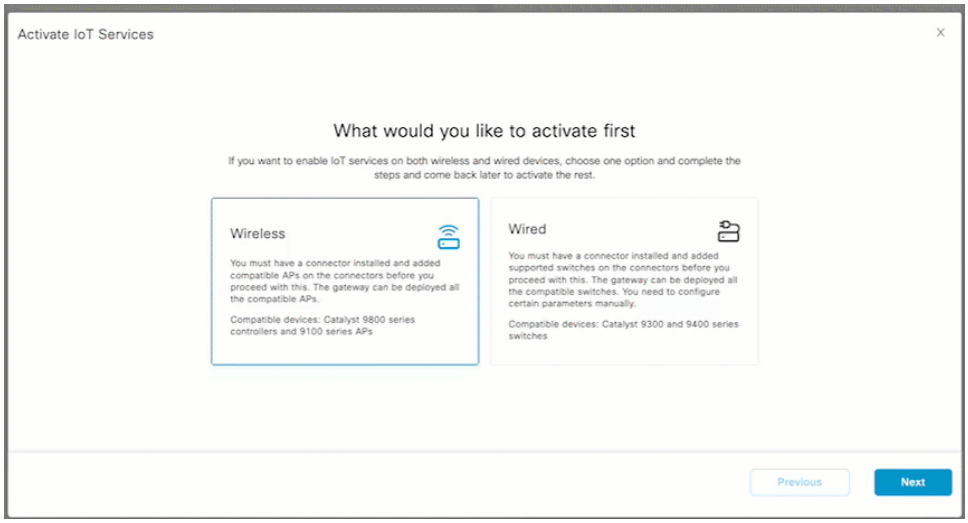
This task enables an access point (AP) to act as a BLE gateway. For more information, see Access Point as a Gateway.

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > IoT Gateways > AP Gateway**.

Step 2 Click **Add New Gateways**.

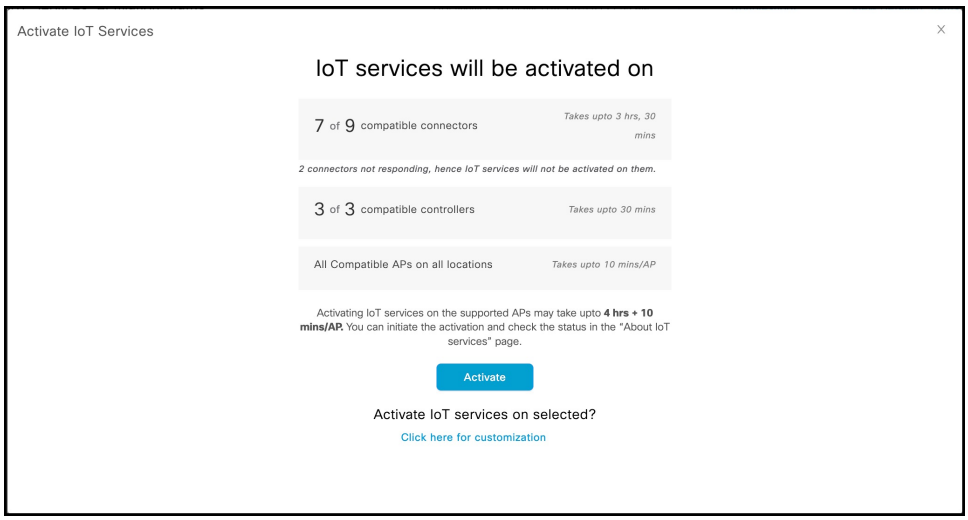
Step 3 In the **Activate IoT Services** window that is displayed, choose **Wireless**.

Figure 28: Activate IoT Service (Wireless)



You can see the list of all devices on which IoT service (wireless) can be activated, along with the activation time.

Figure 29: List of Supported Devices

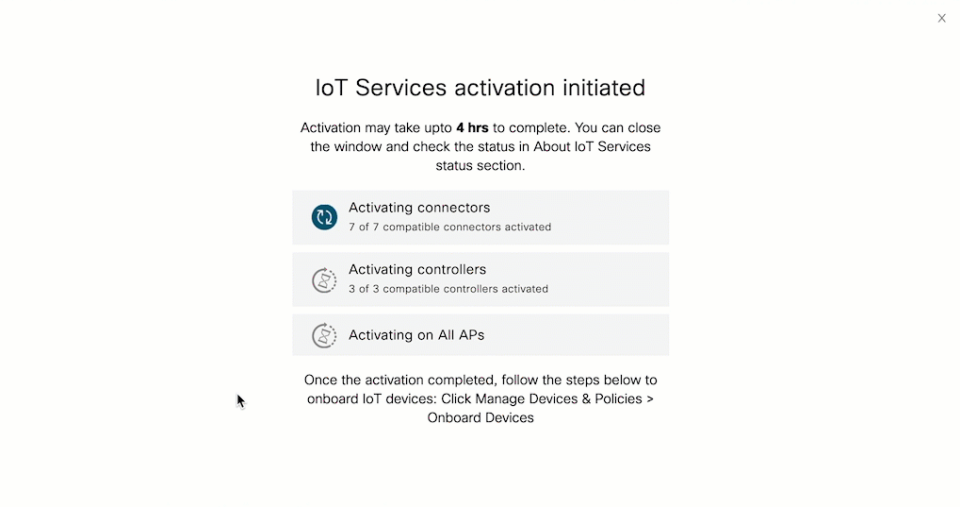


Step 4 To activate IoT service (wireless) on all devices on your network, in the **IoT services will be activated on** window, click **Activate**.

This activation of IoT service (wireless) automates the following tasks:

- Enables IoT streams on the connector
- Enables the wireless controller stream
- Configures APs as a Bluetooth Low Energy (BLE) gateway (this includes turning on the BLE radio, BLE scanning, and deploying the BLE gateway app)

Figure 30: Activate IoT Service (Wireless) on all devices



Step 5 To activate IoT service (wireless) only on specific devices of your network, do the following:

a) Choose one or more connectors to activate IoT service (wireless).

b) To activate the wireless gateway, click **Activate Wireless**.

c) In the **Deploy Wireless Gateway** window, select the APs on which you want to activate IoT service (wireless).

Figure 31: Activate IoT Service (Wireless) on Preferred Devices

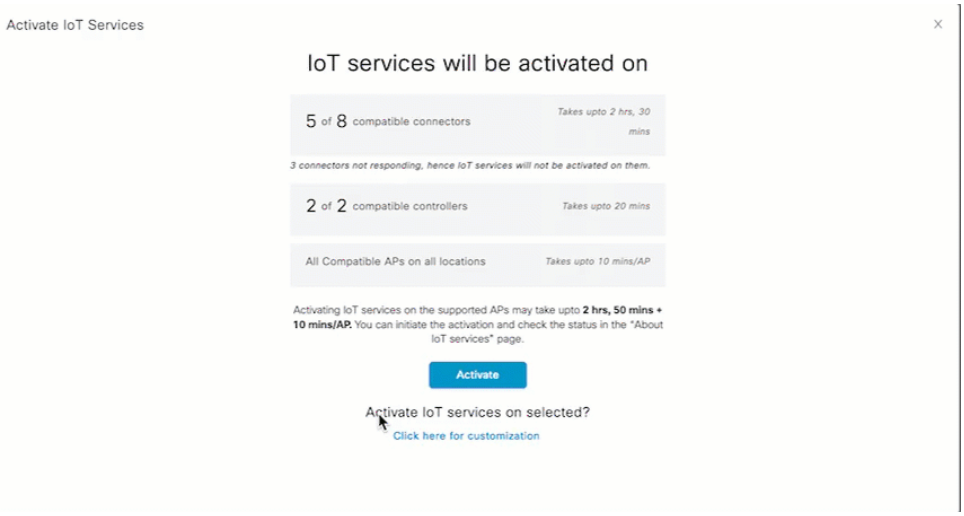
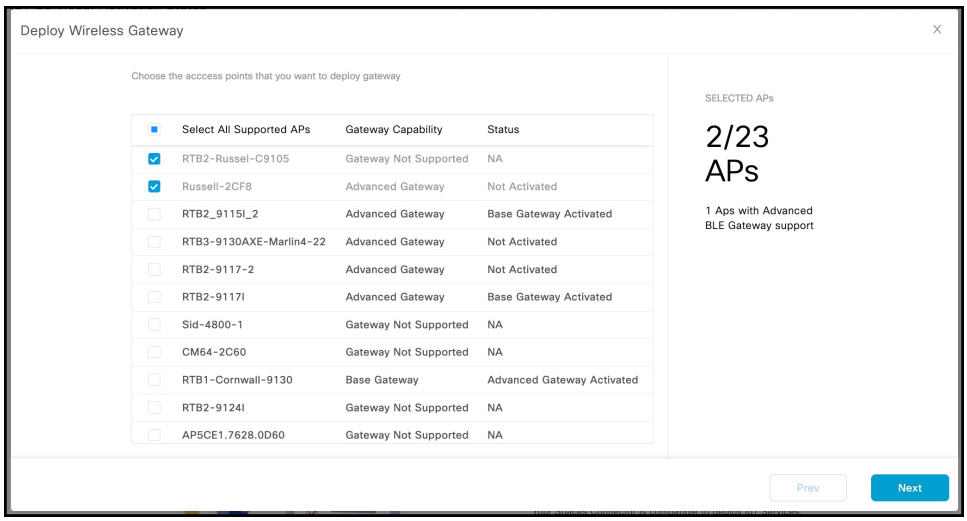


Figure 32: Activate IoT Service (Wireless) on Preferred Devices



What to do next

Once the activation completed, you can onboard the IoT Service (Wireless) devices. Click **Manage Devices & Policies > Onboard Devices**.

Uninstall or Upgrade an IOx Application on an Advanced Gateway

You can uninstall or upgrade IOx applications on advanced gateways. The Cisco Spaces: BLE Management is one such application.

Before you begin

Ensure that you have configured an access point (AP) as an advanced gateway.

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > IoT Gateways > AP Gateways** and click **All APs.**

Step 2 Click the MAC address of the AP to open the detailed **AP** page.

Step 3 In the **App Management** section, you can see the applications available for un-installation or upgrade. Do one of the following:

- To uninstall, click the uninstall icon near Cisco Spaces: BLE Management.
- To upgrade, check if a version is available for upgrade near the Cisco Spaces: BLE Management and click it.
- To upload tech-support files to the connector, click the gear icon.

Figure 33: Uninstall or Upgrade Cisco Spaces: BLE Management

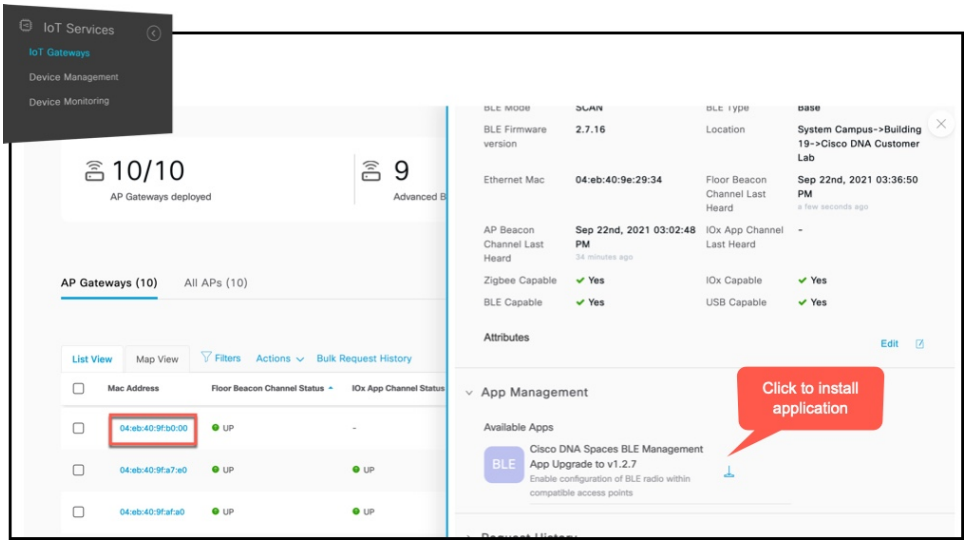


Figure 34: Uninstall Cisco Spaces: BLE Management

A gear icon appears beside the application that allows you to upload log files to connector. You can also download these files to assist a technical support team.

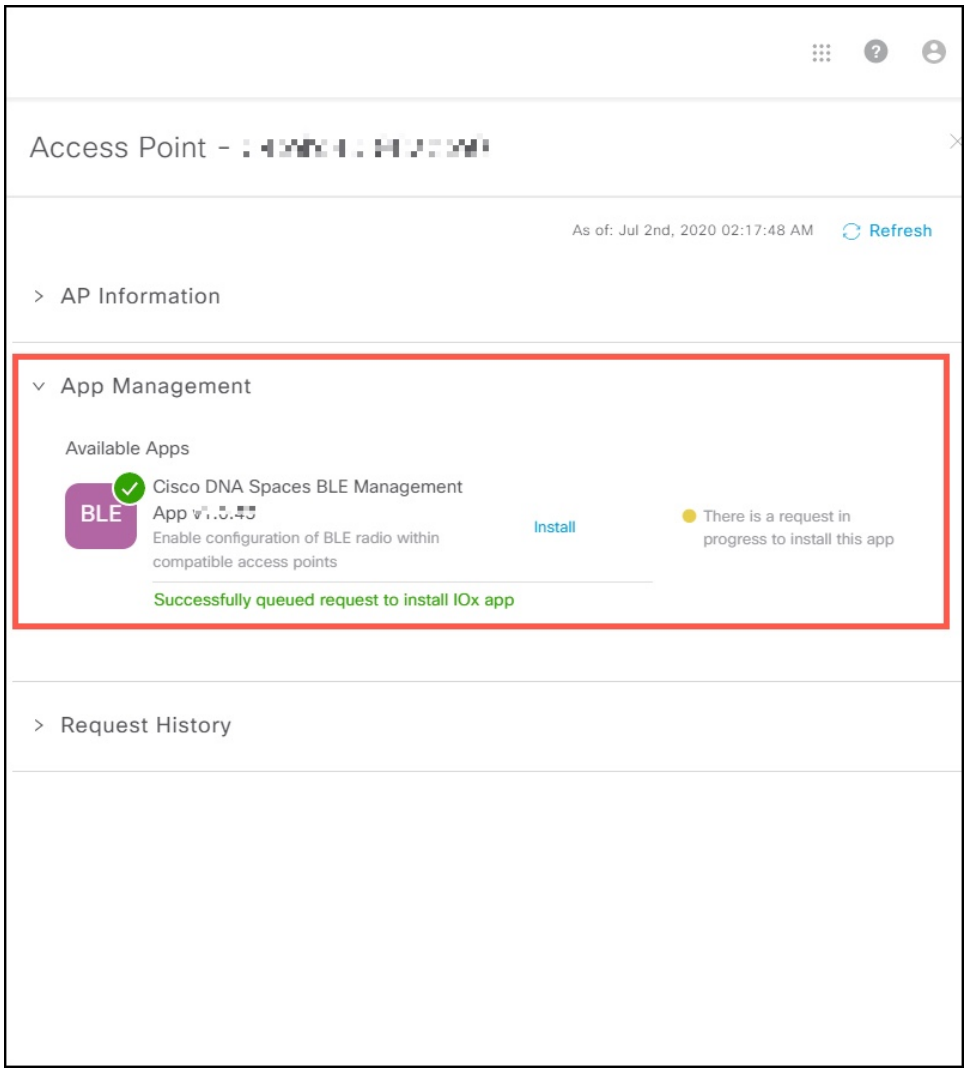
Figure 35: Technical Support Log Files

Step 4 Enter the credentials needed for authentication on the AP.

Note The authentication request to the APs includes these credentials, after which IoT Service does not retain these credentials.

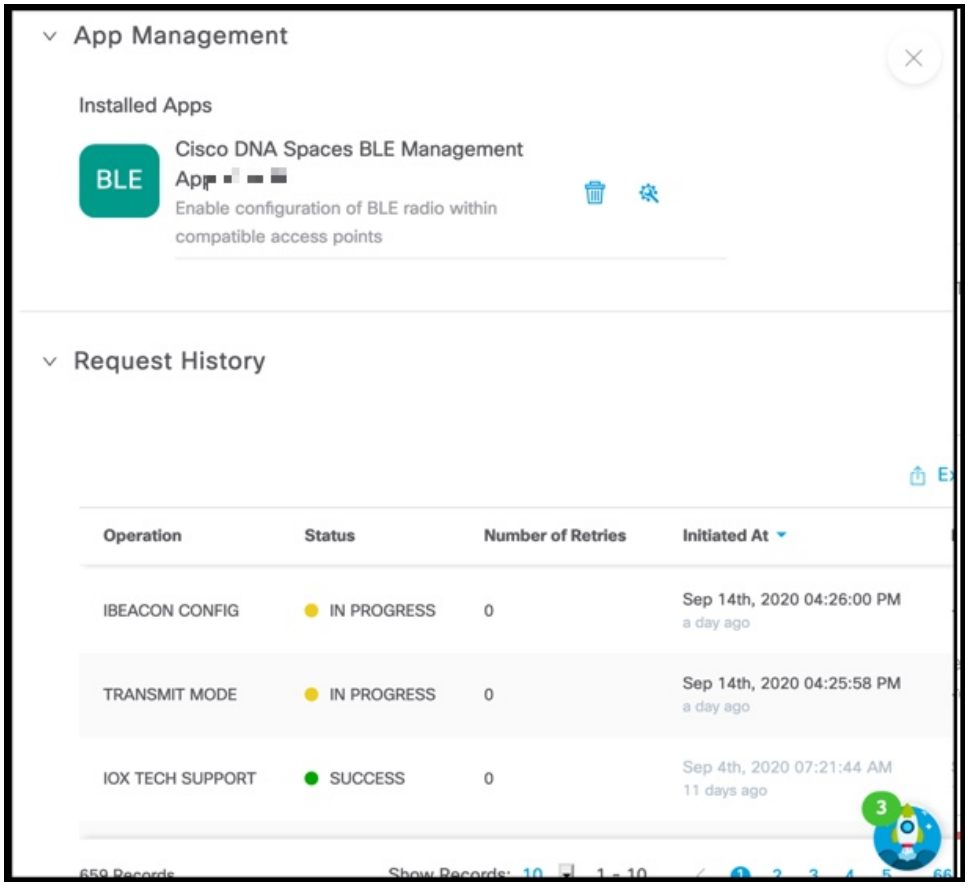
The AP which is the advanced gateway receives these change requests. You can observe the progress on the displayed page.

Figure 36: App Management: Progress of Uninstall or Upgrade



You can also check the status of deployment by clicking **Request History**.

Figure 37: Uninstall or Upgrade Status in the Request History Area



The **Status** column shows the status of Uninstall or Upgrade on each AP.

- SUCCESS: Uninstall or Upgrade of application on the AP was a success.
- FAILURE: Uninstall or Upgrade of application on the AP was a failure.
- IN PROGRESS: Uninstall or Upgrade of application on the AP is still in progress.

You can also check the status of AP gateway deployment by clicking the **Deployment status** icon in the top-right corner of the dashboard (in the **AP Gateways** page). Here you can see the deployment status of a base or advanced gateway at a more detailed level. You can see whether the gateway is enabled, whether it is in the scan or transmit mode, whether configurations are being pushed on to the gateway, or if the gateway is capable, or the status of IOX installation. Unlike bulk history, here you can view the details of an individual AP gateway. If the gateway deployment fails, the reasons are listed here.

Figure 38: Deployment Status

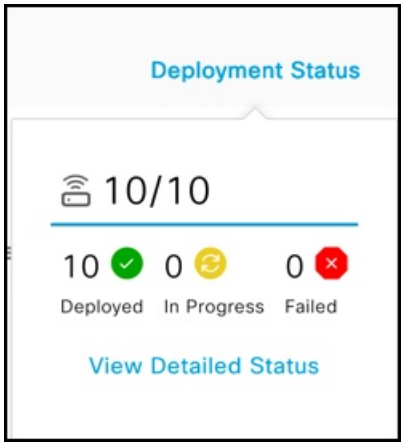
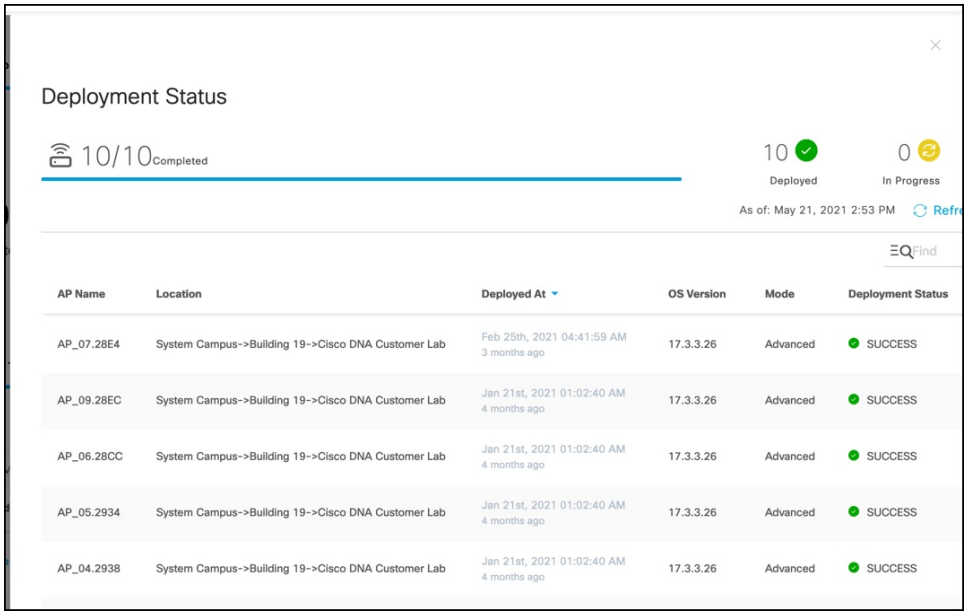
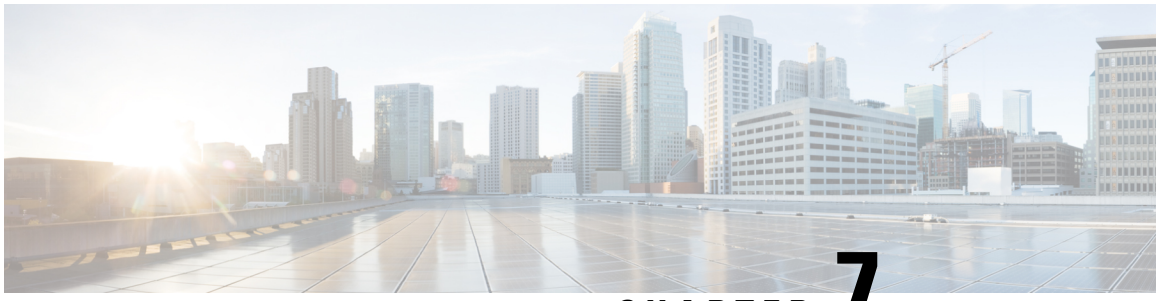


Figure 39: Deployment Status





CHAPTER 7

Beacons and Tags

Discover Beacons

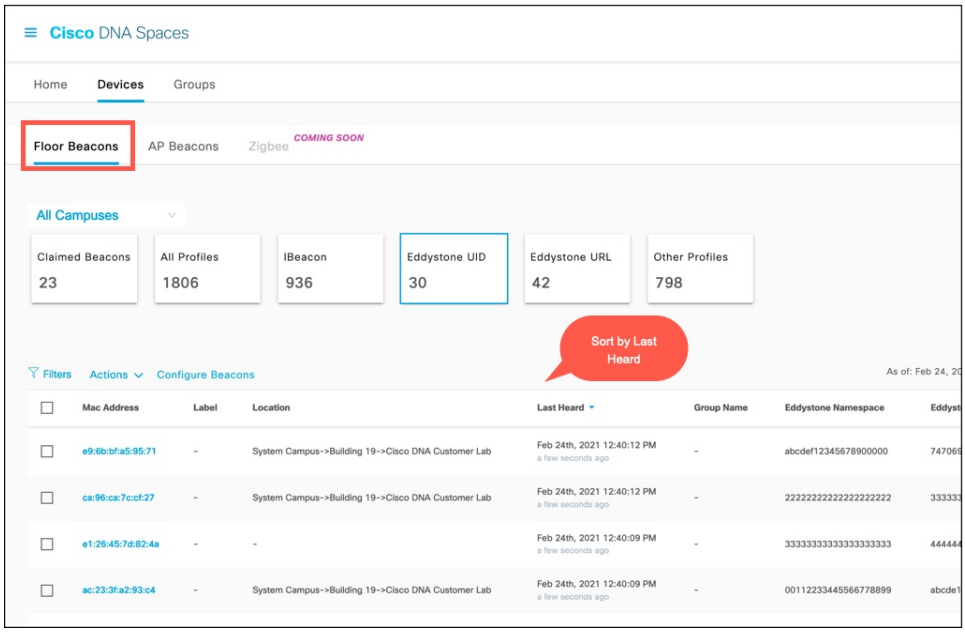
This section shows you how to view the beacons scanned by IoT Service.

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > Device Management > Devices**.

Step 2 Click on **Floor Beacons** to view scanned beacons. Click on one of the following: **All Profiles, iBeacon**, **Eddystone UID**, **Eddystone URL**, **Other Profiles.**

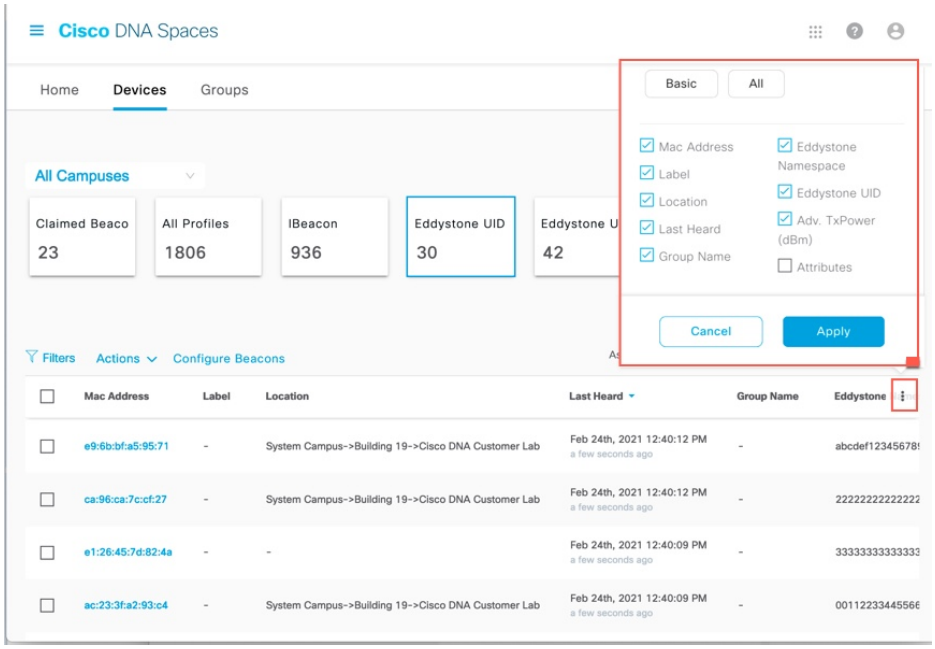
This list is sorted by **Last Heard** by default. You can sort the table by other fields by clicking the arrow beside the column header.

Figure 40: Beacon Details



Step 3 Add or delete columns using the three dots on the right.

Figure 41: Adding or Deleting Columns



Step 4 Click on the MAC address of the beacon to view further details.

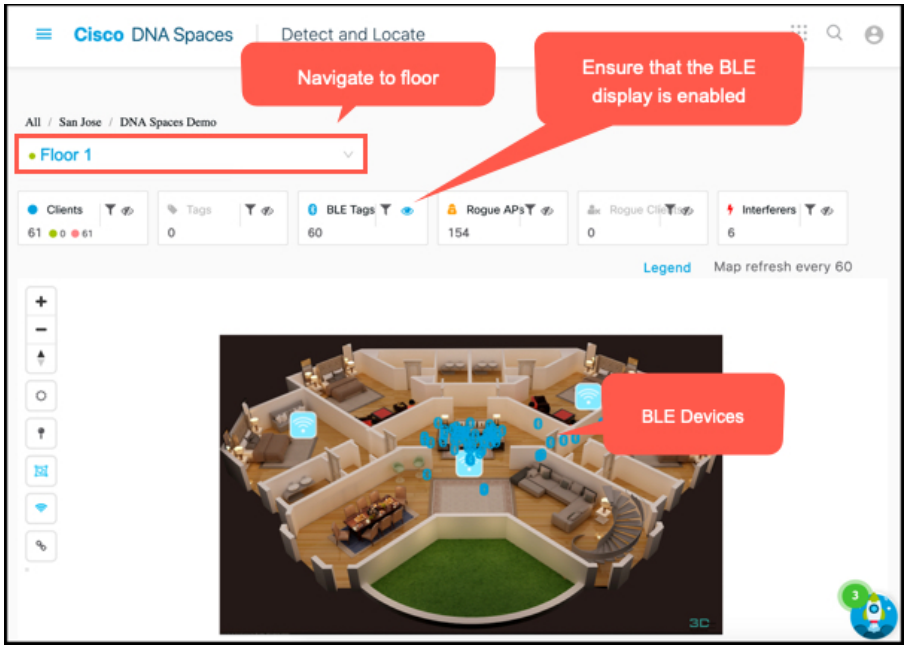
Figure 42: Beacon Details

What to do next

You can view location details of the beacon on Cisco Spaces: Detect and Locate.

Figure 43: Cisco Spaces: Detect and Locate

Figure 44: Cisco Spaces: Detect and Locate



For more information, see Cisco DNA Spaces: Detect and Locate Configuration Guide.

Claiming a Beacon

When you claim a beacon, your IoT Service account claims ownership of the beacon using the order ID of the beacon. If you do not claim the beacon, IoT Service may still detect the beacon. But you cannot configure or manage the beacon.

This procedure shows you how to claim a beacon scanned by IoT Service.

Before you begin

Keep the order ID of the beacon ready. You have received the order ID through an e-mail and physically along with the packaging of the beacon.

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > Device Management**.

Step 2 Click **Onboard Devices** and choose **Floor Beacons**.

Figure 45: Onboard Devices

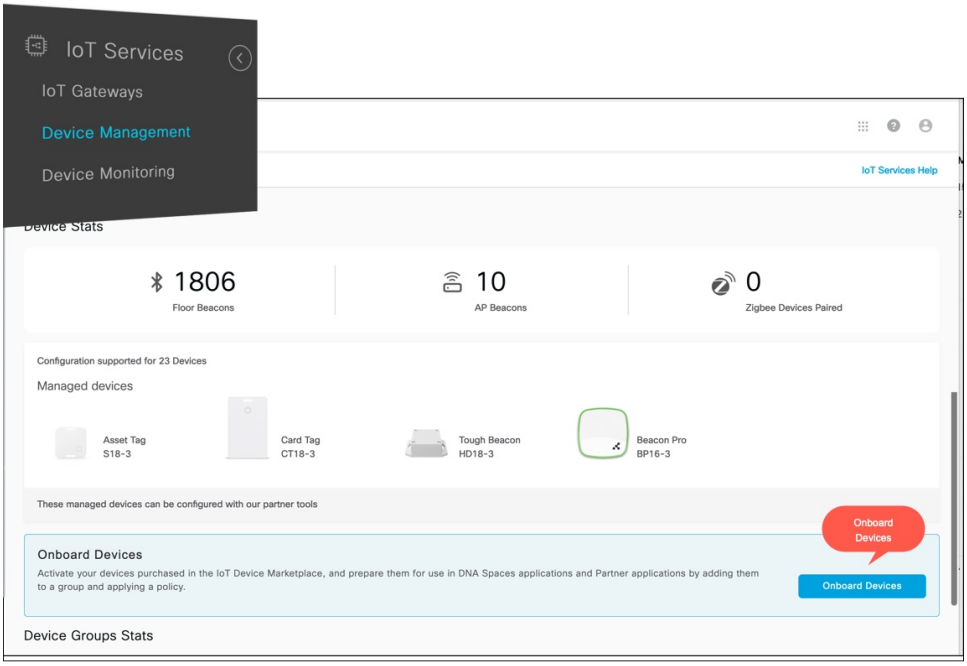
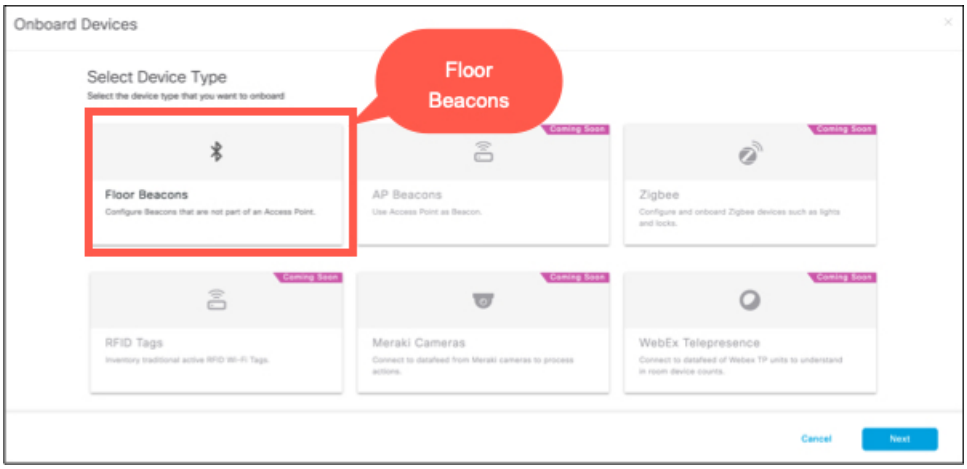


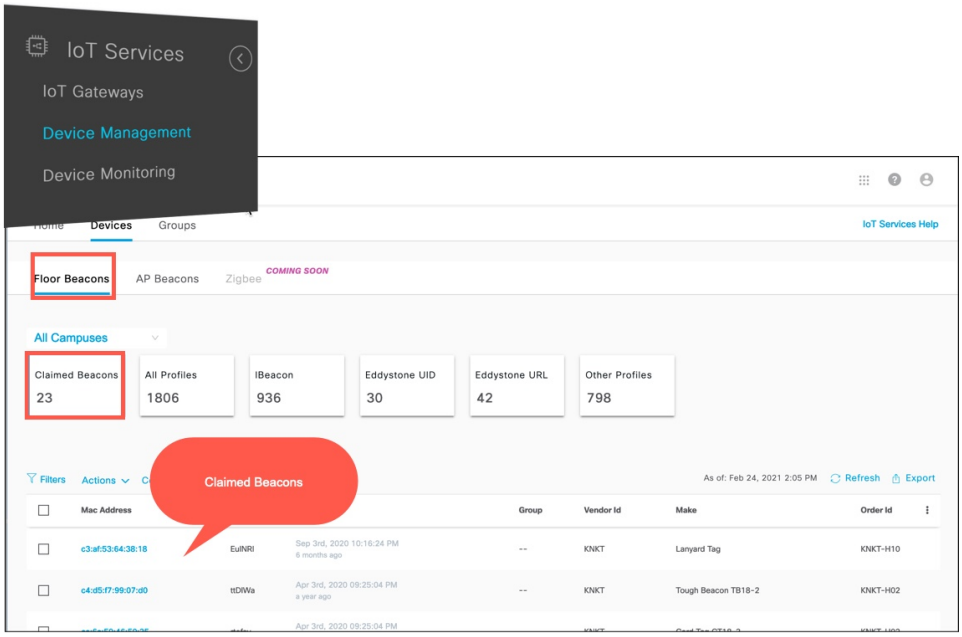
Figure 46: Onboard Floor Beacons



Step 3 In the displayed **Claim Floor Beacons** page, enter the **Order ID** and click **Add to Inventory**.
You can see the beacon in the **IoT Service>Device Management**.

Step 4 In the IoT Service dashboard, navigate to **Device Management**. Under **Floor Beacons > Claimed Beacons**. Verify if the claimed beacon is displayed in this list.

Figure 47: Beacon Details



What to do next

You can now configure the beacons.

Configuring a Beacon on IoT Service

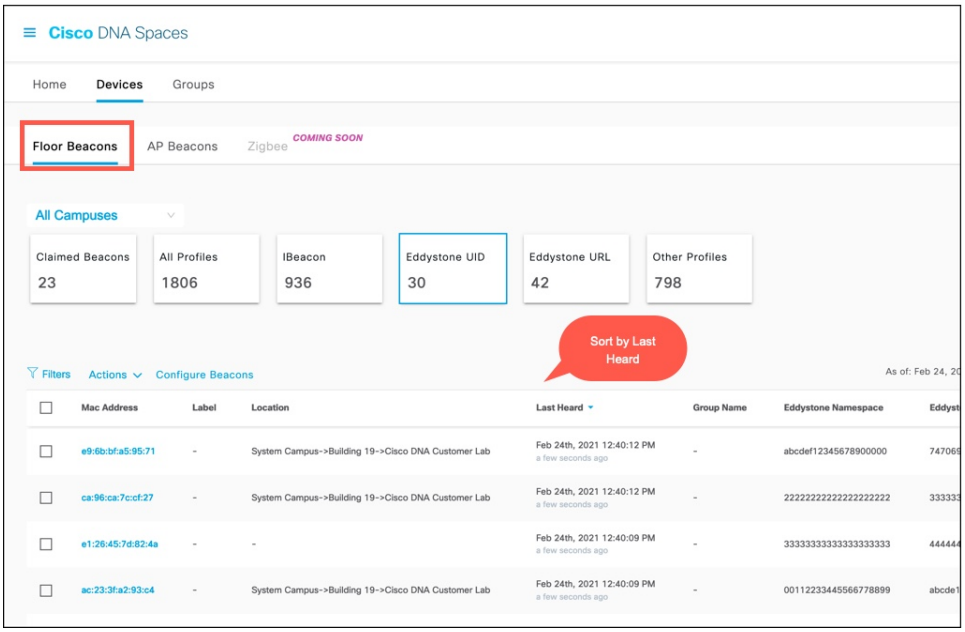
This task shows you how to view the beacons scanned by IoT Service.

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > Device Management > Devices**.

Step 2 Click on **Floor Beacons** to view the scanned beacons.

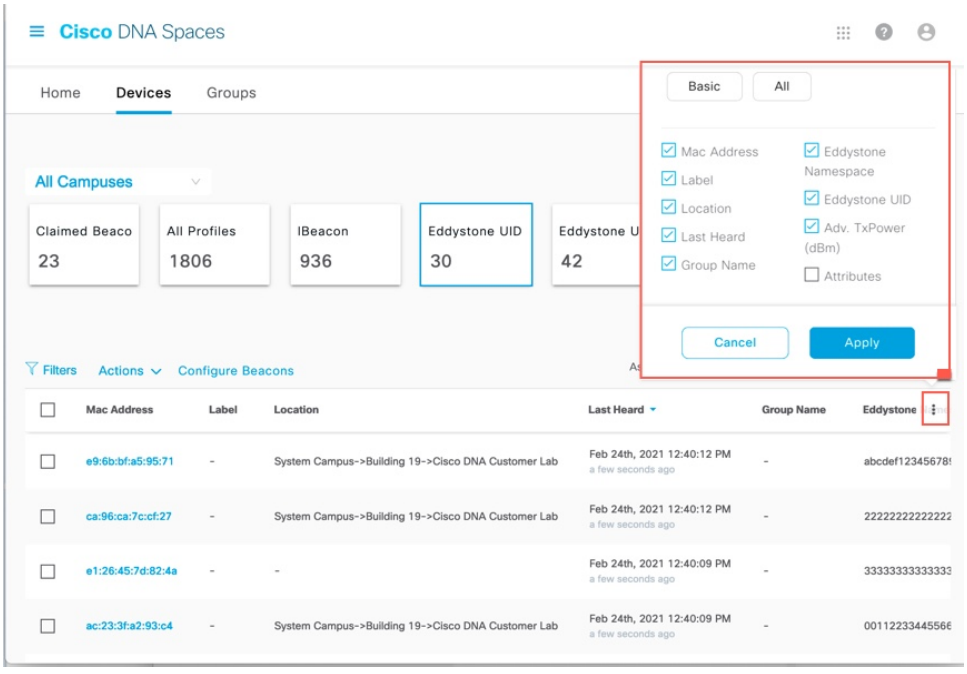
This list is sorted by **Beacon Type**.

Figure 48: Beacon Details



Step 3 Add or delete columns using the three dots on the right.

Figure 49: Adding or Deleting Columns



Step 4 Click on the MAC address of the beacon to view further details.

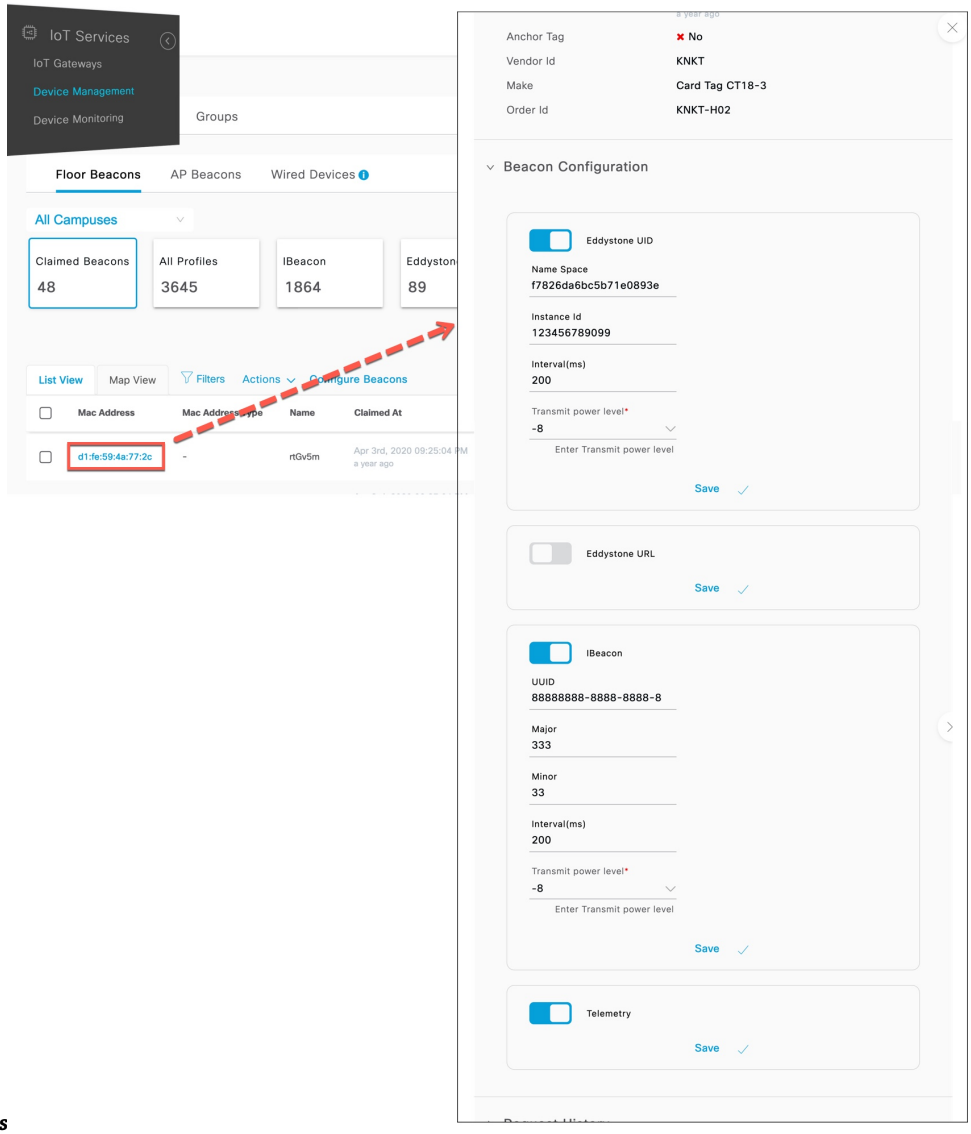


Figure 50: Beacon Details

Step 5 From the **Beacon Information** section, configure the device or enable telemetry.

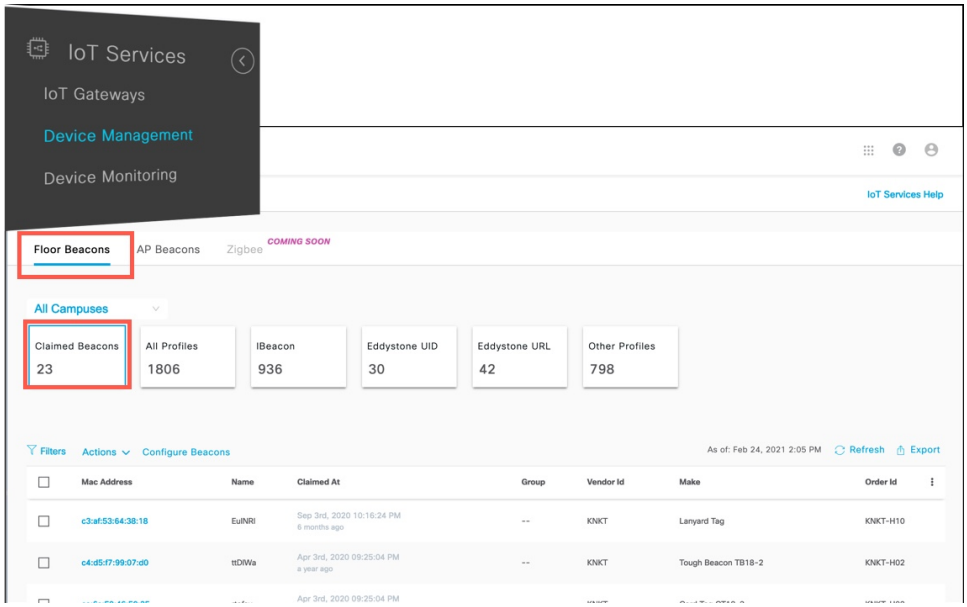
Viewing Sensor Information

Before you begin

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > Device Management > Devices**.

Step 2 Click the **Floor Beacons** tab and click the profile. Choose the floor beacon of your choice.

Figure 51: Beacon Details



Step 3 Click the beacon to see further details. In the **Sensor Information** area, you can see the broadcast sensor data for the beacon.

Figure 52: Status of Configuration on IoT Service

Configuring a Location Anchor

You can configure a claimed beacon as a location anchor for wayfinding. Once a claimed floor beacon is configured as a location anchor, the **Anchor Tag** field in its details indicates the same.

Note Access Points are location anchors by default. Floor beacons must be configured as location anchors.

This task shows you how to configure a claimed floor beacon as a location anchor.

SUMMARY STEPS

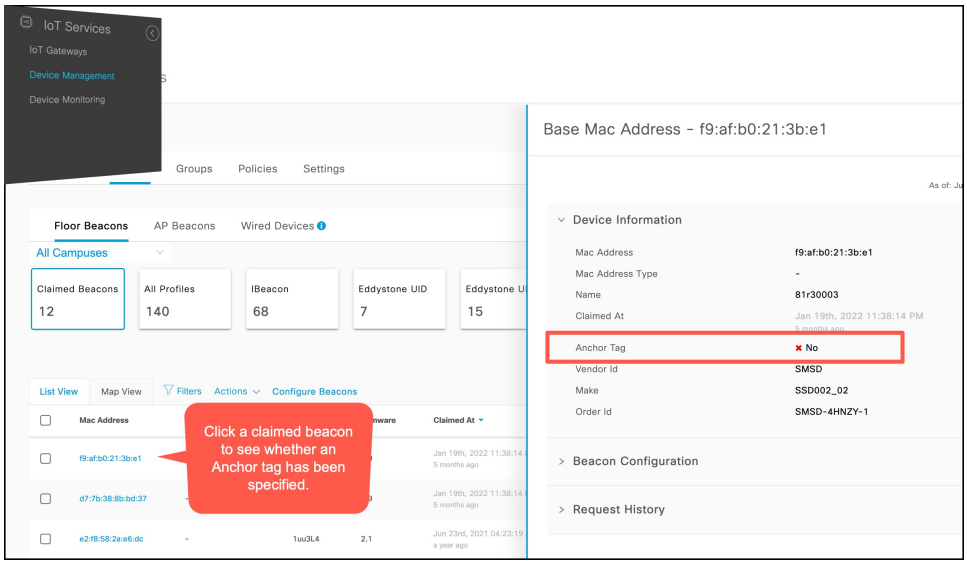
1. From the Cisco Spaces dashboard, navigate to **IoT Service > Device Management > Devices**.
2. Click the **Floor Beacons** tab and click **Claimed Beacons**. Select a floor beacon of your choice to view details. The **Anchor Tag** field indicates if the beacon has a location tag that is associated with it. Close the details page.
3. Click **Map View** and navigate to the required floor. From the list of icons in the left pane, click the **Add Anchor Tag.**
4. Click the position on the map where you want to configure the location anchor. In the **Add anchor tag** page that is displayed, choose the floor beacon by doing one of the following:
   - In the **Claimed Beacon** text field, you can type the first few letters of the floor beacon and choose the correct one from the drop-down that appears.
   - From the **Claimed Beacon** drop-down list, you can choose the floor beacon that you want to configure as a location anchor.

DETAILED STEPS

Step 1 From the Cisco Spaces dashboard, navigate to **IoT Service > Device Management > Devices**.

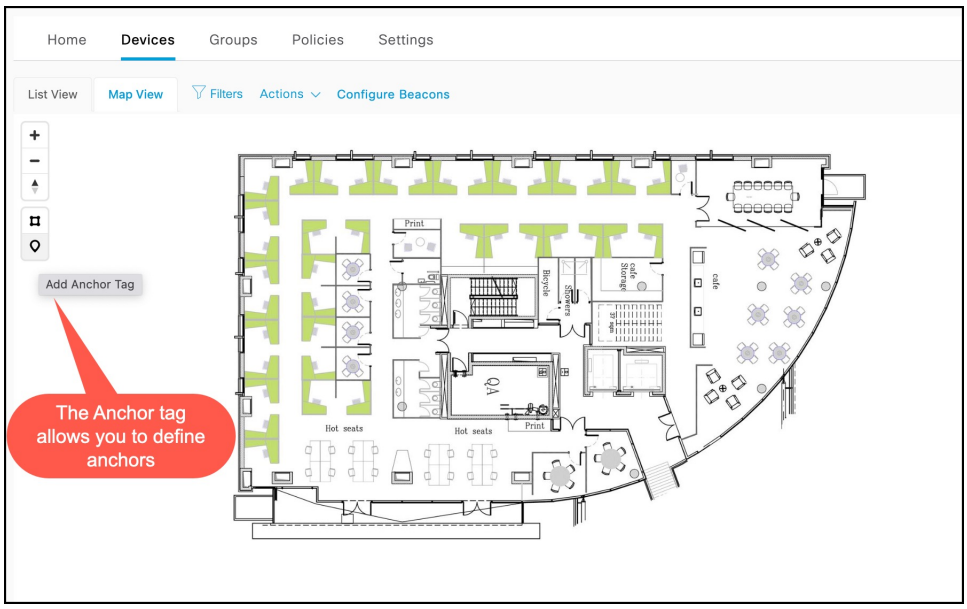
Step 2 Click the **Floor Beacons** tab and click **Claimed Beacons**. Select a floor beacon of your choice to view details. The **Anchor Tag** field indicates if the beacon has a location tag that is associated with it. Close the details page.

Figure 53: Anchor Tag



Step 3 Click **Map View** and navigate to the required floor. From the list of icons in the left pane, click the **Add Anchor Tag.**

Figure 54: Adding Location Anchor in Map View



Step 4 Click the position on the map where you want to configure the location anchor. In the **Add anchor tag** page that is displayed, choose the floor beacon by doing one of the following:

- In the **Claimed Beacon** text field, you can type the first few letters of the floor beacon and choose the correct one from the drop-down that appears.
- From the **Claimed Beacon** drop-down list, you can choose the floor beacon that you want to configure as a location anchor.

Figure 55: Position Anchor Tag

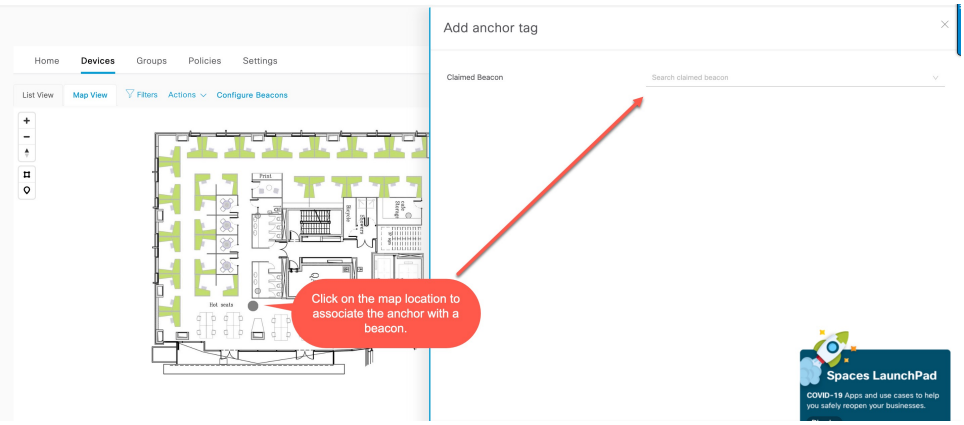
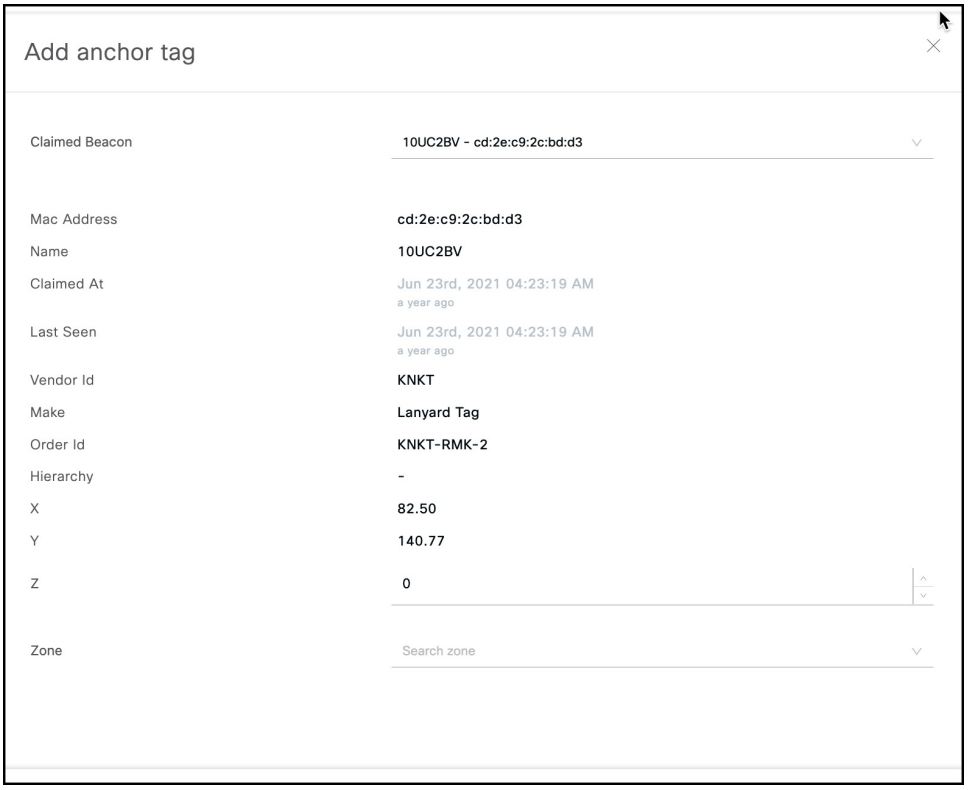
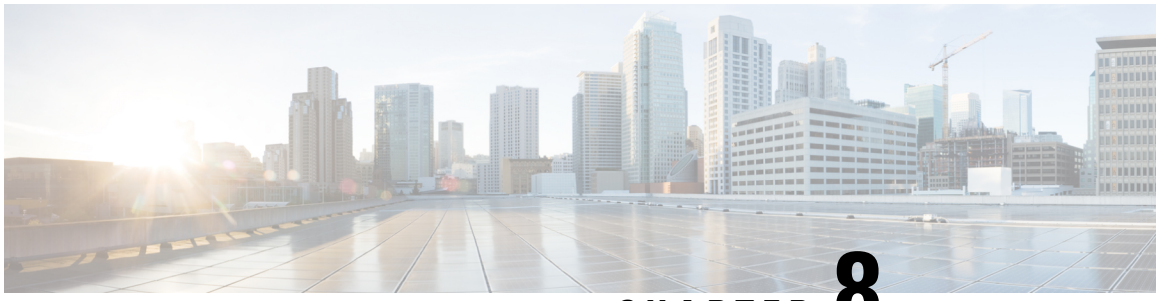


Figure 56: Configure Claimed Beacon as Location Anchor



Once you configure a location anchor, you can use Firehose events to gather location anchor information for wayfinding.



CHAPTER 8

AP as a Sensor

AP as a Sensor

You can now configure the following access points as sensors:

- Cisco Catalyst 9136 Series Access Points
- Cisco Catalyst Wireless 9166I Series Access Points

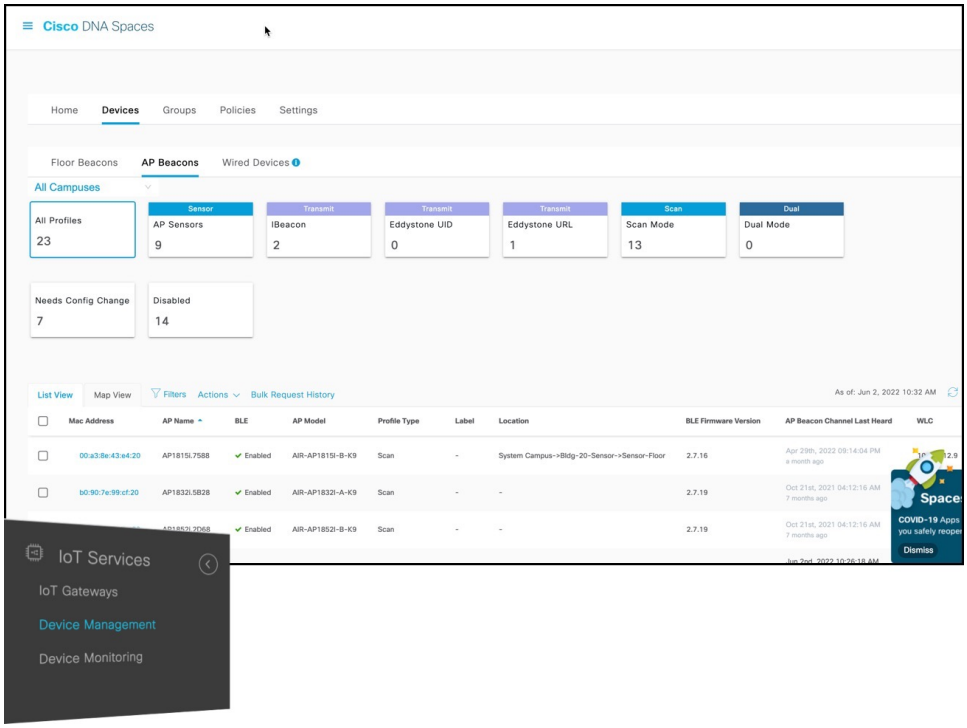
Once configured as a sensor, you can collect telemetry data using this AP. The following sensor values can be configured:

- Temperature
- Relative humidity
- Total volatile organic compound (TVOC), and
- Indoor air quality

Enabling or Disabling an AP Sensor

Step 1 Navigate to Cisco Spaces: IoT Service > **Device Management** > **Devices** > **AP Beacons** > **Sensor**.

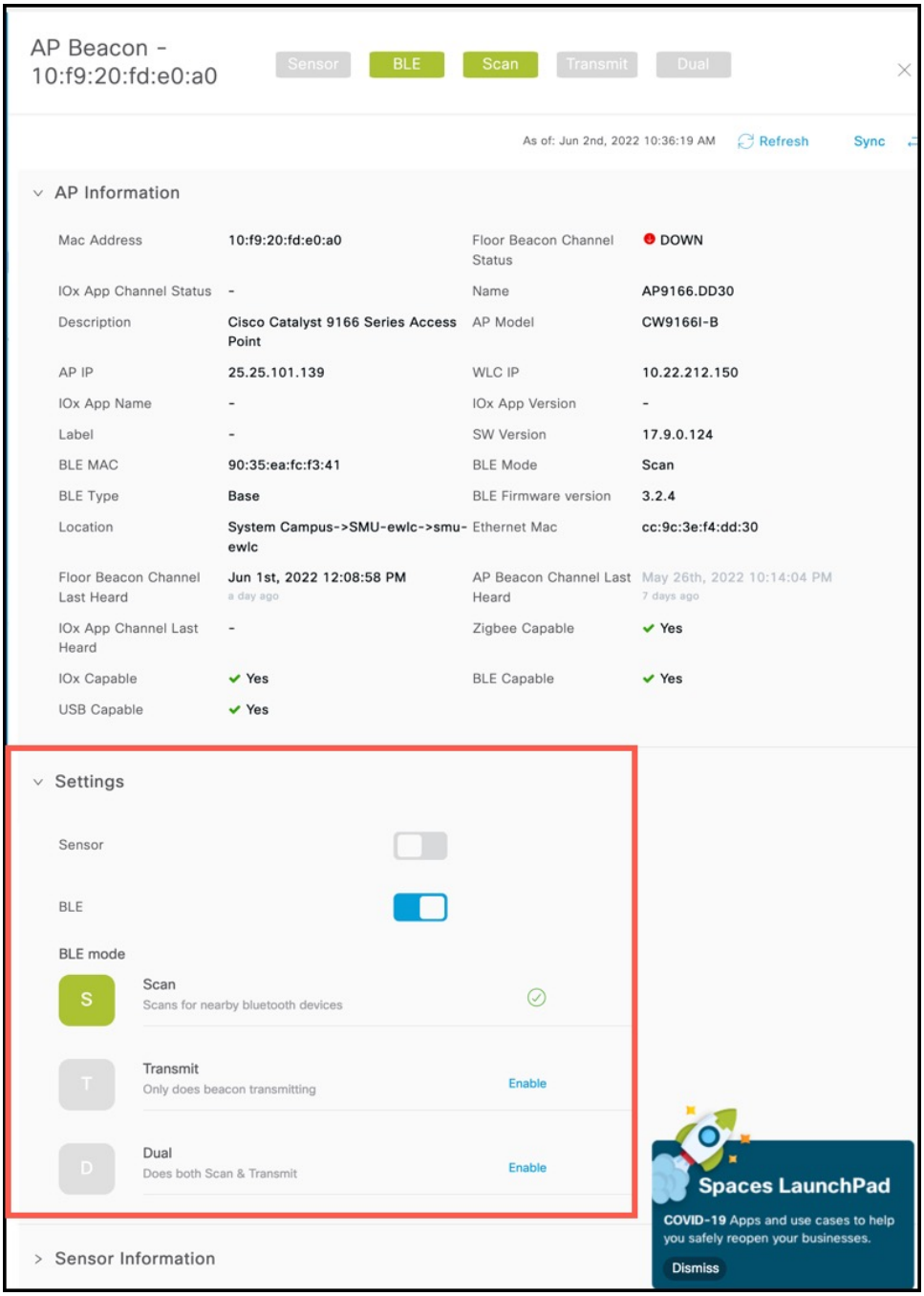
Figure 57: AP as a Sensor



Step 2 Click the AP that you want to configure as a sensor.
The AP Beacons details page opens.

Step 3 In the **Settings** area, click **Sensor** to enable or disable the AP as a sensor.

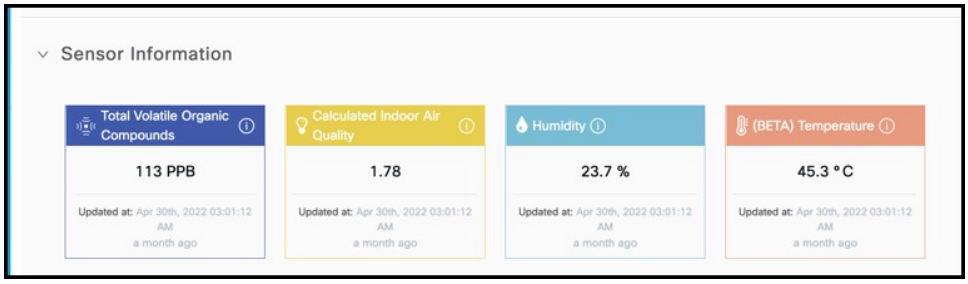
Figure 58: Enabling or Disabling AP as a Sensor



Viewing Sensor Information

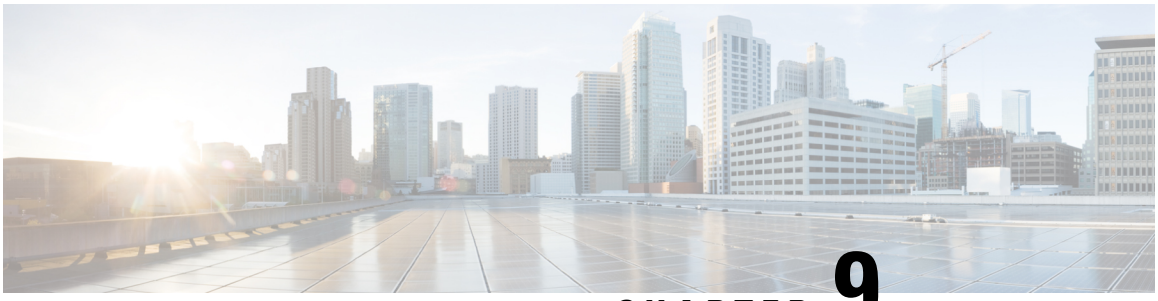
You can view sensor information from the **Sensor Information** area.

Figure 59: Viewing Sensor Information



PART III

Device Management



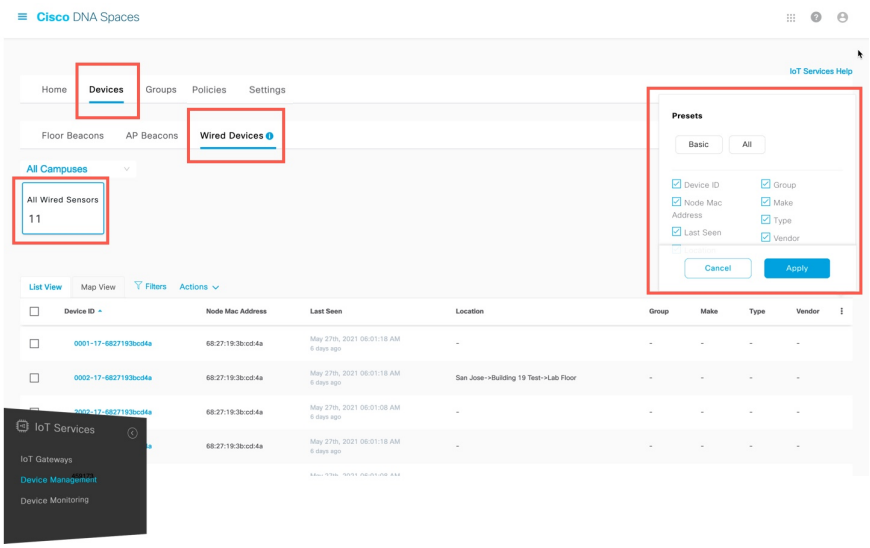
CHAPTER 9

Device Management

Dashboard View of Devices

Choose **IoT Service > Device Management > Devices** and select a device type (**Floor Beacons**, **AP Beacons**, **Wired Devices**) to view an overview of that device.

Figure 60: Dashboard View of Devices

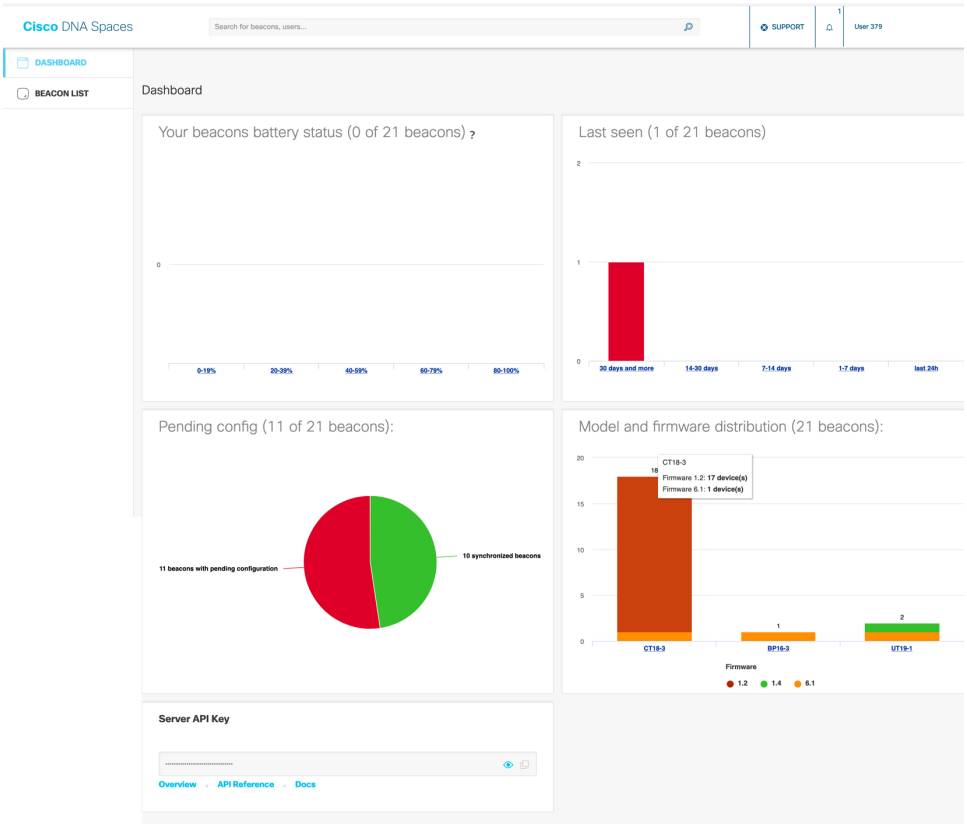


Configuring Beacons

Navigate to **IoT Service** > **Device Management** > **Devices** > **Floor Beacons** > **Configure Beacons**. The window that opens is referred to as the Device Manager in this document.

The Device Manager dashboard gives you a general overview of your beacon infrastructure. All beacons claimed by IoT Service are visible on the Device Manager dashboard. You can see actionable graphs which allow you to navigate quickly to a subset of devices. For example, beacons with 0 to 19 percent battery life, or all beacons with the same underlying firmware or model

Figure 61: The Device Manager Dashboard



Categorizing Devices into Manual Groups

You can create groups and assign devices to them. You can focus attention on certain devices, and view only these devices by filtering them by the group.

The advantages of manual groups are as follows:

- Policies are applied to groups.
- Firehose APIs can filter devices by these groups.
- In the Cisco Spaces: IoT Service dashboard, you can filter devices by groups.

Step 1 In the Cisco Spaces: IoT Service dashboard, navigate to **Device Management** > **Groups**.

Step 2 In the **Add a Group** page, enter **Group Name**, **Description**, and choose **Manual Group** and click **Next**.

Step 3 Click **Create a new group**, and provide a group name and description. Click **Next**.

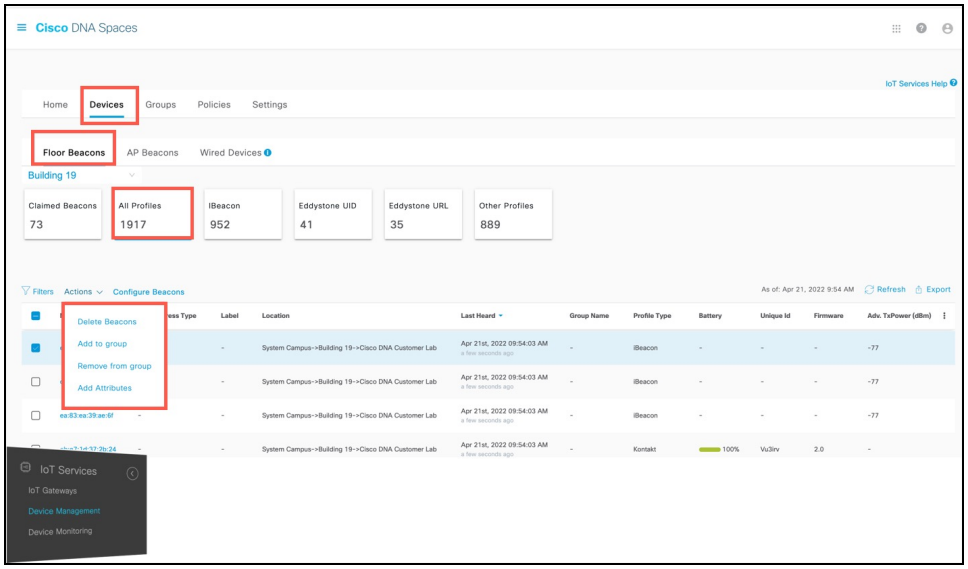
Step 4 In the **Add a group** page that is displayed, choose the type of device (Wireless or Wired), and select the devices to add to this group.

Step 5 Click **Create group**. In the **Done! You have Created a Group** page, click **Close**, or **Create another group**.

On the **Groups** tab, you can see the group that you created. Click the group to see the devices in the group. You can also edit the group from this page.

In the **Devices** > **Floor Beacons** > **All Profiles** tab, you can select devices and click **Actions** to add or remove device(s) to groups.

Figure 62: Adding Devices to a Manual Group from the Devices tab



Categorizing Devices into Groups (Dynamic Groups)

You can configure dynamic groups using parameters like MAC prefix, vendor code, and location hierarchy (floor, building, zone, and so on). New devices are automatically added to the group based on these configured parameters.

The advantages of dynamic groups are as follows:

- Policies are applied to groups. Dynamic groups automatically categorize new devices and apply policies to them.
- Firehose APIs can filter devices by these groups.
- In the Cisco Spaces: IoT Service dashboard, you can filter devices by groups.

Step 1 In the Cisco Spaces: IoT Service dashboard, navigate to **Device Management > Groups**.

Step 2 In the **Add a Group** page, enter **Group Name**, **Description**, and choose **Dynamic Group** and click **Next.**

Step 3 Click **Create a new group**, and provide a group name and description. Click **Next**.

Step 4 In the **Dynamic Grouping** page that is displayed, configure the parameter for this group.

- Group by MAC Prefix
- Group by Vendor Code
- Group by Location Hierarchy

Figure 63: Group by MAC Prefix

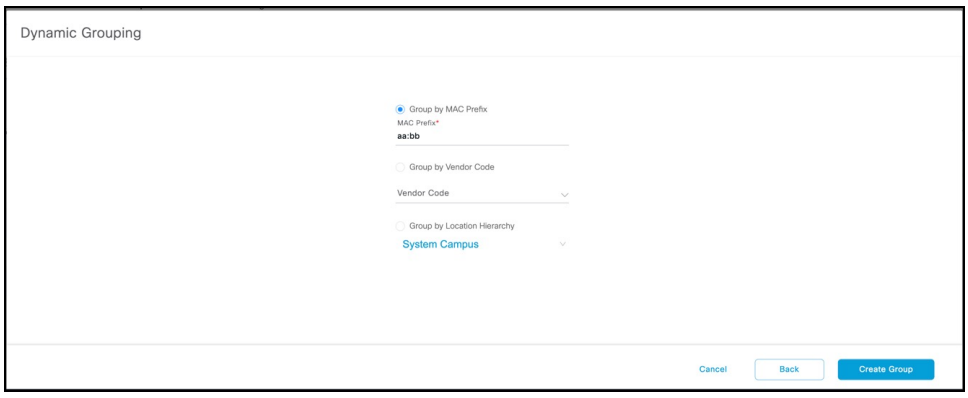


Figure 64: Group by Vendor Code

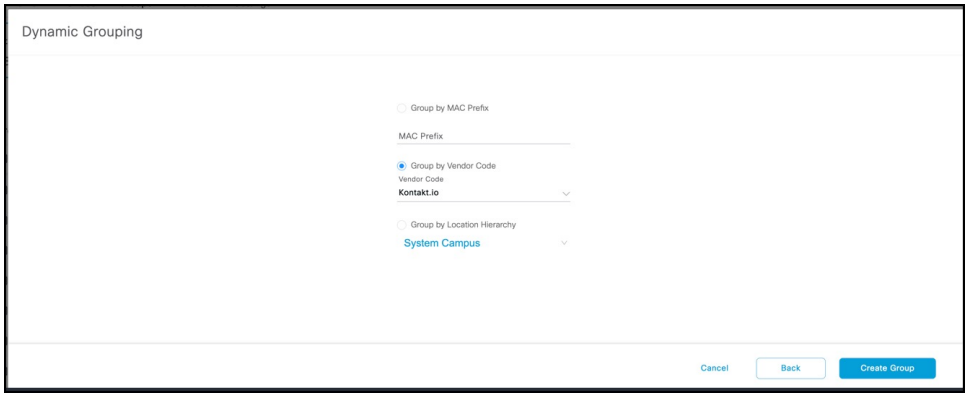


Figure 65: Group by Location Hierarchy

Step 5 Click **Create group**. In the **Done! You have Created a Group** page, click **Close**, or **Create another group**.

On the **Groups** tab, you can see the group that you created. Click the group to see the devices in the group. You can also edit the group from this page.

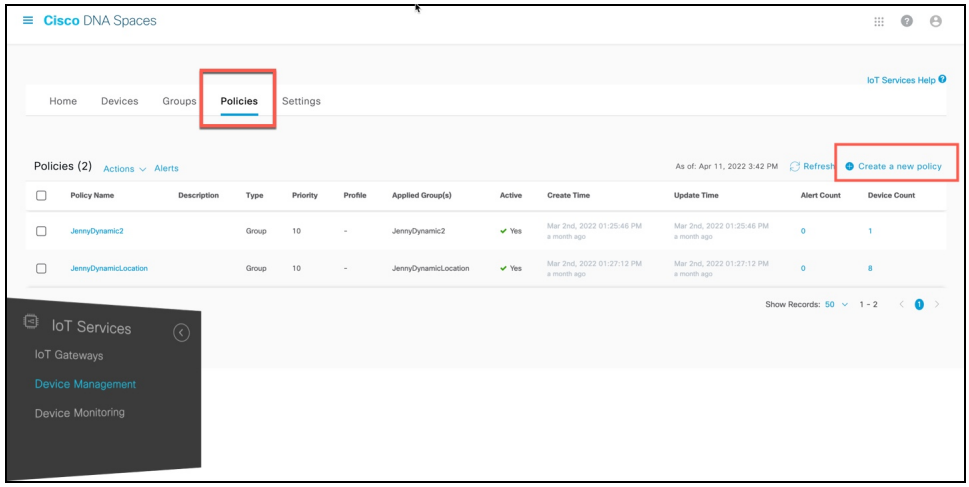
What to do next

You can delete a device by selecting the check box of the group and then selecting **Actions** > **Delete Group**.

Applying Policies to Beacons

Step 1 From the Cisco Spaces: IoT Service dashboard, click **Device Management** > **Policies** and then **Create a new policy**.

Figure 66: Creating a New Policy



Step 2 From the **Configure a Transmit Policy** page that opens, provide a policy name, a description, and choose one of the four policy types.

Figure 67: Choosing One of Four Policies

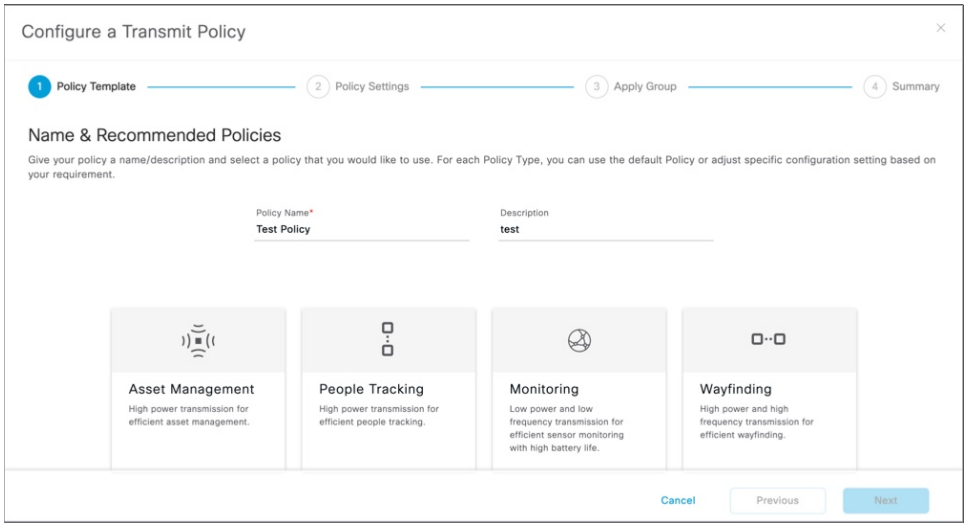
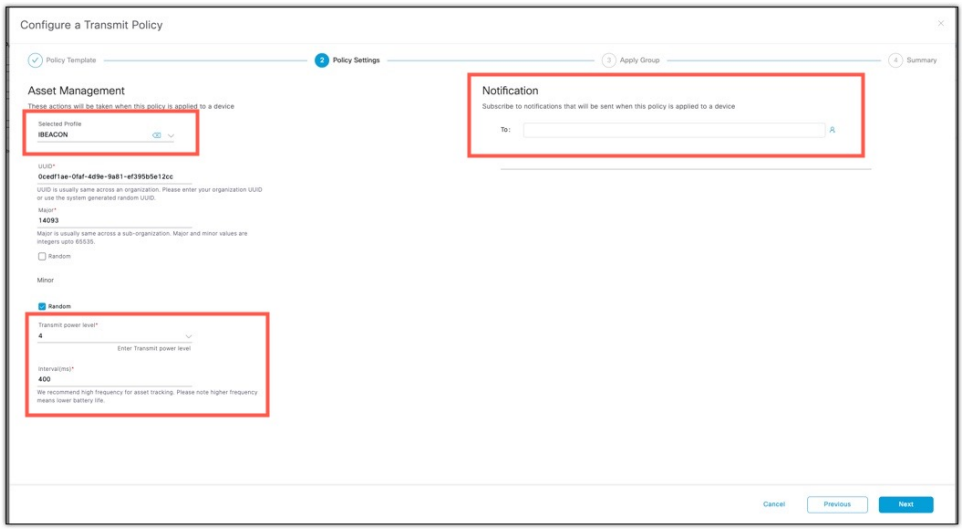


Table 5: Types of Transmit Policy

Policy Type	Transmit Power Level	Interval (ms)
Asset Management: High-Power transmission for efficient asset management	4	400
People Tracking: High-Power transmission for efficient asset management	0	300
Monitoring: Low power and low frequency transmission for efficient sensor monitoring and high battery life.	-8	2000
Wayfinding: High power and high frequency transmission for efficient wayfinding.	4	100

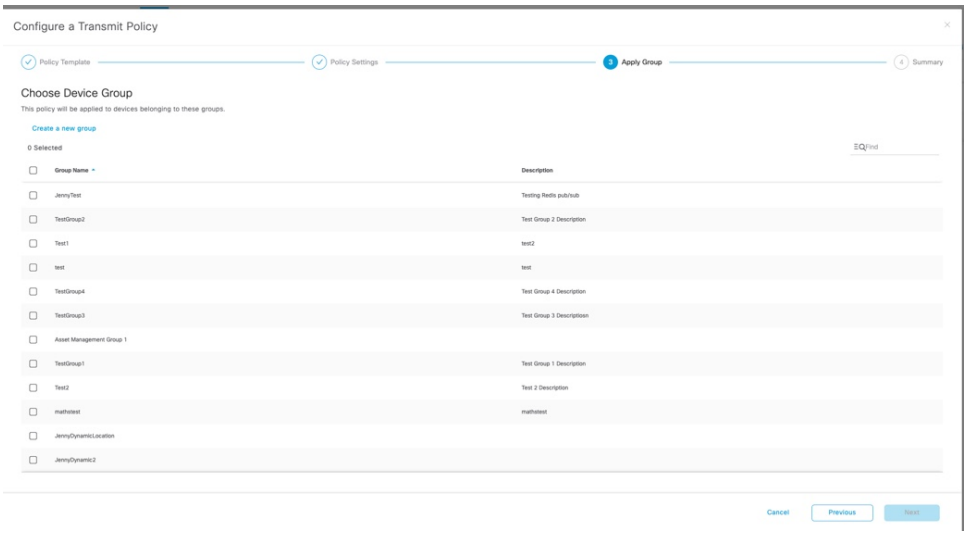
Step 3 From the **Configure a Transmit Policy** page that opens, enter email addresses in the **Notification** field. When this policy is applied to any device, the addresses are notified.

Figure 68: Configure a Transmit Policy



Step 4 From the **Choose Device Group** page, choose a device group. The policy is automatically applied to any device added to this device group.

Figure 69: Choosing a Device Group for Dynamic Policy Application



Step 5 Review the summary and click **Create**. Then click **Close**.

Step 6 In the **Policies** page, you can do any of the following:

- Click a policy to enable or disable the policy.
- From the **Device** column of a policy, click the value to see the list of devices on which the policy is applied.
- From the **Alert Count** column of a policy, click the value to see the list of alerts for the policy.

Figure 70: Enabling or Disabling a Policy

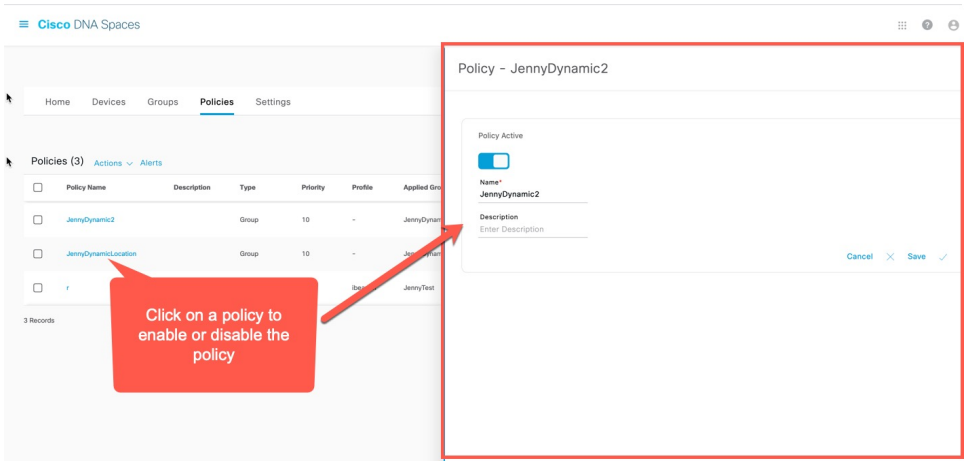
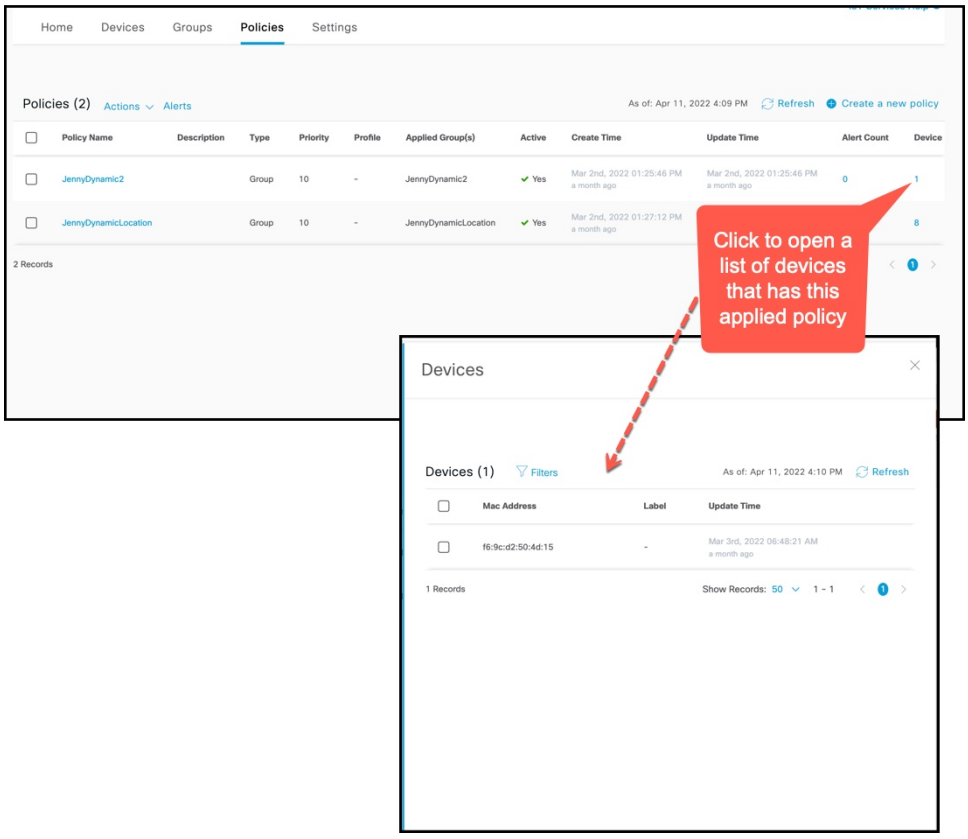
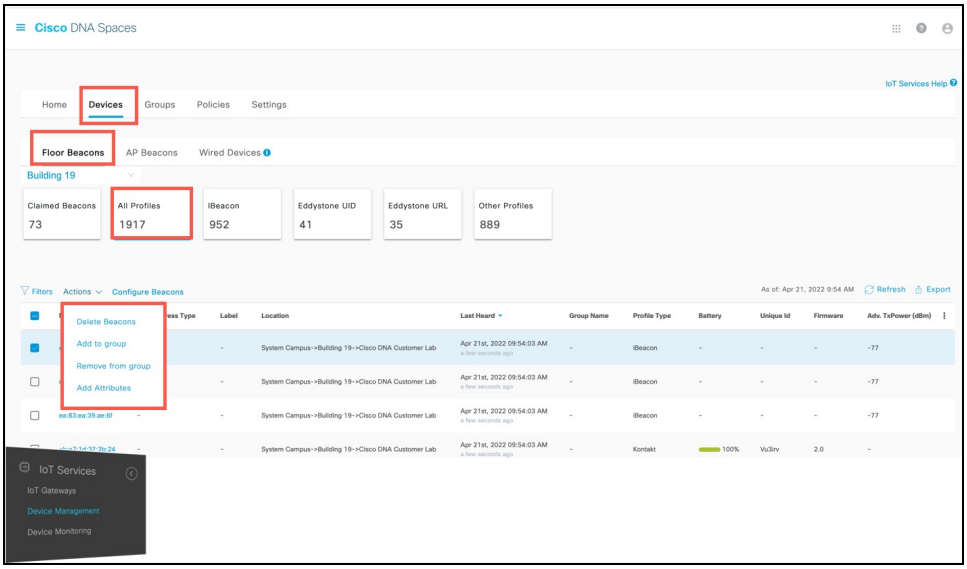


Figure 71: Viewing Devices on Which a Policy Is Applied



You can now apply this policy to a static or dynamic group. If the policy is applied on a static group, you can assign devices to the group, and the policy is automatically applied. To do this, navigate to the Cisco Spaces: IoT Service dashboard, click **Device Management > Devices** and then **Floor Beacons > All Profiles**. Select the devices and click **Actions > Add to group**.

Figure 72: Creating a New Policy

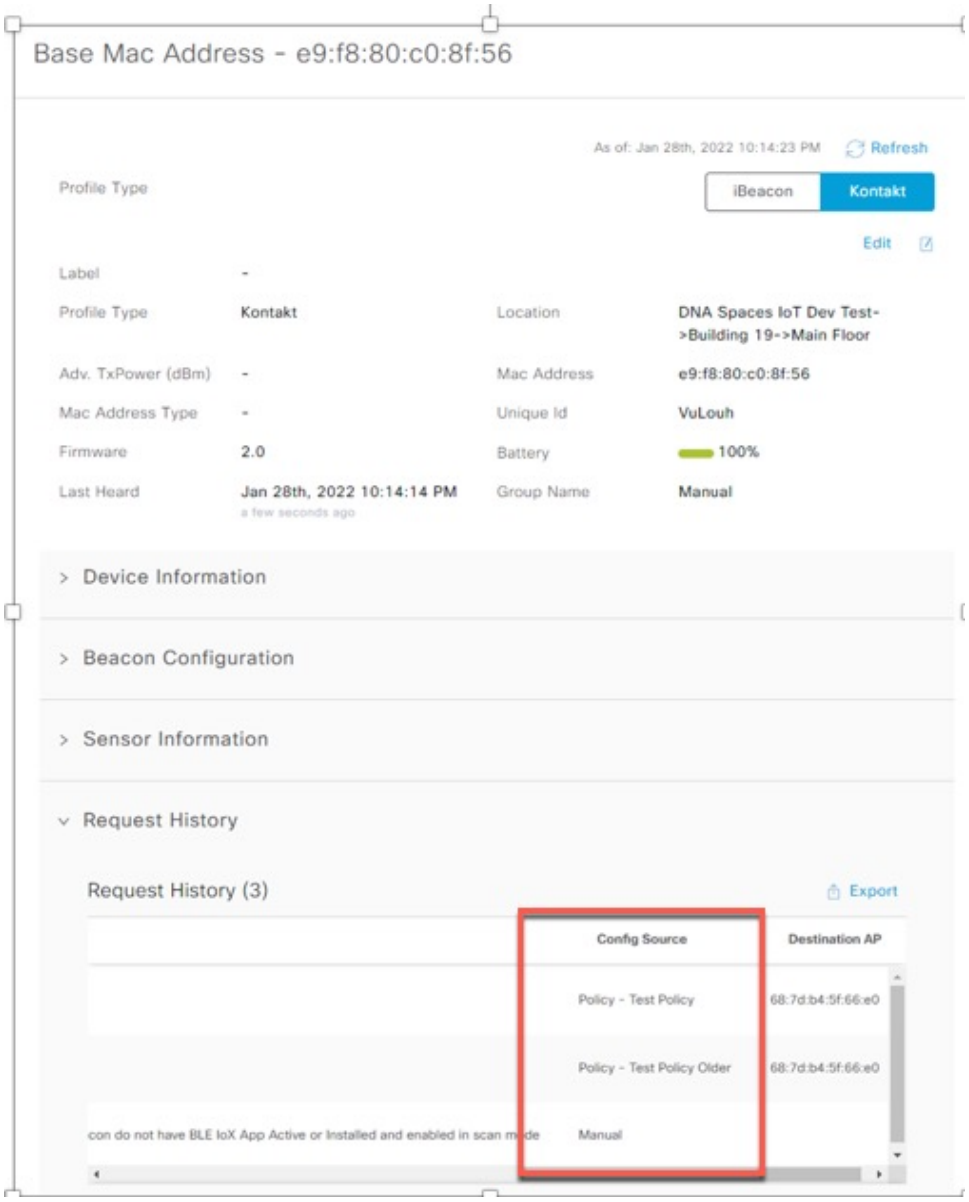


What to do next

You can verify if a policy is applied on a device by checking the request history in the device details. In the **Request History** page, refer to the **Config Source** column.

- **Manual**: Policy change that is made by Cisco Spaces or partner dashboard.
- **<Policy Name >**: Policy has been applied dynamically to the device.

Figure 73: Config Source: Policy



Filtering Devices

While Cisco Spaces: IoT Service scans all devices, you may not want to view certain devices on the dashboard. You can now filter out devices from the Cisco Spaces: IoT Service dashboard using types of MAC addresses. Filtering is currently at the cloud level and not at AP-level. Once filtered, these devices do not appear in the following locations;

- Cisco Spaces: Detect and Locate
- Cisco Spaces: IoT Service

- Output of Firehose API calls

You can filter out devices based on the following MAC address types.

- **Enable Public MAC**: Allows global, fixed MAC addresses that are registered with the IEEE Registration Authority, which does not change during the device's lifetime.
- **Enable Random Static MAC**: Allows random static MAC address, which is a random number generated every time that the device boots up or a value that stays the same for the device's lifetime. However, it does not change within one power cycle of the device.
- **Enable Random Private MAC**: Allows random private MAC addresses of two types:
  - **Resolvable**: These are generated from an identity resolving key (IRK) and a random number. They can be changed often (even during the lifetime of a connection) and prevents an unknown scanning device from identifying and tracking the device. Only scanning devices that possess the IRK distributed by the beaconing device (exchanged using a private resolvable address) can resolve that address, allowing the scanning device to identify the beaconing device.
  - **Unresolvable**: A random number that can change anytime.

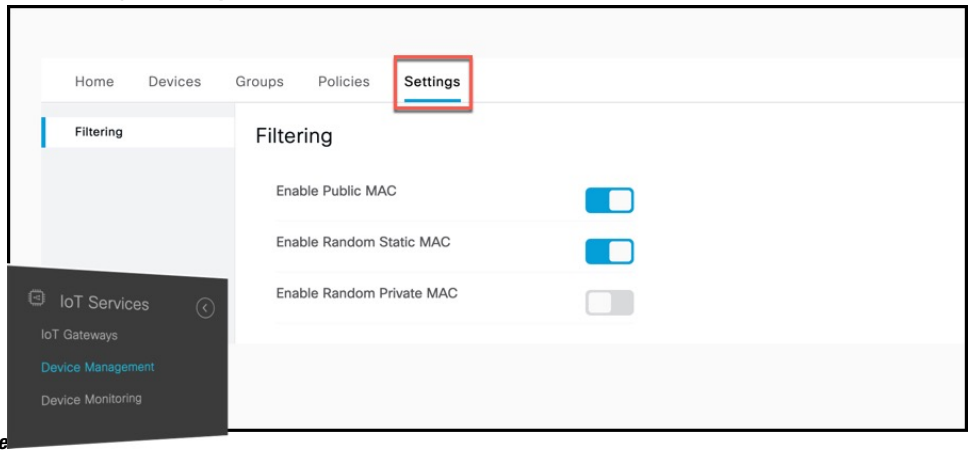
SUMMARY STEPS

1. Navigate to **Device Management** > **Settings**.

DETAILED STEPS

Navigate to **Device Management** > **Settings**.

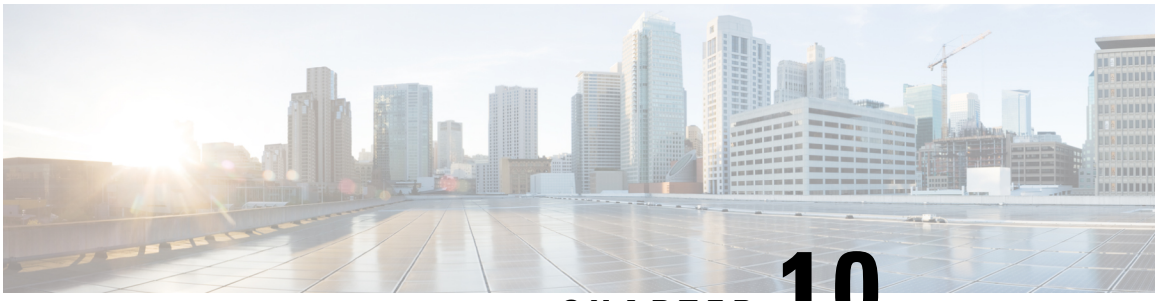
Figure 74: Filtering Devices by MAC Address



Type

PART IV

Device Monitoring



CHAPTER 10

Device Monitoring

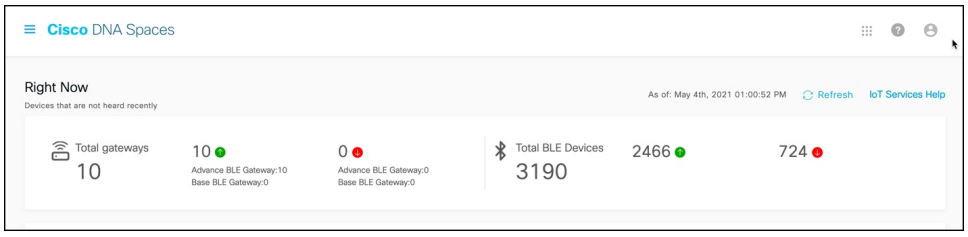
From the IoT Service > **Device Monitoring** page, you can monitor all the IoT devices and gateways, and also get a one-shot categorized view of devices according to their battery life and last heard time.

Right Now

In the **Total gateways** part of this section, you can see an overview of all gateways that are being monitored. You can also see the number of reachable gateways (base and advanced) counted under the green dot, and the number of unreachable gateways counted under the red dot.

In the **Total BLE Devices** part of this section, you can see an overview of all BLE devices that are being monitored. You can also see the number of reachable devices (base and advanced) counted under the green dot, and the number of unreachable devices counted under the red dot.

Figure 75: Right Now



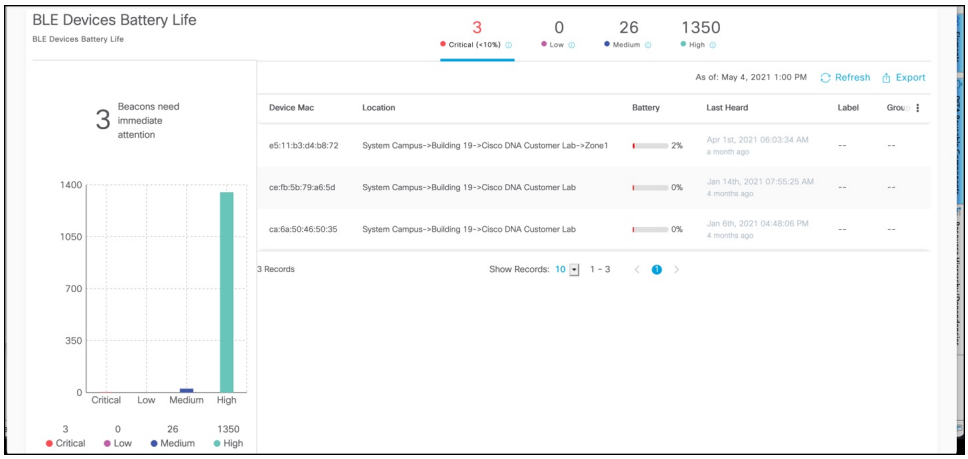
BLE Devices Battery Life

In the section, you get an overview of only those BLE devices (beacons) that can sense their own battery life. The devices are categorized according to their current battery life as:

- Critical
- Low
- Medium

- High

On the top of this section, you can see the number of devices in each category. To the left, you can also see this information represented as a bar chart. You can click either on the category listed on the top or the corresponding bar to see a detailed list of the devices. You can also export this list as a CSV file.

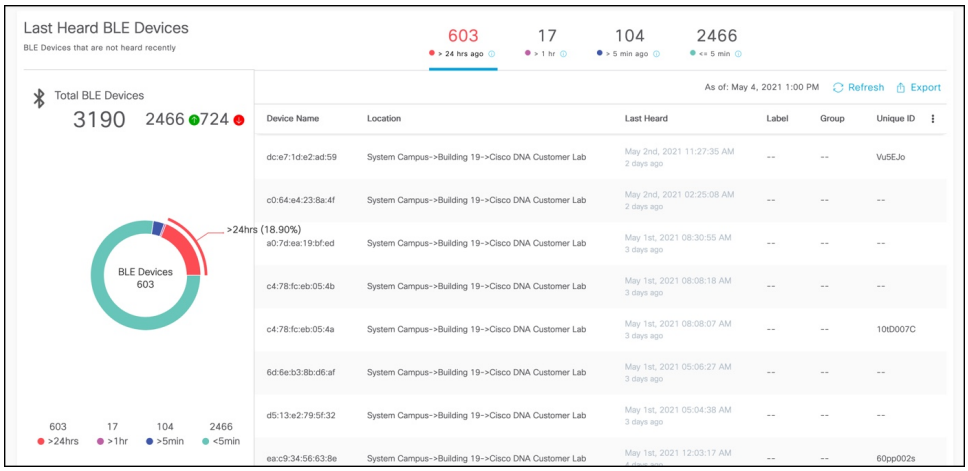


Last Heard BLE Devices

In the section, you get an overview of all BLE devices (beacons). The devices are categorized according to the last time they were heard as the following:

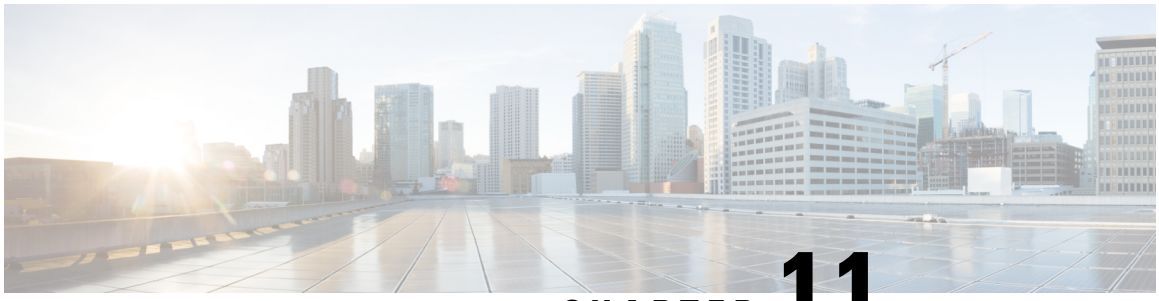
- greater than 24 hrs ago
- greater than one hour ago
- greater than five minutes ago.
- less than or equal to five minutes ago

To the top of this section, you can see this information represented as numbers. To the left of this section, you can also see this information represented as a bar chart. You can click either on the number listed on the top or the corresponding bar to see a detailed list of the devices. You can also export this list as a CSV file.


Last Heard BLE Devices
BLE Devices that are not heard recently
603
> 24 hrs ago
17
> 1 hr
104
> 5 min ago
2466
<= 5 min
Total BLE Devices
3190
2466
724
As of: May 4, 2021 1:00 PM
Refresh
Export
Device Name
Location
Last Heard
Label
Group
Unique ID
dc:e7:1d:e2:ad:59
System Campus->Building 19->Cisco DNA Customer Lab
May 2nd, 2021 11:27:35 AM
2 days ago
Vu5EJo
c0:64:e4:23:8a:4f
System Campus->Building 19->Cisco DNA Customer Lab
May 2nd, 2021 02:25:08 AM
2 days ago
a0:7d:ea:19:bf:ed
System Campus->Building 19->Cisco DNA Customer Lab
May 1st, 2021 08:30:55 AM
3 days ago
c4:78:fc:eb:05:4b
System Campus->Building 19->Cisco DNA Customer Lab
May 1st, 2021 08:08:18 AM
3 days ago
c4:78:fc:eb:05:4a
System Campus->Building 19->Cisco DNA Customer Lab
May 1st, 2021 08:08:07 AM
3 days ago
10tD007C
6d:6e:b3:8b:d6:af
System Campus->Building 19->Cisco DNA Customer Lab
May 1st, 2021 05:06:27 AM
3 days ago
d5:13:e2:79:5f:32
System Campus->Building 19->Cisco DNA Customer Lab
May 1st, 2021 05:04:38 AM
3 days ago
ea:c9:34:56:63:8e
System Campus->Building 19->Cisco DNA Customer Lab
May 1st, 2021 12:03:17 AM
60pp002s
>24hrs (18.90%)
BLE Devices
603
603
>24hrs
17
>1hr
104
>5min
2466
<5min

PART V

Troubleshooting



CHAPTER 11

Troubleshooting IoT Services: Controller

Reprovisioning IoT Services After Failover

What settings are needed to allow access via NETCONF?

To enable access via the Network Configuration Protocol (NETCONF), configure the following settings on your wireless controller:

1. Enable the authentication, authorization, and accounting (AAA) new model by entering the following command in the global configuration mode:

   ```
   aaa new-model
   ```

2. Set the default AAA authentication for login to the local user database with the command:

```
aaa authentication login default local
```

3. Specify the default AAA authorization for exec (shell access) to use the local user database by using the command:

```
aaa authorization exec default local
```

Enter these commands in the global configuration mode of your wireless controller:

```
wireless controller# configure terminal
wireless controller(config)# aaa new-model
wireless controller(config)# aaa authentication login default local
wireless controller(config)# aaa authorization exec default local
```

After executing these commands, your wireless controller should be properly configured to allow access through NETCONF using the local user database for authentication and authorization.

The global configuration for BLE radio has to be enabled on Wireless Controller. How do I verify the setting?

This task shows you how to verify if you have enabled BLE radio on the wireless controller at a global configuration level. This is a necessary setting.

Run the command: **show running-config | include ap dot15**

```
wireless controller# show running-config | include ap dot15
no ap dot15 shutdown
```

Verify if the output is `no ap dot15 shutdown`. This output indicates that the dot15 BLE radios are not shut down.

For the gRPC connection to work, a streaming token is required on the Wireless Controller. How do I view the token?

To establish a functioning gRPC connection, a gRPC streaming token must be present on the wireless controller. To verify the token, execute the **show running-config | include ap cisco-dna** command on the command on the wireless controller

```
wireless-controller# show running-config | include ap cisco-dna

ap cisco-dna token 0 eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aWQiOjE2MjUs
ImNpZCI6Mzc4NTc3ODI1NDI2NzIyNjUwMDAsImVwIjoiMTAuMzAuMTE0LjEwODo4MDAwIiwiaW
F0IjoxNTg1NzA2OTIxfQ.56vXfL1IGrss6TJZDQaWVarAoTWZsIhbe3tGVMEJNYk
```

The resulting output will display the gRPC streaming token. For example:

```
ap cisco-dna token 0 <token_string>
```

Ensure that this token corresponds with the token configured on the access point (AP). You can check the AP's token by running the **show cloud connector key authentication** command.

Additionally, to examine the encoded information contained in the token, you can input the token into a JWT decoder like the one found at http://jwt.io/. Here is an example of the kind of payload data you might see:

```
PAYLOAD:DATA
{
  "tid": 1625,
  "cid": 37857782542672265000,
  "ep": "10.30.114.108:8000",
  "iat": 1585706921
}
```

gRPC must be enabled in the access point join profile. How do I verify the join profile has gRPC enabled?

This procedure demonstrates how to enable gRPC in the AP join profile, a necessary configuration.

To view the active settings, run the **show running-config | begin ap profile default-ap-profile** command.

```
controller# show running-config | begin ap profile default-ap-profileap profile
default-ap-profile
 apphost
 cisco-dna grpc
 description "default ap profile"
 mgmtuser username admin password 0 Cisco123! secret 0 Cisco123!
 ssh
trapflags ap crash
trapflags ap noradiocards
trapflags ap register
netconf-yang
end
```

This output reveals the configuration for the default AP profile. Should you require a different profile, apply the command accordingly, replacing **`default-ap-profile`** with the desired profile name.

Ensure the configuration includes the line `cisco-dna grpc`. This line confirms that gRPC is enabled for all access points utilizing this profile.

How do I verify gRPC is up?

To verify whether gRPC is operational, execute the **show ap grpc summary** command.

This command displays the gRPC connection status for each AP connected to the wireless controller, as shown in the example below:

```
controller# show ap grpc summary
AP Name                          AP Mac               gRPC Status
-----------------------------------------------------------------------------------
AP_10.2830                       04eb.409f.a7e0       Up
AP_02.2898                       04eb.409f.ab20       Up
AP_06.28CC                       04eb.409f.acc0       Up
AP_08.28E0                       04eb.409f.ad60       Up
AP_07.28E4                       04eb.409f.ad80       Up
AP_09.28EC                       04eb.409f.adc0       Up
AP_01.28F0                       04eb.409f.ade0       Up
AP_03.2928                       04eb.409f.afa0       Up
AP_05.2934                       04eb.409f.b000       Up
AP_04.2938                       04eb.409f.b020       Up
```

Each AP's name, MAC address, and gRPC status are listed. A status of `Up` indicates that gRPC is active and running for that AP.

.

How do I verify that TDL subscriptions are created and are valid?

1. To initiate the process of viewing all current telemetry subscriptions and to check their types and validity statuses, input the command below:

   ```
   show telemetry ietf subscription all
   ```

2. After executing the command, the wireless controller presenst a summarized output of the telemetry subscriptions. Enterprise Data Management (EDM) configures six distinct subscriptions, which you can identify by their numbers ranging from 122 to 127.

   Here is a sample of what the command's output might look like:

   ```
   wireless controller# show telemetry ietf subscription all
   Telemetry subscription brief
   ID     Type          State    Filter type
   ---------------------------------------
   122    Configured    Valid    tdl-uri
   123    Configured    Valid    tdl-uri
   124    Configured    Valid    tdl-uri
   125    Configured    Valid    transform-name
   126    Configured    Valid    transform-name
   ```

The output enumerates each subscription's unique ID, its configuration status, the validity of the state, and the applied filter type.

Are the TDL subscriptions created and valid?

Run the command **show telemetry ietf subscription all** command on the wireless controller.

The command displays the subscriptions, the subscription type, and if a subscription is valid. IoT Service creates five different subscriptions 122-126.

```
wireless controller# show telemetry ietf subscription all
  Telemetry subscription brief

  ID               Type        State       Filter type
  ------------------------------------------------------
  122              Configured  Valid       tdl-uri
  123              Configured  Valid       tdl-uri
  124              Configured  Valid       tdl-uri
  125              Configured  Valid       transform-name
  126              Configured  Valid       transform-name
```

What is the TDL status?

Execute the **show telemetry ietf subscription ID receiver** command on the wireless controller.

The command presents the status of Telemetry Description Language (TDL) subscriptions.

```
wireless controller# show telemetry ietf subscription 125 receiver
Telemetry subscription receivers detail:

  Subscription ID: 125
  Address: 10.22.243.33
  Port: 8004
  Protocol: cloud-native
  Profile:
  Connection: 33
  State: Connected
  Explanation:
```

The IoT Service manages five distinct subscriptions, with IDs from 122 to 126. For each subscription:

- Verify that the **Address** matches the IP address of the Cisco Spaces: Connector.
- Confirm that the **State** is **Connected**

.

How do I view the current CAPWAP values for an AP?

1. Enter the command without any dots in the MAC address of the AP:

   ```
   test platform software database get ewlc_oper/capwap_data;wtp_mac=mac_without_dots
   ```

For example:

```
wireless controller# test platform software database get
ewlc_oper/capwap_data;wtp_mac=1cd1e065c340
```

The output presents a table with various records:

- Index 0 contains the AP's MAC address, IP address, model, and other static information.
- The **device_detail.static_info** section includes the AP's model, memory type, CPU type, and memory size, among other details.
- The **device_detail.wtp_version section** includes backup software version, mini iOS version, hardware version, and the current software version that the AP is running.
- The **ap_services** section gives details about monitor mode, DHCP server status, and sniffer interface ID.
- The **tag_info** section indicates whether the AP has any misconfigured tags.
- The **external_module_data** section displays information about any external modules connected to the AP, including product ID and version.
- The **ap_state** section displays administrative and operational states of the AP.
- The **ap_mode_data** section details the current mode and sub-mode of the AP.

```
wireless-controller# test platform software database get
ewlc_oper/capwap_data;wtp_mac=1cd1e065c340
Table Record Index 0 = {
 [0] wtp_mac = 1CD1.E065.C340
 [1] ip_addr = 10.22.243.229
 [2] name = AP84F1.47B2.B868
 [3] device_detail.static_info.board_data.model = C9115AXI-B
 [4] device_detail.static_info.board_data.wtp_serial_num = FJC25331LCY
```

```
 [5] device_detail.static_info.board_data.card_id = 0
 [6] device_detail.static_info.board_data.card_rev = 0
 [7] device_detail.static_info.board_data.wtp_enet_mac = 84F1.47B2.B868
 [8] device_detail.static_info.board_data.ap_sys_info.mem_type = DDR3
 [9] device_detail.static_info.board_data.ap_sys_info.cpu_type =  ARMv8 Processor rev 0
(v8l)
 [10] device_detail.static_info.board_data.ap_sys_info.mem_size = 1971200
 [11] device_detail.static_info.board_data_opt.antenna_type = BSN_INT_ANT_AP
 [12] device_detail.static_info.board_data_opt.wtp_type = BSN_AP_STANDARD
 [13] device_detail.static_info.board_data_opt.remote = true
 [14] device_detail.static_info.board_data_opt.join_priority = 1
 [15] device_detail.static_info.descriptor_data.max_radio_slots = 2
 [16] device_detail.static_info.descriptor_data.radio_slots_in_use = 2
 [17] device_detail.static_info.descriptor_data.encryption_capabilities = true
 [18] device_detail.static_info.ap_prov.is_universal = false
 [19] device_detail.static_info.ap_prov.universal_prime_status = Unprimed
 [20] device_detail.static_info.ap_models.model = C9115AXI-B
 [21] device_detail.static_info.ap_models.ap_model_short = 9115AXI
 [22] device_detail.static_info.num_ports = 1
 [23] device_detail.static_info.num_slots = 2
 [24] device_detail.static_info.wtp_type = 83
 [25] device_detail.static_info.wtp_model_type = 90
 [26] device_detail.static_info.ap_capability = [
        BRIDGE_MODE_CAPABLE,
        CAP_THREE_SPATIAL_STREAMS_CAPABLE,
        ANTENNA_SELECTION_RESTRICTED_CAPABLE,
        AVC_FNF_CAPABLE,
        RXSOP_THRESHOLD_CAPABLE,
        FABRIC_CAPABILITY,
        BARBADOS_INTERNAL_ANTENNA_SKU_CAPABLE,
        REMOTE_LAN_CAPABLE,
        DOT11AC_160MHZ_CHANNEL_WIDTH_CAPABLE,
        AVC_FNF_FABRIC_CAPABLE,
        AP_CTS_CAPABLE,
        AP_QCA_SPECTRUM_INTELLIGENCE_CAPABLE,
        FIPS_CAPABLE,
        IS_DOT1X_PORT_AUTH_CAPABLE,
        AP_TRACING_CAPABLE,
        AP_WPA3_CAPABLE,
        OFFICE_EXTEND_CAPABLE,
        ETH2_RLAN_CAPABLE,
        AP_MEWLC_CAPABLE,
        SNIFFER_MODE_CAPABLE,
        ICAP_PARTIAL_PACKET_TRACE_CAPABLE,
        ICAP_ANOMALY_DETECTION_CAPABLE,
        ICAP_STATISTICS_CAPABLE,
        ICAP_FEATURE_CAPABLE,
        AP_AWIPS_CAPABLE,
        IOX_HARDWARE_CAPABLE,
        AUX_CLIENT_INTERFACE_CAPABLE,
        CLICKOS_FEATURE_SET,
        AP_TRAFFIC_DISTRIBUTION_STATISTICS_CAPABLE
]

 [27] device_detail.static_info.remote_lan.num_rlan_ports = 0
 [28] device_detail.static_info.remote_lan.rlan_slot_id = 0
 [29] device_detail.static_info.remote_lan.rlan_port_can_be_zero = false
 [30] device_detail.static_info.is_cisco_ap = true
 [31] device_detail.static_info.is_mm_opt = false
 [32] device_detail.static_info.ap_image_name =
 [33] device_detail.dynamic_info.ap_crash_data.ap_crash_file =
 [34] device_detail.dynamic_info.ap_crash_data.ap_radio_2g_crash_file =
 [35] device_detail.dynamic_info.ap_crash_data.ap_radio_5g_crash_file =
 [36] device_detail.dynamic_info.led_brightness_level = 8
```

```
 [37] device_detail.dynamic_info.led_state_enabled = true
 [38] device_detail.dynamic_info.reset_button_state = false
 [39] device_detail.dynamic_info.led_flash_enabled = true
 [40] device_detail.dynamic_info.flash_sec = 0
 [41] device_detail.dynamic_info.temp_info.degree = 0
 [42] device_detail.dynamic_info.temp_info.temp_status = AP_TEMP_STATUS_NORMAL
 [43] device_detail.dynamic_info.temp_info.heater_status =
AP_TEMP_HEATER_STATUS_BOTH_HEATERS_OFF
 [44] device_detail.wtp_version.backup_sw_version.version = 17
 [45] device_detail.wtp_version.backup_sw_version.release = 7
 [46] device_detail.wtp_version.backup_sw_version.maint = 1
 [47] device_detail.wtp_version.backup_sw_version.build = 11
 [48] device_detail.wtp_version.backup_sw_version.stringified_ver_info = 17.7.1.11
 [49] device_detail.wtp_version.mini_ios_version.version = 0
 [50] device_detail.wtp_version.mini_ios_version.release = 0
 [51] device_detail.wtp_version.mini_ios_version.maint = 0
 [52] device_detail.wtp_version.mini_ios_version.build = 0
 [53] device_detail.wtp_version.mini_ios_version.stringified_ver_info =
 [54] device_detail.wtp_version.hw_ver.version = 1
 [55] device_detail.wtp_version.hw_ver.release = 0
 [56] device_detail.wtp_version.hw_ver.maint = 0
 [57] device_detail.wtp_version.hw_ver.build = 0
 [58] device_detail.wtp_version.hw_ver.stringified_ver_info = 1.0.0.0
 [59] device_detail.wtp_version.sw_ver.version = 17
 [60] device_detail.wtp_version.sw_ver.release = 3
 [61] device_detail.wtp_version.sw_ver.maint = 5
 [62] device_detail.wtp_version.sw_ver.build = 43
 [63] device_detail.wtp_version.sw_ver.stringified_ver_info = 17.3.5.43
 [64] device_detail.wtp_version.boot_ver.version = 1
 [65] device_detail.wtp_version.boot_ver.release = 1
 [66] device_detail.wtp_version.boot_ver.maint = 2
 [67] device_detail.wtp_version.boot_ver.build = 4
 [68] device_detail.wtp_version.boot_ver.stringified_ver_info = 1.1.2.4
 [69] device_detail.wtp_version.sw_version = 17.3.5.43
 [70] ap_lag_enabled = false
 [71] ap_location.floor = 0
 [72] ap_location.location = default location
 [73] ap_services.monitor_mode_opt_type = ENM_MODE_TYPE_NONE
 [74] ap_services.ap_dhcp_server.is_dhcp_server_enabled = false
 [75] ap_services.sniffer_ap_ifid = 0
 [76] tag_info.misconfigured_tag = APMGR_TAGS_CONFIGURED
 [77] tag_info.tag_source = EWLC_TAG_SRC_DEFAULT
 [78] tag_info.is_ap_misconfigured = false
 [79] tag_info.is_policy_tag_misconfigured = false
 [80] tag_info.is_site_tag_misconfigured = false
 [81] tag_info.is_rf_tag_misconfigured = false
 [82] tag_info.is_flex_profile_misconfigured = false
 [83] tag_info.is_ap_profile_misconfigured = false
 [84] tag_info.is_rf_profile_24_misconfigured = false
 [85] tag_info.is_rf_profile_5_misconfigured = false
 [86] tag_info.is_ap_tag_registration_done = true
 [87] tag_info.resolved_tag_info.resolved_policy_tag = default-policy-tag
 [88] tag_info.resolved_tag_info.resolved_site_tag = default-site-tag
 [89] tag_info.resolved_tag_info.resolved_rf_tag = default-rf-tag
 [90] tag_info.policy_tag_info.policy_tag_name = default-policy-tag
 [91] tag_info.site_tag.site_tag_name = default-site-tag
 [92] tag_info.site_tag.ap_profile = default-ap-profile
 [93] tag_info.site_tag.flex_profile = default-flex-profile
 [94] tag_info.rf_tag.rf_tag_name = default-rf-tag
 [95] tag_info.rf_tag.dot11a_rf_profile = default_rf_5gh
 [96] tag_info.rf_tag.dot11b_rf_profile = default_rf_24gh
 [97] tag_info.filter_info.filter_name =
 [98] tunnel.preferred_mode = PREFERRED_MODE_IPV4
 [99] tunnel.udp_lite = IPV6_CAPWAP_UDPLITE_UNCONFIG
```

```
 [100] external_module_data.xm_data.is_module_present = false
 [101] external_module_data.xm_data.enable = true
 [102] external_module_data.xm_data.xm.goodness_field = [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
]

 [103] external_module_data.xm_data.xm.numeric_id = 12
 [104] external_module_data.xm_data.xm.version = [
        0,
        0
]

 [105] external_module_data.xm_data.xm.product_id = [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
]

 [106] external_module_data.xm_data.xm.serial_number = [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
]

 [107] external_module_data.xm_data.xm.max_power = 0
 [108] external_module_data.xm_data.xm.eeprom_size = [
```

```
        0,
        0,
        0,
        0
]

 [109] external_module_data.xm_data.xm.xm_cookie_version = 0
 [110] external_module_data.xm_data.xm.inventory.prod_id = C9115AXI-B
 [111] external_module_data.xm_data.xm.inventory.ver_id = 05
 [112] external_module_data.xm_data.xm.inventory.serial_num = FJC25331LCY
 [113] external_module_data.xm_data.xm.inventory.ent_name = C9115AX
 [114] external_module_data.xm_data.xm.inventory.ent_desc = Cisco Catalyst 9115AX Series
(IEEE 802.11ax) Access Point
 [115] external_module_data.xm_data.xm.module_name =
 [116] external_module_data.xm_data.xm.version_string =
 [117] external_module_data.xm_data.xm.serial_number_string =
 [118] external_module_data.xm_data.xm.product_id_string =
 [119] external_module_data.xm_data.xm.module_type =
 [120] external_module_data.xm_data.xm.module_description =
 [121] external_module_data.xm_data.xm.module_capabilities =
 [122] external_module_data.xm_data.xm.module_state =
 [123] external_module_data.usb_data.is_module_present = false
 [124] external_module_data.usb_data.enable = true
 [125] external_module_data.usb_data.xm.goodness_field = [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
]

 [126] external_module_data.usb_data.xm.numeric_id = 12
 [127] external_module_data.usb_data.xm.version = [
        0,
        0
]

 [128] external_module_data.usb_data.xm.product_id = [
        85,
        110,
        107,
        110,
        111,
        119,
        110,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
```

```
        0,
        0,
        0
]

 [129] external_module_data.usb_data.xm.serial_number = [
        85,
        110,
        107,
        110,
        111,
        119,
        110,
        0,
        0,
        0,
        0
]

 [130] external_module_data.usb_data.xm.max_power = 0
 [131] external_module_data.usb_data.xm.eeprom_size = [
        0,
        0,
        0,
        0
]

 [132] external_module_data.usb_data.xm.xm_cookie_version = 0
 [133] external_module_data.usb_data.xm.inventory.prod_id =
 [134] external_module_data.usb_data.xm.inventory.ver_id =
 [135] external_module_data.usb_data.xm.inventory.serial_num =
 [136] external_module_data.usb_data.xm.inventory.ent_name =
 [137] external_module_data.usb_data.xm.inventory.ent_desc =
 [138] external_module_data.usb_data.xm.module_name = Unknown
 [139] external_module_data.usb_data.xm.version_string = V00
 [140] external_module_data.usb_data.xm.serial_number_string = Unknown
 [141] external_module_data.usb_data.xm.product_id_string = Unknown
 [142] external_module_data.usb_data.xm.module_type = USB Module
 [143] external_module_data.usb_data.xm.module_description = Unknown
 [144] external_module_data.usb_data.xm.module_capabilities =
 [145] external_module_data.usb_data.xm.module_state = Not Detected
 [146] external_module_data.usb_override = false
 [147] external_module_data.is_ext_module_enabled = false
 [148] external_module_data.expansion_module_extended_info.power_sufficient = 0
 [149] external_module_data.expansion_module_extended_info.antenna_product_id = [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
```

```
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
]

 [150] external_module_data.expansion_module_extended_info.antenna_serial_number = [
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0,
        0
]

 [151] external_module_data.expansion_module_extended_info.antenna_prod_ID =
 [152] ipv6_joined = 0
 [153] wtp_ip_addr = 10.22.243.229
 [154] ap_state.ap_admin_state = ENM_ADMINSTATE_ENABLED
 [155] ap_state.ap_operation_state = EWLC_ENM_AP_STATE_REG
 [156] ap_mode_data.home_ap_enabled = false
 [157] ap_mode_data.clear_mode = false
 [158] ap_mode_data.ap_sub_mode = AP_SUB_MODE_NONE
 [159] ap_mode_data.wtp_mode = EWLC_ENM_SPAM_AP_MODE_LOCAL
 [160] ap_mode_data.ap_fabric_data.is_fabric_ap = false
 [161] ap_mode_data.ap_fabric_data.lisp_state = EWLC_ENM_LISP_QUERY_NOT_NEEDED
 [162] ap_time_info.boot_time = Fri, 05 Aug 2022 06:47:33 +0000
```

```
 [163] ap_time_info.join_time = Fri, 05 Aug 2022 06:50:13 +0000
 [164] ap_time_info.join_time_taken = 159
 [165] ap_time_info.last_up_time = 1
 [166] country_code = US
 [167] ap_security_data.lsc_provision_inprogress = false
 [168] ap_security_data.fips_enabled = false
 [169] ap_security_data.wlancc_enabled = false
 [170] ap_security_data.cert_type = EWLC_CERT_MIC
 [171] ap_security_data.lsc_ap_auth_type = EWLC_ENM_LSC_AP_AUTH_CAPWAP_DTLS
 [172] num_radio_slots = 2
 [173] dart_is_connected = false
 [174] dart_is_connected_str = Not Connected
 [175] is_master = false
 [176] sliding_window.multi_window_support = true
 [177] sliding_window.window_size = 1
 [178] ap_vlan.vlan_tag_state = VLAN_TAGGING_DISABLED
 [179] ap_vlan.vlan_tag_id = 0
 [180] capwap_iifid = 2415919114
 [181] hyperlocation_data.hyperlocation_method = HYPERLOCATION_METHOD_NONE
 [182] hyperlocation_data.per_ap_hl_tlv_rcvd = HYPERLOCATION_AP_TLV_RECEIVED
 [183] hyperlocation_data.cmx_ip = null
 [184] cdp_enable = true
 [185] cdp_cache_index_list.buffer = [
        1,
        0,
        0,
        0
]

 [186] ap_stationing_type = EWLC_ENM_INDOOR_AP
 [187] int_if_num = 0
 [188] radio_key = [
        {wtp_mac : 1CD1.E065.C340, radio_slot_id : 0},
        {wtp_mac : 1CD1.E065.C340, radio_slot_id : 1},
        {wtp_mac : 0000.0000.0000, radio_slot_id : 0},
        {wtp_mac : 0000.0000.0000, radio_slot_id : 0}
]

 [189] reboot_stats.reboots = 9
 [190] reboot_stats.ac_initiated = 4
 [191] reboot_stats.link_failure = 0
 [192] reboot_stats.sw_failure = 0
 [193] reboot_stats.hw_failure = 0
 [194] reboot_stats.unknown_failure = 0
 [195] reboot_stats.reboot_reason = AP_REBOOT_REASON_IMG_UPGRADE
 [196] reboot_stats.reboot_types = AP_REBOOT_SPAM_INITIATED
 [197] reboot_stats.reboot_type = AP_REBOOT_SPAM_INITIATED
 [198] slot_type = [
        0,
        0,
        0,
        0
]

 [199] mesh_profile_inuse =
 [200] mesh_ap_role = ENM_EWLC_AP_ROLE_MESH
 [201] wtp_cfg_reval_data.wtp_revalidate = false
 [202] wtp_cfg_reval_data.pending_wtp_notifies = 0
 [203] me_internal_ap = false
 [204] ap_type = AP_TYPE_CAPWAP
 [205] is_mewlc_candidate = false
 [206] is_invalid_master = false
 [207] is_callback_success = false
 [208] proxy_info.hostname =
```

```
 [209] proxy_info.port = 0
 [210] proxy_info.no_proxy_list =
 [211] grpc_enabled = true
 [212] ap_image_size = 0
 [213] ap_cur_bytes = 0
 [214] image_size_eta = 0
 [215] image_size_start_time = Thu, 01 Jan 1970 00:00:00 +0000
 [216] image_size_percentage = 0
 [217] dual_dfs_capable = false
 [218] mdns_group_id = 0
 [219] mdns_rule_name =
 [220] ap_keepalive_state = true
 [221] local_dhcp = false
 [222] ipv4_pool.network = 0.0.0.0
 [223] ipv4_pool.lease_time = 0
 [224] ipv4_pool.netmask = 0.0.0.0
 [225] wlc_image_size_eta = 0
 [226] wlc_image_size_start_time = Thu, 01 Jan 1970 00:00:00 +0000
 [227] wlc_image_size_percentage = 0
 [228] matching_ewc_image = false
 [229] disconnect_detail.ext_disconnect_reason_capable = false
 [230] disconnect_detail.disconnect_reason = UNKOWN
 [231] antenna_monitor.support = false
 [232] antenna_monitor.enabled = false
 [233] antenna_monitor.rssi_fail_threshold = 0
 [234] antenna_monitor.weak_rssi = 0
 [235] antenna_monitor.detection_time = 0
 [236] wtp_ip = 10.22.243.229
}
```

How do I view the current TDL values for an AP?

1. Execute the command on the wireless controller to retrieve the current configuration for an AP:

   ```
   test platform software database get ewlc_oper/ble_ltx_ap;ap_mac=<mac-without-dots>
   ```

   Replace <*mac-without-dots*> with the actual MAC address of the AP, removing any periods. For example:

   ```
   wireless controller# test platform software database get
   ewlc_oper/ble_ltx_ap;ap_mac=04eb409ec3c0
   ```

The output presents a list of parameters, such as:

- The AP's MAC address, without any delimiters.
- The administrative state of the AP.
- Details of the scan configuration, including intervals and states.
- Settings for the iBeacon and Eddystone profiles.
- Information on viBeacons profiles.
- Statistics on the types of scans performed.
- Host device data, such as the name and BLE MAC address.
- Current feature modes and the operational status of the device.
- Capabilities of the device, including support for technologies like BLE and Zigbee.

Each parameter provides details including the last report time and the validity of the status.

```
wireless controller# test platform software database get
ewlc_oper/ble_ltx_ap;ap_mac=04eb409ec3c0
Table Record Index 0 = {
 [0] ap_mac = 04EB.409E.C3C0
 [1] admin.state = BLE_LTX_ADMIN_STATE_ON
 [2] admin.feedback.state_status = 0
 [3] admin.report.last_report_time = Fri, 05 Jun 2020 07:26:18 +0000
 [4] admin.report.valid = true
 [5] scan_config.interval_sec = 1
 [6] scan_config.state = BLE_LTX_SCAN_STATE_ON
 [7] scan_config.max_value = 8
 [8] scan_config.window_msec = 800
 [9] scan_config.filter = BLE_LTX_SCAN_FILTER_ON
 [10] scan_config.feedback.interval_sec_status = 0
 [11] scan_config.feedback.state_status = 0
 [12] scan_config.feedback.max_value_status = 0
 [13] scan_config.feedback.window_msec_status = 0
 [14] scan_config.feedback.filter_status = 0
 [15] scan_config.report.last_report_time = Fri, 05 Jun 2020 07:26:18 +0000
 [16] scan_config.report.valid = true
 [17] profile_ibeacon.uuid = 00000000-0000-0000-0000-000000000000
 [18] profile_ibeacon.major = 0
 [19] profile_ibeacon.minor = 0
 [20] profile_ibeacon.tx_power = 0
 [21] profile_ibeacon.frequency_msec = 0
 [22] profile_ibeacon.adv_tx_power = 65
 [23] profile_ibeacon.feedback.uuid_status = 0
 [24] profile_ibeacon.feedback.major_status = 0
 [25] profile_ibeacon.feedback.minor_status = 0
 [26] profile_ibeacon.feedback.tx_power_status = 0
 [27] profile_ibeacon.feedback.frequency_msec_status = 0
 [28] profile_ibeacon.feedback.adv_tx_power_status = 0
 [29] profile_ibeacon.report.last_report_time = Fri, 05 Jun 2020 02:18:30 +0000
 [30] profile_ibeacon.report.valid = true
 [31] profile_eddy_url.url =
 [32] profile_eddy_url.feedback.url_status = 0
 [33] profile_eddy_url.report.last_report_time = Thu, 01 Jan 1970 00:00:00 +0000
 [34] profile_eddy_url.report.valid = false
 [35] profile_eddy_uid.namespace =
 [36] profile_eddy_uid.instance_id =
 [37] profile_eddy_uid.feedback.namespace_status = 0
 [38] profile_eddy_uid.feedback.instance_id_status = 0
 [39] profile_eddy_uid.report.last_report_time = Thu, 01 Jan 1970 00:00:00 +0000
 [40] profile_eddy_uid.report.valid = false
 [41] profile_vibeacons.common.interval_msec = 0
 [42] profile_vibeacons.common.feedback.interval_msec_status = 0
 [43] profile_vibeacons.common.report.last_report_time = Thu, 01 Jan 1970 00:00:00 +0000
 [44] profile_vibeacons.common.report.valid = false
 [45] profile_vibeacons.vibeacons = [
        {beacon_id : 0, uuid : , tx_power : 0, major : 0, minor : 0, adv_tx_power : 0,
status : BLE_LTX_VIBEACON_OFF,
feedback.beacon_id_status : 0, feedback.uuid_status : 0, feedback.tx_power_status : 0,
feedback.major_status : 0,
feedback.minor_status : 0, feedback.status_status : 0, feedback.adv_tx_power_status : 0,
report.last_report_time : Thu, 01 Jan 1970 00:00:00 +0000,
report.valid : false},
        {beacon_id : 1, uuid : , tx_power : 0, major : 0, minor : 0, adv_tx_power : 0,
status : BLE_LTX_VIBEACON_OFF,
feedback.beacon_id_status : 0, feedback.uuid_status : 0, feedback.tx_power_status : 0,
feedback.major_status : 0,
feedback.minor_status : 0, feedback.status_status : 0, feedback.adv_tx_power_status : 0,
report.last_report_time : Thu, 01 Jan 1970 00:00:00 +0000,
```

```
report.valid : false},
        {beacon_id : 2, uuid : , tx_power : 0, major : 0, minor : 0, adv_tx_power : 0,
status : BLE_LTX_VIBEACON_OFF,
feedback.beacon_id_status : 0, feedback.uuid_status : 0, feedback.tx_power_status : 0,
feedback.major_status : 0,
feedback.minor_status : 0, feedback.status_status : 0, feedback.adv_tx_power_status : 0,
report.last_report_time : Thu, 01 Jan 1970 00:00:00 +0000,
report.valid : false},
        {beacon_id : 3, uuid : , tx_power : 0, major : 0, minor : 0, adv_tx_power : 0,
status : BLE_LTX_VIBEACON_OFF,
feedback.beacon_id_status : 0, feedback.uuid_status : 0, feedback.tx_power_status : 0,
feedback.major_status : 0,
feedback.minor_status : 0, feedback.status_status : 0, feedback.adv_tx_power_status : 0,
report.last_report_time : Thu, 01 Jan 1970 00:00:00 +0000,
report.valid : false},
        {beacon_id : 4, uuid : , tx_power : 0, major : 0, minor : 0, adv_tx_power : 0,
status : BLE_LTX_VIBEACON_OFF,
feedback.beacon_id_status : 0, feedback.uuid_status : 0, feedback.tx_power_status : 0,
feedback.major_status : 0,
feedback.minor_status : 0, feedback.status_status : 0, feedback.adv_tx_power_status : 0,
report.last_report_time : Thu, 01 Jan 1970 00:00:00 +0000,
report.valid : false}
]

 [46] profile_vibeacons.report.last_report_time = Thu, 01 Jan 1970 00:00:00 +0000
 [47] profile_vibeacons.report.valid = false
 [48] scan_counters.total = 0
 [49] scan_counters.dna_ltx = 0
 [50] scan_counters.system_tlm = 0
 [51] scan_counters.event_tlm = 0
 [52] scan_counters.regular_tlm = 0
 [53] scan_counters.emergency = 0
 [54] scan_counters.event_emergency = 0
 [55] scan_counters.other = 0
 [56] scan_counters.report.last_report_time = Fri, 05 Jun 2020 07:26:18 +0000
 [57] scan_counters.report.valid = true
 [58] host_data.device_name = Developme
 [59] host_data.ble_mac = 806F.B031.E024
 [60] host_data.api_version = 1
 [61] host_data.fw_version = FF020710
 [62] host_data.advertise_count = 0
 [63] host_data.uptime_dsec = 10
 [64] host_data.active_profile = BLE_LTX_PROFILE_NO_ADV
 [65] host_data.report.last_report_time = Fri, 05 Jun 2020 07:26:18 +0000
 [66] host_data.report.valid = true
 [67] feature_mode.feature = BLE_LTX_FEATURE_ZIGBEE
 [68] feature_mode.mode = BLE_LTX_MODE_IOX
 [69] feature_mode.report.last_report_time = Fri, 05 Jun 2020 07:26:19 +0000
 [70] feature_mode.report.valid = true
 [71] device_status.device = BLE_LTX_DEVICE_MSM1
 [72] device_status.state = BLE_LTX_DEVICE_STATE_IOX_BLE_MODE
 [73] device_status.report.last_report_time = Fri, 05 Jun 2020 07:26:18 +0000
 [74] device_status.report.valid = true
 [75] capability.ble = true
 [76] capability.zigbee = true
 [77] capability.thread = false
 [78] capability.usb = true
 [79] capability.report.last_report_time = Wed, 03 Jun 2020 08:08:20 +0000
 [80] capability.report.valid = true
}
```

How do I get the telemetry connection status?

This procedure shows you how to check the telemetry connection status.

1. Enter the command:

   ```
   show telemetry internal protocol cloud-native manager <connector-ip-address> 8004
   source-address <source-IP-address>
   ```

   Replace <*connector-ip-address*> with the IP address of the connector and <*source-IP-address*> with the source IP address of your wireless controller.

2. In the output displayed, look for the **State** field to determine the telemetry connection status.

The following is a sample output of the command. The **State** is **CNDP_STATE_CONNECTED** and that indicates that the connection is successfully established

```
wireless controller# show telemetry internal protocol cloud-native manager 10.22.243.53
8004 source-address 10.22.243.52
Telemetry protocol manager stats:

Con str               : 10.22.243.53:8004:0:10.22.243.52
Sockfd                : 97
Protocol              : cloud-native
State                 : CNDP_STATE_CONNECTED
Table id              : 0
Wait Mask             :
Connection Retries    : 0
Send Retries          : 0
Pending events        : 0
Session requests      : 1
Session replies       : 1
Source ip             : 10.22.243.52
Bytes Sent            : 1121093
Msgs Sent             : 17613
Msgs Received         : 0
Creation time:        : Wed Jun  3 23:16:22:830
Last connected time:  : Wed Jun  3 23:16:22:892
Last disconnect time: :
Last error:           :
Connection flaps:     : 0
Last flap Reason:     :
Keep Alive Timeouts:  : 0
Last Transport Error  : No Error
```

How do I view IOx AP state and mode?

To view the Bluetooth Low Energy (BLE) state and mode for each AP connected to the wireless controller, you can perform the following steps:

1. On the wireless controller, enter the following command:

   ```
   show ap ble summary
   ```

The following example shows how to view the BLE state and mode for each AP.

This output provides a summary of each AP's BLE status, indicating whether it is active (**Up**) and the current BLE mode, which is **IOx** for all APs in this example.

```
wireless-controller# show ap ble summary
AP Name                          BLE AP State          BLE mode
-----------------------------------------------------------------
AP_10.2830                       Up                    IOx
AP_02.2898                       Up                    IOx
AP_06.28CC                       Up                    IOx
AP_08.28E0                       Up                    IOx
AP_07.28E4                       Up                    IOx
AP_09.28EC                       Up                    IOx
AP_01.28F0                       Up                    IOx
AP_03.2928                       Up                    IOx
AP_05.2934                       Up                    IOx
AP_04.2938                       Up                    IOx
```

How do I view gRPC details?

To view detailed gRPC (gRPC Remote Procedure Calls) statistics for a specific Access Point (AP), follow these steps:

1. Run the following command after replacing the <*AP Name*>:

   ```
   show ap name <AP Name> grpc detail
   ```

2. The output provides detailed gRPC statistics for the specified AP.

   In this output, the **gRPC channel status** indicates whether the connection is active (**Up**). The output also shows various packet statistics such as transmit attempts, transmit failures, packets received, and receive failures.

   The following is a sample output of the command:

   ```
   wireless-controller# show ap name ap-name grpc detail

   gRPC channel status        : Up
   Packets transmit attempts  : 818411
   Packets transmit failures  : 2651788
   Packets receive count      : 2711
   Packets receive failures   : 0
   ```

How do I view AP BLE configuration details?

To understand the Bluetooth Low Energy (BLE) configuration details for an AP, you can examine the output provided by your wireless controller. Run the following command, and replace <*ap-name*>.

```
show ap name <ap-name> ble detail
```

The command displays the detailed BLE configuration settings for an AP.

```
wireless-controller# show ap name ap-name grpc detail

Mode report time          : 06/25/2020 21:30:54
Mode                      : Advanced (IOx)
Radio mode                : BLE
Admin state report time   : 06/25/2020 21:31:14
Admin state               : Up
Interface report time     : 06/25/2020 21:30:58
Interface                 : MSM1
Interface state           : Open
Type                      : Integrated
```

```
Capability report time    : 06/25/2020 21:16:25
Capability                : BLE, Zigbee, USB,
Host data report time     : 06/25/2020 21:31:14
Host data
  Device name                 : AP_102830
  Dot15 Radio MAC             : 18:04:ed:c5:02:bc
  API version                 : 256
  FW version                  : 2.7.16
  Broadcast count             : -1844445184
  Uptime                      : 838860800 deciseconds
  Active profile              : No Advertisement
Scan Statistics report time    : 06/25/2020 21:30:36
Scan statistics
  Total scan records          : 0
Scan role report time  : 06/25/2020 21:31:14
Scan role
  Scan state                  : Enable
  Scan interval               : 1 seconds
  Scan window                 : 800 milliseconds
  Scan max value              : 8
  Scan filter                 : Enable
Broadcaster role
  Current profile type: iBeacon
  Last report time            : N/A
    UUID                      : Unknown
    Major                     : Unknown
    Minor                     : Unknown
    Transmit power            : Unknown
    Frequency                 : Unknown
    Advertised transmit power : Unknown
  Current profile type: Eddystone URL
  Last report time            : 06/25/2020 21:27:50
    URL                       : http://dnaspaces.io/edm
  Current profile type: Eddystone UID
  Last report time            : N/A
    Namespace                 : Unknown
    Instance id               : Unknown
  Current profile type: viBeacon
  Last report time            : N/A
    Interval                  : Unknown
    Beacon ID                 : 0
      UUID                      : Unknown
      Major                     : Unknown
      Minor                     : Unknown
      Transmit power            : Unknown
      Advertised transmit power : Unknown
      Enable                    : Unknown
    Beacon ID                 : 1
      UUID                      : Unknown
      Major                     : Unknown
      Minor                     : Unknown
      Transmit power            : Unknown
      Advertised transmit power : Unknown
      Enable                    : Unknown
    Beacon ID                 : 2
      UUID                      : Unknown
      Major                     : Unknown
      Minor                     : Unknown
      Transmit power            : Unknown
      Advertised transmit power : Unknown
      Enable                    : Unknown
    Beacon ID                 : 3
      UUID                      : Unknown
      Major                     : Unknown
```

```
    Minor                      : Unknown
    Transmit power             : Unknown
    Advertised transmit power  : Unknown
    Enable                     : Unknown
  Beacon ID                 : 4
    UUID                       : Unknown
    Major                      : Unknown
    Minor                      : Unknown
    Transmit power             : Unknown
    Advertised transmit power  : Unknown
    Enable                     : Unknown
```

Some of the output descriptors are described below:

1. **Mode Report Time:** This timestamp, **06/25/2020 21:30:54**, indicates when the AP mode was last reported.
2. **Mode:** The AP is set to an **Advanced (IOx)** operational mode.
3. **Radio Mode:** The radio is operating in **BLE** mode.
4. **Admin State Report Time:** As of **06/25/2020 21:31:14**, the administrative state of the AP was last reported.
5. **Admin State:** The AP is currently **Up** and operational.
6. **Interface Report Time:** The interface status was last reported on **06/25/2020 21:30:58**.
7. **Interface:** The interface identifier is **MSM1**.
8. **Interface State:** The interface is **Open** for connections.
9. **Type:** The AP has an **Integrated** interface type.
10. **Capability Report Time:** The capabilities were last reported on **06/25/2020 21:16:25**.
11. **Capability:** The AP supports **BLE**, **Zigbee**, and **USB** functionalities.
12. **Host Data Report Time:** This timestamp, **06/25/2020 21:31:14**, shows when the host data was last reported.
13. **Host Data:** It includes the AP's name **AP_102830**, its Dot15 radio MAC address **18:04:ed:c5:02:bc**, API version **256**, firmware version **2.7.16**, and other operational details.
14. **Scan Statistics Report Time:** The scan statistics were last reported on **06/25/2020 21:30:36**.
15. **Scan Statistics:** Indicates no total scan records are available.
16. **Scan Role Report Time:** The scan role was last reported on **06/25/2020 21:31:14**.
17. **Scan Role:** The AP is set to enable scanning with a **1-second** interval and an **800-millisecond** window. The maximum value is **8** and the scan filter is enabled.

How do I view the current TDL values for AP air quality?

To view the current Total Dissolved Load (TDL) values for AP air quality, perform the following steps:

1. Run the command to retrieve the TDL values:

```
test platform software database get-n all ewlc_oper/ap_air_quality
```

2. The command displays the current TDL values for all APs with air quality sensors. For example:

```
wireless controller# test platform software database get-n all ewlc_oper/ap_air_quality
Table Record Index 0 = {
[0] ap_mac = 687D.B45E.E7C0
[1] last_update = Tue, 12 Oct 2021 15:08:19 +0530
[2] rmox_0 = 5.62121e+07
[3] rmox_1 = 6.12815e+06
[4] rmox_2 = 1.26038e+06
[5] rmox_3 = 579564
[6] rmox_4 = 398259
[7] rmox_5 = 280246
[8] rmox_6 = 201467
[9] rmox_7 = 370324
[10] rmox_8 = 680235
[11] rmox_9 = 1.29709e+06
[12] rmox_10 = 3.18129e+06
[13] rmox_11 = 1.06436e+07
[14] rmox_12 = 6.10561e+07
[15] iaq = 1
[16] etoh = 0.0094
[17] eco2 = 400.212
[18] tvoc = 0.0178
}
```

In this example, the output provides the air quality data for an AP, including the MAC address, last update time, various rmox values, indoor air quality (iaq), ethanol (etoh), equivalent carbon dioxide (eco2), and total volatile organic compounds (tvoc).

How do I view the current TDL values for AP temperature and humidity?

To view the current Total Dissolved Load (TDL) values for AP temperature and humidity, please follow these steps:

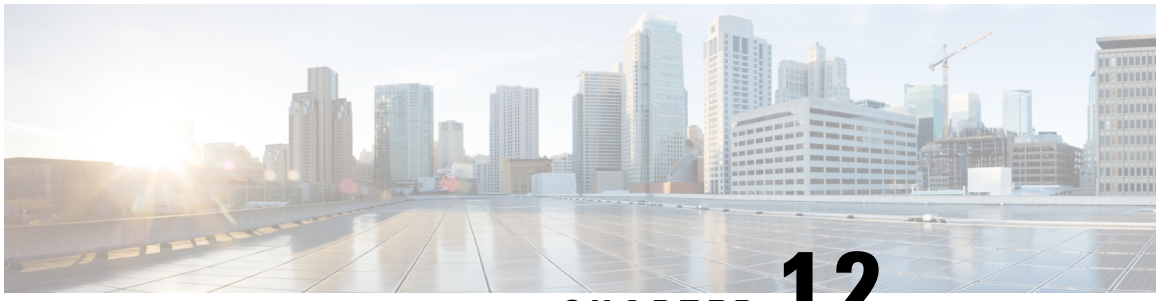
1. Execute the command to fetch the TDL values for temperature and humidity:

```
test platform software database get-n all ewlc_oper/ap_temp
```

2. This command shows the TDL values for all APs equipped with temperature and humidity sensors. For example:

```
wireless controller# test platform software database get-n all ewlc_oper/ap_temp

Table Record Index 0 = {
[0] ap_mac = 687D.B45E.E7C0
[1] last_update = Tue, 12 Oct 2021 15:08:19 +0530
[2] temp = 233.382
[3] humidity = 0
}
```

In this example, the output lists the temperature and humidity values, along with the MAC address of the AP and the last update timestamp.



CHAPTER 12

Troubleshooting IoT Services: IOx Application

How do I verify the IOx application is running on the AP?

Run the command: **show iox applications**

App State should be *RUNNING* to indicate if it is running.

```
AP# show iox applications
Total Number of Apps : 1
--------------------------
App Name                      : cisco_dnas_ble_iox_app
   App Ip                     : 192.168.11.2
   App State                  : RUNNING
   App Token                  : 02fb3e98-ac02-4356-95ba-c43e8a1f4217
   App Protocol               : ble
   App Grpc Connection        : Up
   Rx Pkts From App           : 3878345
   Tx Pkts To App             : 6460
   Tx Pkts To Wlc             : 0
   Tx Data Pkts To DNASpaces  : 3866864
   Tx Cfg Resp To DNASpaces   : 1
   Rx KeepAlive from App      : 11480
   Dropped Pkts               : 0
   App keepAlive Received On  : Mar 24 05:56:49
```

How do I debug the IOx application installation failure?

1. Ensure that the Network Time Protocol (NTP) server is synchronized with the Wireless Controller and APs in use.

2. Cisco Spaces: Connector installs the IoX Application. Download the AP image bundle from Cisco Spaces to Connector. Next, use the Cisco Application Framework (CAF) to install the image and launch the application from Cisco Spaces, primarily utilizing the ioxclient tool. For more information, see What is ioxclient?.
3. To examine the logs, you can either upload them to the Cisco Spaces or log into Cisco Spaces: Connector using SSH.
4. Observe the following critical logs:
   - `/opt/spaces-connector/runtime/logs/iot-services/server.log` : Records the initiation and completion of requests. It indicates when the main installation begins and the parameters it uses.
   - `/opt/spaces-connector/runtime/logs/iot-services/dnas_iox_app_manage.log`: Provides detailed information on the installation process.
5. To monitor the logs in real-time, do the following:
   - As a spacesadmin user, run the command, `tail -F /opt/spaces-connector/runtime/logs/iot-services/server.log`.
   - As a spacesadmin user, run the command, `tail -F /opt/spaces-connector/runtime/logs/iot-services/dnas_iox_app_manage.log`.

How do I verify the IoX Application AP bundle download from Cisco Spaces?

The IoX Application installation is done from the Cisco Spaces: Connector. The AP image bundle is downloaded from Cisco Spaces to Cisco Spaces: Connector. To verify if the IoX Application was downloaded accurately, you can check the log files. See How do I debug the IOx application installation failure?, on page 107

If the logs suggest a problem with the download, you can attempt to manually download the image. To manually download the image, log into Cisco Spaces: Connector via SSH. As a **spacesadmin** user, use the **wget** command:

```
spacesadmin# wget
"https://dnaspaces.io/api/edm/v1/device/iox-app/download?id=cisco_dnas_ble_iox_app&version=1.1.16"
```

How do I start an interactive shell session for the IOx application?

Run the command: **connect iox application**

This starts a shell which is running inside the IOx application container.

```
AP# connect iox application
/ #
```

How can I see the logs for the IOx application?

First, start an interactive shell using the **show iox application** command.

Then, run the command: **tail -F /data/logs/dnas_ble.log**

You can see the logs for the IOx application.

```
AP# tail -F /data/logs/dnas_ble.log
Tue Mar 24 06:55:21 2020 [INFO]: Starting DNA Spaces BLE IOx Application
Tue Mar 24 06:55:21 2020 [INFO]: Auth token file contents:
db26a8ab-e800-4fe9-a128-80683ea17b12
Tue Mar 24 06:55:21 2020 [INFO]: Setting gRPC endpoint to: 1.1.7.101:57777
Tue Mar 24 06:55:21 2020 [INFO]: Auth with token: db26a8ab-e800-4fe9-a128-80683ea17b12
Tue Mar 24 06:55:21 2020 [INFO]: Attempt to connect to DNAS Channel
Tue Mar 24 06:55:21 2020 [INFO]: Starting to run metrics
Tue Mar 24 06:55:21 2020 [INFO]: Starting to run Channel Keepalive
Tue Mar 24 06:55:21 2020 [INFO]: Initialize DNAS Reader Channel
Tue Mar 24 06:55:21 2020 [INFO]: Start listener for messages
Tue Mar 24 06:55:21 2020 [INFO]: Running BLE scan thread
```

How do I monitor metrics in the IOx application?

First, start an interactive shell using the **show iox application** command.

Run the command: **tail -F /data/logs/dnas_ble_metrics.log**

This command begins watching the log file for IOx application metrics. Metrics are updated every 30 seconds in the log file.

Table 6: Monitor Metrics

Metrics Name	Metrics Description
Application Version	The version number of the IOx application that is currently in use.
Start Time	The timestamp of when the application was initialized on the AP.
Up Time	The total time the application has been running since it was last started.
Total Physical Memory	The total RAM allocated to the application's container.
Physical Memory Free	The amount of RAM that remains unused in the application's container.
Physical Memory Used	The amount of RAM that is currently being used by the application's container.
Total Physical Shared Memory	The amount of memory shared amongst processes or containers.
Total Physical Buffer Memory	The memory dedicated to buffering, which aids in optimizing I/O operations.
Total AP Percent CPU Used	The percentage of the AP's CPU that is consumed by the application's container.

Metrics Name	Metrics Description
Process Virtual Memory	The virtual memory used by the application's process.
Process Physical Memory	The amount of physical RAM occupied by the application's process.
Process CPU Used	The CPU usage of the application's process.
gRPC Reconnect Count	The number of times a gRPC (remote procedure call) connection has been reestablished.
CAPWAP Restart Count	The number of restarts of the Control And Provisioning of Wireless Access Points (CAPWAP) protocol connection.
Last CAPWAP Restart Time	The timestamp marking the most recent CAPWAP connection restart.
BLE Device Open Count	The number of instances a Bluetooth Low Energy (BLE) device connection has been established.
Last BLE Device Open Time	The timestamp indicating the last occasion a BLE device was connected.
BLE Device Close Count	A count of disconnections of a BLE device.
Last BLE Device Close Time	The timestamp of the most recent closure of a BLE device connection.
Log Rotation Count	The frequency with which the log file (`dnas_ble.log`) has been archived and a new log started.
Floor Beacon Scan Data Message Count	The total count of BLE scan data messages since the application began.
Floor Beacon Scan Data Message Rate Per Second	The average creation rate of BLE scan data messages per second.
Floor Beacon Scan Data Write Count	The total number of BLE scan data packets transmitted since the start of the application.
Floor Beacon Scan Data Write Rate Per Second	The transmission rate of BLE scan data packets per second.
Floor Beacon Scan Data Message Count Per Write	The average count of BLE scan data messages included in each write operation.
Floor Beacon Scan Data Message Avg Write Time	The average duration it takes to write a BLE scan data packet.
Floor Beacon Config Request Count	The total number of floor beacon configuration requests since the application started.
Last Floor Beacon Config Request Time	The timestamp of the most recent request for floor beacon configuration.
Floor Beacon Config Success Count	The total number of successful floor beacon configuration requests.

Metrics Name	Metrics Description
Last Floor Beacon Config Success Time	The timestamp indicating the completion of the most recent successful floor beacon configuration.
Floor Beacon Config Failure Count	The count of floor beacon configuration requests that did not succeed.
Last Floor Beacon Config Failure Time	The timestamp of the last unsuccessful floor beacon configuration request.

```
AP# tail -F /data/logs/dnas_ble_metrics.log
Wed Oct 6 17:03:49 2021 [INFO]: Application Version: 1.2.5Wed Oct  6 17:03:49 2021 [INFO]:
 Start Time: Fri Sep 17 15:54:11 2021 Up Time:
        0019D:01H:09M:38S
Wed Oct 6 17:03:49 2021 [INFO]: Total Physical Memory: 1557 MBWed Oct 6 17:03:49 2021 [INFO]:
 Physical Memory Free: 786 MBWed Oct 6 17:03:49 2021 [INFO]: Physical Memory Used: 770 MBWed
 Oct 6 17:03:49 2021 [INFO]: Total Physical Shared Memory: 170 MBWed Oct 6 17:03:49 2021
[INFO]: Total Physical Buffer Memory: 0 MBWed Oct 6 17:03:49 2021 [INFO]: Total AP Percent
 CPU Used: 1.934973Wed Oct 6 17:03:49 2021 [INFO]: Process Virtual Memory: 108696 kBWed Oct
 6 17:03:49 2021 [INFO]: Process Physical Memory: 8828 kBWed Oct 6 17:03:49 2021 [INFO]:
Process CPU Used: 0.004167Wed Oct 6 17:03:49 2021 [INFO]: gRPC Reconnect Count: 0Wed Oct 6
 17:03:49 2021 [INFO]: CAPWAP Restart Count: 1Wed Oct  6 17:03:49 2021 [INFO]: Last CAPWAP
 Restart Time: Fri Sep 17 15:54:11
       2021
Wed Oct 6 17:03:49 2021 [INFO]: BLE Device Open Count: 1Wed Oct  6 17:03:49 2021 [INFO]:
Last BLE Device Open Time: Fri Sep 17 15:54:11
        2021
Wed Oct 6 17:03:49 2021 [INFO]: BLE Device Close Count: 1Wed Oct  6 17:03:49 2021 [INFO]:
Last BLE Device Close Time: Sat Sep 18 05:48:12
        2021
Wed Oct 6 17:03:49 2021 [INFO]: Log Rotation Count: 0Wed Oct  6 17:03:49 2021 [INFO]: Floor
 Beacon Scan Data Message Count:
       10896160
Wed Oct  6 17:03:49 2021 [INFO]: Floor Beacon Scan Data Message Rate Per Second:
       00
Wed Oct  6 17:03:49 2021 [INFO]: Floor Beacon Scan Data Write Count:
       217955
Wed Oct  6 17:03:49 2021 [INFO]: Floor Beacon Scan Data Write Rate Per Second:
       00
Wed Oct  6 17:03:49 2021 [INFO]: Floor Beacon Scan Data Message Count Per Write:
       50
Wed Oct  6 17:03:49 2021 [INFO]: Floor Beacon Scan Data Message Avg Write Time
         (milliseconds): 12
Wed Oct 6 17:03:49 2021 [INFO]: Floor Beacon Config Request Count: 0Wed Oct  6 17:03:49
2021 [INFO]: Last Floor Beacon Config Request Time:
       None
Wed Oct 6 17:03:49 2021 [INFO]: Floor Beacon Config Success Count: 0Wed Oct  6 17:03:49
2021 [INFO]: Last Floor Beacon Config Success Time:
       None
Wed Oct 6 17:03:49 2021 [INFO]: Floor Beacon Config Failure Count: 0Wed Oct  6 17:03:49
2021 [INFO]: Last Floor Beacon Config Failure Time:
    None
```

How do I monitor BLE scans in the IoX Application?

1. To monitor the IoX Application scan log file in real-time, execute the following command:

   ```
   tail -F /data/logs/dnas_ble_scans.log
   ```

2. This command will continuously display the log file's output as it updates with new scan information.
3. The IoX Application scans update every 5 minutes, but they may occur more frequently if the scan table becomes full.

Table 7: Output Descriptions

Field	Description
Profile	Beacon profile such as iBeacon, Eddystone URL, Eddystone UID, or Unknown.
MAC	MAC address of the beacon scanned.
RSSI	Last Received Signal Strength Indicator (RSSI) of the beacon detected.
Count	Number of times the beacon was heard since the last scan values were dumped.
Interval	Average interval between detections of the beacon.
Last-heard	Time elapsed since the beacon was last detected based on the latest scan values.

```
AP# tail -F /data/logs/dnas_ble_scans.log

Sat Sep 18 05:44:57 2021 [INFO]: Profile        MAC               RSSI  Count  Interval
Last-heard
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        00:00:00:00:00:0F 63    16           1S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Secure F1:01:AF:4E:8A:3B 55    1            0S
0000D:00H:00M:02S
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Telem  F1:01:AF:4E:8A:3B 55    1            0S
0000D:00H:00M:03S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        F1:01:AF:4E:8A:3C 56    1            0S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        D1:03:15:95:D6:F3 77    1            0S
0000D:00H:00M:03S
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Secure DF:03:AB:CD:C2:DB 86    2            3S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        DF:03:AB:CD:C2:DC 76    2            2S
0000D:00H:00M:02S
Sat Sep 18 05:44:57 2021 [INFO]: Unknown        18:04:ED:04:1C:5F 62    7            1S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Secure C3:05:7E:BD:25:D4 81    1            0S
0000D:00H:00M:04S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        C3:05:7E:BD:25:D5 85    3            1S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        CB:06:D8:B5:A7:97 86    1            0S
0000D:00H:00M:03S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        D8:06:04:DE:80:59 88    1            0S
0000D:00H:00M:04S
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Secure FF:07:D0:2F:6A:AF 79    1            0S
0000D:00H:00M:02S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        FF:07:D0:2F:6A:B0 79    3            1S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: Unknown        36:08:36:6C:DA:E8 81    5            1S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Secure C6:09:26:9D:4D:94 73    2            2S
0000D:00H:00M:01S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        C6:09:26:9D:4D:95 73    1            0S
0000D:00H:00M:02S
Sat Sep 18 05:44:57 2021 [INFO]: iBeacon        C1:0A:21:02:A7:D8 77    3            1S
0000D:00H:00M:02S
```

```
Sat Sep 18 05:44:57 2021 [INFO]: Kontakt Secure FD:0C:9B:17:A2:22 88    1           0S
0000D:00H:00M:03S
```

What files exist in the IOx application?

The following log files are generated when the application is running and are located in the directory **/data/logs**

Table 8: Log Files

Log File Name	Description
dnas_ble_scans.log	Active log file recording data on BLE devices scanned by the application.
dnas_ble.log	Active log file for debug messages, located in the temporary directory due to its high write frequency and to utilize the partition's I/O capabilities.
dnas_ble_metrics.log	Active log file that records metric messages related to the IOx application's performance and operations.
dnas_ble_last_restart.log	When the IOx application restarts, the current dnas_ble.log file is copied to this file to help troubleshoot the cause of the restart.
dnas_ble_metrics_last_restart.log	When the IOx application restarts, dnas_ble_metrics.log is copied to this file to aid in diagnosing the reasons for the restart based on the metrics recorded before it occurred.
dnas_ble_scans_last_restart.log	When the IOx application restarts, dnas_ble_scans.log is copied to this file to aid in diagnosing the reasons for the restart based on BLE scanning activity recorded prior to the restart.
dnas_ble_scans_1.log	A rotated log file for BLE device scans. It is part of the log file management system that helps control file size by archiving older entries.
dnas_ble_metrics_1.log	Rotated log file containing historical metric messages, also part of the log rotation strategy.
dnas_ble_1.log	Rotated log file that includes debug messages for the application, ensuring older logs are archived for size management.
dnas_ble_stdout.log	Log file capturing the standard output and error streams of the IOx application, which is useful for reviewing the application's console output and any error messages.

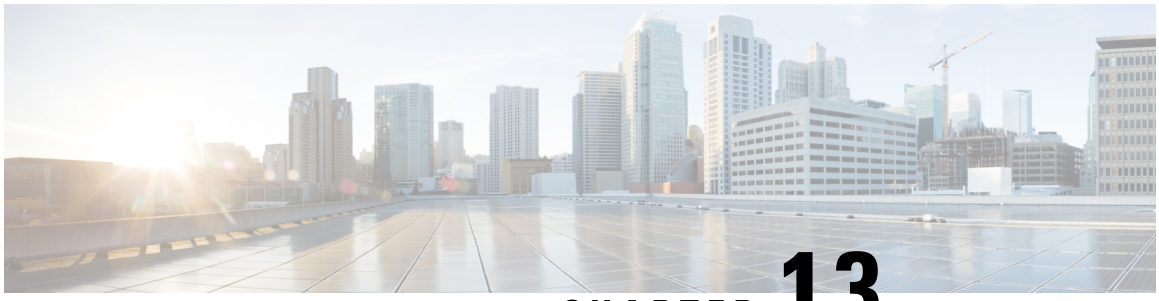
The following configuration files are generated when the application is running and are located in the directory **/data/logs**

Table 9: Configuration Files

Configuration File Name	Description
dnas_ble_config.json	Configuration settings for the BLE radio; these settings are used to reload the last configuration upon restart.

The following are binary files installed specifically for the IOx application. All the files are located in the directory: **/var/dnas_ble**

File Name	Description
/dnas_ble_iox_app	IOx application binary used to scan and configure floor beacons.
dnas_ble_iox_app_start.sh	Script to start the application and restart it in case of failure.



CHAPTER 13

Troubleshooting IoT Services: Cisco Spaces Connector

What are the metrics available on the Connector GUI for IoT Service (Wireless) ?

You can monitor these metrics on the connector GUI for the tile for IoT Service (Wireless).

Table 10: Monitor Metrics

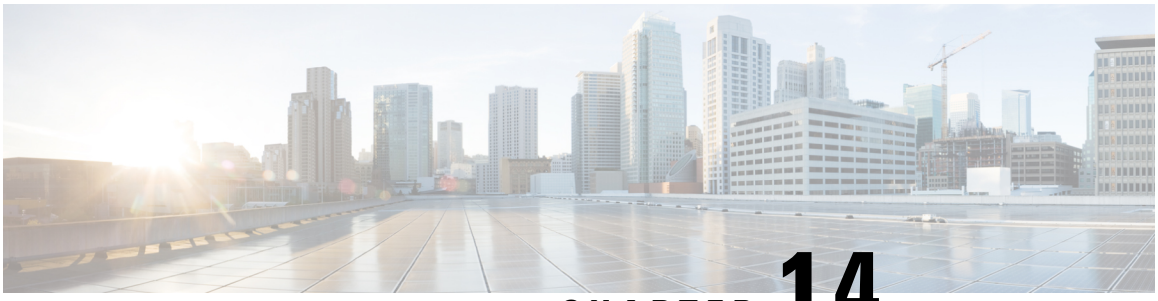
Metrics Name	Metrics Description
Mac Address	MAC address of the IoT Service (Wireless) on the connector
IP Address	IP address of the IoT Service (Wireless) on the connector
Log Level	Logging level set for the IoT Service (Wireless)
Incoming gRPC rate	The number of gRPC Remote Procedure Calls (gRPC) events the connector receives each second.
Incoming TDL rate	The number of TDL (Telemetry Definition Language) events the connector receives each second.
Incoming TDL failed rate	The number of TDL events per second that fail to be processed by the connector.
Last five minutes Incoming gRPC rate	The average rate of incoming gRPC events for the past five minutes.
Last five minutes TDL rate	The average rate of incoming TDL events for the past five minutes.
Last five minutes TDL failed rate	The average rate of incoming TDL events that failed in the last five minutes.
Active gRPC connection count	The current count of active gRPC connections to the connector.

What are the log files created on the Connector for IoT Service (Wireless)?

The following log files are located in the directory /opt/spaces-connector/runtime/logs/iot-services/.

Table 11: Log Files

Log File Name	Description
apgrpcchannel.log	Active log file recording data on BLE devices scanned by the application.
apgrpcchannel.log	This log file records the connection status of the Access Point's gRPC (gRPC Remote Procedure Calls) channel.
boot.log	This log file contains boot information such as CPU and memory details.
control-channel.log	This log file monitors the status of the control channel connection.
dnas_iox_app_manage.log	This log file pertains to the management of the IoX Application environment, including installation, uninstallation, and technical support actions.
filter.log	This log file is related to the filter configuration activities.
heartbeat.log	This log file captures heartbeat messages sent to the service manager.
highavailability.log	This log file details the status of high availability features.
metrics.log	This log file contains metric data formatted in JavaScript Object Notation (JSON).
netconf-service/server.log	This log file records operations related to Network Configuration Protocol (NETCONF).
nginx-access.log	This log file captures access records for NGINX.
nginx-error.log	This log file documents error messages related to NGINX.
server.log	This log file includes general messages and information.
status.log	This log file provides updates on the status of the system or service.



CHAPTER 14

Troubleshooting IoT Services: Access Point

How do I check the gRPC connection status on the access point?

Run the command: **show cloud connector connection detail**

This command returns information about the connection. ***Connection State*** should be READY. ***Connection Url*** should be the IP address of the Cisco Spaces: Connector on port 8000. ***Certificate Available*** should be true. ***Controller Ip*** should be the controller the AP is associated with.

```
AP# show cloud connector connection detail
Connection State           : READY
Connection Url             :  10.22.243.33:8000
Certificate Available      :  true
Controller Ip              :  10.22.243.31
Stream Setup Interval      :  30
Keepalive Interval         :  30
Last Keepalive Rcvd On     :  2020-04-01 00:32:47.891433113 +0000 UTC m=+345985.338898246
Number of Dials               :  2
Number of Tx Pkts             :  2788175
Number of Rx Pkts             :  11341
Number of Dropped Pkts        :  0
Number of Rx Keepalive        :  11341
Number of Tx Keepalive        :  11341
Number of Rx Cfg Request      :  0
Number of Tx AP Cfg Resp      :  0
Number of Tx APP Cfg Resp     :  0
Number of Tx APP state pkts   :  5
Number of Tx APP data pkts    :  2776829
```

How do I check the stream token on the access point?

Run the command: **show cloud connector key access**

This command returns information about the stream token. ***Token Valid*** should be Yes. The ***Last Success on*** time should be more recent than the ***Last Failure on*** time. If there are failures, the ***Last Failure reason*** field details the reason for the failure.

```
AP# show cloud connector key access
Token Valid : Yes
Token Stats :
        Number of Attempts  : 44
        Number of Failures  : 27
        Last Failure on     : 2020-03-28 02:02:15.649556818 +0000 UTC m=+5753.097022576
        Last Failure reason : curl: SSL connect error
        Last Success on     : 2020-04-01 00:48:37.313511596 +0000 UTC m=+346934.760976625
        Expiration time     : 2020-04-02 00:48:37 +0000 UTC
Connection Retry Interval : 30
```

Also run the command: **show cloud connector key authentication**.

This command returns the authentication token used initially to set up the connection. ***Token Valid*** should be Yes. ***Token Endpoint*** should be the IP address of the Cisco Spaces connector on port 8000. ***Token Content*** should be the token set on the wireless controller using this configuration command: **ap cisco-dna token 0** *token-content*.

```
AP# show cloud connector key authentication
Token Valid    : Yes
Token Endpoint : 10.22.243.33:8000
Token Content  :
----------------------------------------
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ
0aWQiOjEwMTM3LCJjaWQiOjExNTM5MzM4MDQ5NDM
3MzM2MDAwLCJlcCI6IjEwLjIyLjI0My4zMzo4MDA
wIiwiaWF0IjoxNTg1MzYwNzk5fQ.tj0LQDzXorHz
30Q_ffRWWY6Ege3xyF_tgQOmVFBkG3k
----------------------------------------
```

How do I view the gRPC server logs on the access point?

Run the command: **show grpc server log**

```
AP# show grpc server log
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] spaces conn url 10.22.243.33:8000"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] entering stopDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] exiting stopDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] entering startDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] launching token request cycle"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] exiting startDNAspacesTmpTokenRoutine"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] spaces token expiration time 2020-04-02
 01:36:52 +0000 UTC"
time="2020-04-01T01:36:52Z" level=info msg=" Calling startDNASpacesConn routine "
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] Receive Success status"
time="2020-04-01T01:36:52Z" level=info msg="[DNAS] Connection not in ready state sleeping
for 10 seconds"
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] Setup Stream for the gRPC connection"
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] Connect RPC Succeeded."
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] RX  routine got enabled "
time="2020-04-01T01:37:02Z" level=info msg="[DNAS] TX routine got enabled "
```

How do I view the beacons scanned by an access point running in Native Mode?

Run the command: **show controllers ioTRadio ble 0 scan brief**

```
<access-point># show controllers ioTRadio ble 0 scan brief

      Profile               MAC  RSSI(-dBm)  RSSI@1meter(-dBm)        Last-heard
      Unknown 3C:1D:AF:62:EC:EC          88                 0 0000D:00H:00M:01S
      iBeacon 18:04:ED:04:1C:5F          86                65 0000D:00H:00M:01S
      Unknown 18:04:ED:04:1C:5F          78                65 0000D:00H:00M:01S
      Unknown 04:45:E5:28:8E:E7          85                65 0000D:00H:00M:01S
      Unknown 2D:97:FA:0F:92:9A          91                65 0000D:00H:00M:01S
      iBeacon E0:7D:EA:16:35:35          68                65 0000D:00H:00M:01S
      Unknown E0:7D:EA:16:35:35          68                65 0000D:00H:00M:01S
      iBeacon 04:EE:03:53:74:22          45               256 0000D:00H:00M:01S
      Unknown 04:EE:03:53:74:22          45               256 0000D:00H:00M:01S
              04:EE:03:53:6A:3A          72               N/A 0000D:00H:00M:01S
      Unknown 04:EE:03:53:6A:3A          72                65 0000D:00H:00M:01S
      iBeacon E0:7D:EA:16:35:35          68                65 0000D:00H:00M:01S
      Unknown E0:7D:EA:16:35:35          67                65 0000D:00H:00M:01S
      iBeacon 04:EE:03:53:74:22          60               256 0000D:00H:00M:01S
      Unknown 04:EE:03:53:74:22          60               256 0000D:00H:00M:01S
Eddystone URL 04:EE:03:53:6A:3A          72               N/A 0000D:00H:00M:01S
```

How do I view the beacon broadcast setting for an access point running in Native Mode?

Run the command: **show controllers ioTRadio ble 0 broadcast**

Following is an example of two *iBeacons* configured in the v-iBeacon profile. Beacon 1 has ***UUID***: 11111111-1111-1111-1111-111111111111 ***Major***: 22222***Minor***: 33 ***Transmit Power***: -21 and ***Advertisement Power***: -60 ***Interval***: 100. Beacon 1 has ***UUID***: 22222222-2222-2222-2222-222222222222 ***Major***: 3333 ***Minor***: 44 ***Transmit Power***: -21 and ***Advertisement Power***: -65 ***Interval***: 100.

```
AP# show controllers ioTRadio ble 0 broadcast

BLE Profile Config
------------------
Active profile           : v-iBeacon
Profile 0 (iBeacon)
UUID                     : 00001000000000000000000000000000
Interval (ms)            : 100
Power (dBm)              : -21
Advertised Power (dBm)   : -65
Minor                    : 0
Major                    : 0
TxPower byte             : bfbfbfbfbfbfbfbfbfbfbfbf

Profile 1 (Eddystone UID)
Namespace (hex)          : 000000000005446089c
Instance-ID (hex)        : 7f0000001f00

Profile 2 (Eddystone URL)
URL                      : http://www.
```

```
Profile 3 (v-iBeacon)
v-iBeacon status         : Chirping
Chirping interval (ms)   : 100
Profile 4 (Custom Profile)
Adv Data                 :
0000001800000000000000ecb255ad550000000c0000000000000000000000
Scan Data                :
00000000ae01000000000000005446089c7f000000190000000000000004cb5
Simulator mode           : Disabled
Beacon-ID               Mac                                UUID Major Minor Status
        1 C0:64:E4:23:7F:2F 11111111111111111111111111111111 22222    33      1
        2 C0:64:E4:23:7F:2E 22222222222222222222222222222222 33333    44      1
        3 C0:64:E4:23:7F:2D 00000000000000000000000000000000     0     0      0
        4 C0:64:E4:23:7F:2C 00000000000000000000000000000000     0     0      0
        5 C0:64:E4:23:7F:2B 00000000000000000000000000000000     0     0      0

Beacon-ID Transmit power(dBm) Advertised power(dBm)
        1                -21                   -60
        2                -21                   -65
        3                -21                   -65
        4                -21                   -65
        5                -21                   -65
```

Following is an example of *Eddystone UID* profile. Beacon has ***Namespace***: 44444444444444444444 ***Instance-ID***: 555555555555 ***Transmit Power***: -21and ***Advertisement Power***: -65 ***Interval***: 100.

```
AP# show controllers ioTRadio ble 0 broadcast

BLE Profile Config
------------------
Active profile           : Eddystone UID
Profile 0 (iBeacon)
UUID                     : 00001000000000000000000000000000
Interval (ms)            : 100
Power (dBm)              : -21
Advertised Power (dBm)   : -65
Minor                    : 0
Major                    : 0
TxPower byte             : bfbfbfbfbfbfbfbfbfbfbfbf

Profile 1 (Eddystone UID)
Namespace (hex)          : 44444444444444444444
Instance-ID (hex)        : 555555555555

Profile 2 (Eddystone URL)
URL                      : http://www.

Profile 3 (v-iBeacon)
v-iBeacon status         : Chirping
Chirping interval (ms)   : 100
Profile 4 (Custom Profile)
Adv Data                 :
0000001800000000000000ecb255ad550000000c0000000000000000000000
Scan Data                :
00000000ae01000000000000005446089c7f000000190000000000000004cb5
Simulator mode           : Disabled
Beacon-ID               Mac                                UUID Major Minor Status
        1 C0:64:E4:23:7F:2F 11111111111111111111111111111111 22222    33      1
        2 C0:64:E4:23:7F:2E 22222222222222222222222222222222  3333    44      1
        3 C0:64:E4:23:7F:2D 00000000000000000000000000000000     0     0      0
        4 C0:64:E4:23:7F:2C 00000000000000000000000000000000     0     0      0
        5 C0:64:E4:23:7F:2B 00000000000000000000000000000000     0     0      0
```

```
Beacon-ID Transmit power(dBm) Advertised power(dBm)
        1                -21                   -60
        2                -21                   -65
        3                -21                   -65
        4                -21                   -65
        5                -21                   -65
```

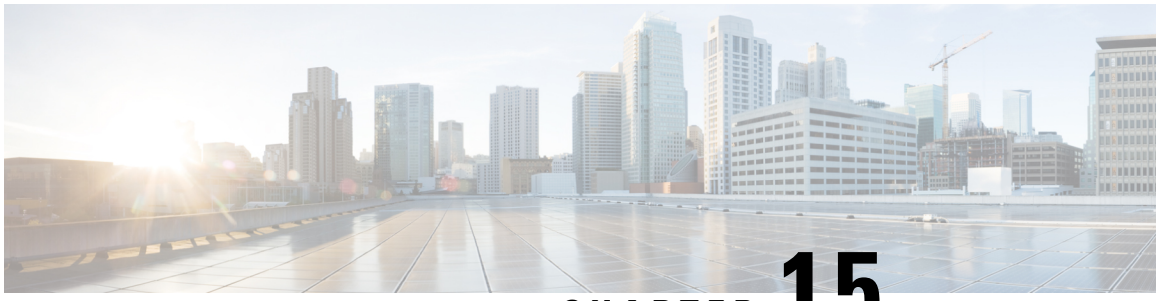
Following is an example of *Eddystone URL* profile. Beacon has ***URL***: http://www.cisco.com/ ***Transmit Power***: -21 and ***Advertisement Power***: -65 ***Interval***: 100.

```
AP# show controllers ioTRadio ble 0 broadcast

BLE Profile Config
------------------
Active profile           : Eddystone URL
Profile 0 (iBeacon)
UUID                     : 00001000000000000000000000000000
Interval (ms)            : 100
Power (dBm)              : -21
Advertised Power (dBm)   : -65
Minor                    : 0
Major                    : 0
TxPower byte             : bfbfbfbfbfbfbfbfbfbfbfbf

Profile 1 (Eddystone UID)
Namespace (hex)          : 44444444444444444444
Instance-ID (hex)        : 555555555555

Profile 2 (Eddystone URL)
URL                      : http://www.cisco.com/

Profile 3 (v-iBeacon)
v-iBeacon status         : Chirping
Chirping interval (ms)   : 100
Profile 4 (Custom Profile)
Adv Data                 :
0000001800000000000000ecb255ad550000000c00000000000000000000
Scan Data                :
00000000ae0100000000000005446089c7f000000190000000000000004cb5
Simulator mode           : Disabled
Beacon-ID              Mac                                 UUID Major Minor Status
        1 C0:64:E4:23:7F:2F 11111111111111111111111111111111 22222    33      1
        2 C0:64:E4:23:7F:2E 22222222222222222222222222222222  3333    44      1
        3 C0:64:E4:23:7F:2D 00000000000000000000000000000000     0     0      0
        4 C0:64:E4:23:7F:2C 00000000000000000000000000000000     0     0      0
        5 C0:64:E4:23:7F:2B 00000000000000000000000000000000     0     0      0

Beacon-ID Transmit power(dBm) Advertised power(dBm)
        1                -21                   -60
        2                -21                   -65
        3                -21                   -65
        4                -21                   -65
        5                -21                   -65
```

How do I view the beacon broadcast setting for an access point running in Native Mode?

PART VI

Appendix



CHAPTER 15

Cisco Catalyst 9800 Series Wireless Controller

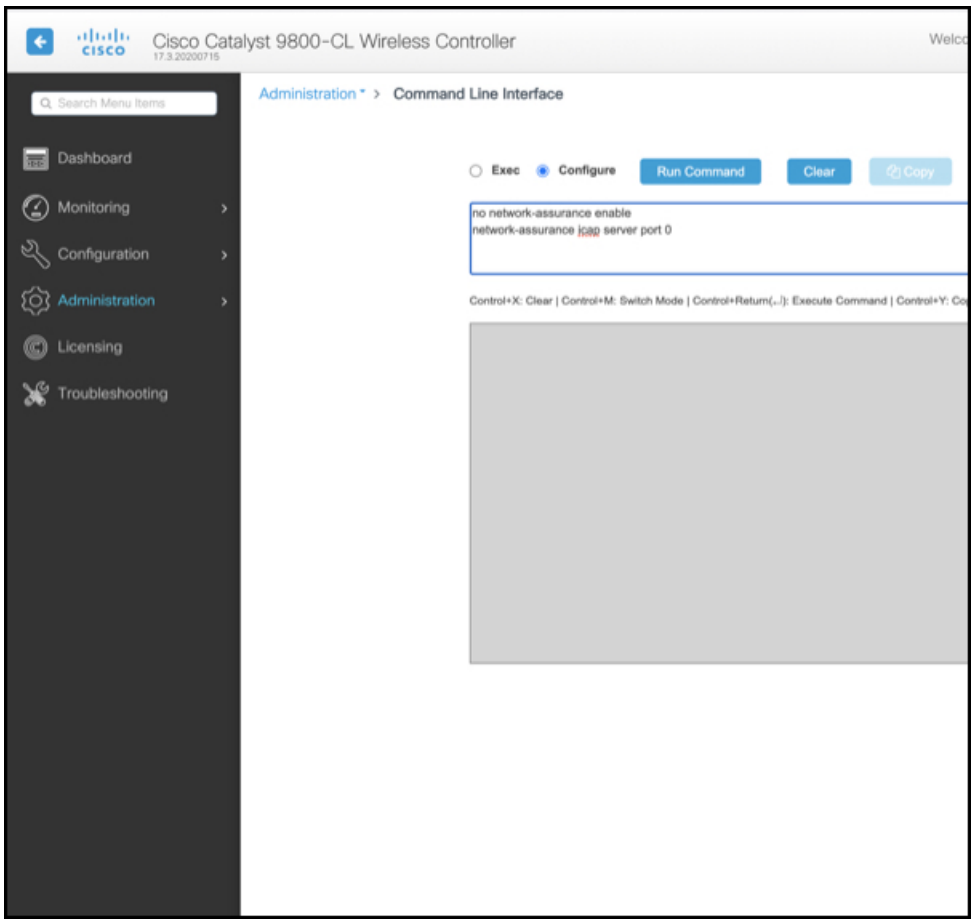
Disable Assurance with iCAP using GUI (Versions 17.3.1 or lower)

This task is applicable only for Cisco Catalyst 9800 Series Wireless Controller versions 17.3.1 or lower.

Disable Assurance with Intelligent Capture (iCAP) in order to enable IoT Service. With the wireless controller WebUI, you can issue CLI commands to disable assurance and iCAP.

Step 1 Log in to the Cisco Catalyst 9800 Series Wireless Controller GUI and navigate to **Administration > Command Line Interface.** Click Configure and enter the **no network-assurance enable** command and the **network-assurance icap server port 0** command.

Figure 76: Entering the commands to enable BLE



Step 2 Click **Run Command**.
If the command runs successfully, you can see a success message displayed.

What to do next

Assurance and iCAP are now disabled. You can add this Cisco Catalyst 9800 Series Wireless Controller to Cisco Spaces. If the Cisco Catalyst 9800 Series Wireless Controller was previously added to Catalyst Center (version 2.2 and above), the Catalyst Center can automatically categorize this device as a noncompliant device. No further action is thus required to make the Cisco Catalyst 9800 Series Wireless Controller work on Cisco Spaces.

Disable Assurance with iCAP using CLI (Versions 17.3.1 or lower)

This task is applicable only for Cisco Catalyst 9800 Series Wireless Controller versions 17.3.1 or lower.

This task uses the CLI to disable assurance including internet Content Adaptation Protocol (iCAP). Login to the Cisco Catalyst 9800 Series Wireless Controller CLI and enter the following commands.

SUMMARY STEPS

1. configure terminal
2. no network-assurance enable
3. network-assurance icap server port 0
4. end

DETAILED STEPS

Step 1 configure terminal

Step 2 no network-assurance enable

Step 3 network-assurance icap server port 0

Step 4 end

What to do next

Assurance and iCAP are now disabled. You can add this Cisco Catalyst 9800 Series Wireless Controller to Cisco Spaces. If the Cisco Catalyst 9800 Series Wireless Controller was previously added to Catalyst Center (version 2.2 and above), the Catalyst Center can automatically categorize this device as a noncompliant device. No further action is thus required to make the Cisco Catalyst 9800 Series Wireless Controller work on Cisco Spaces.

Disable iCAP using WEBUI (Versions 17.3.2 or higher)

This task is applicable only for Cisco Catalyst 9800 Series Wireless Controller versions 17.3.2 or higher.

Cisco Catalyst 9800 Series Wireless Controller running Cisco IOS XE Amsterdam 17.3.x supports only one of the following:

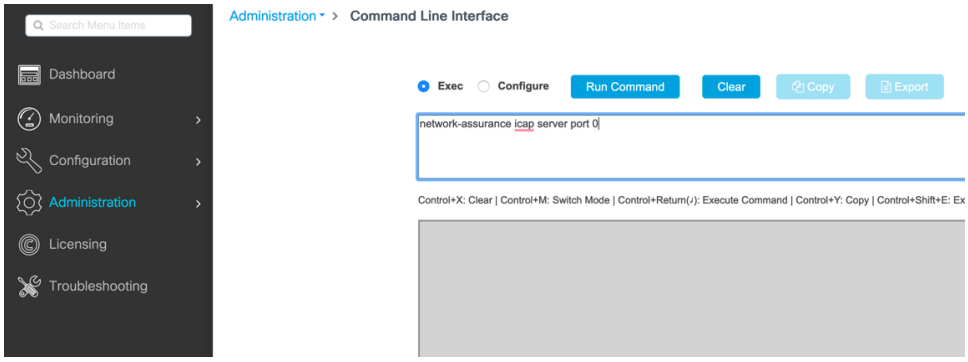
- IoT service (wireless) with Cisco Spaces.
- Network Assurance solution on Catalyst Center using Intelligent Capture (iCAP)

IoT service (wireless) and Intelligent Capture (iCAP) can co-exist from Cisco IOS XE Cupertino 17.7.x or higher.

Disable Intelligent Capture (iCAP) in order to enable IoT service (wireless). With the wireless controller GUI, you can issue CLI commands to disable iCAP.

Step 1 Log in to the Cisco Catalyst 9800 Series Wireless Controller WebUI and navigate to **Administration > Command Line Interface.** Click Configure and enter the **network-assurance icap server port 0** command.

Figure 77: Entering the commands to enable IoT Service



Step 2 Click **Run Command**.
If the command runs successfully, you can see a success message displayed.

What to do next

Intelligent Capture (iCAP) feature is now disabled. You can now add this Cisco Catalyst 9800 Series Wireless Controller to Cisco Spaces

If this wireless controller was previously added to Catalyst Center (version 2.2 and above), Catalyst Center now categorizes this device as a noncompliant device allowing Cisco Spaces to push the necessary configurations to the device. No further action is thus required to make the wireless controller work on Cisco Spaces.

Disable iCAP using CLI (Versions 17.3.2 or higher)

This task uses the CLI to disable Intelligent Capture (iCAP). Login to the Cisco Catalyst 9800 Series Wireless Controller CLI and enter the following commands.

This task is applicable only for Cisco Catalyst 9800 Series Wireless Controller versions 17.3.2 or higher.

Cisco Catalyst 9800 Series Wireless Controller running Cisco IOS XE Amsterdam 17.3.x supports only one of the following:

- IoT service (wireless) with Cisco Spaces.
- Network Assurance solution on Catalyst Center using Intelligent Capture (iCAP)

SUMMARY STEPS

1. configure terminal
2. network-assurance icap server port 0
3. end

DETAILED STEPS

Step 1 configure terminal

Step 2 network-assurance icap server port 0

Step 3 end

What to do next

Intelligent Capture (iCAP) feature is now disabled. You can now add this Cisco Catalyst 9800 Series Wireless Controller to Cisco Spaces

If this wireless controller was previously added to Catalyst Center (version 2.2 and above), Catalyst Center now categorizes this device as a noncompliant device allowing Cisco Spaces to push the necessary configurations to the device. No further action is thus required to make the wireless controller work on Cisco Spaces.

Enable or Disable iCAP or Assurance using DNAC (Versions 17.3.2 or higher)

This task shows you how you can disable or enable the network-assurance or iCAP feature using the Catalyst Center templates.

Step 1 From the Catalyst Center dashboard, use the template editor to create a template with the required configuration. Specify the template name, description, software type, and device type.

Step 2 Save and commit the template.

Step 3 Add the template to the respective site.

Step 4 Select the device from the site and provision the device.

Step 5 In **Advanced Configuration**, select the template and apply to the device.