Aguardio

# HPE ARUBA INTEGRATION GUIDE

## AOS 8. X Controller- BLE Config Workflow Summary



Aguardio's smart IoT sensors deliver unique data from water pipes and bathrooms. Digitalization of pipes with sensors enhances water & energy management plus optimizes buildings & operations via data (both for cold & hot water plus water for heating). The Pipe Sensor e.g. monitors water activity for water pipes and toilets (e.g. flushes), detects leaks, and enables Legionella risk management plus delivers data for optimization of heating.

Click here to download Aguardio HUB guide

### 1. Controller Config Workflow Checklist
### (items 1-3 are required, item 4 needed only if AP is a beacon)

- Radio Profile - Make sure Operational mode is set to scan or both.
- Transport Profile - Check if the correct filtering is configured and the websocket connection is alive. Make sure the authentication method is correctly set.
- Trusted Root Certificate Assignment - Configured for all controllers that transport IoT data. Make sure that the correct DigiCert certificate is uploaded.
- BLE Service Profile - Optional (configured for all relevant AP groups). Requires radio profile beacon mode enabled and a defined Advertisement format. Note: Option only if AP needs to be in beacon mode.

## 2. Controller Setup Workflow

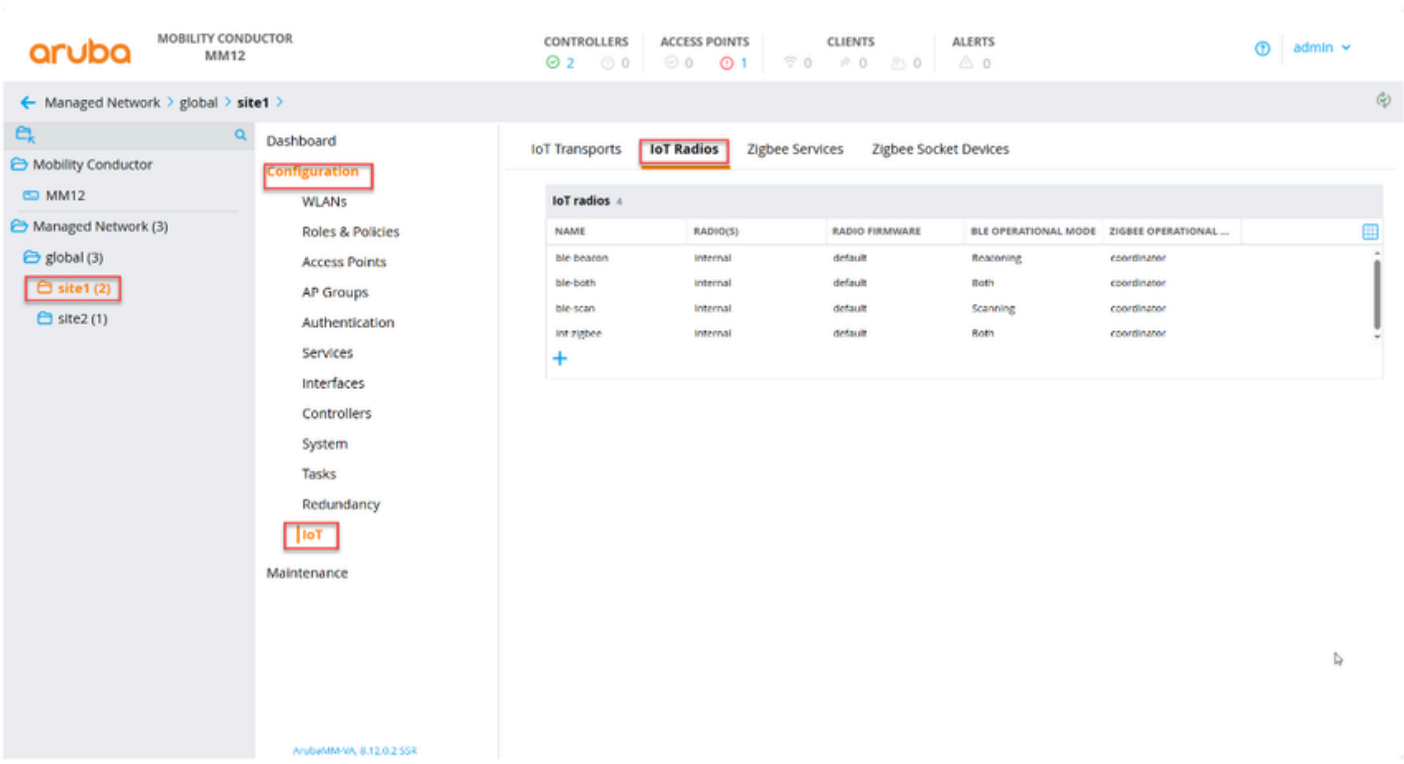
### 2.1 Radio Profile – configure for all AP group(s) in the proper config hierarchy in your Mobility Conductor:

**a.** Navigation Path: Managed Network – Node – Configuration – IoT – IoT Radios

**b.** Add a new profile with a local name that describes the function (e.g. ble-scan, ble-beacon, etc.). Provide a table of necessary parameters and values for your integration.

**c.** Select "**Submit**" then "**Pending Changes/Deploy Changes**" to enable the configuration push.

**d.** Repeat steps b-c for each config folder in your network hierarchy as required.




Example Navigation Path (site1 controller cluster)

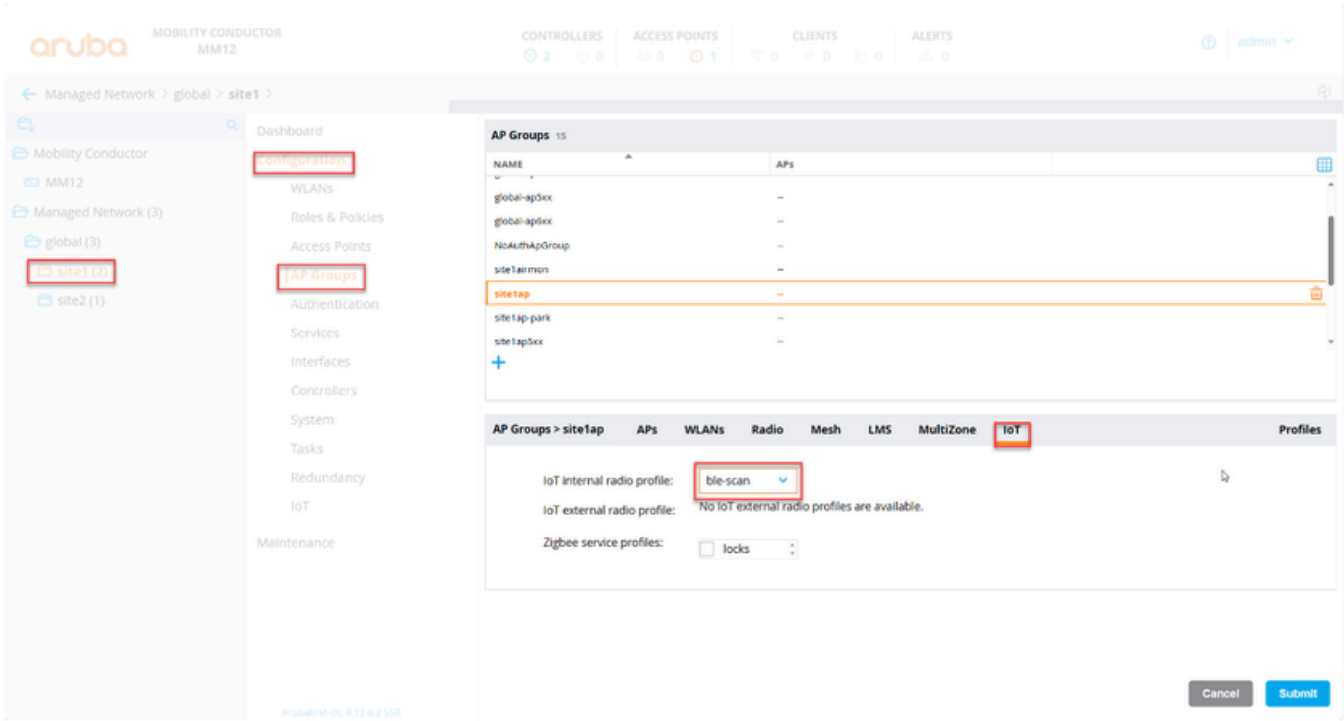
## 2.2 Radio Profile Assignment - Assign Radio Profile to each AP group(s) per previous step, as required.

**a.** Navigation Path: Managed Network – Node – Configuration – AP Groups – Select AP Group – IoT.

**b.** Select the radio profile previously configured (from step 1)

**c.** Select "**Submit**" then "**Pending Changes/Deploy Changes**" to enable the configuration push.

**d.** Repeat steps a-c for each required AP Group assignment.




Example Radio Profile AP Group Assignment (site1 AP Group)

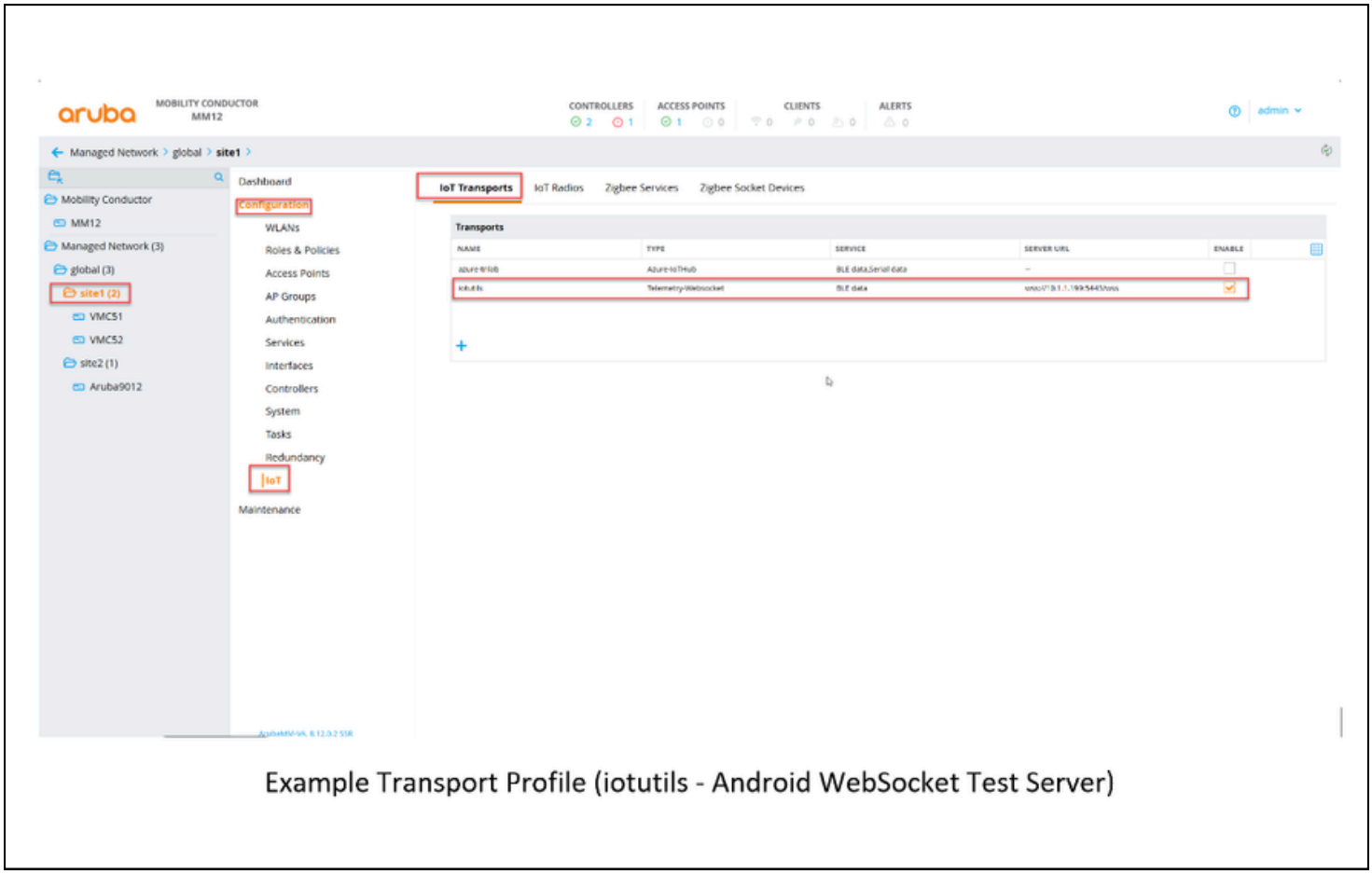
## 2.3 Transport Profile – config data types, filters, server endpoint and authentication to application.

**a.** Navigation Path: Managed Network – Node – Configuration – AP Groups – Select AP Group – IoT.

**b.** Select the radio profile previously configured (from step 1)

**c.** Select "**Submit**" then "**Pending Changes/Deploy Changes**" to enable the configuration push.

**d.** Repeat steps a-c for each required AP Group assignment.

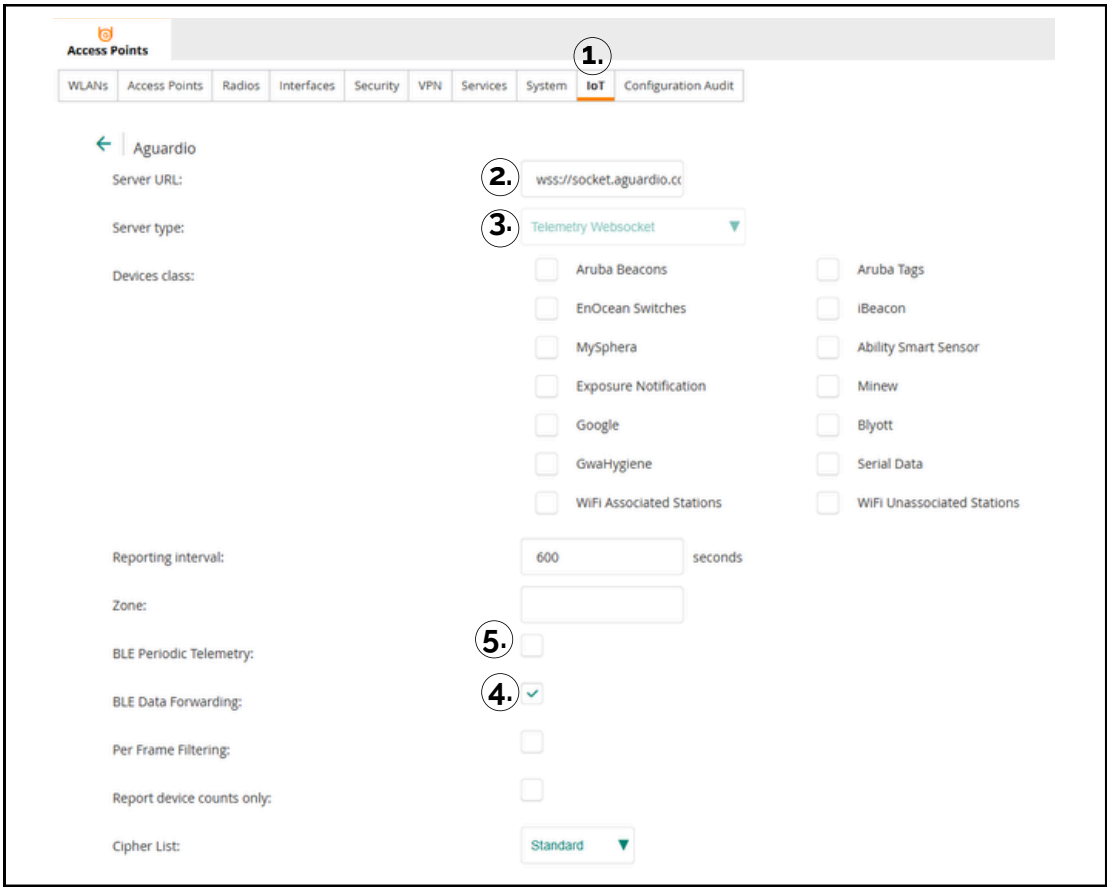



Example Transport Profile (iotutils - Android WebSocket Test Server)

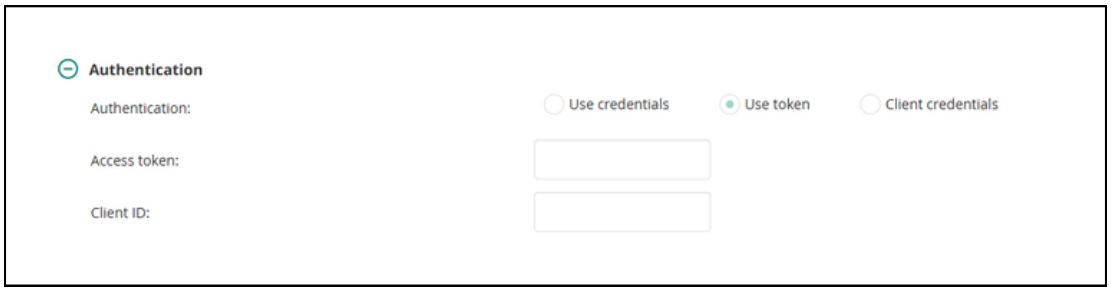
## 2.4 IoT Transport Stream

### 2.4.1 Create and configure IoT Transport Stream

1. Create a new IoT Transport Stream by clicking the '+' symbol on the right.
2. Server URL: **wss://socket.aguardio.com/sensoraos8**
3. Server Type: **Telemetry Websocket**
4. Check BLE Data Forwarding
5. Uncheck BLE Periodic Telemetry

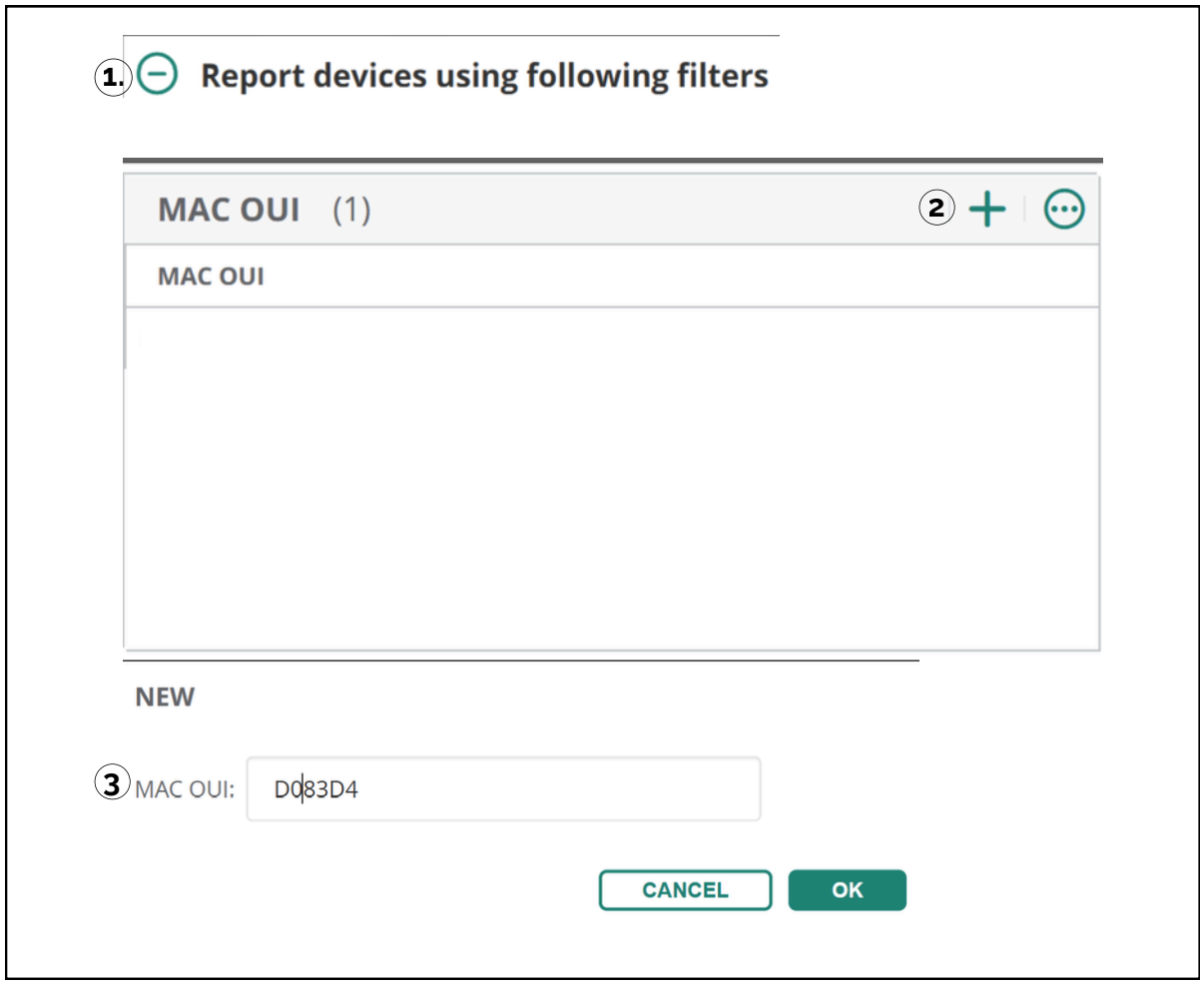


2.4.2 Set authentication to 'Token', and add your generated API key as the authenticator token. You may leave 'Client ID' empty.




2.4.3 **Create a filter for data.**

1. Click on the '***Report Devices using following filters***'
2. Add a new filter by clicking the +' sign
3. Type ''D083D4''



## 2.5 Certificate Assignment - Configuration Workflow:

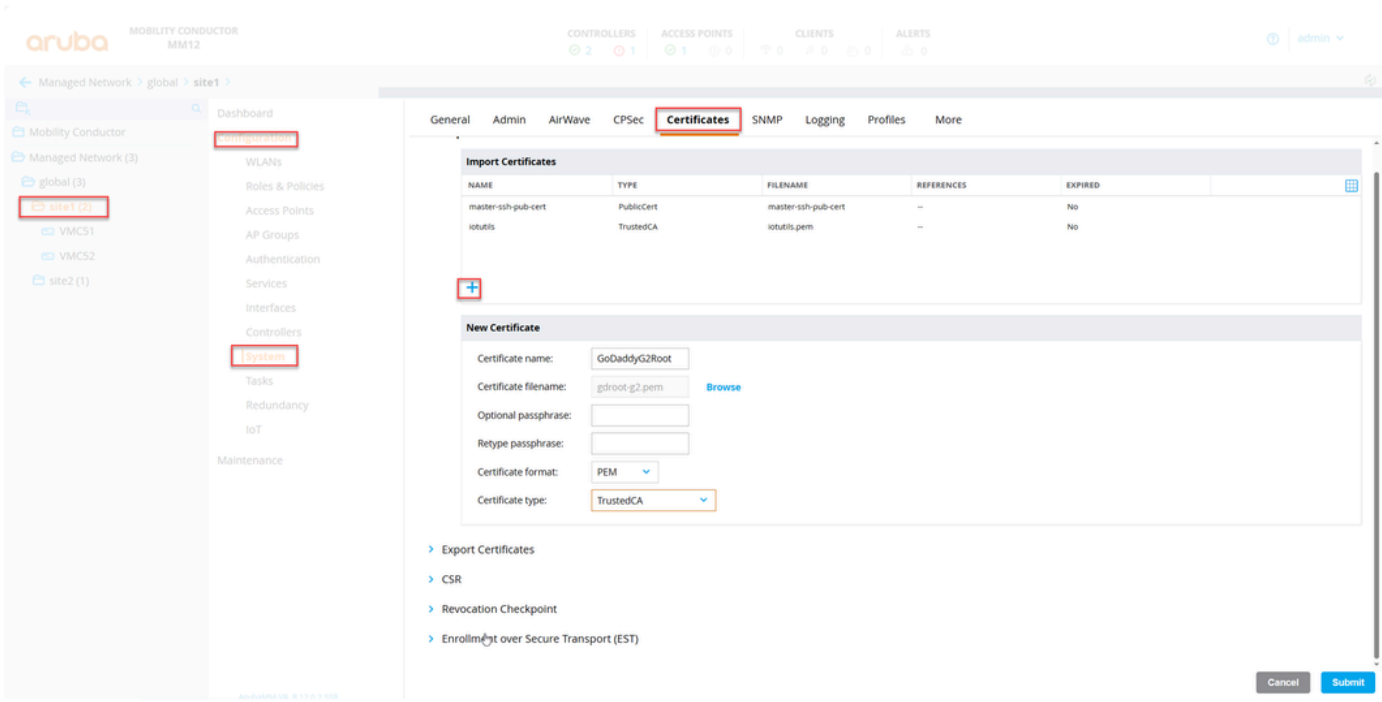
**a.** Navigate to the appropriate network hierarchy in your Mobility Conductor that contains all controllers that need certificate assignments. Navigation Path:Managed Network – Node – Configuration – System – Certificates. ***Note: Every controller at this level or below will inherit this configuration.***

**b.** Import your cert (self-signed or the Trusted Root from your public CA) and apply configuration

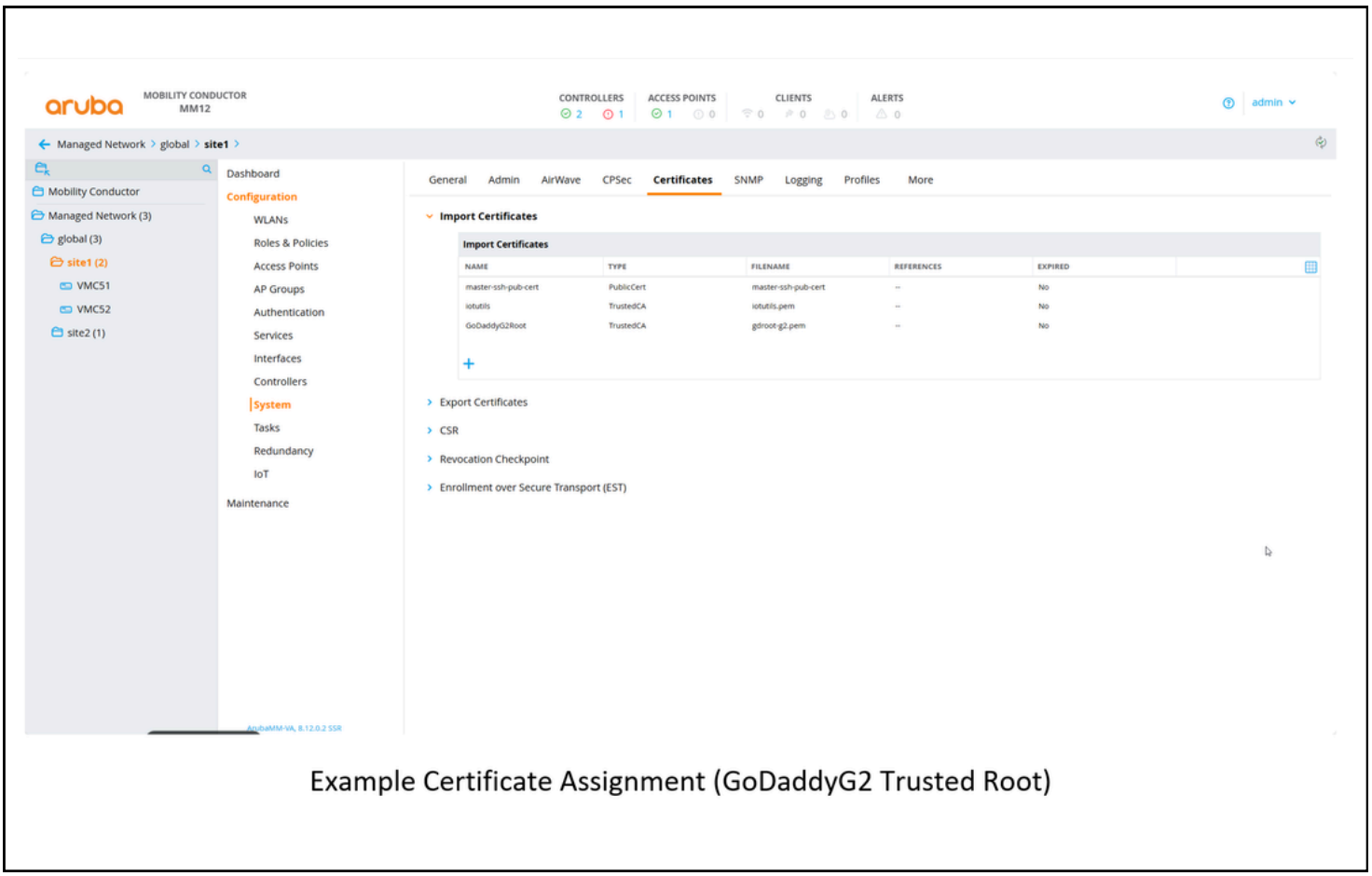
- Certificate name: **Aguardio - DigiCert Global Root G2**
- Certificate filename: **DigiCertGlobalRootG2.crt.pem**
- Download URL: **https://cacerts.digicert.com/DigiCertGlobalRootG2.crt.pem**
- Certificate format: **PEM**
- Certificate type: "**TrustedCA**"

**c.** Hit "**Submit**" then "**Pending Changes/Deploy Changes**" to enable the configuration push.

**d.** Repeat steps a-c for each managed network node (config folder) to ensure all appropriate controllers in your network configuration are configured.




Example Navigation Path (site1 controller cluster)




Example Certificate Assignment (GoDaddyG2 Trusted Root)

## 2.6 BLE Service Profile - Configuration Workflow:

**a.** Navigate to the Mobility Conductor service level. Navigation Path: Mobility Conductor ➡ Configuration ➡ services ➡ IoT

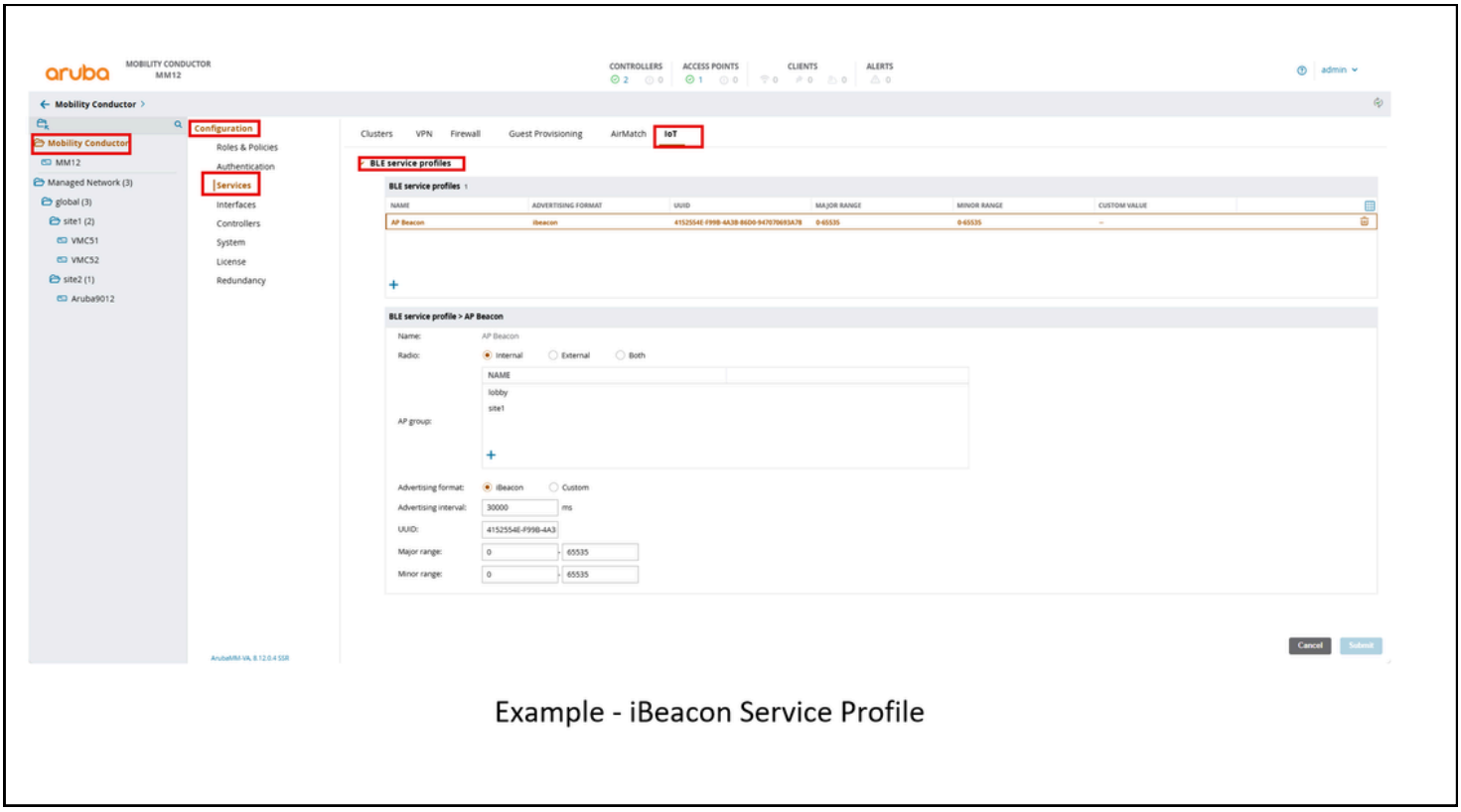
**b.** Create profile specific integration requirements (iBeacon or Custom Payload)

**c**. For iBeacon Config:

1. Profile Name:
2. Radio: {Internal, External, Both}
3. AP Group: (add all AP groups that have radio profile configs set to beacon) mode}
4. Adv Format: iBeacon Note: Series 300 only supports iBeacon
5. Adv interval (ms): 100-30000 max (increments of 100)
6. UUID:
7. Major range: 1-65535
8. Minor range: 1-65535
9. Hit "**Submit**" then "**Pending Changes/Deploy Changes**" to enable the configuration push.

**d.** For Custom Payload Config:

1. Profile Name:
2. Radio: Internal, External, Both
3. AP Group: (add all AP groups that have radio profile configs set to beacon) mode]
4. Adv Format: Custom
5. Adv interval (ms): 100-30000 max (increments of 100)
6. Customer Value: 31 byes (HEX octet)
7. Hit "**Submit**" then "**Pending Changes/Deploy Changes**" to enable the configuration push.




Example - iBeacon Service Profile

## 3. Configuration Validation

Perform the following commands on each controller to verify config settings applied in GUI are present:

1. Radio Profile: **show iot radio-profile <profile name>**
2. Transport Profile: **show iot transportProfile <profile name>**
3. Certificate Assignment: **show certificates /mm/mynode**
4. BLE Service Profiles:

**a.** Mobility Conductor GUI.

i. Navigate to: Managed Network ➡ Dashboard ➡ IoT ➡ BLE Beacons

ii. Observe AP configs. Set view and export to CSV available from GUI.

**b.** Command Line.

i. Show Beacon Configs: **show iot-manager ble-services beacon-info {all, custom-beacon, ibeacon}** Note: CLI command performed on Mobility Conductor

ii. iBeacon CSV export options: **ibeacon-info {ap-group, ble-profile}** Note:CSV file available from Mobility Conductor filesystem (can download from GUI (Diagnostics area in menu)

## 4. Troubleshooting Commands

For 8.x/Controller-based deployment, the Controller makes the connection to a partner application. Perform the following commands on each controller to verify data and connection establishment:

1. Verify controller date/time is set properly: **show clock**

2. Verify DNS Servers configured: **show ip domain-name** (needed to resolve wss and auth endpoints)
   - **a.** Verify WSS endpoint is reachable: **ping <wss endpoint FQDN>**
   - **b.** Verify Auth endpoint is reachable: **ping <auth endpoint FQDN>**

3. Verify certificates are installed on controller(s): **show certificates /mm/mynode** (check on each controller)

4. Verify BLE scan data present: **show ap debug ble-table ap-name <ap_name> all**

5. Check Connection Status: **show ble_relay report <profile name|blank>** (verify transport profile connection status)

6. Review Connection Error Log: **show ble_relay ws-log <transport_profile>** Note: Contact support to help debug error messages. Note: append **| include <error, connection, certificate>** options to command line to filter for specific messages in the log file.

7. Check Connection Buffer Status: **show ble_relay tag report** Note: look for indication of dropped messages or WSS queues not available. This would indicate possible buffer size increase is needed on partner application server (WSS Server) or additional processing capacity to keep the queues from filling up (based on data rates).