Certificate:

Install OpenSSL

mkdir ~/certificates

# navigate to certificates directory

cd ~/certificates

# download helper script files

curl https://raw.githubusercontent.com/Azure/azure-iot-sdk-c/master/tools/CACertificates/certGen.sh --output certGen.sh

curl https://raw.githubusercontent.com/Azure/azure-iot-sdk-c/master/tools/CACertificates/openssl\_device\_intermediate\_ca.cnf --output openssl\_device\_intermediate\_ca.cnf

curl https://raw.githubusercontent.com/Azure/azure-iot-sdk-c/master/tools/CACertificates/openssl\_root\_ca.cnf --output openssl\_root\_ca.cnf

# update script permissions so user can read, write, and execute it

chmod 700 certGen.sh

#### **Generate**

./certGen.sh create\_root\_and\_intermediate

download ~/certificates/certs/azure-iot-test-only.root.ca.cert.pem

1. On the **Device Provisioning Service** blade, on the left-side menu under **Settings**, click **Certificates**.

Replace the exist rootCA by the azure-iot-test-only.root.ca.cert.pem

1. On the **Certificate Details** pane, click **Generate Verification Code**.
2. ./certGen.sh create\_verification\_certificate <verification-code>
3. download ~/certificates/certs/verification-code.cert.pem

#### **Task 5: Create an Enrollment Group**

1. Ensure that the **Attestation Type** is set to **Certificate**.

Device Twin State field

"telemetryDelay": "1"

### **Simulate Devices**

generate and download 9 device certificates.

mkdir monitoring

cd monitoring

# create a new file

touch gen-dev-certs.sh

chmod +x gen-dev-certs.sh

#!/bin/bash

# Generate 9 device certificates

# Rename for each device

# download from the Cloud CLI

pushd ~/certificates

for i in {1..9}

do

chmod +w ./certs/new-device.cert.pem

./certGen.sh create\_device\_certificate asset-track$i

sleep 5

cp ./certs/new-device.cert.pfx ./certs/sensor-thl-200$i.cert.pfx

download ./certs/sensor-thl-200$i.cert.pfx

done

popd

./gen-dev-certs.sh