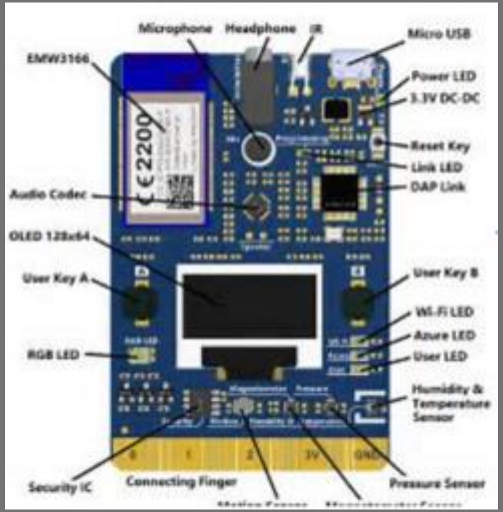
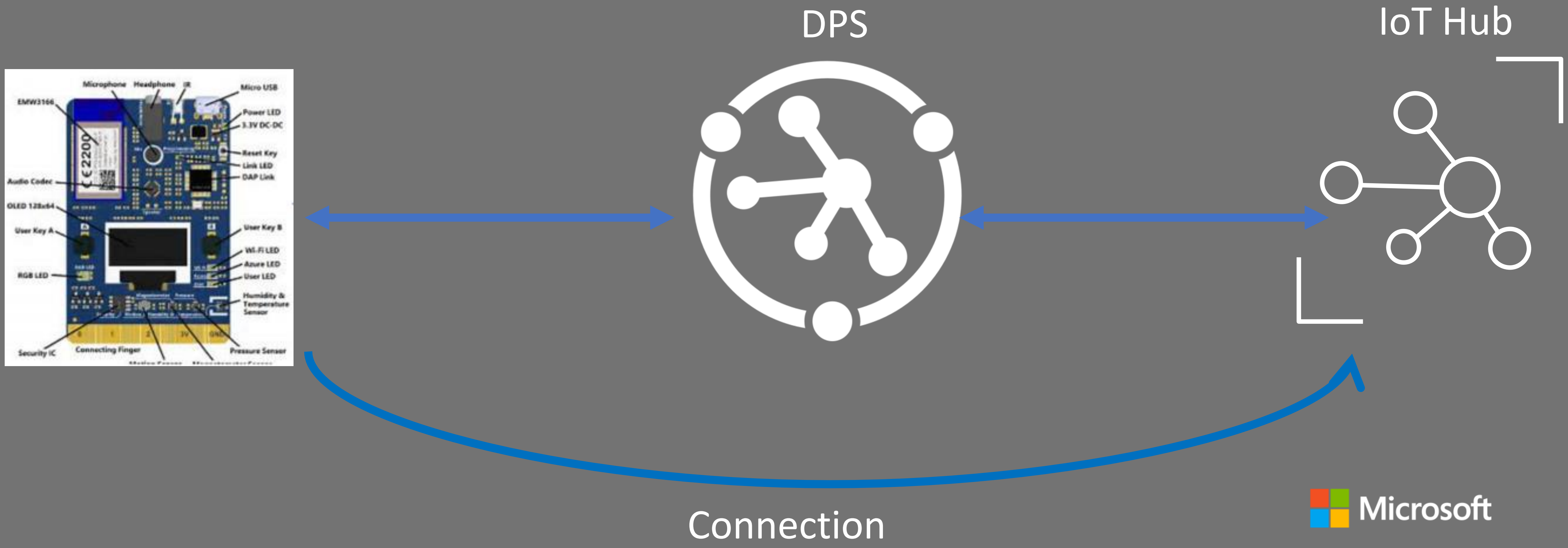
DPS Hands-on Lab

- What you will learn are:
  - Configure global endpoint of DPS on device
  - Use Unique Device Secret (UDS) to generate X.509 certificate
  - Use MXChip to enroll individual device
  - Verify MXChip is provisioned with zero-touch

Prepare the MXCHIP



What are we doing?

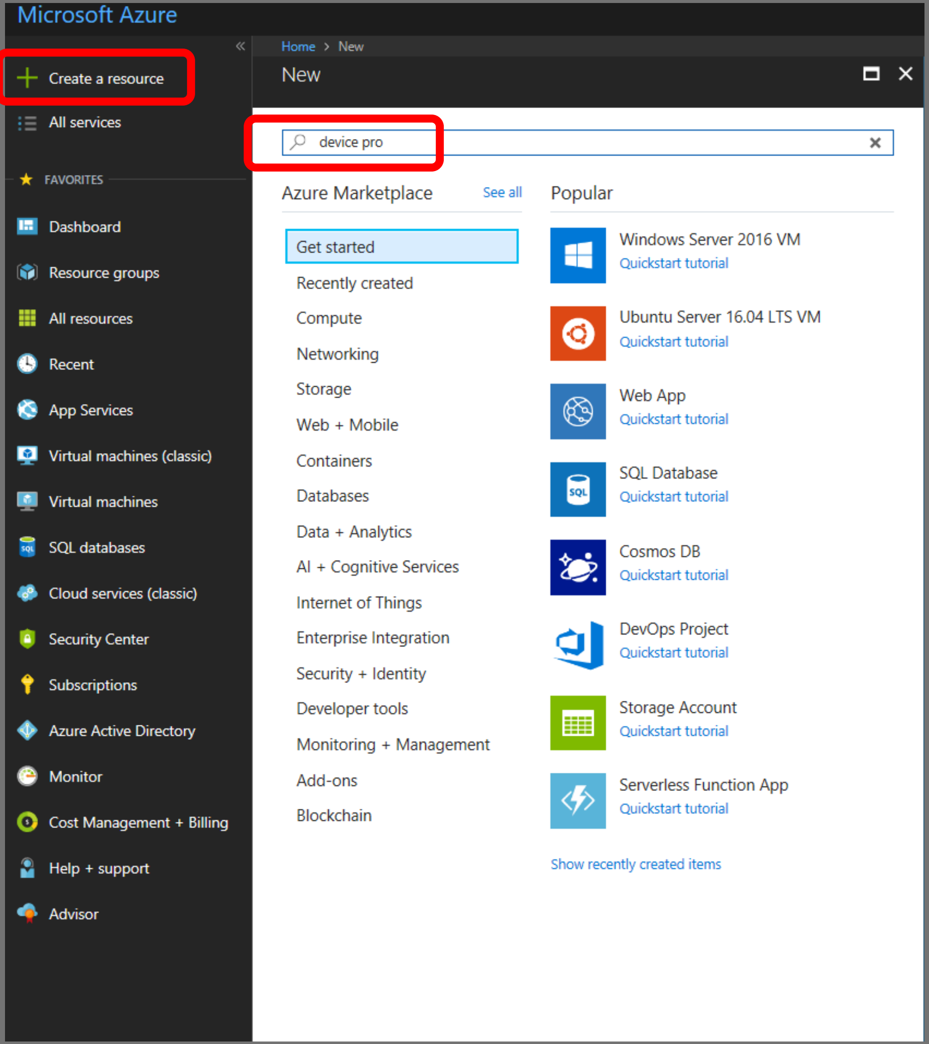


Requirements

- VSCode
- Prepare the Hardware
  - Plug the device to your machine
  - Connect to WIFI
  - Start the DevKit and update firmware (ver 1.3.2 – We need it !)
- Prepare development environment (prerequisites sent prior the training)
  - Download the latest package: DOWNLOAD
  - Run the installation script and verify everything is successfully installed
- Make sure Git client tools are installed and have the latest version
  - Software Freedom Conservancy's Git client tools
- Install Putty https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

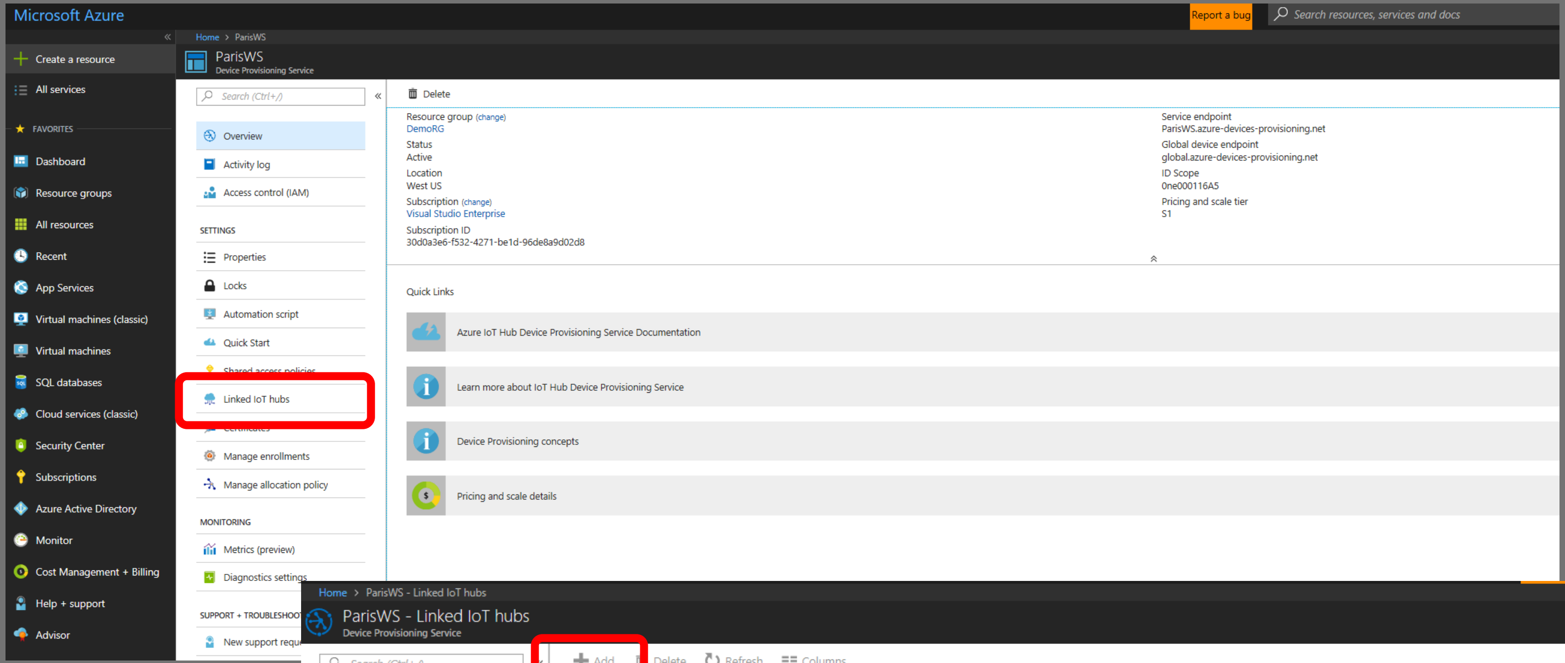
Microsoft

Create IoT Hub DPS with the Azure portal



Microsoft

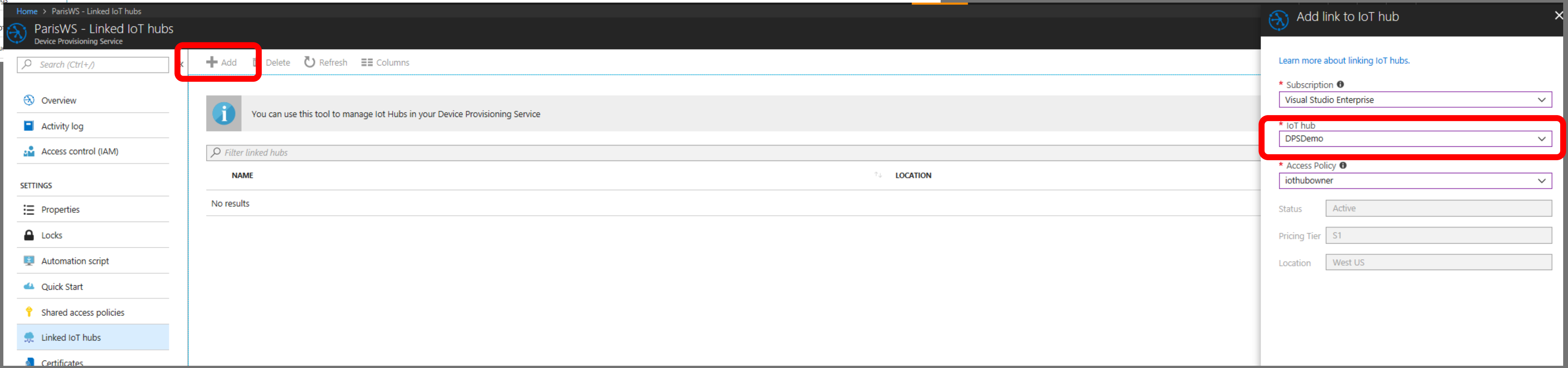
Link to your exiting IoT Hub and your DPS



ParisWS
Device Provisioning Service

Resource group (change)
DemoRG
Status
Active
Location
West US
Subscription (change)
Visual Studio Enterprise
Subscription ID
30d0a3e6-f532-4271-be1d-96de8a9d02d8

Service endpoint
ParisWS.azure-devices-provisioning.net
Global device endpoint
global.azure-devices-provisioning.net
ID Scope
0ne000116A5
Pricing and scale tier
S1



ParisWS - Linked IoT hubs
Device Provisioning Service

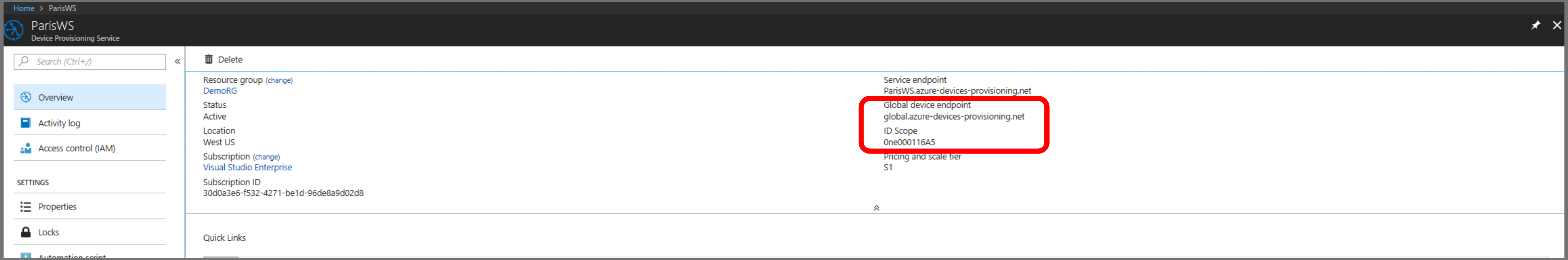
You can use this tool to manage Iot Hubs in your Device Provisioning Service

No results

Add link to IoT hub

* Subscription: Visual Studio Enterprise
* IoT hub: DPSDemo
* Access Policy: iothubowner
Status: Active
Pricing Tier: S1
Location: West US

Note down 'Global device endpoint' and ID Scope for DPS instance info



Resource group (change)
DemoRG
Status
Active
Location
West US
Subscription (change)
Visual Studio Enterprise
Subscription ID
30d0a3e6-f532-4271-be1d-96de8a9d02d8

Service endpoint
ParisWS.azure-devices-provisioning.net
Global device endpoint
global.azure-devices-provisioning.net
ID Scope
0ne000116A5
Pricing and scale tier
S1

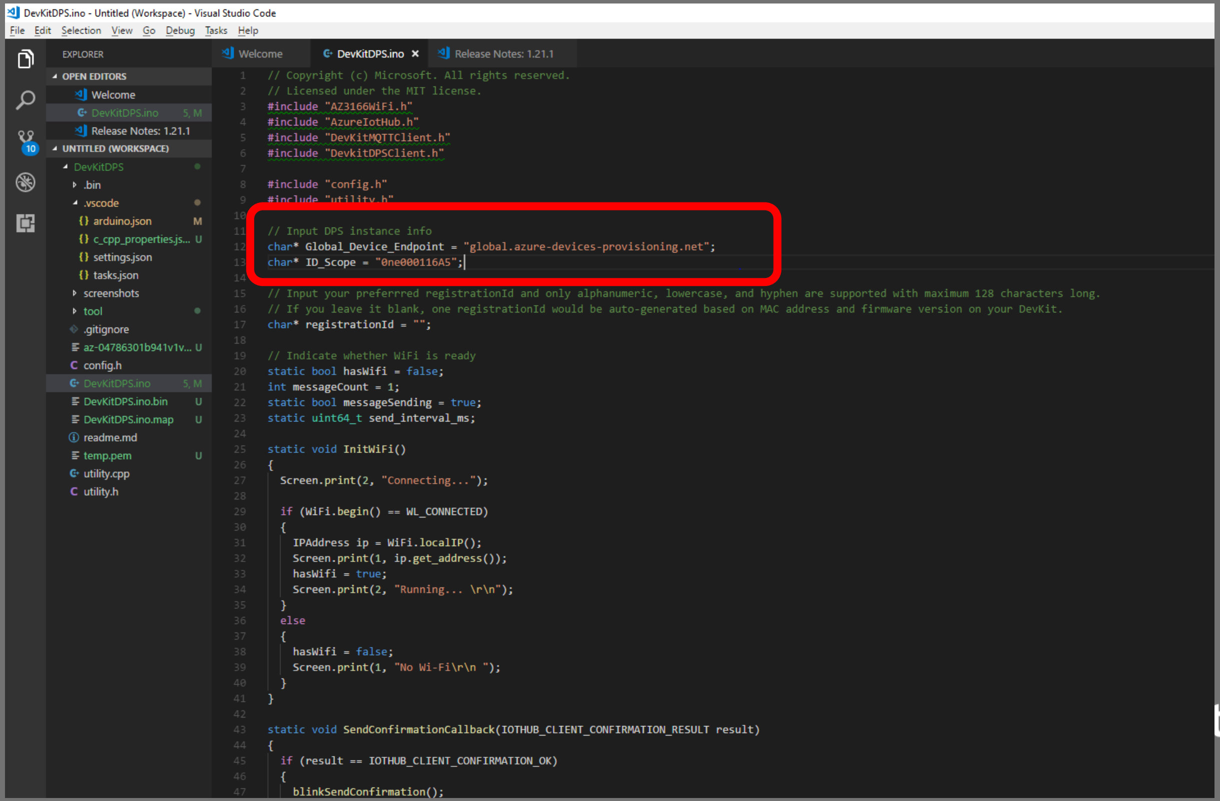
Get the latest DPS sample code from GitHub

- git clone https://github.com/DevKitExamples/DevKitDPS.git

Microsoft

Launch VS Code

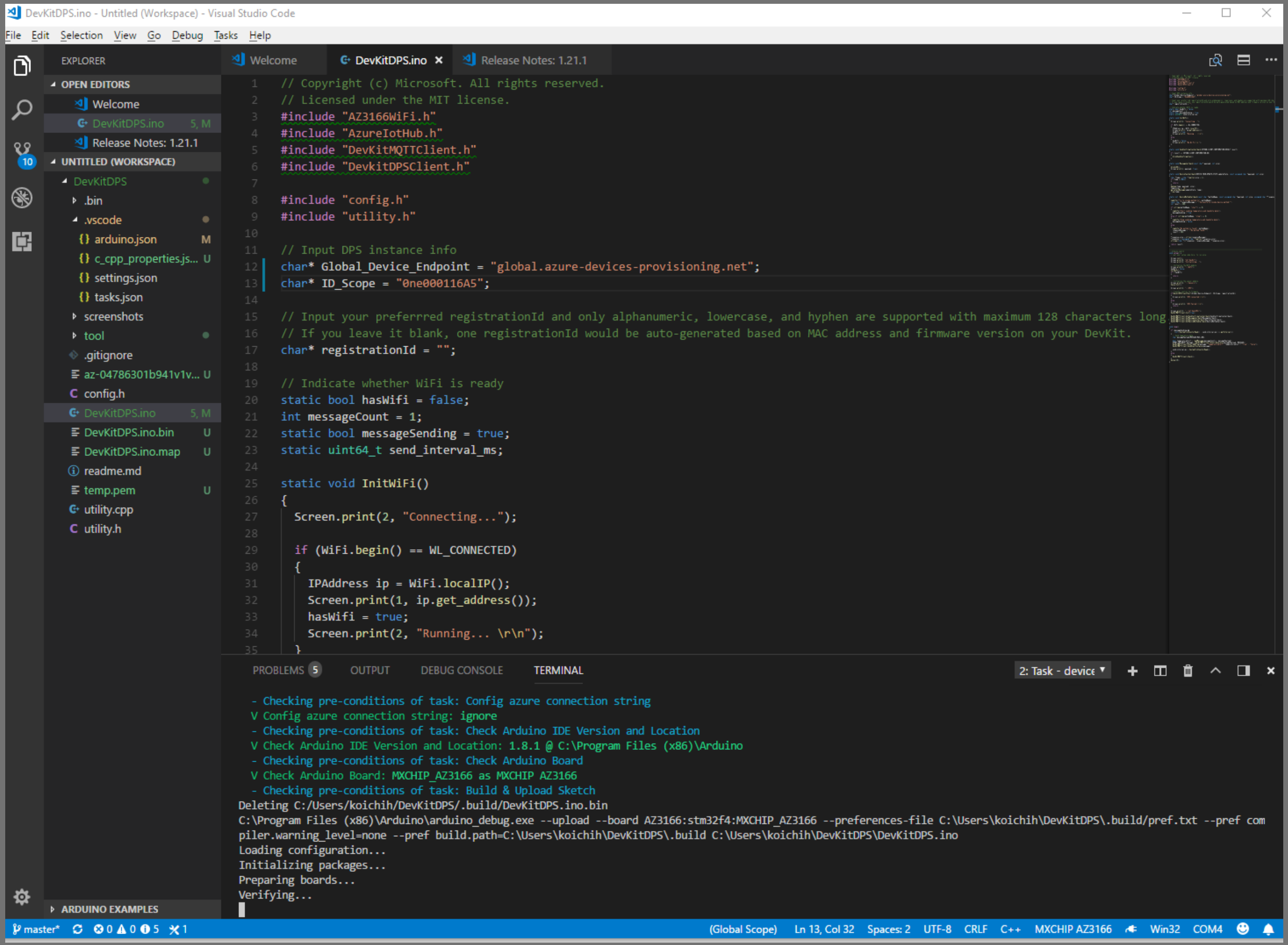
- Make sure MXChip is connected
- Open the folder that contains the code you cloned
- Open DevKitDPS.ino



```
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license.
#include "AZ3166WiFi.h"
#include "AzureIotHub.h"
#include "DevKitMQTTClient.h"
#include "DevkitDPSClient.h"

#include "config.h"
#include "utility.h"

// Input DPS instance info
char* Global_Device_Endpoint = "global.azure-devices-provisioning.net";
char* ID_Scope = "0ne000116A5";

// Input your preferrred registrationId and only alphanumeric, lowercase, and hyphen are supported with maximum 128 characters long.
// If you leave it blank, one registrationId would be auto-generated based on MAC address and firmware version on your DevKit.
char* registrationId = "";

// Indicate whether WiFi is ready
static bool hasWifi = false;
int messageCount = 1;
static bool messageSending = true;
static uint64_t send_interval_ms;

static void InitWiFi()
{
  Screen.print(2, "Connecting...");

  if (WiFi.begin() == WL_CONNECTED)
  {
    IPAddress ip = WiFi.localIP();
    Screen.print(1, ip.get_address());
    hasWifi = true;
    Screen.print(2, "Running... \r\n");
  }
  else
  {
    hasWifi = false;
    Screen.print(1, "No Wi-Fi\r\n ");
  }
}

static void SendConfirmationCallback(IOTHUB_CLIENT_CONFIRMATION_RESULT result)
{
  if (result == IOTHUB_CLIENT_CONFIRMATION_OK)
  {
    blinkSendConfirmation();
```

Build and upload the code to the DevKit

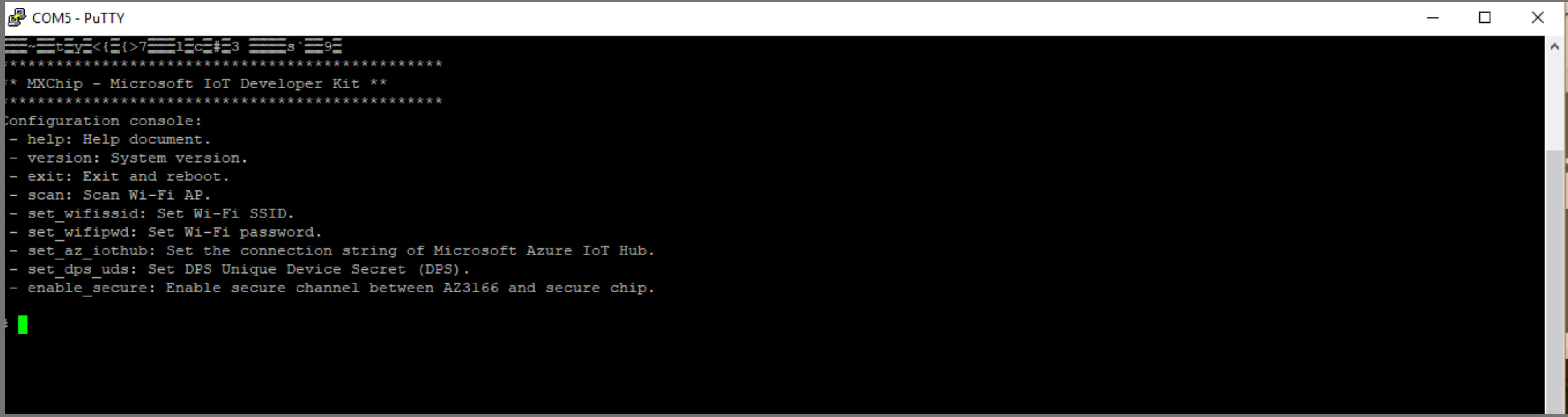
- Ctrl+P
- task device-upload
- Build & Upload success -> Auto reboot
- Screen will display "DPS Failed" and it's OK!



```cpp
// Copyright (c) Microsoft. All rights reserved.
// Licensed under the MIT license.
#include "AZ3166WiFi.h"
#include "AzureIotHub.h"
#include "DevKitMQTTClient.h"
#include "DevkitDPSClient.h"

#include "config.h"
#include "utility.h"

// Input DPS instance info
char* Global_Device_Endpoint = "global.azure-devices-provisioning.net";
char* ID_Scope = "0ne000116A5";

// Input your preferrred registrationId and only alphanumeric, lowercase, and hyphen are supported with maximum 128 characters long
// If you leave it blank, one registrationId would be auto-generated based on MAC address and firmware version on your DevKit.
char* registrationId = "";

// Indicate whether WiFi is ready
static bool hasWifi = false;
int messageCount = 1;
static bool messageSending = true;
static uint64_t send_interval_ms;

static void InitWifi()
{
  Screen.print(2, "Connecting...");

  if (WiFi.begin() == WL_CONNECTED)
  {
    IPAddress ip = WiFi.localIP();
    Screen.print(1, ip.get_address());
    hasWifi = true;
    Screen.print(2, "Running... \r\n");
  }
```

```
 - Checking pre-conditions of task: Config azure connection string
 V Config azure connection string: ignore
 - Checking pre-conditions of task: Check Arduino IDE Version and Location
 V Check Arduino IDE Version and Location: 1.8.1 @ C:\Program Files (x86)\Arduino
 - Checking pre-conditions of task: Check Arduino Board
 V Check Arduino Board: MXCHIP_AZ3166 as MXCHIP AZ3166
 - Checking pre-conditions of task: Build & Upload Sketch
Deleting C:/Users/koichih/DevKitDPS/.build/DevKitDPS.ino.bin
C:\Program Files (x86)\Arduino\arduino_debug.exe --upload --board AZ3166:stm32f4:MXCHIP_AZ3166 --preferences-file C:\Users\koichih\DevKitDPS\.build/pref.txt --pref com
piler.warning_level=none --pref build.path=C:\Users\koichih\DevKitDPS\.build C:\Users\koichih\DevKitDPS\DevKitDPS.ino
Loading configuration...
Initializing packages...
Preparing boards...
Verifying...
```

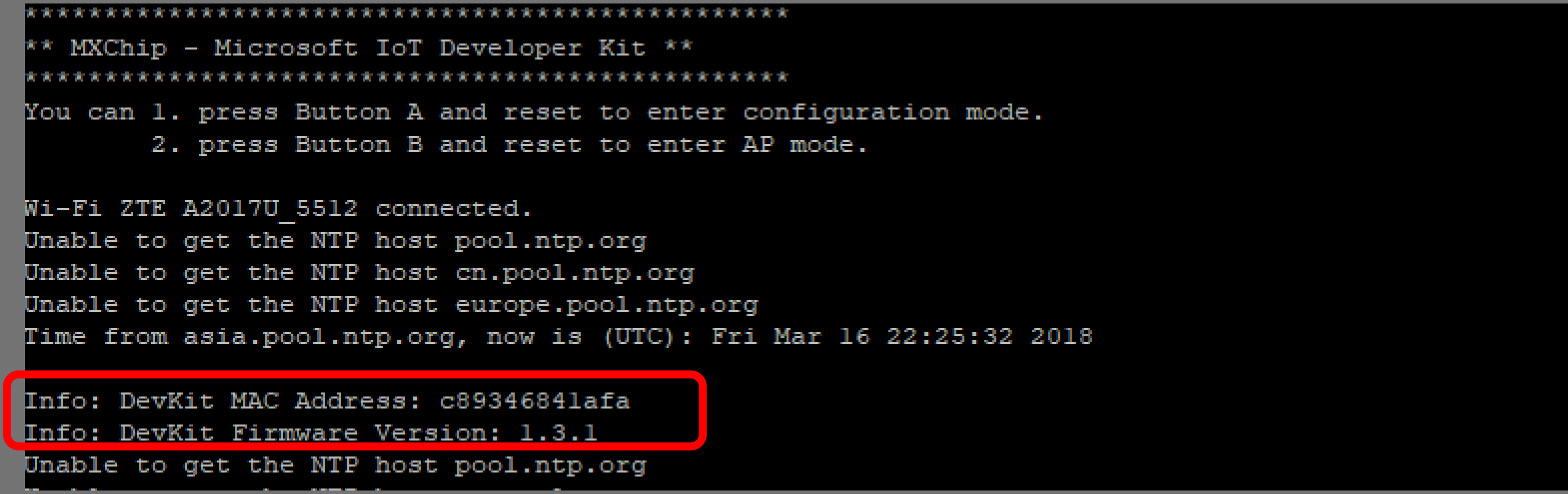
Save UDS (Unique Device Secret) on security chip on DevKit



COM5 - PuTTY

```
[illegible]
************************************************
* MXChip - Microsoft IoT Developer Kit **
************************************************
onfiguration console:
 - help: Help document.
 - version: System version.
 - exit: Exit and reboot.
 - scan: Scan Wi-Fi AP.
 - set_wifissid: Set Wi-Fi SSID.
 - set_wifipwd: Set Wi-Fi password.
 - set_az_iothub: Set the connection string of Microsoft Azure IoT Hub.
 - set_dps_uds: Set DPS Unique Device Secret (DPS).
 - enable_secure: Enable secure channel between AZ3166 and secure chip.

```

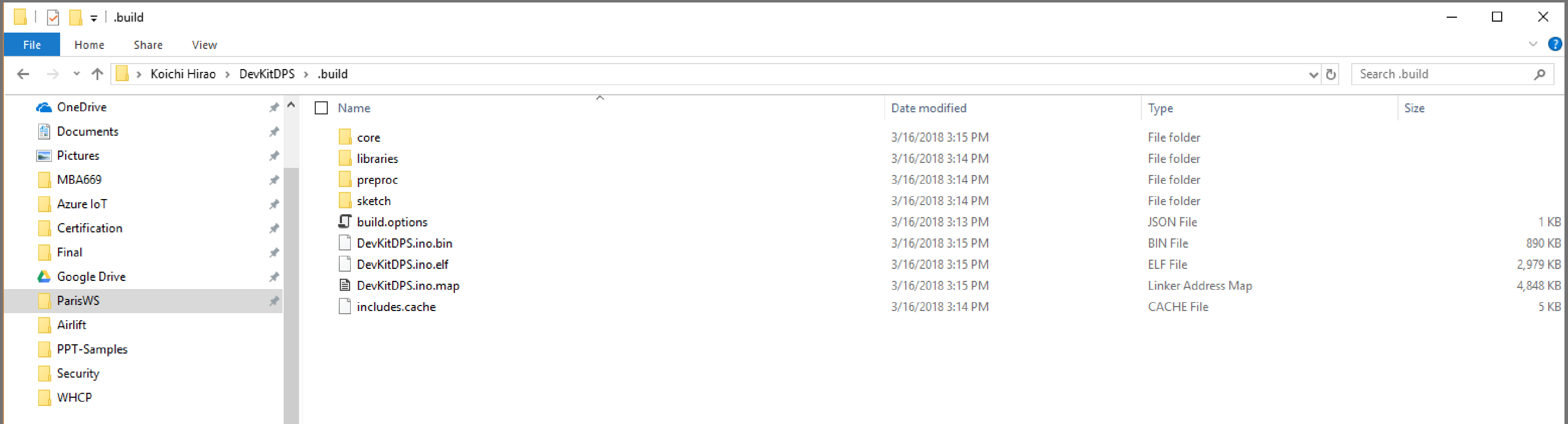
- Hold A + reset to put configuration mode
- Launch Putty
- Make sure COM port #
- Baud rate = 115200
- set_dps_uds 999301f1283cd8e7453f328b8f4c2b15cbbe07c669957517a70a575973a331fa
    - Saving UDS to the security chip
- <u>DO NOT </u>close PUTTY.
- Reset
- Note down
    - MAC Address
    - Firmware version
- Screen will display "DPS Failed" and it's OK!



```
************************************************
** MXChip - Microsoft IoT Developer Kit **
************************************************
You can 1. press Button A and reset to enter configuration mode.
        2. press Button B and reset to enter AP mode.

Wi-Fi ZTE A2017U_5512 connected.
Unable to get the NTP host pool.ntp.org
Unable to get the NTP host cn.pool.ntp.org
Unable to get the NTP host europe.pool.ntp.org
Time from asia.pool.ntp.org, now is (UTC): Fri Mar 16 22:25:32 2018

Info: DevKit MAC Address: c89346841afa
Info: DevKit Firmware Version: 1.3.1
Unable to get the NTP host pool.ntp.org
```

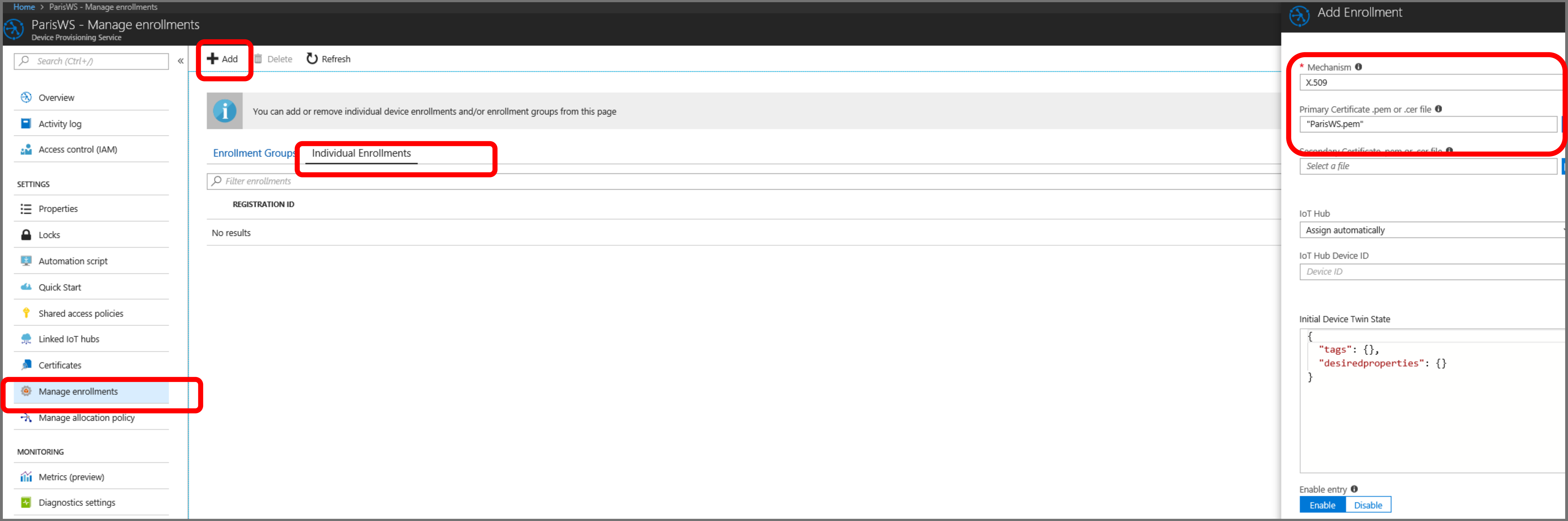
Generate X.509 Certificate

- From DevKitDPS\.build folder, copy **DPS.ino.bin** and **DPS.ino.map** in it.
- Paste these two files into DevKit\Tools
- Run dps_cert_gen.exe from Tools folder
- Enter UDS(project name: leave as default)
- Enter MAC address and firmware



Name	Date modified	Type	Size
core	3/16/2018 3:15 PM	File folder	
libraries	3/16/2018 3:14 PM	File folder	
preproc	3/16/2018 3:14 PM	File folder	
sketch	3/16/2018 3:14 PM	File folder	
build.options	3/16/2018 3:13 PM	JSON File	1 KB
DevKitDPS.ino.bin	3/16/2018 3:15 PM	BIN File	890 KB
DevKitDPS.ino.elf	3/16/2018 3:15 PM	ELF File	2,979 KB
DevKitDPS.ino.map	3/16/2018 3:15 PM	Linker Address Map	4,848 KB
includes.cache	3/16/2018 3:14 PM	CACHE File	5 KB

Create a device enrollment entry in DPS



Home > ParisWS - Manage enrollments

ParisWS - Manage enrollments
Device Provisioning Service

You can add or remove individual device enrollments and/or enrollment groups from this page

Enrollment Groups | Individual Enrollments

Filter enrollments

REGISTRATION ID
No results

Add Enrollment

* Mechanism: X.509

Primary Certificate .pem or .cer file: "ParisWS.pem"

Secondary Certificate .pem or .cer file: Select a file

IoT Hub: Assign automatically

IoT Hub Device ID: Device ID

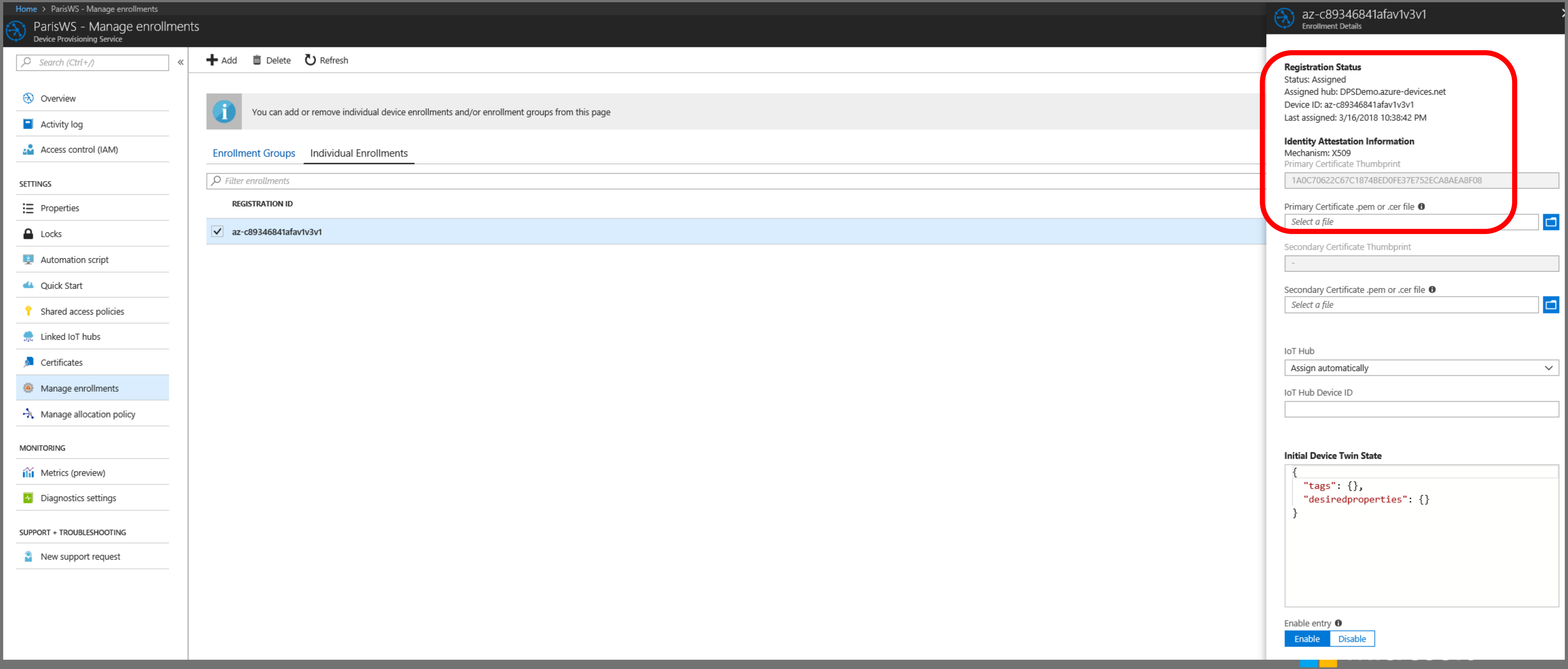
Initial Device Twin State

```
{
  "tags": {},
  "desiredproperties": {}
}
```

Enable entry: [✓ Enable] Disable

Microsoft

Confirm the Identity attestation info



Home > ParisWS - Manage enrollments

ParisWS - Manage enrollments
Device Provisioning Service

You can add or remove individual device enrollments and/or enrollment groups from this page

REGISTRATION ID
✓ az-c89346841afav1v3v1

az-c89346841afav1v3v1
Enrollment Details

Registration Status
Status: Assigned
Assigned hub: DPSDemo.azure-devices.net
Device ID: az-c89346841afav1v3v1
Last assigned: 3/16/2018 10:38:42 PM

Identity Attestation Information
Mechanism: X509
Primary Certificate Thumbprint: 1A0C70622C67C1874BED0FE37E752ECA8AEA8F08
Primary Certificate .pem or .cer file: Select a file
Secondary Certificate Thumbprint: -
Secondary Certificate .pem or .cer file: Select a file
IoT Hub: Assign automatically
IoT Hub Device ID:

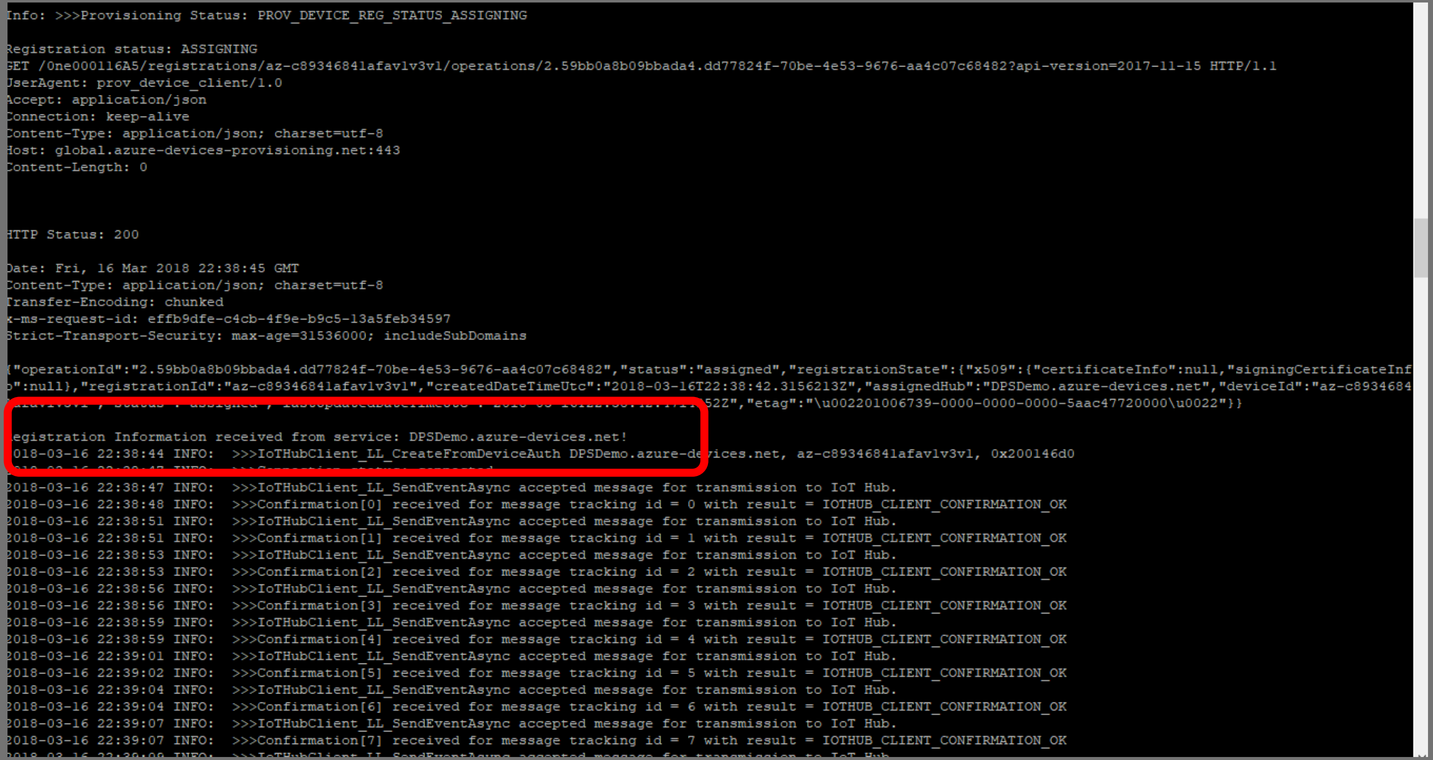
Initial Device Twin State

```
{
  "tags": {},
  "desiredproperties": {}
}
```

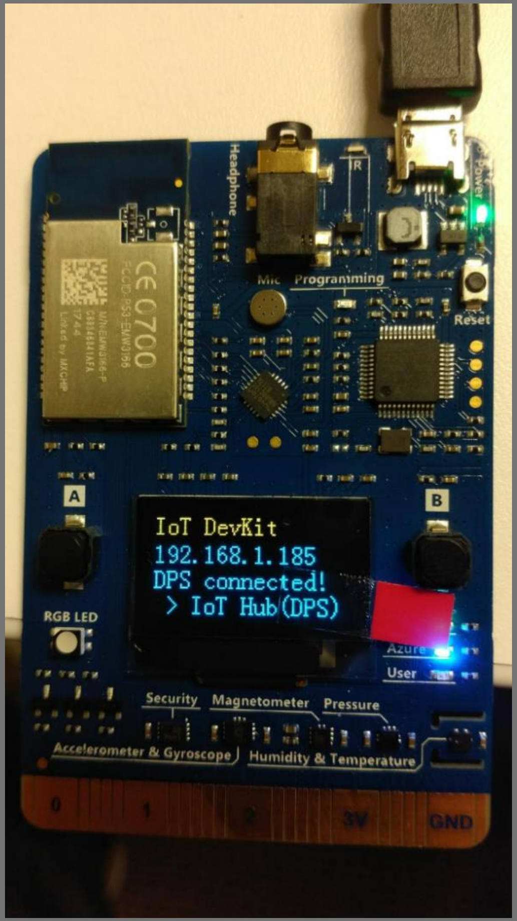
Enable entry: [✓ Enable] Disable

Start the DevKit

- You should see either on Putty or VS Code spew that the registration was success to DPS.



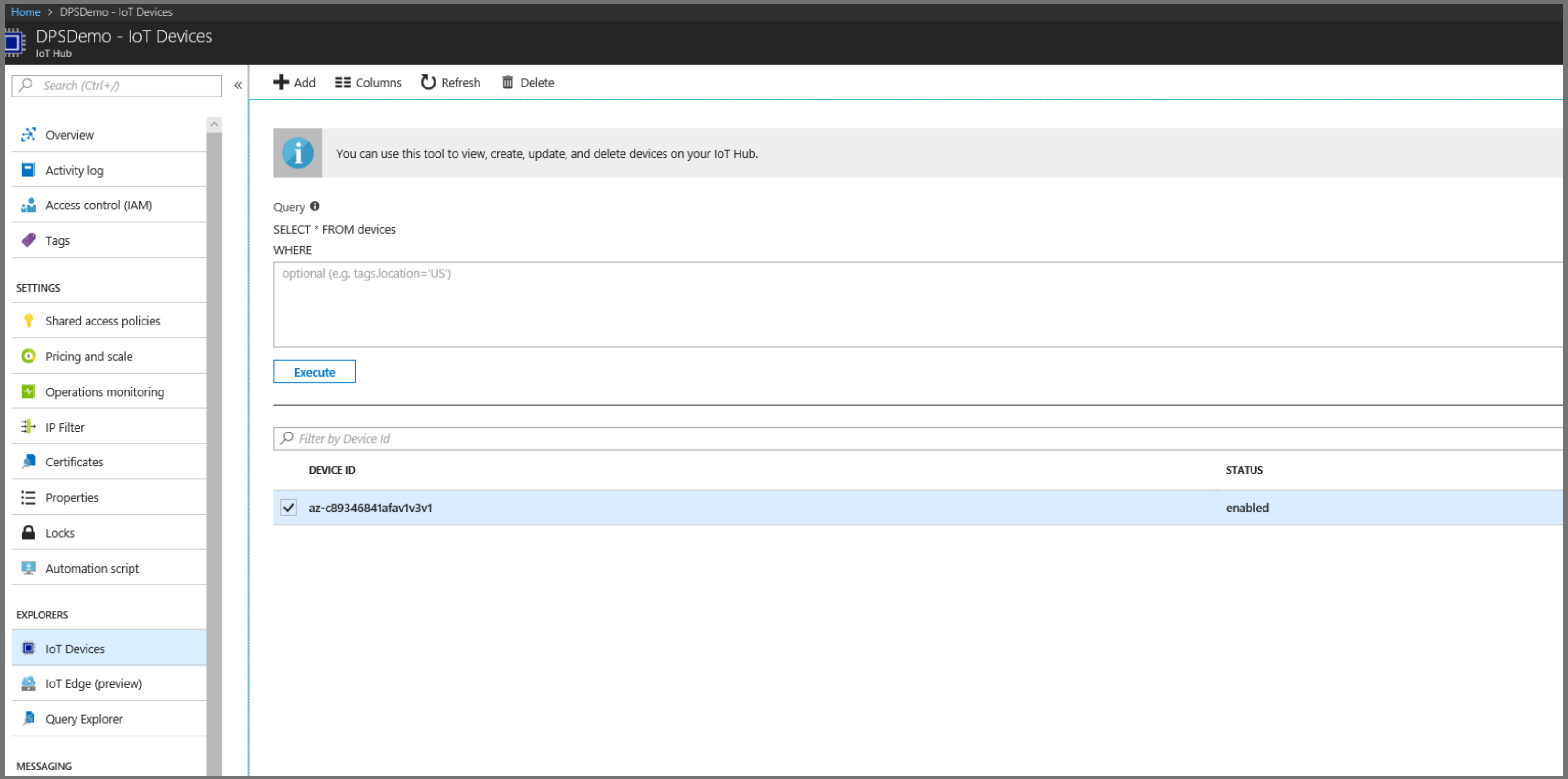
```
Info: >>>Provisioning Status: PROV_DEVICE_REG_STATUS_ASSIGNING

Registration status: ASSIGNING
GET /0ne000116A5/registrations/az-c89346841afav1v3v1/operations/2.59bb0a8b09bbada4.dd77824f-70be-4e53-9676-aa4c07c68482?api-version=2017-11-15 HTTP/1.1
UserAgent: prov_device_client/1.0
Accept: application/json
Connection: keep-alive
Content-Type: application/json; charset=utf-8
Host: global.azure-devices-provisioning.net:443
Content-Length: 0



HTTP Status: 200

Date: Fri, 16 Mar 2018 22:38:45 GMT
Content-Type: application/json; charset=utf-8
Transfer-Encoding: chunked
x-ms-request-id: effb9dfe-c4cb-4f9e-b9c5-13a5feb34597
Strict-Transport-Security: max-age=31536000; includeSubDomains

{"operationId":"2.59bb0a8b09bbada4.dd77824f-70be-4e53-9676-aa4c07c68482","status":"assigned","registrationState":{"x509":{"certificateInfo":null,"signingCertificateInf
o":null},"registrationId":"az-c89346841afav1v3v1","createdDateTimeUtc":"2018-03-16T22:38:42.3156213Z","assignedHub":"DPSDemo.azure-devices.net","deviceId":"az-c8934684
[illegible]522Z","etag":"\u002201006739-0000-0000-0000-5aac47720000\u0022"}}

Registration Information received from service: DPSDemo.azure-devices.net!
2018-03-16 22:38:44 INFO:  >>>IoTHubClient_LL_CreateFromDeviceAuth DPSDemo.azure-devices.net, az-c89346841afav1v3v1, 0x200146d0
[illegible]
2018-03-16 22:38:47 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:38:48 INFO:  >>>Confirmation[0] received for message tracking id = 0 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:38:51 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:38:51 INFO:  >>>Confirmation[1] received for message tracking id = 1 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:38:53 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:38:53 INFO:  >>>Confirmation[2] received for message tracking id = 2 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:38:56 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:38:56 INFO:  >>>Confirmation[3] received for message tracking id = 3 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:38:59 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:38:59 INFO:  >>>Confirmation[4] received for message tracking id = 4 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:39:01 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:39:02 INFO:  >>>Confirmation[5] received for message tracking id = 5 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:39:04 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:39:04 INFO:  >>>Confirmation[6] received for message tracking id = 6 with result = IOTHUB_CLIENT_CONFIRMATION_OK
2018-03-16 22:39:07 INFO:  >>>IoTHubClient_LL_SendEventAsync accepted message for transmission to IoT Hub.
2018-03-16 22:39:07 INFO:  >>>Confirmation[7] received for message tracking id = 7 with result = IOTHUB_CLIENT_CONFIRMATION_OK
```



Microsoft

Check your IoT Hub for registration

- Send some messages to the DevKit



Home > DPSDemo - IoT Devices

DPSDemo - IoT Devices
IoT Hub

You can use this tool to view, create, update, and delete devices on your IoT Hub.

Query
SELECT * FROM devices
WHERE
optional (e.g. tags.location='US')

DEVICE ID	STATUS
✓ az-c89346841afav1v3v1	enabled

Microsoft

Microsoft