Azure IOT for WLSOM Full Bring-Up Instructions

Section1. Building Image with Script

1. Quick Instructions:

-In a new Ubuntu 18.04 environment, first get the install script by cloning the script repo:

sudo apt install git-core

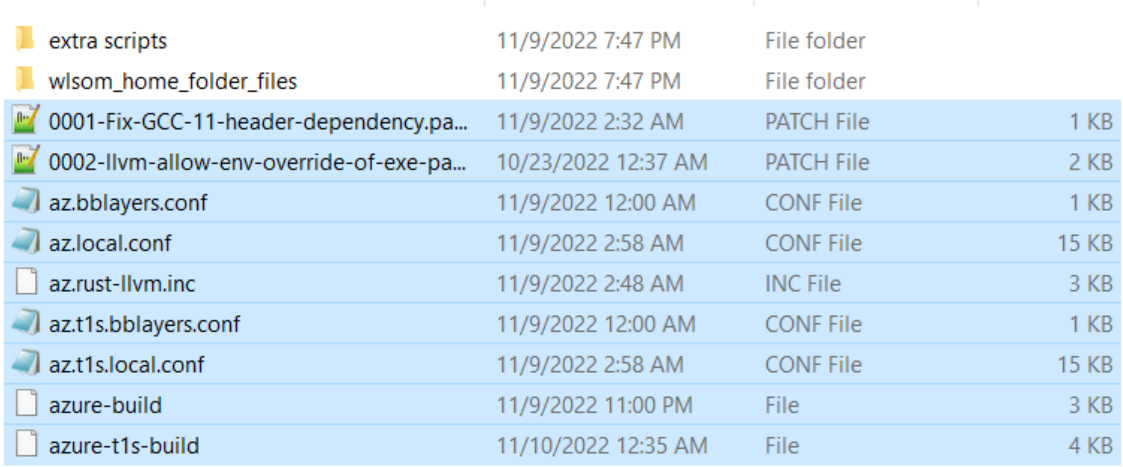
git clone https://github.com/k-mchp/azure-wlsom-ub18.git

-In a new Mint21 environment, first get the install script by cloning the script repo:

sudo apt install git-core

git clone https://github.com/k-mchp/azure-wlsom-mint21.git

-Copy all files from the top folder into a new folder that you create for your build.



-Run the script with the following command

. azure-build

(You will be prompted for your password one time, this is your login password)

-To build azure with t1s use azure-t1s-build and run:

. azure-t1s-build

(Note - If you would like to do all of this manually, just follow the commands in azure-build script)

2. Prerequisites:

-The scripts in azure-wlsom-ub18 folder were tested in an ubuntu 18.04 VM with 70GB of HDD space, and 8GB of allocated RAM.

-The scripts in the azure-wlsom-mint21 folder were tested on a new Mint21 machine.

(Testing other Linux Distributions now)

3. Description:

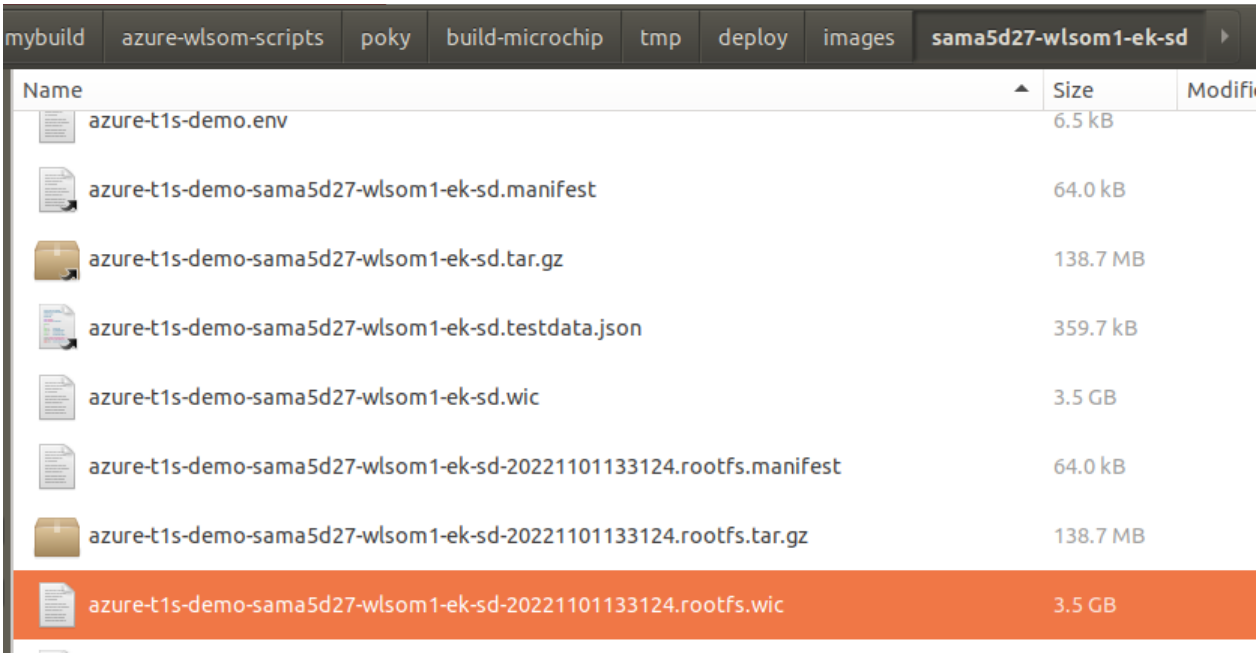
-The script downloads everything needed, places it all in the correct directories and builds the final

image (The local.conf forces this to build in a single thread due to limitations of building rust build, which fails when building with multiple threads. Due to this limitation the initial build can take up to 20hrs in a VM)

Notes:
This document is based on documents written by Swapna Gurumani, John Haroian and Matt Wood

Section2. Installing Image on WLSOM

-After the build is finished, locate wic file(~3.5GB) in poky/build-microchip/tmp/deploy/images/sama5d27-wlsom-ek-sd/

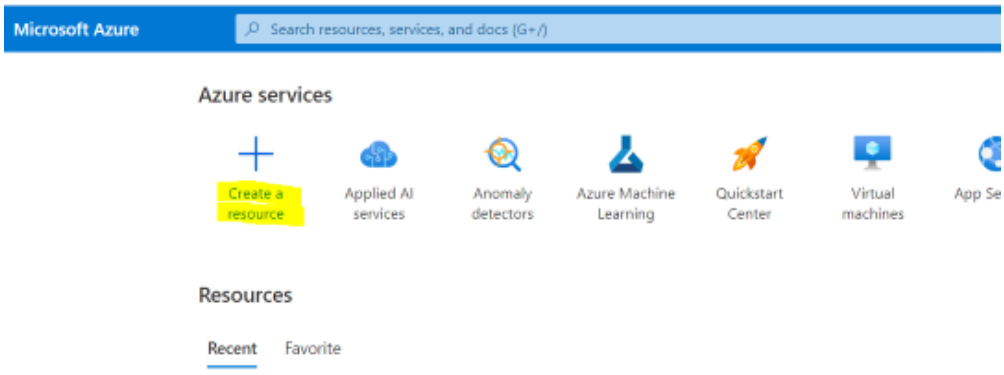


-Copy this file to Windows host machine using shared folder or removable storage
-Use Balena etcher program to copy the wic image onto an SD card that will fit the WLSOM(8GB or larger)
-See instructions on creating an SD card here: https://www.linux4sam.org/bin/view/Linux4SAM/DemoSD
-Insert SD card into WLSOM

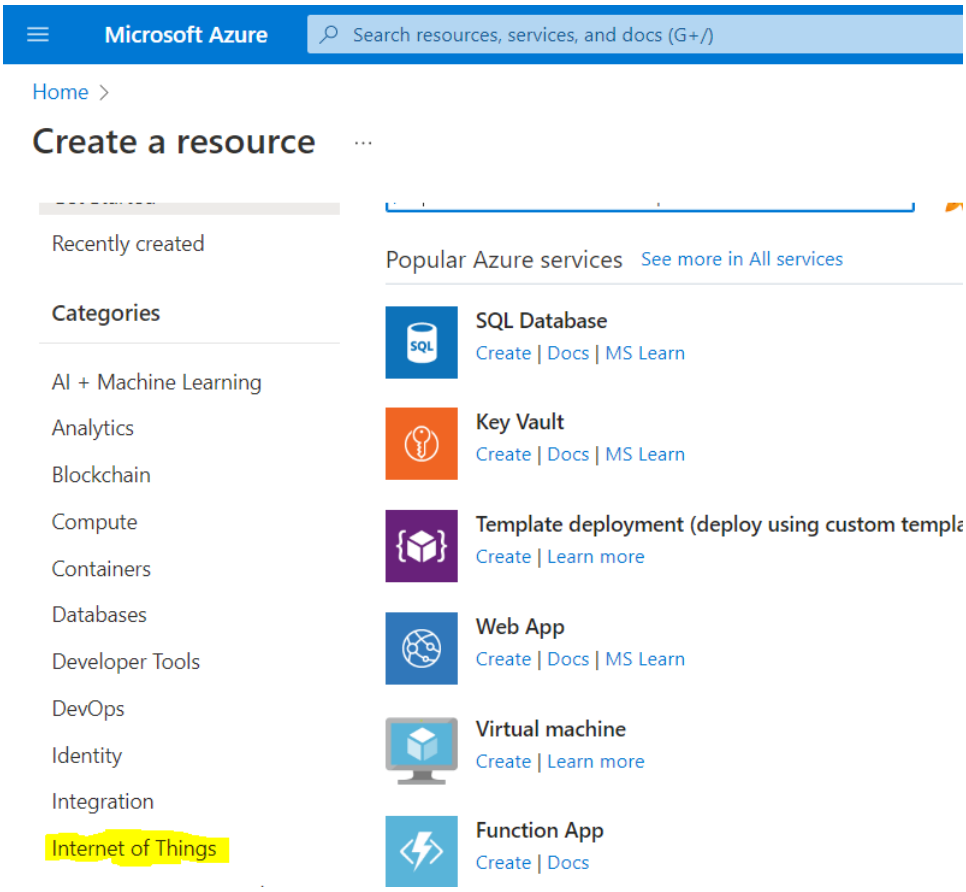
Section3. Create IOT Edge Device and Module

1. Sign in to your Azure portal and navigate to your IoT Hub
   If you need to create a new IoT Hub, please follow the instructions here

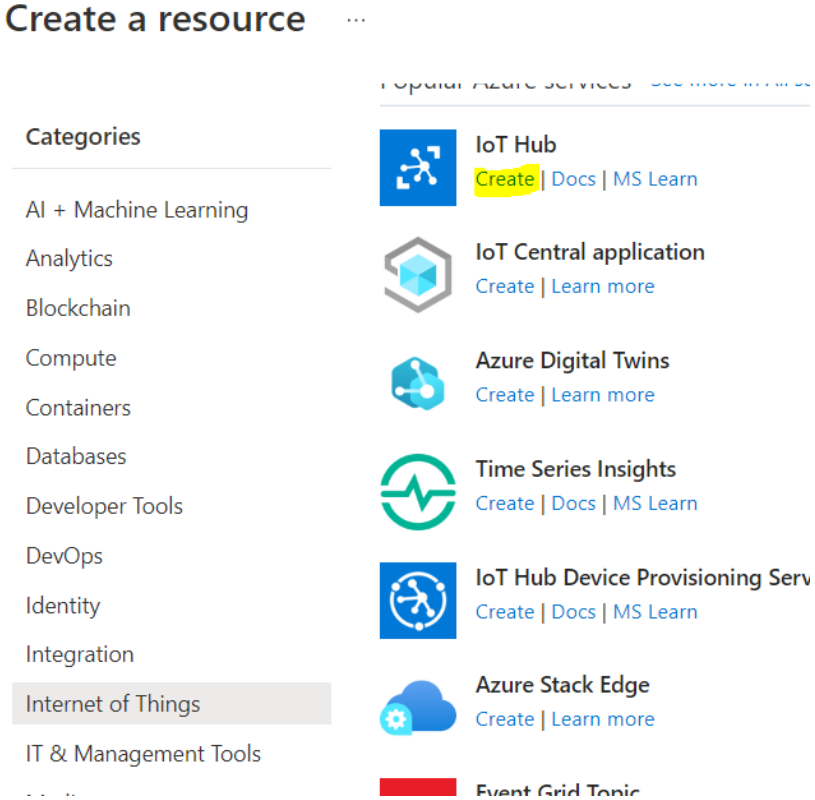
2. Click Create a resource:



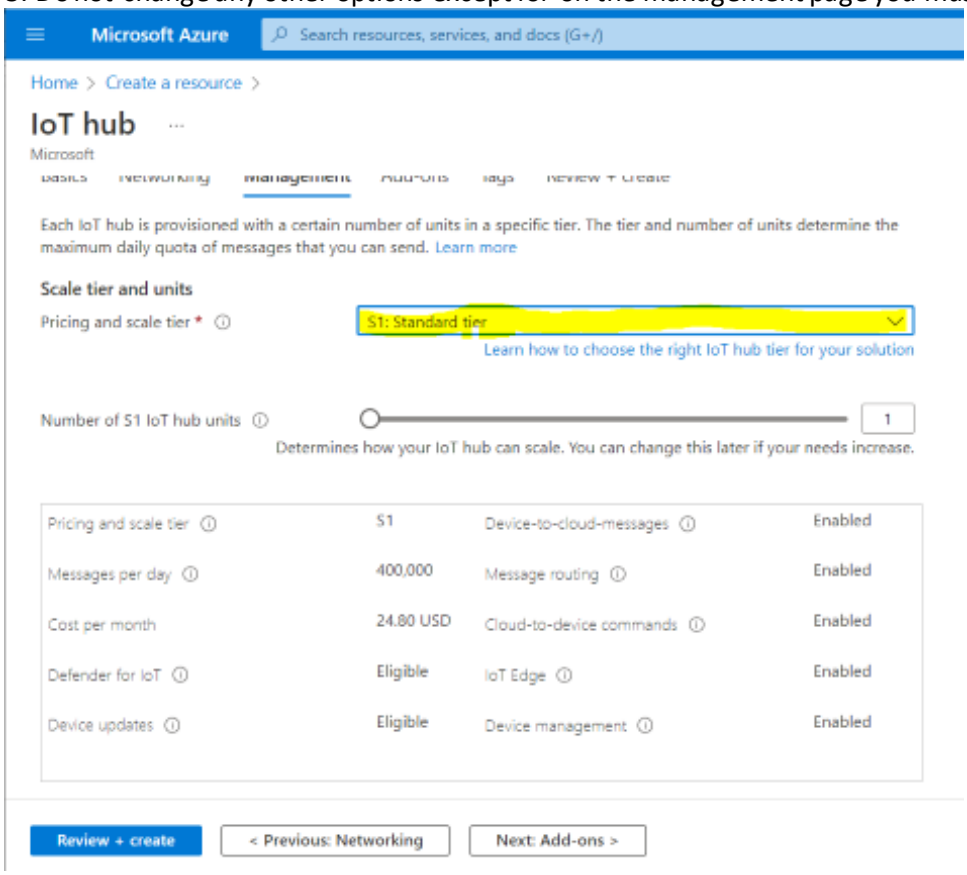
3. Click Internet of things:



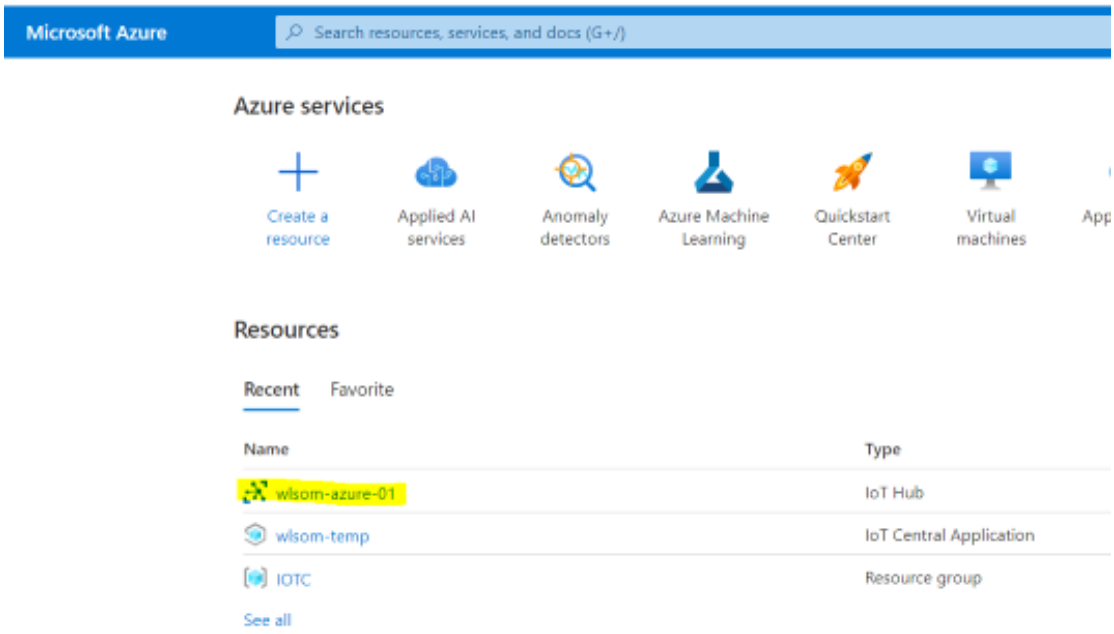
4. Then create under IoT Hub:



5. Do not change any other options except for on the management page you must select your pricing tier:

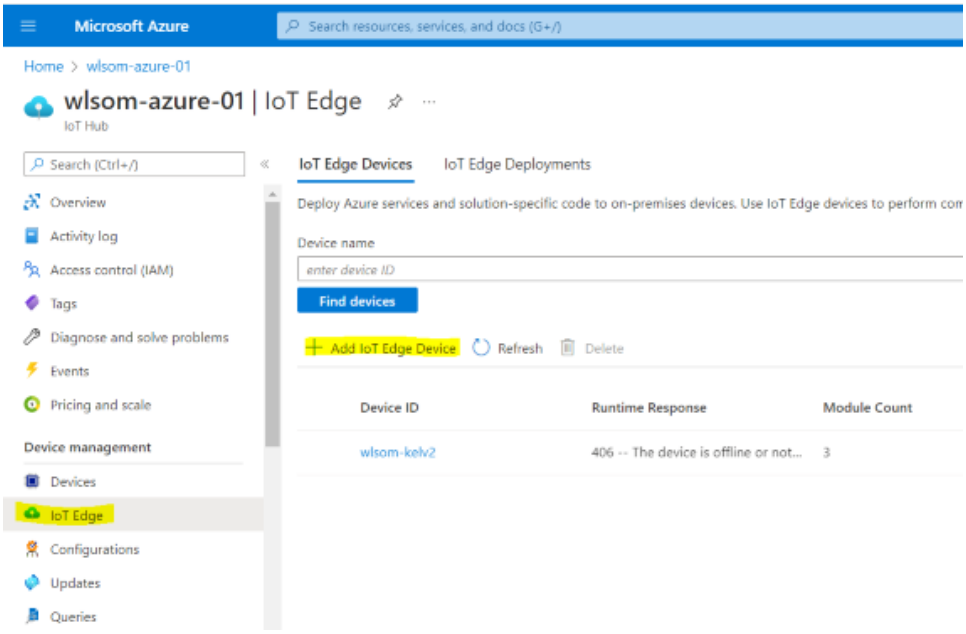


6. The newly created IoT hub resource should show up in the portal. Click on it:



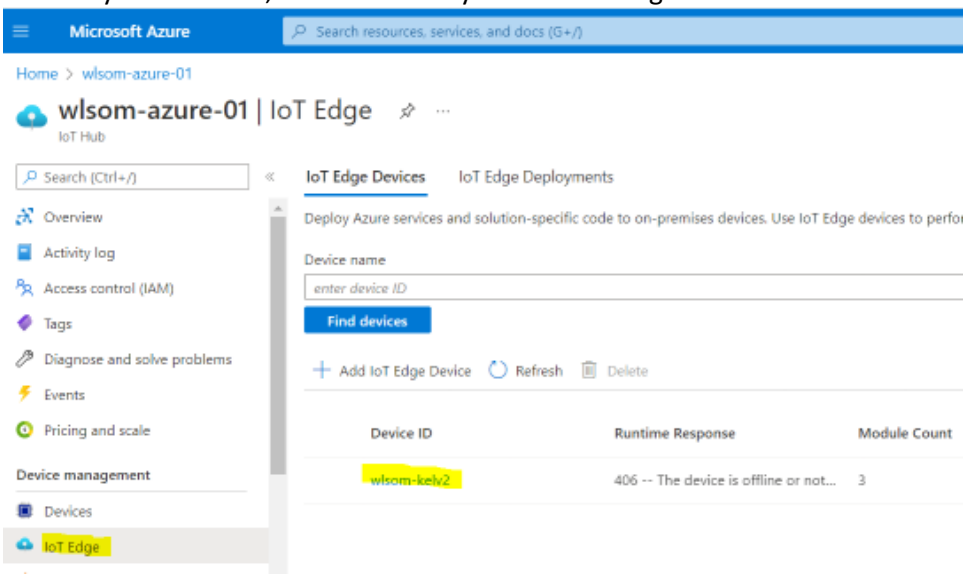
7. Click **Add an IoT Edge Device**

-After selecting the newly created IoT Hub, select IoT Edge and Add IoT Edge Device:

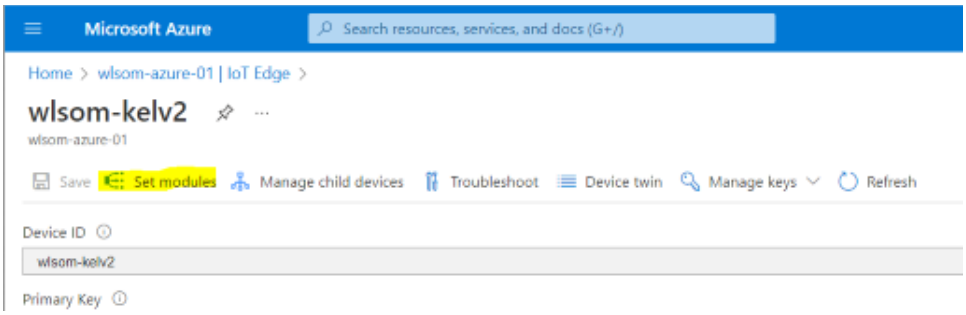


8. Type in the **Device ID(choose any name)** and keep the other configurations as default
    a. Authentication Type: **Symmetric Key**
    b. Auto-Generate Keys: **Enabled**
    c. Connect this device to an IoT Hub: **Enable**

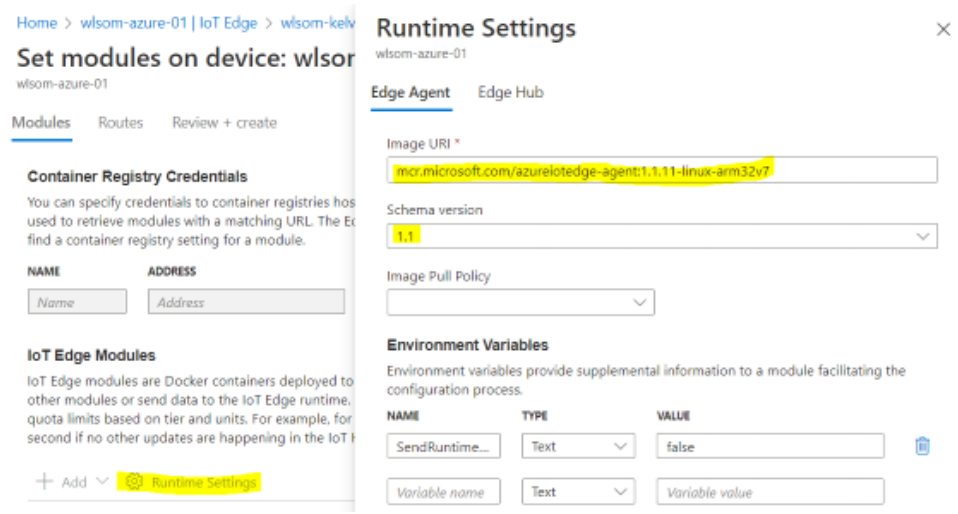
9. Once you click Save, select the newly created IoT Edge Devices from the list



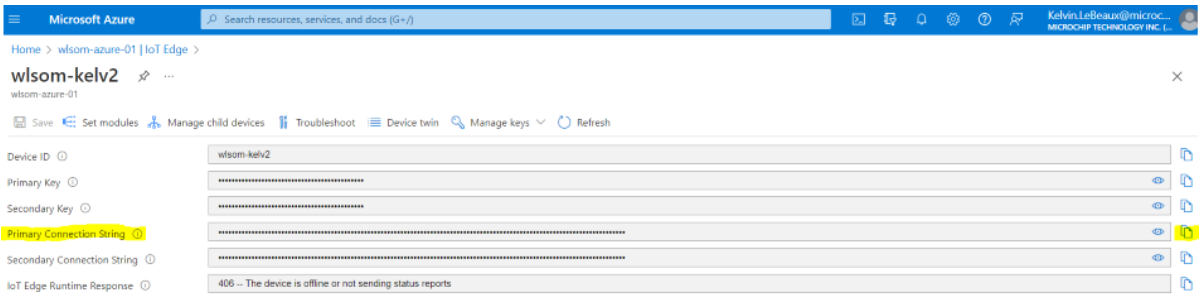
10. Select **Set Modules**



11. Click **Add** and then select **IoT Edge Module** to add a new module
12. Configure the module with the following settings and click **Save:**
    a. Name: **SimulatedTemperatureSensor**
    b. Image URI: **mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.1.11-linux-arm32v7**
13. Click **Runtime Settings**, Change the image value of **Edge Hub** and **Edge Agent** like these, then **Save** it.
    c. Edge Hub Image URI - **mcr.microsoft.com/azureiotedge-hub:1.1.11-linux-arm32v7**
    d. Schema Version – **1.1**
    e. Edge Agent Image - **mcr.microsoft.com/azureiotedge-agent:1.1.11-linux-arm32v7**
    f. Schema Version – **1.1**

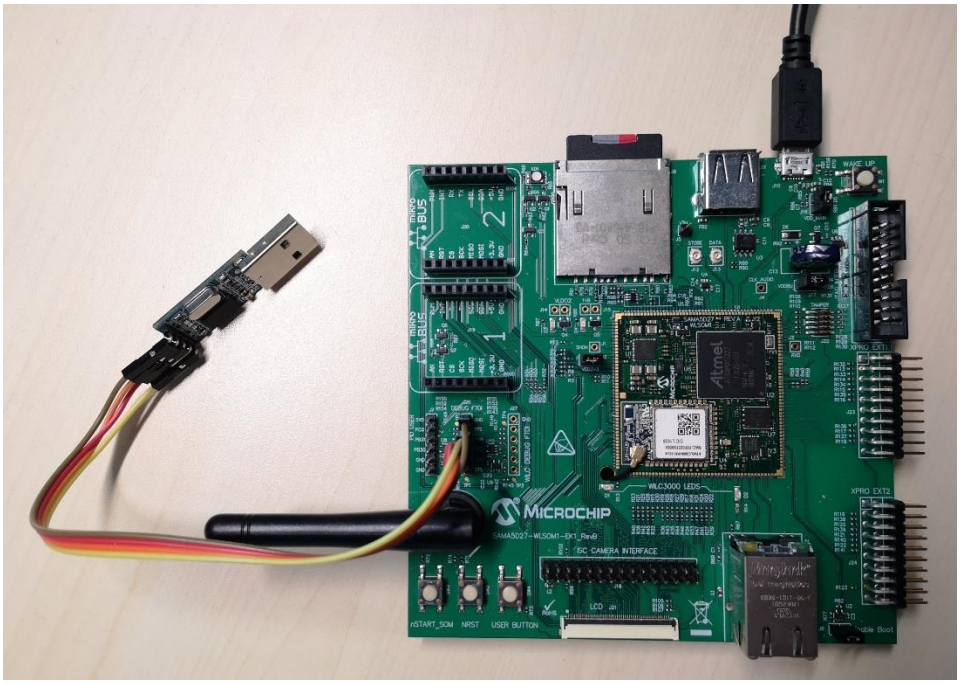


14. Click **Review + create** to review deployment settings.
15. Click **Create** to deploy it.
16. Go back to your created IoT Edge Device and copy the **Primary Connection String** with the icon on the right **(This will be needed to add to the config.yaml file on the WLSOM later)**



Section 4. Bringing Up Image

1. Start and Login:
   -Connect 5V supply into J10 connector
   -Connect a USB-serial adapter to the debug connector J26 of the WLSOM and USB port of computer
   -Connect ethernet cable to J6 on WLSOM and other end to a running Access Point
   (For WiFi - see Appendix 1 - Setting Up WiFi)



-Open serial terminal program of choice and connect to USB adapter serial port with settings: 115200 bps 8-N-1

-Press “nSTART_SOM” button on the WSLOM (There should be messages appearing on the screen)

-When messages are finished enter “root” to login:

```
login as: root
Last login: Tue Aug 16 21:25:46 2022 from 192.168.1.144
root@sama5d27-wlsom1-ek-sd:~#
```

For all of the lines that mention commands must be run on each startup or only run once, there exist shell script files to make this easier. “azurt_start.sh”, and “azure_restart.sh” exist to run the proper commands to start Azure properly from root login of the WLSOM. “azure_start.sh” only needs to run one time, while “azure_restart.sh” must be run on every start/restart.

2. Setting up swap drive:

-Enter the following commands:

sudo fallocate -l 1G /swapfile

sudo chmod 600 /swapfile

sudo mkswap /swapfile

sudo swapon /swapfile **(only this command has to be run on each restart, all others on first time start)**

3. Change ownership of iotedge files **(These commands must be run every restart):**

-Enter the following commands:

chown iotedge:iotedge /var/run/iotedge/
chown iotedge:iotedge /var/lib/iotedge/

4. Modify iotedge configuration file(This only needs to be done once):

-Type the following command to edit the config.yaml file

vi /etc/iotedge/config.yaml

-Paste the primary connection string into the config file (located towards the beginning of the file)

```
Manual provisioning configuration
provisioning:
  source: "manual"
  device_connection_string: "<ADD DEVICE CONNECTION STRING HERE>"
```

-Change the value of hostname, listen.management_uri and listen.workload_uri (located towards the end of the file)

```
hostname: "sama5d27-wlsom1-ek-sd"
listen:
  management_uri: "unix:///var/run/iotedge/mgmt.sock"
  workload_uri: "unix:///var/run/iotedge/workload.sock"
```

5. Starting IoT Edge **(Must be run on each restart):**

-Enter the following command to start IoT edge on the WLSOM:

(May take up to 5 minutes for an error-free connection to be made to the portal)

systemctl restart iotedge

-(Optional) Enter the following command to watch log of iotedge:

(Some warnings/errors may show up, but eventually fix themselves upon pulling image and updating)

journalctl -u iotedge –f

On first startup, messages pulling image tempsensor and edge running shows good connection to Azure cloud(may take up to 30min for a complete connection on first startup):

```
Oct 17 03:40:59 sama5d27-wlsom1-ek-sd iotedged[502]: 2022-10-17T03:40:59Z [INFO]
 - Pulling image mcr.microsoft.com/azureiotedge-simulated-temperature-sensor:1.1
.11-linux-arm32v7...
Oct 17 03:41:49 sama5d27-wlsom1-ek-sd iotedged[502]: 2022-10-17T03:41:49Z [INFO]
 - Checking edge runtime status
Oct 17 03:41:50 sama5d27-wlsom1-ek-sd iotedged[502]: 2022-10-17T03:41:50Z [INFO]
 - Edge runtime is running.
```

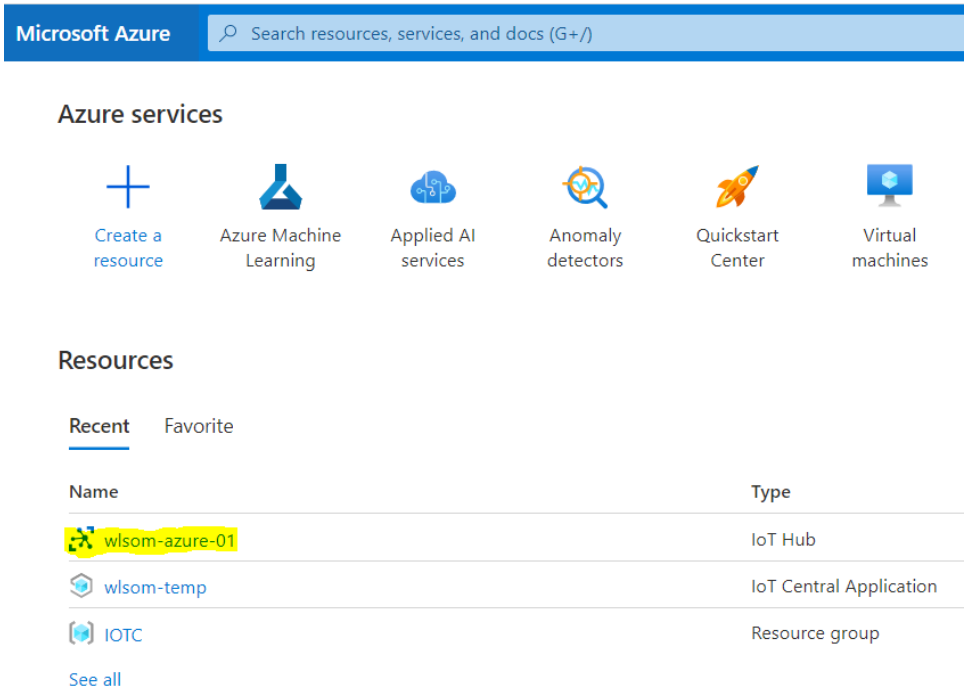
On any other restart, messages that show successful logs for the tempsensor indicates a good connection to Azure cloud (may take up to 15min for a complete connection on other startups):

```
Oct 17 04:49:43 sama5d27-wlsom1-ek-sd iotedged[468]: 2022-10-17T04:49:43Z [INFO]
 - Querying system resources...
Oct 17 04:49:44 sama5d27-wlsom1-ek-sd iotedged[468]: 2022-10-17T04:49:44Z [INFO]
 - [mgmt] - - - [2022-10-17 04:49:44.564915349 UTC] "GET /modules?api-version=20
20-07-07 HTTP/1.1" 200 OK 1767 "-" "-" auth_id(-)
Oct 17 04:49:44 sama5d27-wlsom1-ek-sd iotedged[468]: 2022-10-17T04:49:44Z [INFO]
 - Getting logs for module tempsensor...
Oct 17 04:49:44 sama5d27-wlsom1-ek-sd iotedged[468]: 2022-10-17T04:49:44Z [INFO]
 - Successfully got logs for module tempsensor
Oct 17 04:49:44 sama5d27-wlsom1-ek-sd iotedged[468]: 2022-10-17T04:49:44Z [INFO]
 - [mgmt] - - - [2022-10-17 04:49:44.642813886 UTC] "GET /modules/tempsensor/log
s?api-version=2020-07-07&follow=false&since=15m&tail=1500 HTTP/1.1" 200 OK - "-"
 "-" auth_id(-)
```

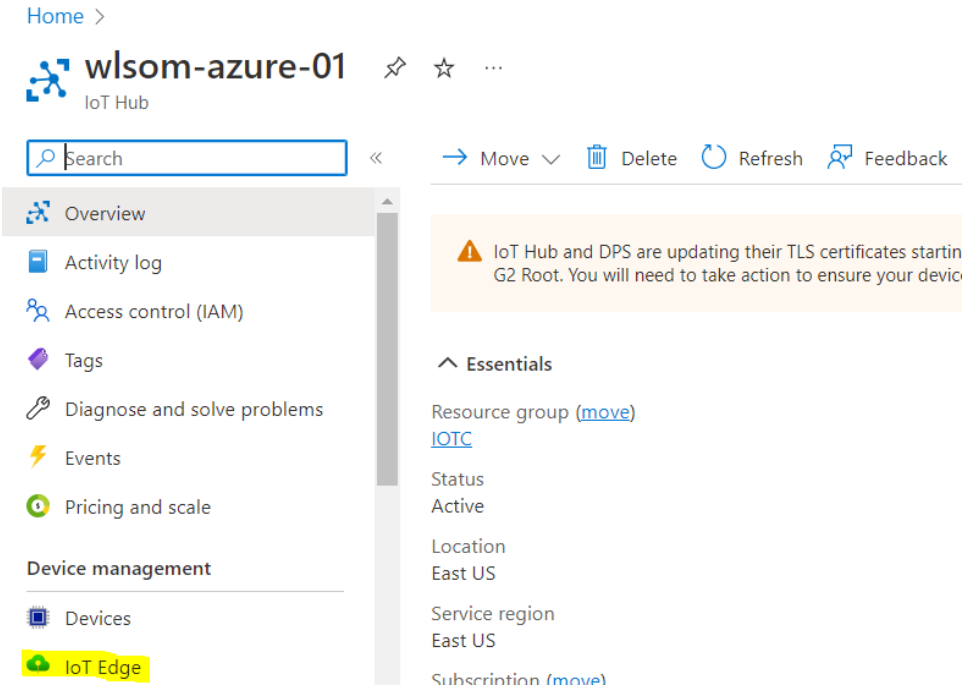
Section 6. Verify WLSOM Data in the Cloud

1. Sign in to your [Azure portal](#) and navigate to your IoT Hub

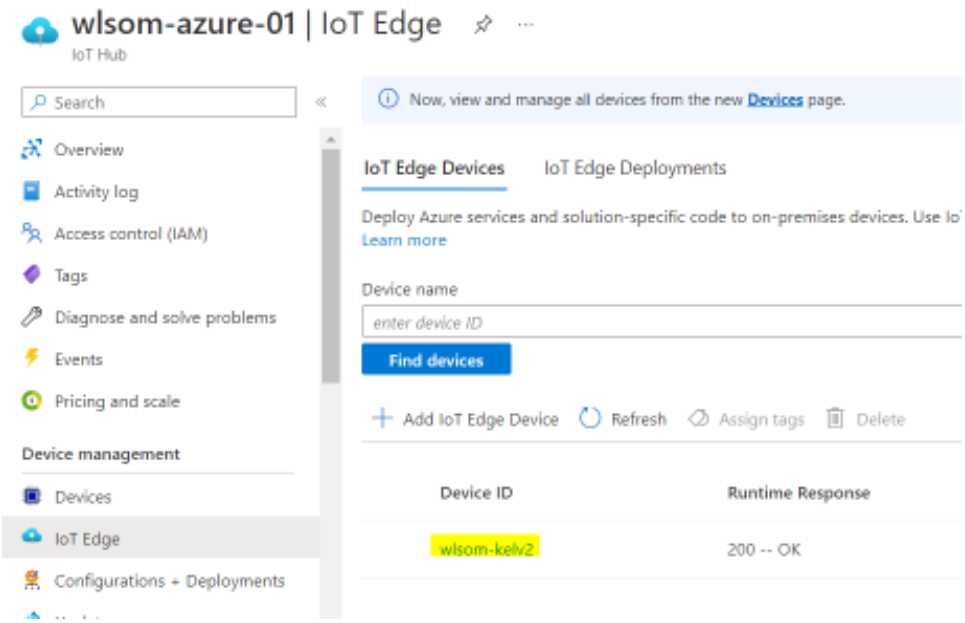
2. Select the IOT hub device created earlier:



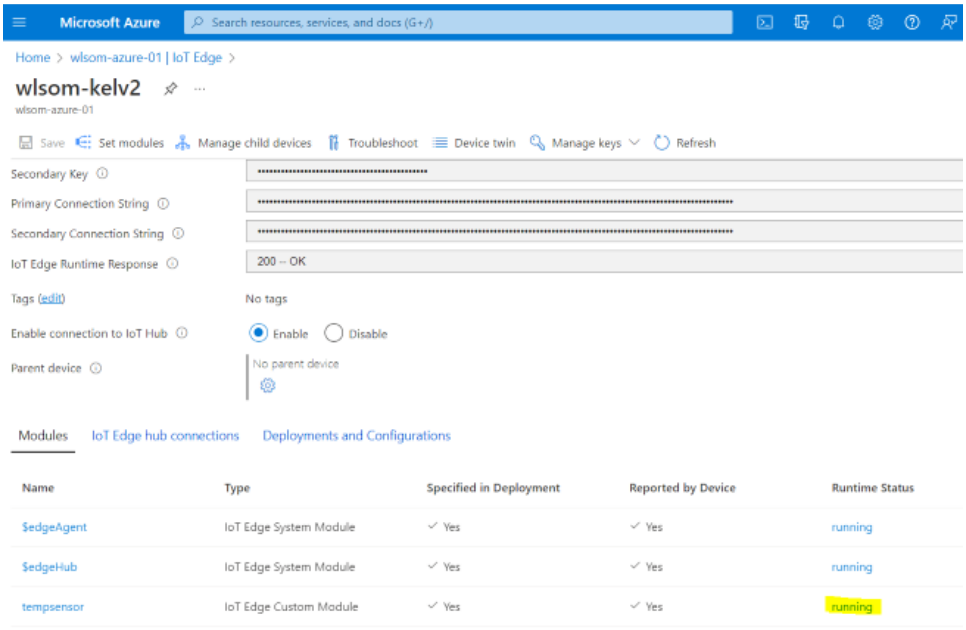
3. Click on the IOT Edge Icon on the left:



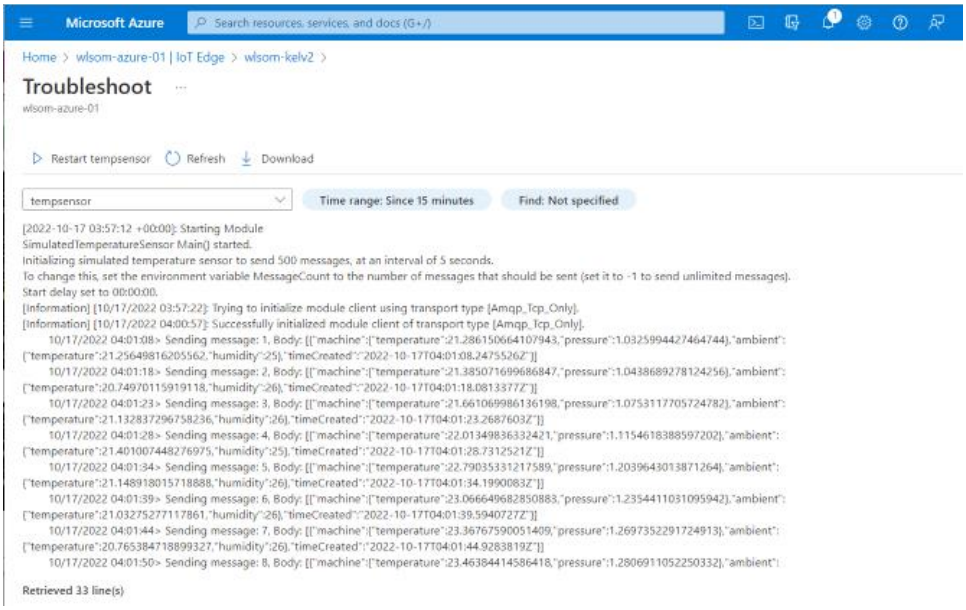
4. Click on the device name of the ito Edge device:



5. Select the running link next to the temperature Sensor module:

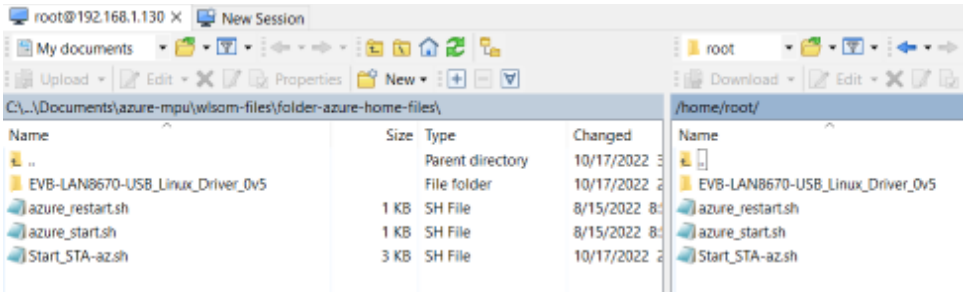


6. The simulated temperature data sent from the wlsom should show up in the log:



Appendix 1. - Setting Up WiFi

All of these steps have been put into a script called Start_STA-az.sh. It will just prompt you for your routers ssid and password then setup the WiFi automatically. You can find this script in a file called azure-home-files.zip. This archive contains this shell script for bringing up wifi as well as scripts for bringing up azure iotedge and the lan687x t1s usb dongle. Just unzip contents and copy the files you need somewhere into the home folder of the wlsom. You can use an SCP program, like WinSCP to connect over the eth0 to your PC for transferring files back and forth.

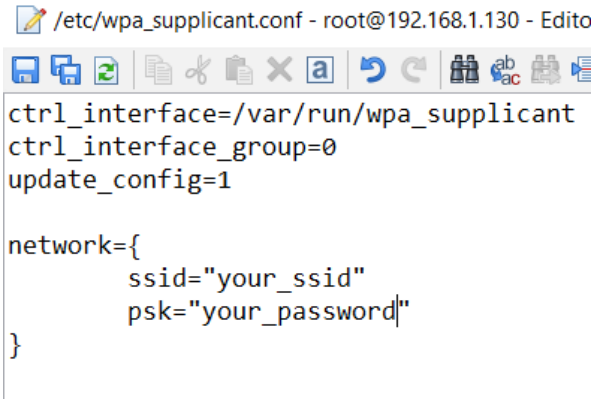


This example was tested on WAP with a default gateway 192.168.1.1 (The user must select an IP address that is within the same subnet as the gateway and rest of the network. 192.168.1.xxx

1. Update wlan0 information in /etc/network/interfaces, if this file doesn't exist, create it

```
root@sama5d27-wlsom1-ek-sd:~# cat /etc/network/interfaces

# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
auto lo
iface lo inet loopback

# Wireless interfaces
auto wlan0
iface wlan0 inet dhcp
        wireless_mode managed
        wireless_essid YOUR_SSID
        wireless_key YOUR_PSK
        wpa-driver wext
        wpa-conf /etc/wpa_supplicant.conf

iface atml0 inet dhcp
```

2. Update /etc/wpa_supplicant.c with your routers ssid and password:



3. Reboot the board

```
root@sama5d27-wlsom1-ek-sd:~# reboot
```

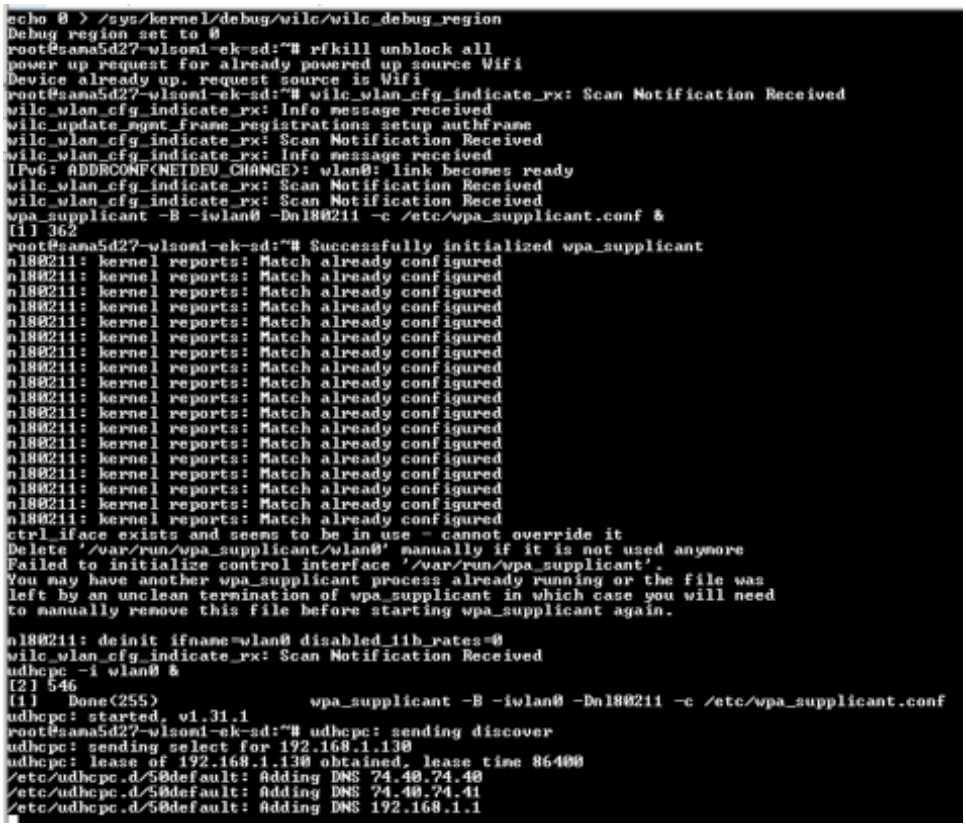
4. Start wpa supplicant:

The following command silences the flood of debug messages that you may see when you turn the wifi on. If you want to see these messages then just omit the following command. It is optional.
echo 0 > /sys/kernel/debug/wilc/wilc_debug_region

The next command allows the wlan0 to turn on. Without this command the wlan0 is blocked from turning on:
rfkill unblock all

This command uses the previously modified file to read your routers SSID and password:
wpa_supplicant -B -iwlan0 -Dnl80211 -c /etc/wpa_supplicant.conf &

5.Start DHCP client, which allows the router to issue an IP address:
udhcpc -i wlan0 &



6. Set wlan0 IP address:

ifconfig wlan0 192.168.1.105

7. Verify whether network can be connected to external sites: