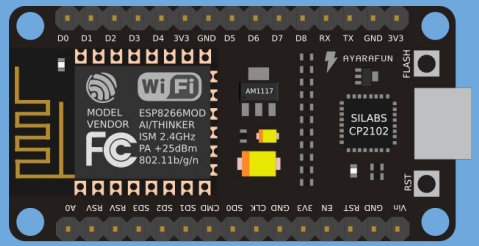
Working With NodeMCU

Speaker -
Shrinit Patil
CSE 3rd Year
GCOEN

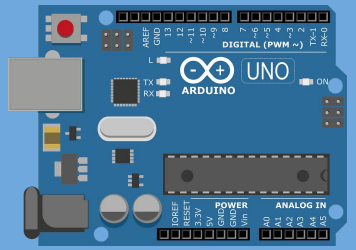
Introduction to NodeMCU



The NodeMCU (Node MicroController Unit) is an open source software and hardware development environment that is built around a very inexpensive System-on-a-chip (SoC) called ESP8266 and ESP32.

- Programmable pins
- Power via micro USB and low energy consumption
- Built-in Wi-Fi
- Program using Arduino IDE as well as RTOS(Real Time Operating System)
- Community support
- Low cost

I know Arduino environment then why to use NodeMCU for IoT?

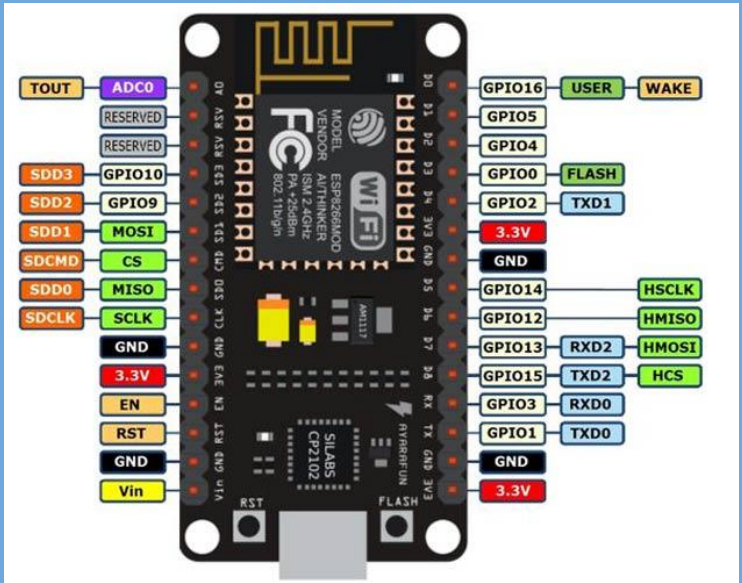


Arduino UNO

- No on board Wi - Fi (available options are costlier)
- Memory limitations (only 2KB SRAM for runtime)
- Much costlier than NodeMCU boards with same function

NodeMCU (ESP8266) Specifications & Features

Pinout of NodeMCU (ESP8266)



- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- GPIO Pins (DIO): 16
- Analog Input Pins (ADC): 1
- UARTs: 1
- SPIs: 1
- I2Cs: 1
- Flash Memory: 4 MB
- SRAM: 64 KB
- Clock Speed: 80 MHz
- Wireless connectivity: Wi-Fi

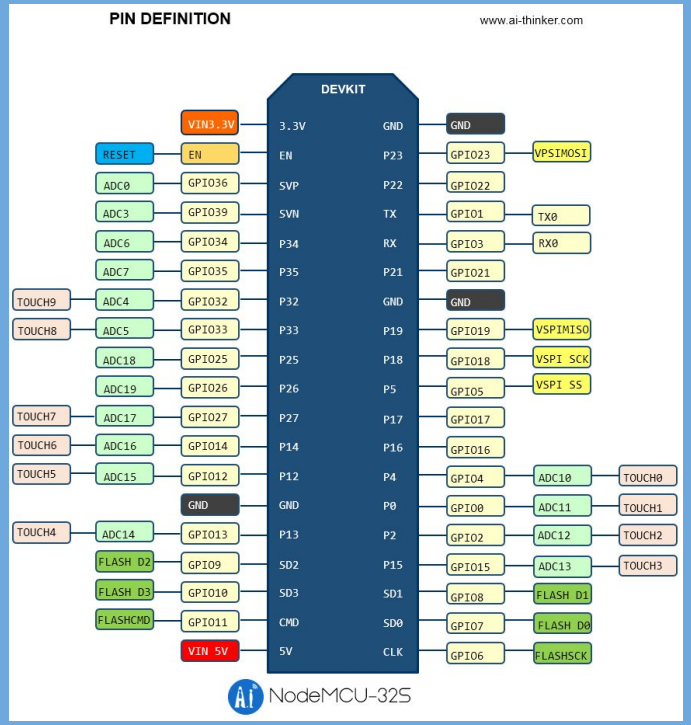
Reference - https://docs.zerynth.com/latest/official/board.zerynth.nodemcu3/docs/index.html

Pin Category	Name	Description
Power	Micro-USB, 3.3V, GND, Vin	**Micro-USB:** NodeMCU can be powered through the USB port **3.3V:** Regulated 3.3V can be supplied to this pin to power the board **GND:** Ground pins **Vin:** External Power Supply
Control Pins	**EN, RST**	The pin and the button resets the microcontroller
Analog Pin	A0	Used to measure analog voltage in the range of 0-3.3V
GPIO Pins	GPIO1 to GPIO16	NodeMCU has 16 general purpose input-output pins on its board
SPI Pins	SD1, CMD, SD0, CLK	NodeMCU has four pins available for SPI communication.
UART Pins	TXD0, RXD0, TXD2, RXD2	NodeMCU has two UART interfaces, UART0 (RXD0 & TXD0) and UART1 (RXD1 & TXD1). UART1 is used to upload the firmware/program.
I2C Pins		NodeMCU has I2C functionality support but due to the internal functionality of these pins, you have to find which pin is I2C.

Reference : https://bit.ly/37mLeMb

NodeMCU (ESP32) Specifications & Features

Pinout of NodeMCU (ESP32)



- Operating Voltage: 3.3V
- Input Voltage: 7-12V
- GPIO Pins (DIO): 28
- Analog Input Pins (ADC): 8
- UARTs: 3
- SPIs: 2
- I2Cs: 3
- Flash Memory: 4 MB
- SRAM: 520 KB
- Clock Speed: 240 MHz
- Wireless connectivity: Wi-Fi, Bluetooth

Reference - https://docs.zerynth.com/latest/official/board.zerynth.nodemcu_esp32/docs/index.html

Pin Category	Pin Name	Details
Power	**Micro-USB, 3.3V, 5V, GND**	**Micro-USB:** ESP32 can be powered through USB port **5V:** Regulated 5V can be supplied to this pin which is we be again regulated to 3.3V by on board regulator, to power the board. **3.3V:** Regulated 3.3V can be supplied to this pin to power the board. **GND:** Ground pins.
Enable	**En**	The pin and the button resets the microcontroller.
Analog Pins	**ADC1_0 to ADC1_5 and ADC2_0 to ADC2_9**	Used to measure analog voltage in the range of 0-3.3V. 12-bit 18 Channel ADC
DAC pins	**DAC1 and DAC2**	Used for Digital to analog Conversion
Input/Output Pins	**GPIO0 to GPIO39**	Totally 39 GPIO pins, can be used as input or output pins. 0V (low) and 3.3V (high). But pins 34 to 39 can be used as input only
Capacitive Touch pins	**T0 to T9**	These 10 pins can be used a touch pins normally used for capacitive pads
RTC GPIO pins	**RTCIO0 to RTCIO17**	These 18 GPIO pins can be used to wake up the ESP32 from deep sleep mode.

Serial	Rx, **Tx**	Used to receive and transmit TTL serial data.
External Interrupts	All GPIO	Any GPIO can be use to trigger an interrupt.
PWM	All GPIO	16 independent channel is available for PWM any GPIO can be made to work as PWM though software
VSPI	GPIO23 (MOSI), GPIO19(MISO), GPIO18(CLK) and GPIO5 (CS)	Used for SPI-1 communication.
HSPI	GPIO13 (MOSI), GPIO12(MISO), GPIO14(CLK) and GPIO15 (CS)	Used for SPI-2 communication.
IIC	GPIO21(SDA), GPIO22(SCL)	Used for I2C communication.
AREF	**AREF**	To provide reference voltage for input voltage.

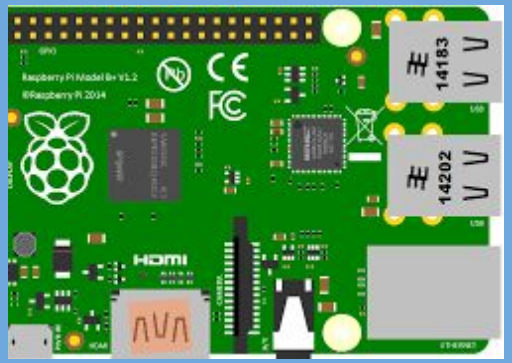
Reference : https://components101.com/microcontrollers/esp32-devkitc

THANK YOU

Raspberry Pi

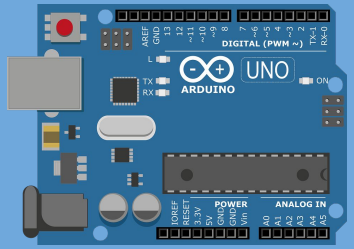
Speaker -
Swanand Jugade
Mech Final Year
GCOEN

Introduction to Raspberry Pi

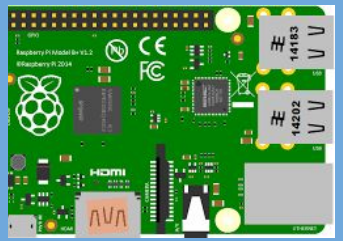


The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. The Raspberry Pi operates in the open source ecosystem: it runs Linux (a variety of distributions), and its main supported operating system, Raspbian, is open source and runs a suite of open source software.

Arduino vs RPi



Arduino is a microcontroller that can run one simple program over and over again.



Raspberry Pi is a general-purpose computer, usually with a linux operating system. It has ability to run multiple programs.

Different Models of RPi

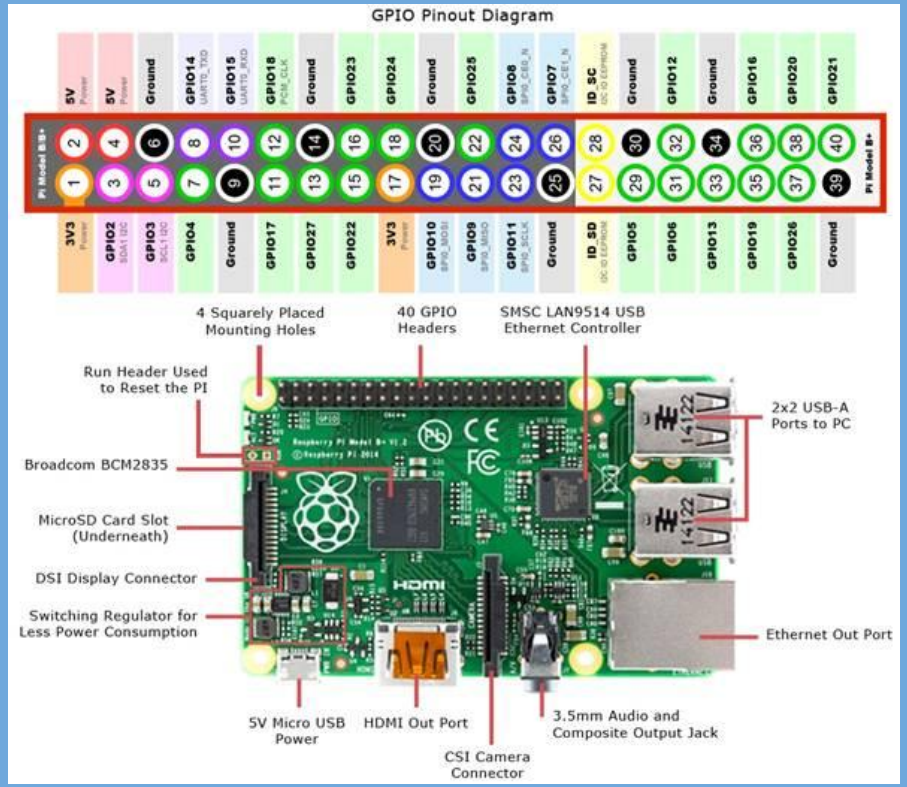
Raspberry Pi Model B Comparison

Raspberry Pi Model B Specifications

Feature	Pi 1 B	Pi 1 B+	Pi 2 B	Pi 2 B V1.2	Pi 3 B	Pi 3 B+
Released	Apr 2012	Jul 2014	Feb 2015	Oct 2016	Feb 2016	Mar 2018
Architecture	ARMv6Z 32-bit	ARMv6Z 32-bit	ARMv7-A 32-bit	ARMv8-A 64/32-bit	ARMv8-A 64/32-bit	ARMv8-A 64/32-bit
SoC	Broadcom BCM2835	Broadcom BCM2835	Broadcom BCM2836	Broadcom BCM2837	Broadcom BCM2837	Broadcom BCM2837B0
CPU	700 MHz ARM1176JZF-S	700 MHz ARM1176JZF-S	900 MHz ARM Cortex-A7	900 MHz ARM Cortex-A53	1.2 GHz ARM Cortex-A53	1.4 GHz ARM Cortex-A53
Cores	1	1	4	4	4	4
GPU	Broadcom VideoCore IV HD 1080p	Broadcom VideoCore IV HD 1080p	Broadcom VideoCore IV HD 1080p	Broadcom VideoCore IV HD 1080p	Broadcom VideoCore IV HD 1080p	Broadcom VideoCore IV HD 1080p
Memory RAM	512 MB	512 MB	1 GB	1 GB	1 GB	1 GB
Operating System	Primarily Linux based	Primarily Linux based	Primarily Linux based	Primarily Linux based	Primarily Linux based	Primarily Linux based
USB 2.0 Ports	2	4	4	4	4	4
Camera Input	15-pin CSI (Camera Serial Interface)	15-pin CSI (Camera Serial Interface)	15-pin CSI (Camera Serial Interface)	15-pin CSI (Camera Serial Interface)	15-pin CSI (Camera Serial Interface)	15-pin CSI (Camera Serial Interface)
Video Output	Composite 3.5 mm RCA and HDMI	HDMI, Composite, DSI (Display Serial Interface)	HDMI, Composite, DSI (Display Serial Interface)	HDMI, Composite, DSI (Display Serial Interface)	HDMI, Composite, DSI (Display Serial Interface)	HDMI, Composite, DSI (Display Serial Interface)
Audio Output	Analog 3.5 mm jack, Digital via HDMI	Analog 3.5 mm jack, Digital via HDMI	Analog 3.5 mm jack, Digital via HDMI	Analog 3.5 mm jack, Digital via HDMI	Analog 3.5 mm jack, Digital via HDMI	Analog 3.5 mm jack, Digital via HDMI
Storage	SD slot	Micro SD slot	Micro SD slot	Micro SD slot	Micro SD slot	Micro SD slot
Ethernet	10/100 Mbps	10/100 Mbps	10/100 Mbps	10/100 Mbps	10/100 Mbps	10/100/1000 (max 300) Mbps
Onboard WiFi	None	None	None	None	2.4 Ghz 802.11 b/g/n	2.4 Ghz and 5 Ghz 802.11 b/g/n/ac
Onboard Bluetooth ®	None	None	None	None	4.1 BLE	4.2 BLE
Input/Output Pins	26	40	40	40	40	40
Power (less peripherals)	5v 700 ma	5v 320 ma	5v 750 ma	5v 750 ma	5v 850 ma	5v 950 ma
Size	85 mm x 56 mm	85 mm x 56 mm	85 mm x 56 mm	85 mm x 56 mm	85 mm x 56 mm	85 mm x 56 mm

Reference - https://www.rs-online.com/designspark/raspberry-pi-model-comparison-table

Raspberry Pi 3B+ Hardware Specifications



Reference - https://www.watelectronics.com/know-all-about-raspberry-pi-board-technology/

THANK YOU

Programming NodeMCU and Raspberry Pi

Speaker -
Ameya Shahu
CSE 4th Year
GCOEN

Programming of NodeMCU

Programming Framework - Mongoose OS (MOS tool)

What is Mongoose OS?

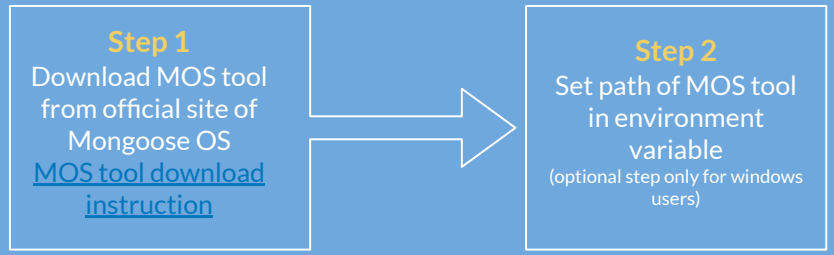
Mongoose OS is a cross-platform IoT operating system, providing a generic infrastructure layer for smart products, targeting production environment.

Why to choose Mongoose OS over other platforms?

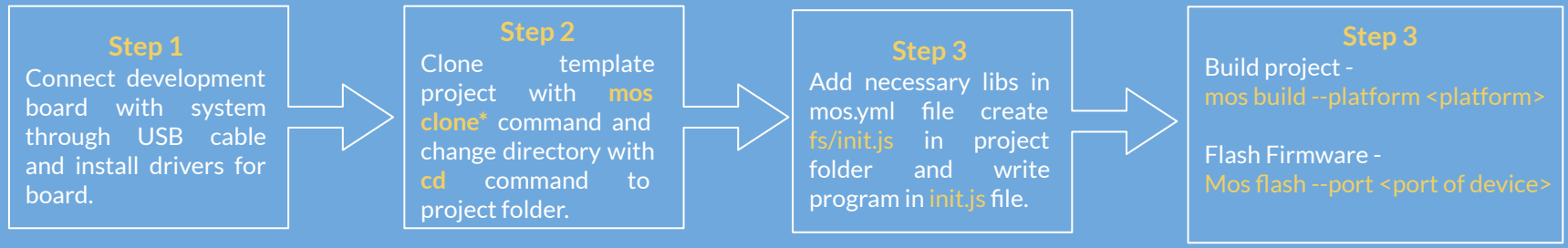
- secure backend integration
- mutual cloud authentication
- remote management
- Over The Air firmware update
- Quick-and-easy setup process, outstanding experience of the MOS UI tool
- Development in JavaScript , C or both

Reference - https://blog.squix.org/2017/09/esp8266-and-esp32-interview-with-sergey-lyubka-from-mongoose-os.html

Installation of Mongoose OS

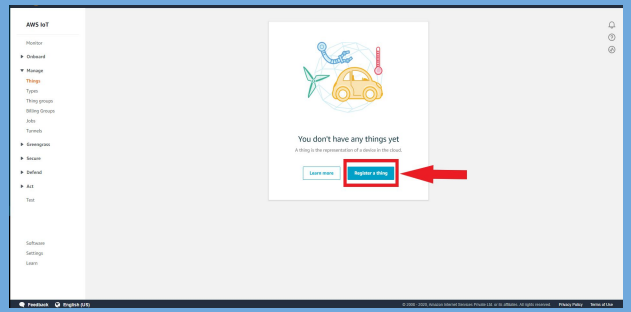


New App Development with Mongoose OS

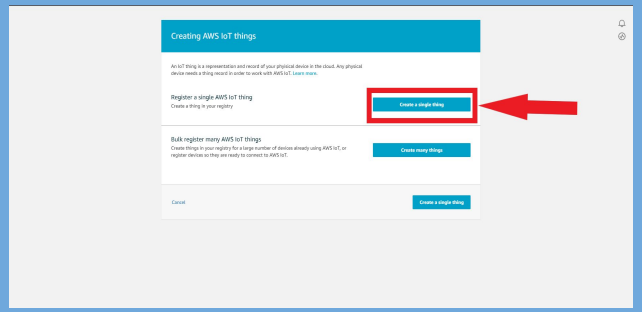


* For empty project - mos clone https://github.com/mongoose-os-apps/empty <name of project>

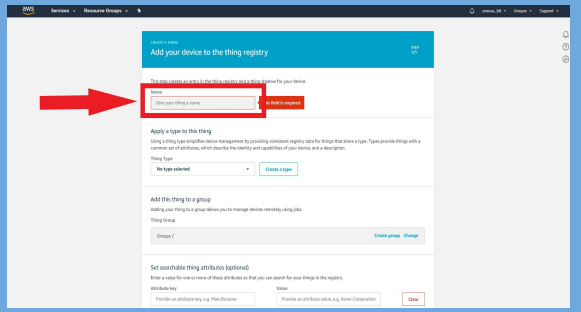
Creating Thing on AWS IoT Core platform



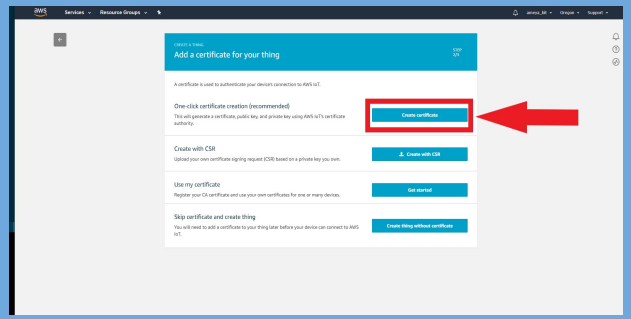
Step 1 :- Login into aws console go to IoT Core and click on create things



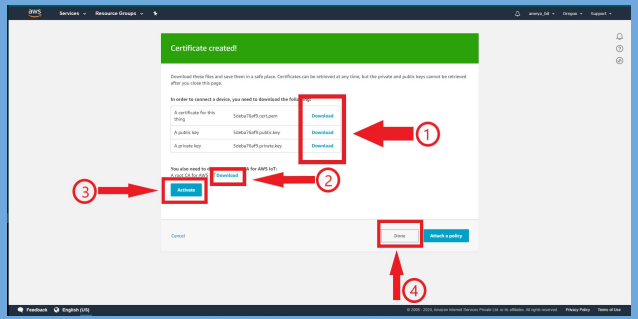
Step 2 :- Click on create things



Step 3:- Enter name for thing (this name will be use as deviceID)

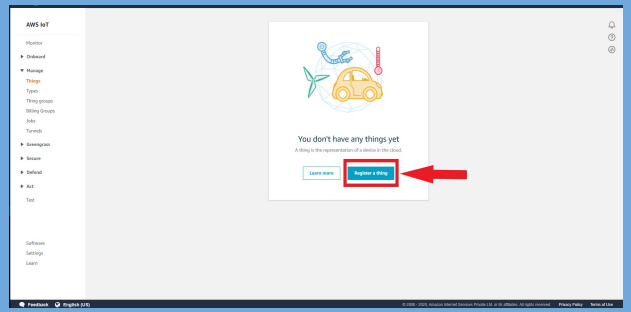


Step 4 :- Generate certificates and keys.

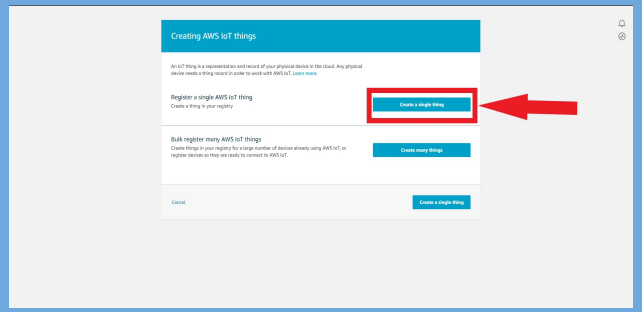


Step 5 :- Download SSL certificate, private and public key and RootCA certificate. Activate certificates and no need to attach policy at this time

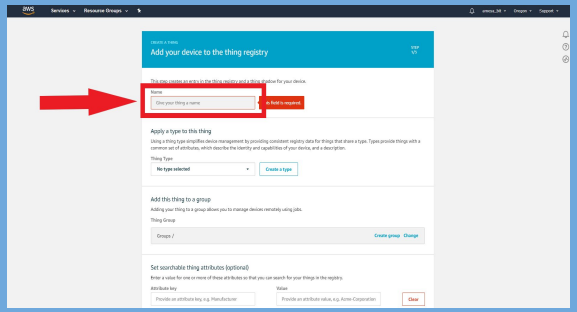
Create and attach policy to thing



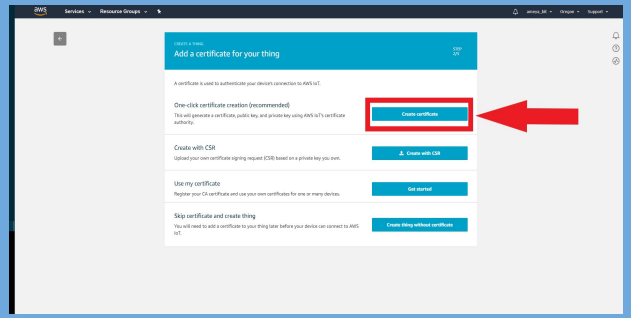
Step 1 :- Login into aws console go to IoT Core and click on create things



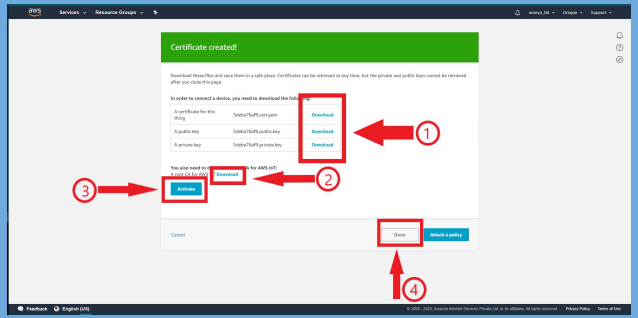
Step 2 :- Click on create things



Step 3:- Enter name for thing (this name will be use as deviceID)



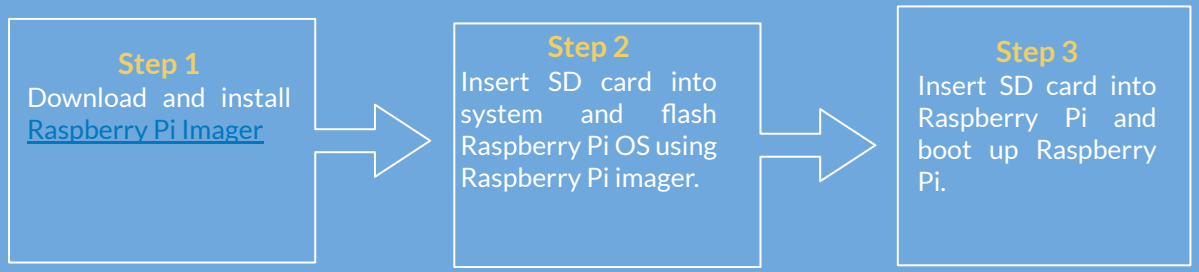
Step 4 :- Generate certificates and keys.



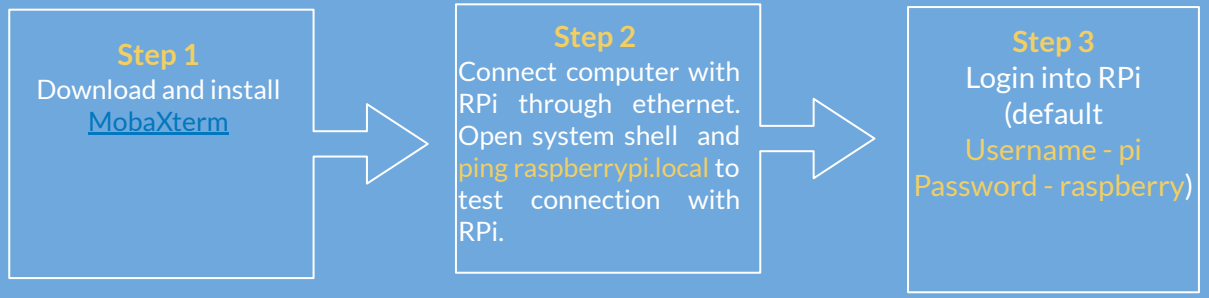
Step 5 :- Download SSL certificate, private and public key and RootCA certificate. Activate certificates and no need to attach policy at this time

Working with Raspberry Pi

Setup of Raspberry Pi



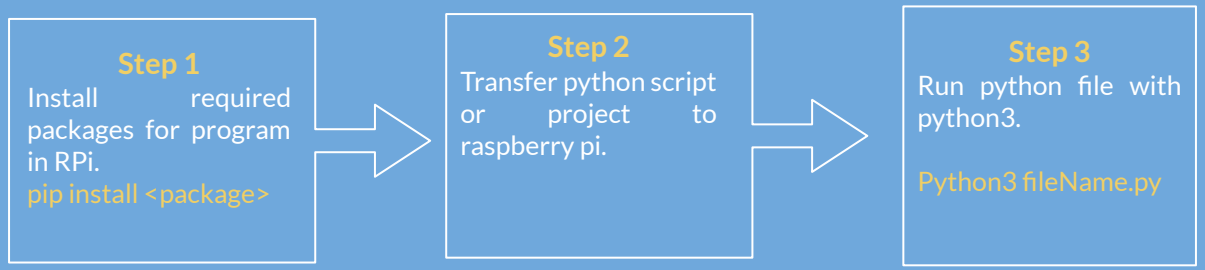
SSH connection to Raspberry Pi



Wireless LAN setup in Raspberry pi

1. Open the wpa-supplicant configuration file in nano:
   sudo nano /etc/wpa_supplicant/wpa_supplicant.conf
2. Go to the bottom of the file and add the following:
   network={
   ssid="wifi name"
   pass="password of wifi"
   }
3. Restart wifi
   sudo service networking restart

Executing Python Program in Raspberry Pi



THANK YOU

Questions and feedbacks are welcome

Connect with me -


ameya-shahu


ameya-shahu-873035184


ameya_shahu