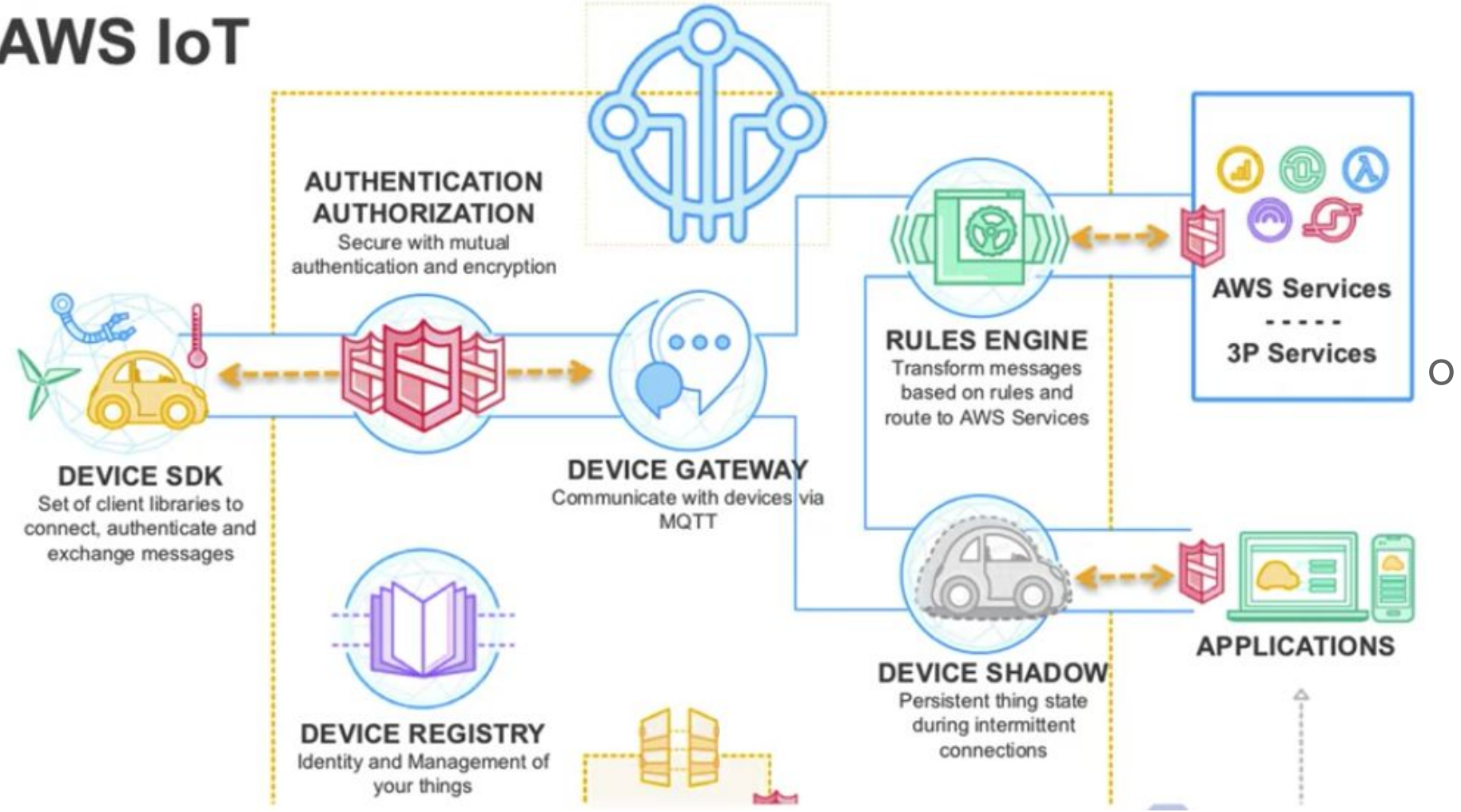
ESP32 & AWS

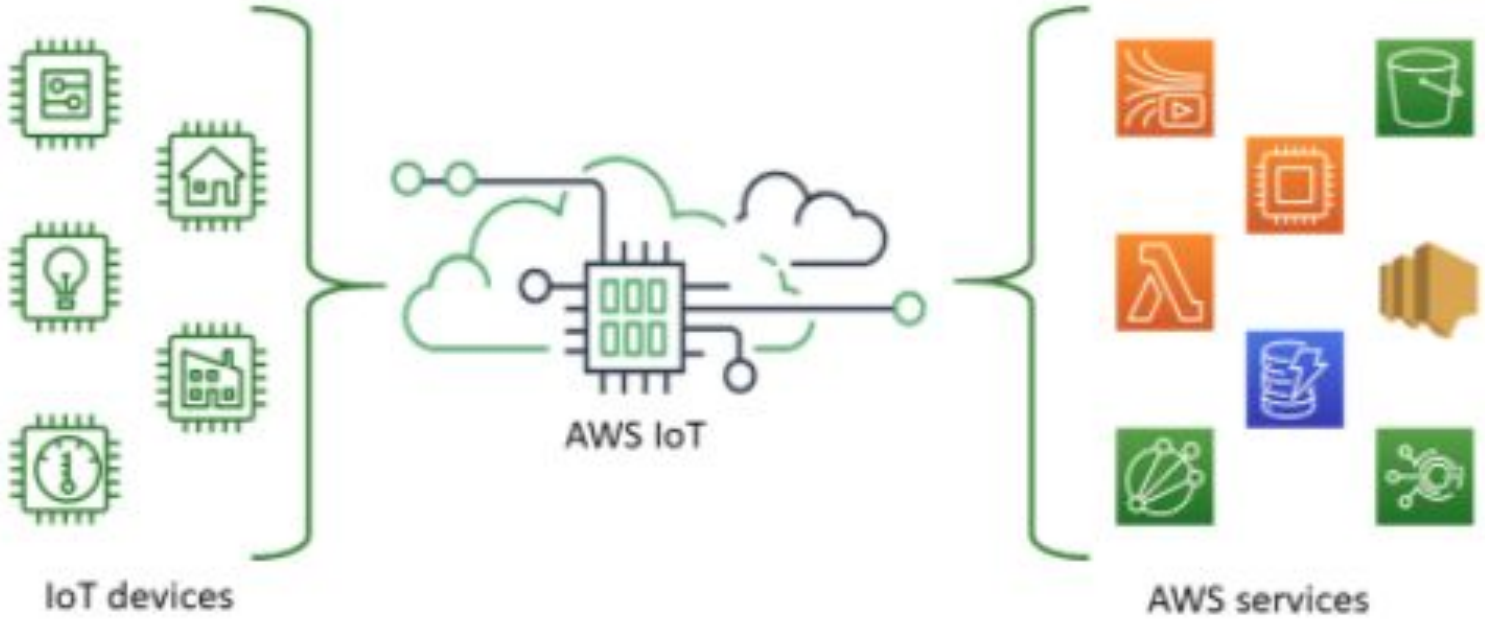
Luis Rey Lara

llara_ptc@upjr.edu.mx

AWS IoT

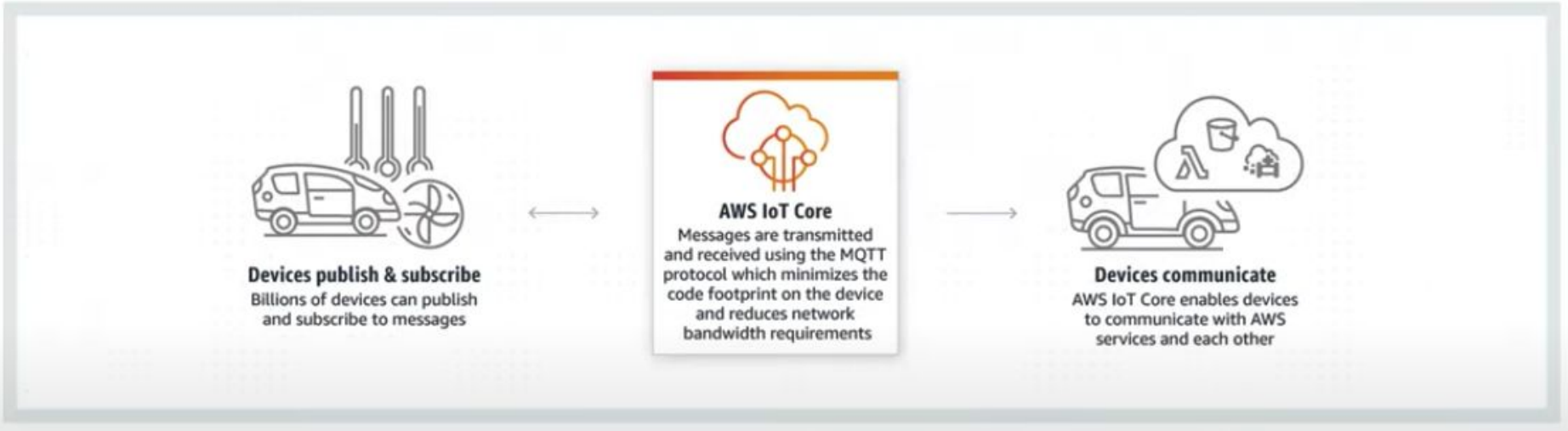

AUTHENTICATION
AUTHORIZATION
Secure with mutual authentication and encryption
RULES ENGINE
Transform messages based on rules and route to AWS Services
AWS Services
- - - - -
3P Services
DEVICE SDK
Set of client libraries to connect, authenticate and exchange messages
DEVICE GATEWAY
Communicate with devices via MQTT
DEVICE SHADOW
Persistent thing state during intermittent connections
APPLICATIONS
DEVICE REGISTRY
Identity and Management of your things

O


AWS IoT
IoT devices
AWS services

Requisitos:

- Cuenta de AWS Educate
- Curso básico de ESP32
- MQTT Básico
  - https://github.com/luisreylaragonzalez/ESP32_AWS


Devices publish & subscribe
Billions of devices can publish and subscribe to messages
AWS IoT Core
Messages are transmitted and received using the MQTT protocol which minimizes the code footprint on the device and reduces network bandwidth requirements
Devices communicate
AWS IoT Core enables devices to communicate with AWS services and each other

Arduino Nano 33

Arduino Nano Every

Arduino Nano 33 Ble

Arduino UNO

Teensy 4.0

Tessel 2

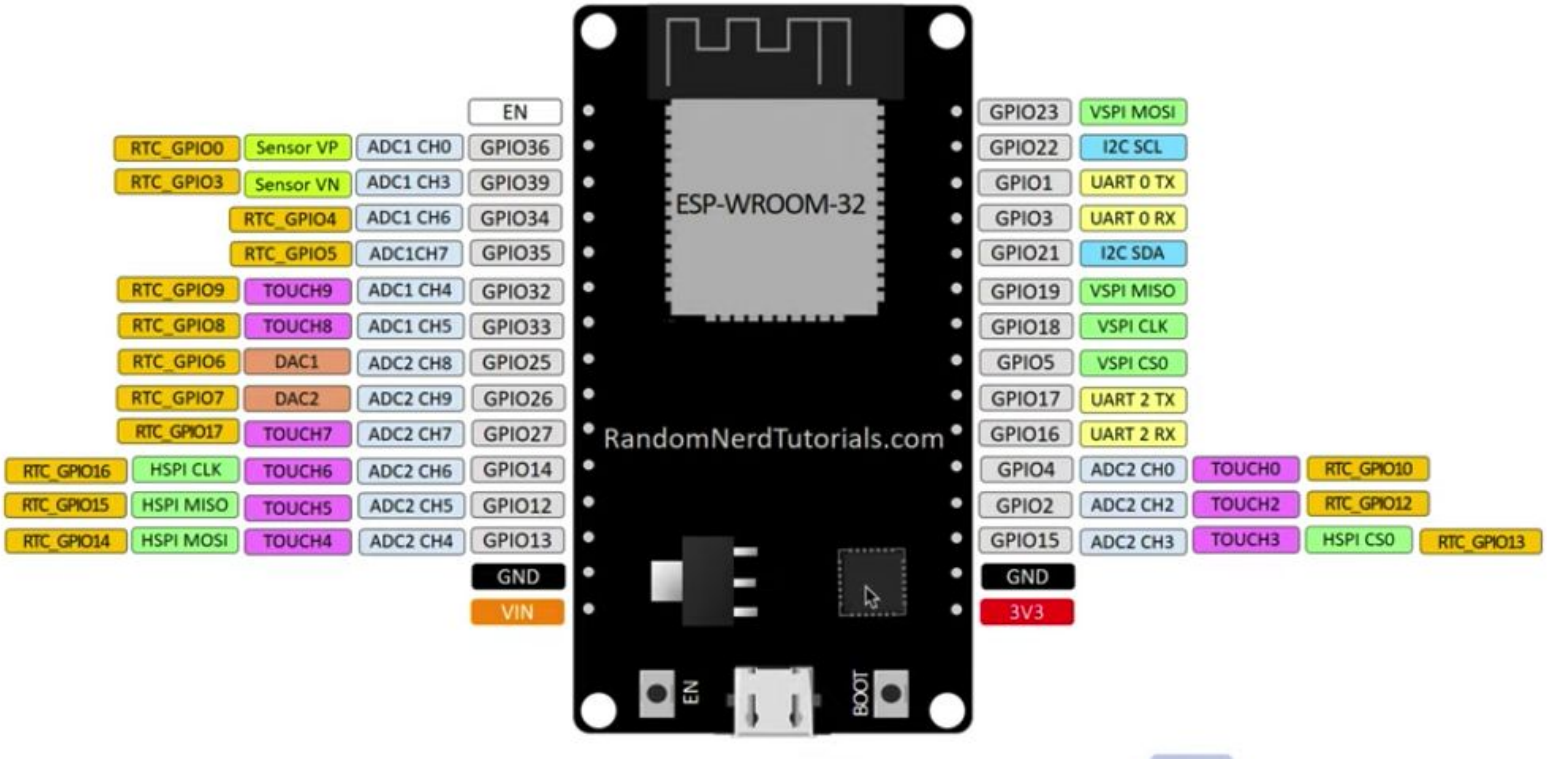
Onion Omega2s+

ESP32

Particle Argon

ESP32 DEVKIT V1 – DOIT

version with 30 GPIOs



Communication (MQTT)

Publish – Topic, Message

Subscribe – Topic

MQTT Broker

AWS IoT Core

Host
IP address
*AWS_IoT_Endpoint Hostname

Port
MQTT: 1883
*MQTTS: 8883

1.- Device Authentication

Vamos a configurar el sistema embebido para que se identifique en AWS

1. Device Authentication

AWS IoT Core

AWS IoT

Monitor
Activity
Onboard
Manage
Greengrass
Secure
Defend
Act
Test

Software
Settings
Learn
Documentation

AWS IoT

AWS IoT is a managed cloud platform that lets connected devices - cars, light bulbs, sensor grids, and more - easily and securely interact with cloud applications and other devices.

Connect and manage your devices

Connect devices to the cloud using the protocol that best fits your requirements - HTTP, MQTT, or WebSocket. Devices can

Process and act upon device data

Filter, transform, and act upon data from devices on the fly, based on business rules. AWS IoT can be easily integrated with AWS

Read and set device state at any time

AWS IoT stores the latest state of a device so that it can be read or set anytime, even when the device is offline.

1a. Create Thing in AWS IoT Core, save certificate and keys

AWS IoT Core

1b. Download the Amazon Root Certificate

AWS IoT Core

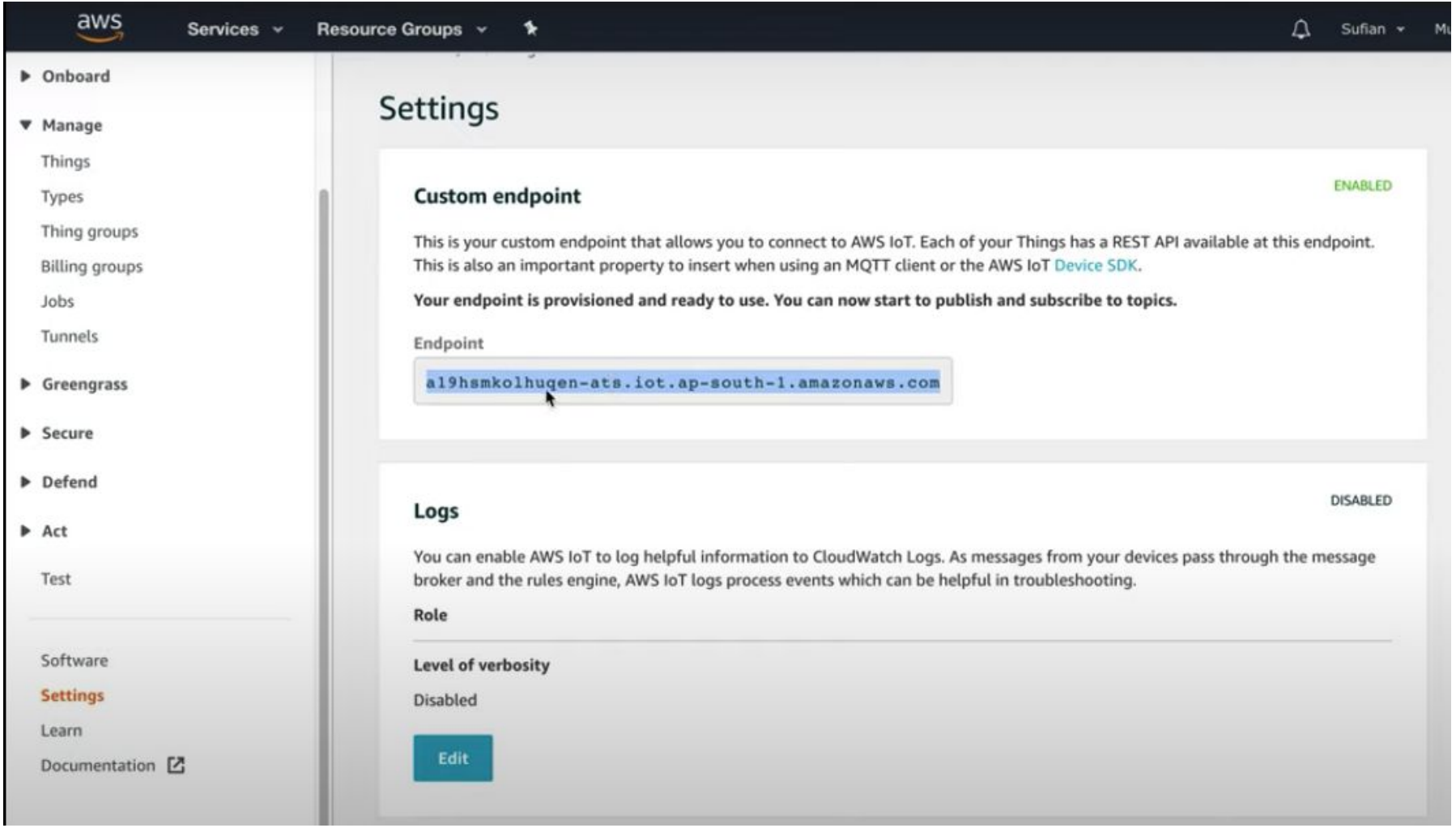
1c. Prepare the Credentials

AWS IoT Core

1c. Prepare the Credentials

- Device Certificate
- Device Private Key
- Amazon Root Certificate
- Thing name
- AWS IoT Endpoint Host
- AWS IoT Endpoint Port

AWS IoT Core


aws
Services
Resource Groups
Sufian
Onboard
Manage
Things
Types
Thing groups
Billing groups
Jobs
Tunnels
Greengrass
Secure
Defend
Act
Test
Software
Settings
Learn
Documentation
Settings
Custom endpoint
ENABLED
This is your custom endpoint that allows you to connect to AWS IoT. Each of your Things has a REST API available at this endpoint. This is also an important property to insert when using an MQTT client or the AWS IoT Device SDK.
Your endpoint is provisioned and ready to use. You can now start to publish and subscribe to topics.
Endpoint
a19hsmkolhugen-ats.iot.ap-south-1.amazonaws.com
Logs
DISABLED
You can enable AWS IoT to log helpful information to CloudWatch Logs. As messages from your devices pass through the message broker and the rules engine, AWS IoT logs process events which can be helpful in troubleshooting.
Role
Level of verbosity
Disabled
Edit

2. Device Authorization

AWS IoT Core

2a. Create a Policy

Policy is a document to authorize the actions allowed to a resource

```
{
        "Version": "Date"
        "Statement": "List of Authorizations"
}
```

2b. Policy Structure

A Statement in policy version "2012-10-17" for AWS IoT service contains the following fields

Effect

`Allow or Deny`

Action

`iot:Connect, iot:Publish, iot:Subscribe, iot:Receive`

Resource

`arn:aws:iot:region:AWS_Account_ID: Resource_type/Resource_Name`

2c. Example Policy to Connect

To authorize a device to only connect to AWS IoT Core, we can use the following policy

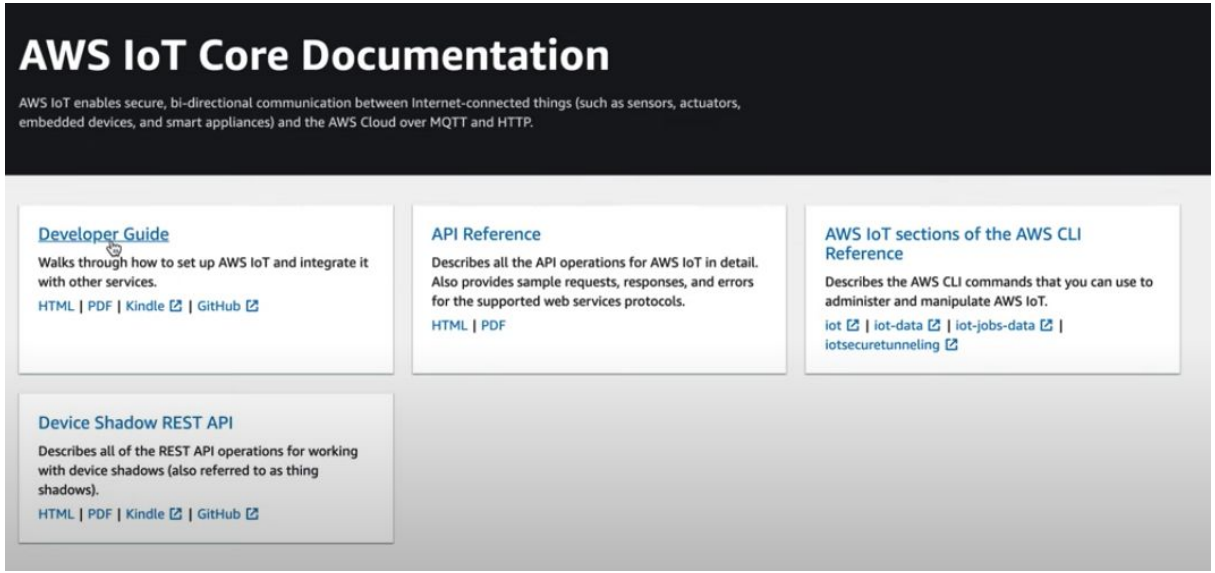
Effect

```
{
    “Version”: “2012-10-17”,
    “Statement”: [
    {
        “Effect”: “Allow”,
        “Action”: “iot:Connect”,
        “Resource”: “arn:aws:iot:region:AWS_AccID:Resource_type/Resource_Name”
    }
    ]
}
```

AWS IoT Core

3. Demo

AWS IoT Core



¿Qué voy a hacer ahora?

Documentación

https://docs.aws.amazon.com/iot/

https://docs.aws.amazon.com/iot/latest/developerguide/what-is-aws-iot.html

¡Gracias!
¿Preguntas?

Luis Rey Lara
Profesor de la Academia Redes y Telecomunicaciones
llara_ptc@upjr.edu.mx