

Challenge AWS IoT Integration

aws



Semester 4 IOT

Discover AWS IoT

April 16, 2023.

Student: Andre Sanao

Course: Smart Industry

Table of Contents

Introduction.....	4
Procedure	5
Conclusion	7
Reference	7

Acronyms

<i>Acronym</i>	<i>Meaning</i>
<i>IOT</i>	→ Internet of Things
<i>ESP32</i>	→ Expressif32
<i>TTN</i>	→ The Things Network
<i>AWS</i>	→ Amazon Web Services

Table 1 – List of acronyms used throughout the report.

Introduction

The assignment on which this document presents a small challenge of IOT subject. In this subject, we will learn how to use hardware to connect to the internet using internet protocols and such. The hardware that is going to be used to demonstrate these protocols is an ESP32. The ESP32 is a microcontroller that is used in embedded systems with an inbuilt wireless connectivity. In the following sections will provide the procedure and conclusion of the assignment.

Procedure

In this assignment, we are going to learn how to use one of the popular cloud service platform AWS. AWS also known as Amazon Web Service is a subsidiary of Amazon that offers a wide range of cloud computing service to individuals, companies and governments. Some core offerings include compute power, storage, databases, networks, analytics, machine learning and IoT solutions and more. AWS enables end users to access these services on-demand which allows for flexibility and scalability in terms of resource usage.

In this assignment, we are going to publish from Node-Red to AWS IoT. We will have to follow several steps in order to publish AWS IoT. We have to set the AWS server to Frankfurt first before following the steps. We first need to create a thing and create a certificate key for the thing. These certificates are necessary for secure connection between AWS and anything else AWS supports only secure connections. (See figure 1)

Thing details


Name iot-andre	Type ESP32
ARN  arn:aws:iot:eu-central-1:349940433542:thing/iot-andre	Billing group -


Figure 1 The Thing

After creating a thing, we are ready to create and attach a policy to the thing. A thing policy is a JSON document that defines permissions for an IoT device (also called a “Thing”) to interact with AWS IoT services The policy specifies which actions the device is allowed to perform and the resources it can access. For now, we are going to allow the device to access everything for the purpose of learning. (See figure 2)

PolicyAndre [Info](#)

Edit active versionDelete

Details

Policy ARN  arn:aws:iot:eu-central-1:349940433542:policy/PolicyAndre	Active version 1	Created April 11, 2023, 10:30:35 (UTC+02:00)	Last updated April 11, 2023, 10:30:35 (UTC+02:00)
---	---------------------	---	--

VersionsTargetsNoncomplianceTags

Active version: 1 [Info](#)

BuilderJSON

Policy effect	Policy action	Policy resource
Allow	*	*

Figure 2 Thing Policy

Now that we have a thing created and a policy attached to the thing, we can create an AWS MQTT interface in Node-Red. To be able to connect our Node-Red flow to AWS server, we need to obtain its address. The address can be obtained by creating a device shadow. Device shadows serves as a virtual representation of the state of an IoT device. It allows you to store and retrieve the current state and desired state of a device even when the device is offline. (See figure 3)

Shadow_Andre

Device Shadow details

ARN arn:aws:iot:eu-central-1:349940433542:thing/iot-andre/Shadow_Andre	Last updated April 11, 2023, 10:44:34 (UTC+02:00)
MQTT topic prefix \$aws/things/iot-andre/shadow/name/Shadow_Andre	Version 1
Device Shadow URL https://a31q7h1d2sxx4v-ats.iot.eu-central-1.amazonaws.com/things/iot-andre/shadow?name=Shadow_Andre	Prefix for Fleet indexing query shadow.name.Shadow_Andre.
	Fleet indexing status Not indexed

Figure 3 Device Shadow

In Node-Red, we have to use MQTT out and fill in the details and upload the obtained certificate keys in the node. The MQTT node uses the device shadow URL to connect to AWS server. (See figure 4)

Properties

Name AWS

Connection **Security** **Messages**

Server a31q7h1d2sxx4v-ats.iot.eu-central-1.amazona **Port** 8883

☒ Connect automatically

☒ Use TLS AWSCertificate

Protocol MQTT V3.1 (legacy)

Client ID iot-andre

Keep Alive 60

Session ☒ Use clean session

Figure 4 Node-Red MQTT node config

After configuring the node, the flow can be deployed to establish connection to AWS server.

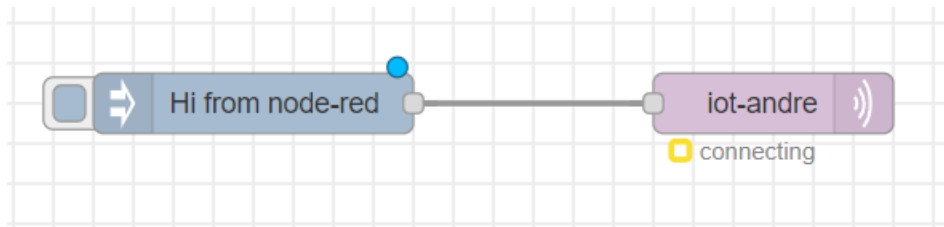


Figure 5 Node-Red Flow

As you can see on the figure above, an inject node is used to send message payload to AWS server. Due to some error somewhere in the steps, I could not manage to establish a connection with the MQTT node. Even with Docker and local Node-Red, I could not manage to find the problem and therefore could not show the message sent.

Conclusion

To conclude this assignment, we gained knowledge about one of the leading providers of cloud-based infrastructure and service. AWS provide a wide range of resources for smart industries and applications, as it is robust, scalable and secure platform for connecting and managing IoT devices compared to Azure, both offer a suite of service for IoT and share a lot of similarities in terms of features, tools and ecosystem. With everything considered, configuring AWS to Node-Red was a great experience and I hope we get to use AWS for future projects.

Reference

- Pal, S., Díaz Vicente García, & Le, D.-N. (2022). *IOT: Security and privacy paradigm*. Amazon. Retrieved April 17, 2023, from <https://eu-central-1.console.aws.amazon.com/iot/home?region=eu-central-1#/home>
- Pal, S., Díaz Vicente García, & Le, D.-N. (2022). *IOT: Security and privacy paradigm*. Amazon. Retrieved April 17, 2023, from <https://eu-central-1.console.aws.amazon.com/iot/home?region=eu-central-1#/tutorial?step=1>
- Pal, S., Díaz Vicente García, & Le, D.-N. (2022). *IOT: Security and privacy paradigm*. Amazon. Retrieved April 17, 2023, from <https://eu-central-1.console.aws.amazon.com/iot/home?region=eu-central-1#/learnHub>
- Red. Node. (n.d.). Retrieved April 16, 2023, from <https://nodered.org/>