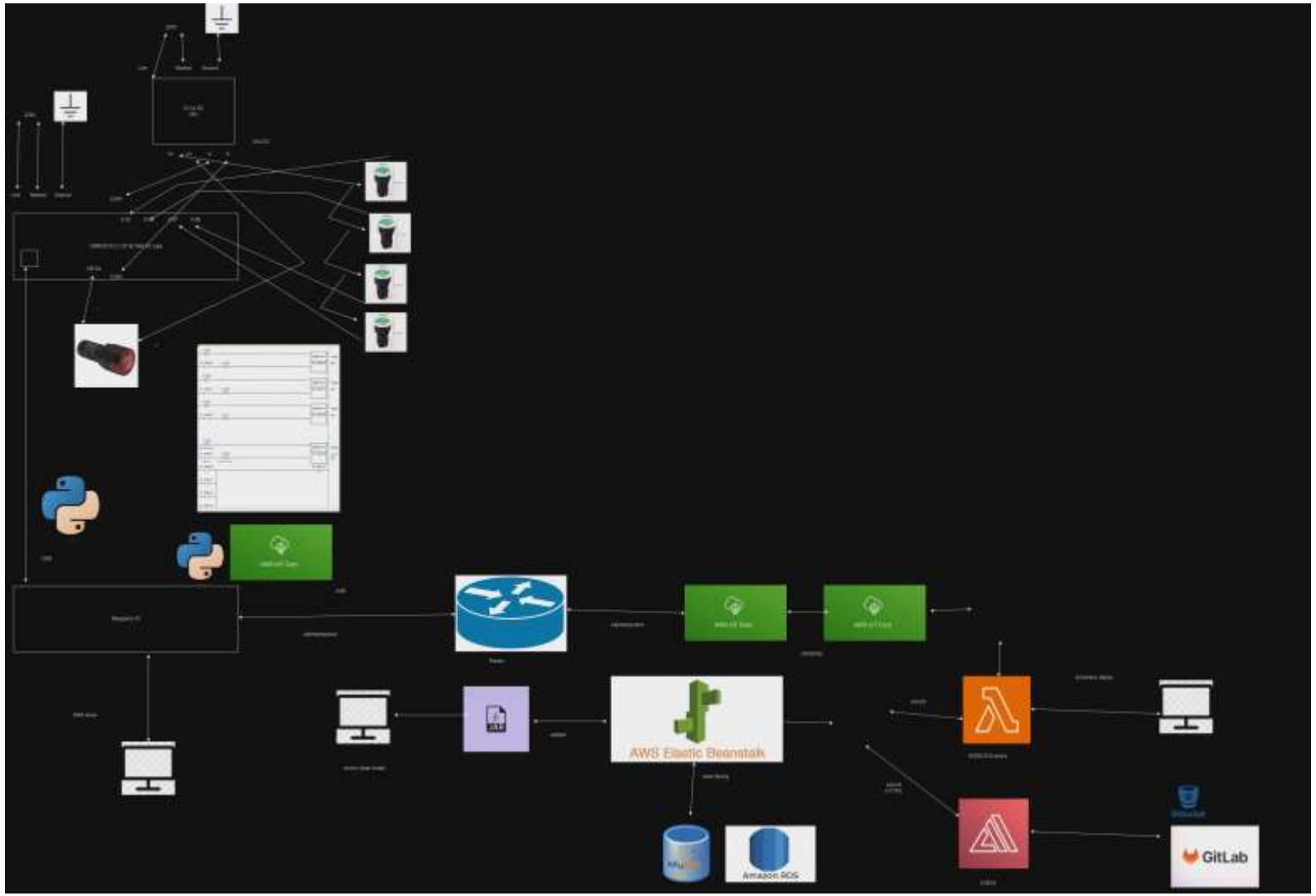
Overview SDLC

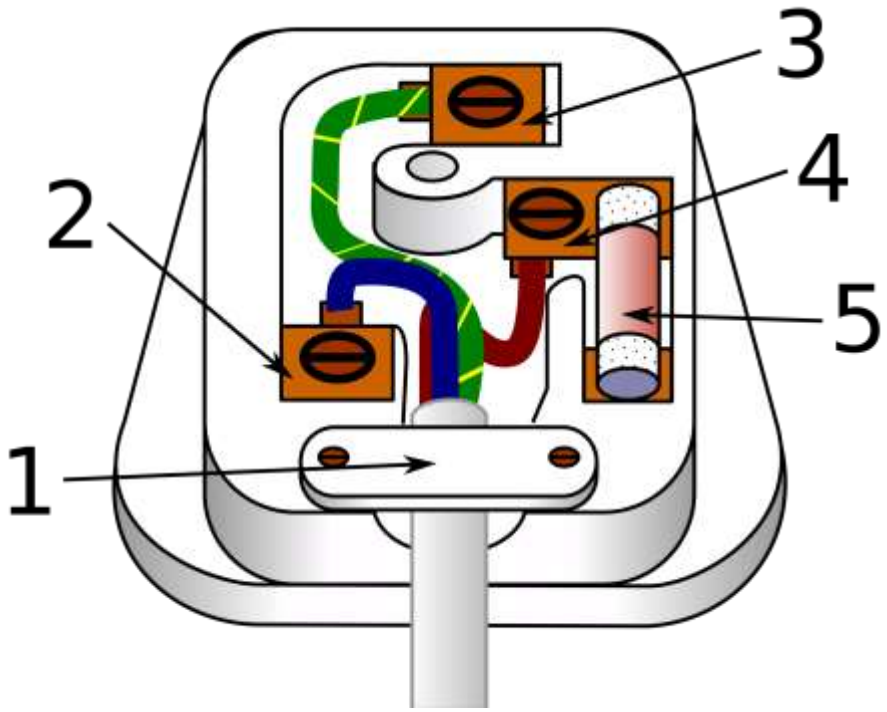
written by junxian428
25/7/2023
Include

- Requirement
- Planning
- Implementation
- Testing
- Deploy

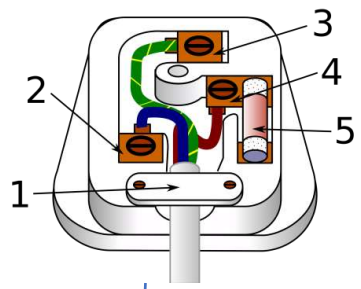
FE/BE architecture and
Event-driven architecture
Serverless
*Microservices
* Java Spring, Python, Qt, VueJS, MySQL
* PLC, Raspberry Pi
*AWS



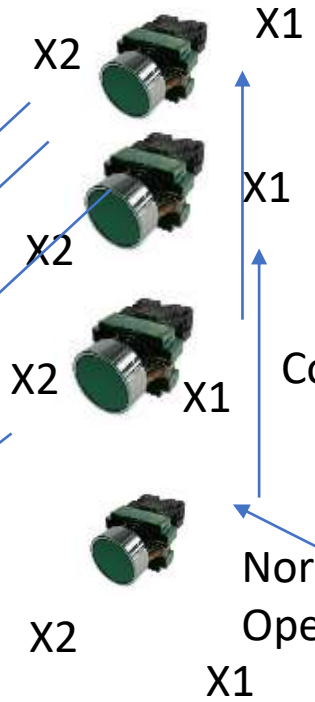
Wiring



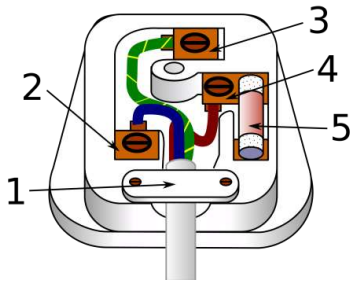
Product Size:



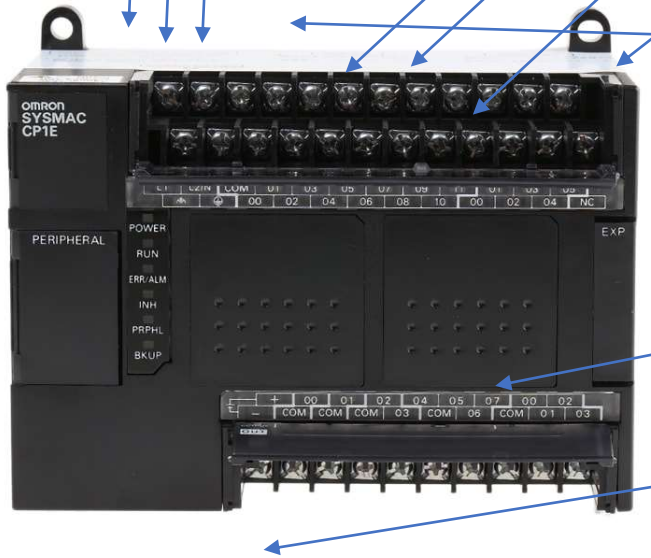
Input	Type
0.03	Button1
0.05	Button2
0.07	Button3
0.09	Button4



+
V



Product Size:



-V

LED 24DC V with
Buzzer

-V

X2

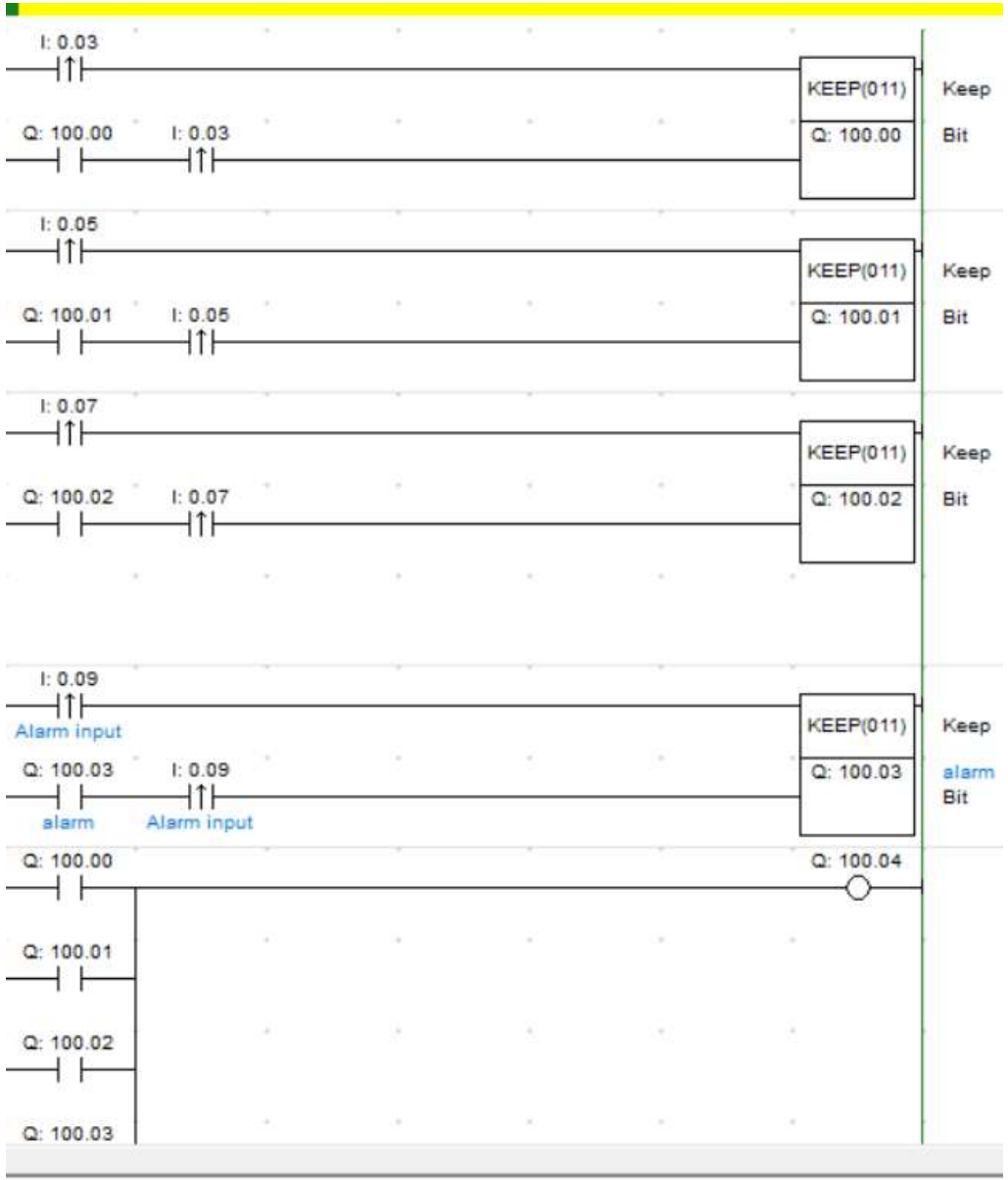
X1

+V

-V

Output	Type
100.04	Led with buzzer

PLC & Ladder Diagram

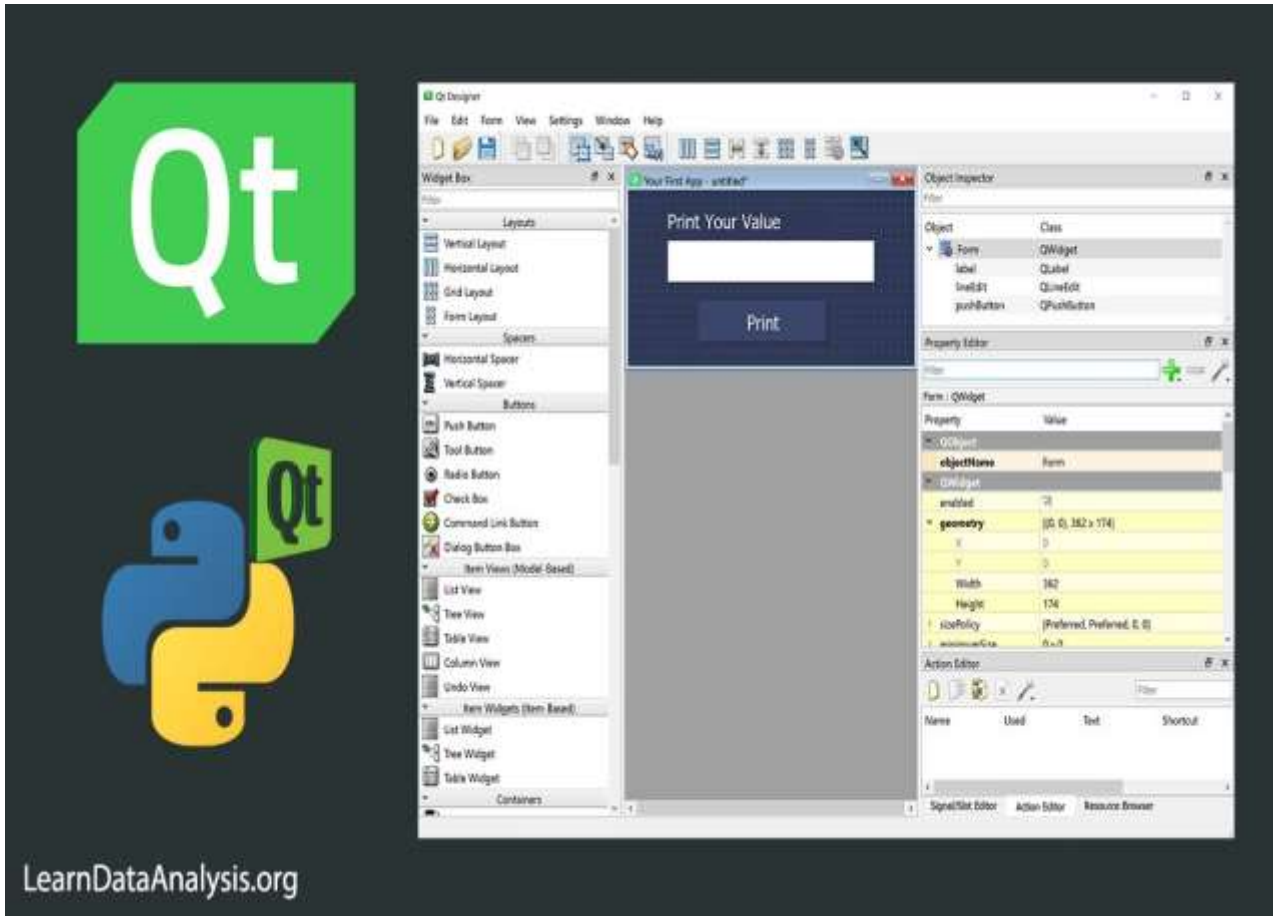


Raspberry Pi

Burn OS into SD card

Raspberry Pi OS

Qt Frontend Python



Python C-command to PLC

- Sudo apt-get install code (visual studio code)
- Sample
- https://github.com/junxian428/DesktopApp_PLC_Raspberry

End of hardware

Begin of software

Raspberry Pi to AWS IoT Thing

- Install AWS IoT SDK, zip file and unzip then run ./start.sh
- Modify code in pubsub.py

To Change Code (in aws-iot-device-sdk-python-v2/samples/pubsub.py)

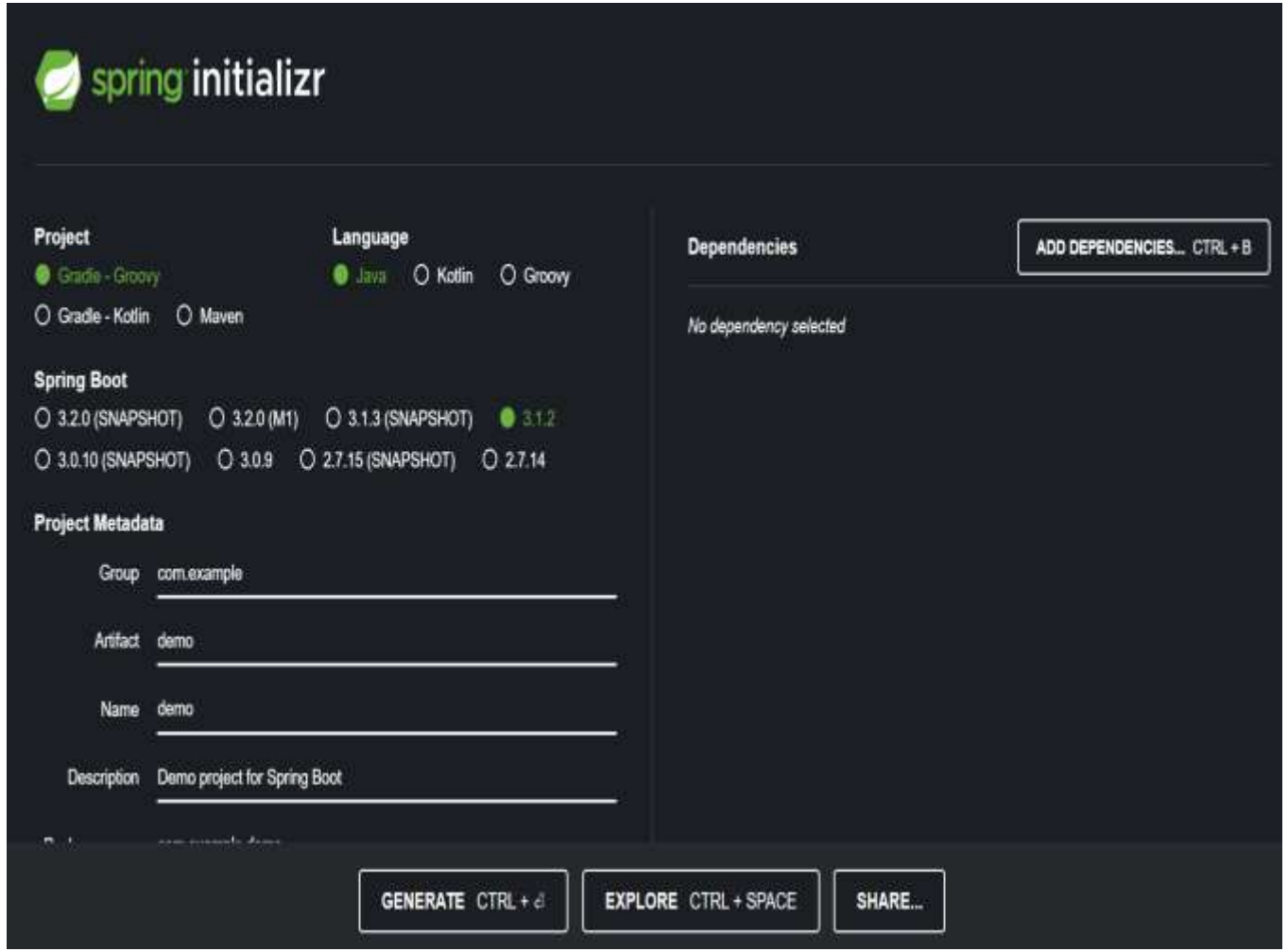
Change the

Message = “your message”

- In order to run,

./start.sh

Backend First Methodology



https://start.spring.io/

JWT

https://github.com/junxian428/Java_Spring_JWT

CRUD

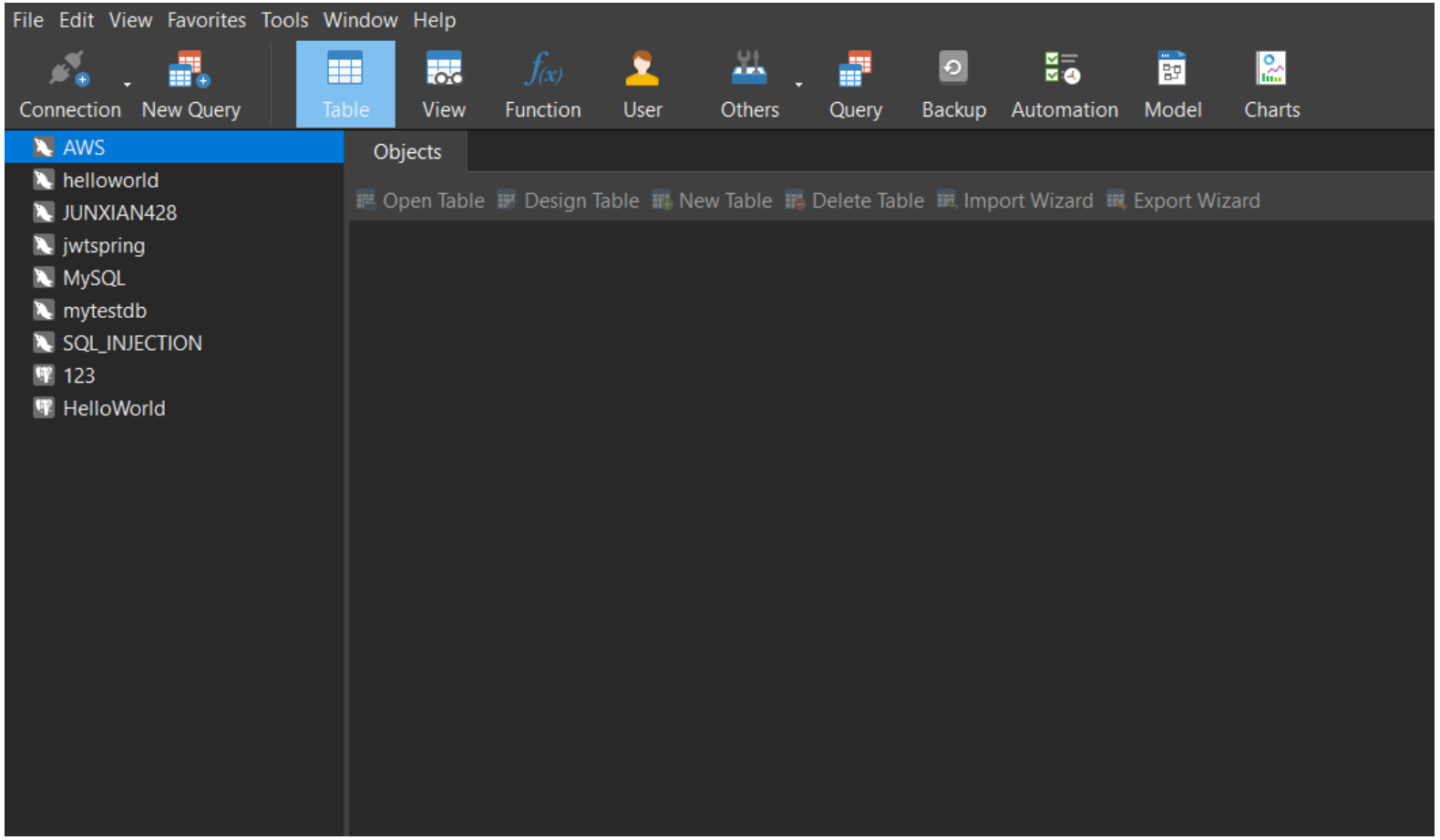
https://github.com/junxian428/Spring_RESTAPI_CRUD_STOCK

Database RDS (MySQL)

```
spring:
  datasource:
    url: jdbc:mysql://localhost:3306/jwt
    username: root
    password:
    driver-class-name: com.mysql.cj.jdbc.Driver
  jpa:
    hibernate:
```

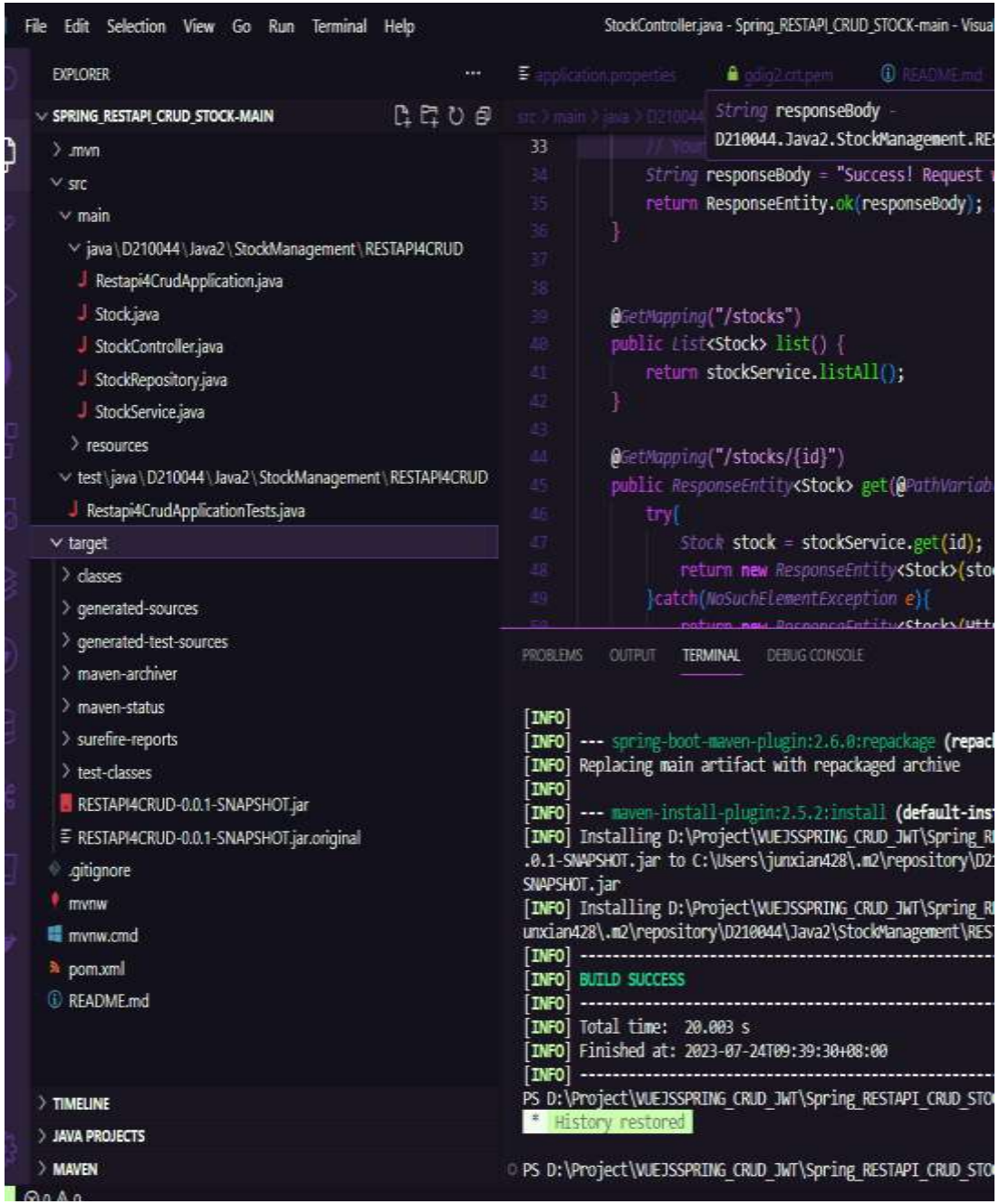
* Change URL to AWS

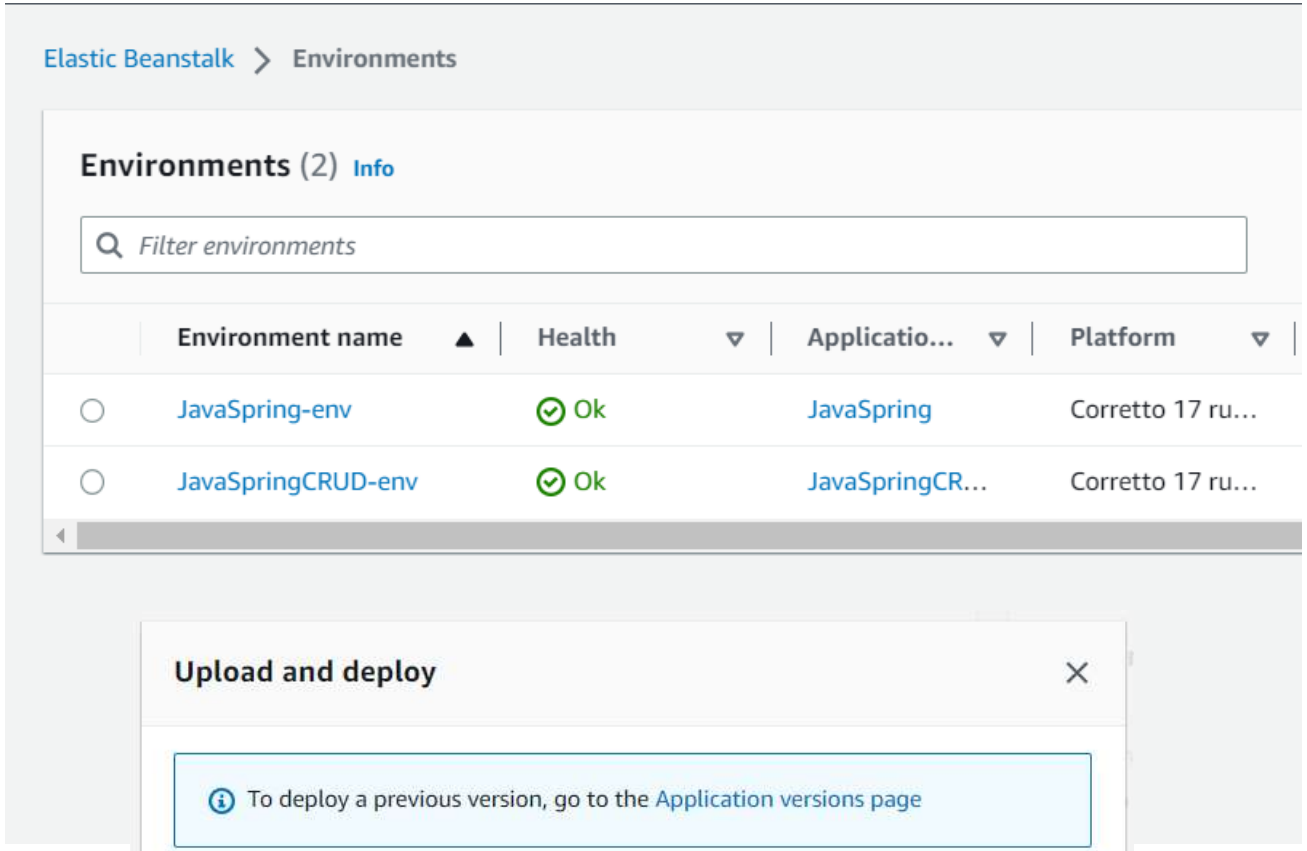
Navicat connects database

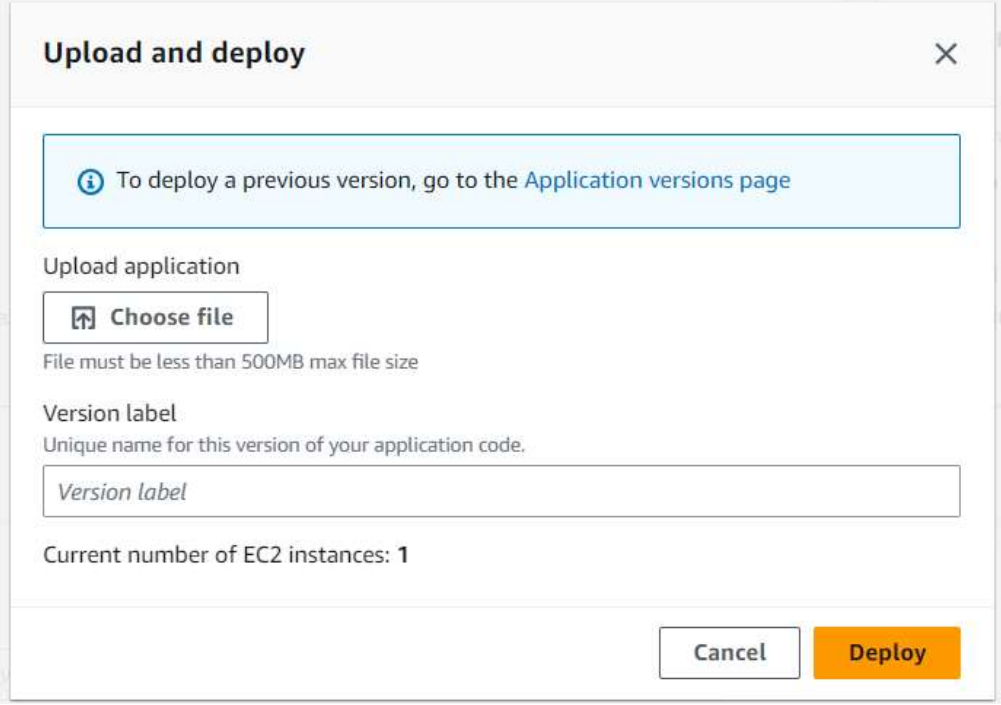


AWS Elastic BeanStalk

- ./mvnw clean install
- Create jar file then upload

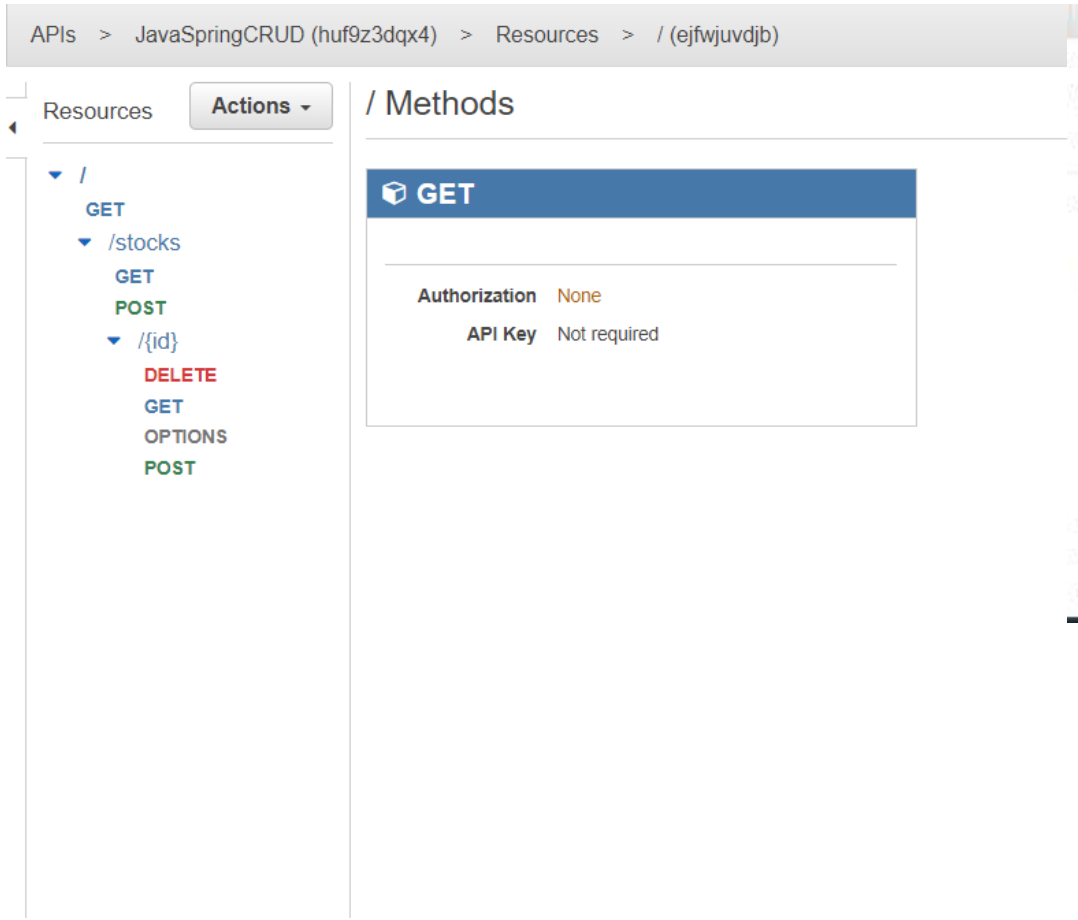


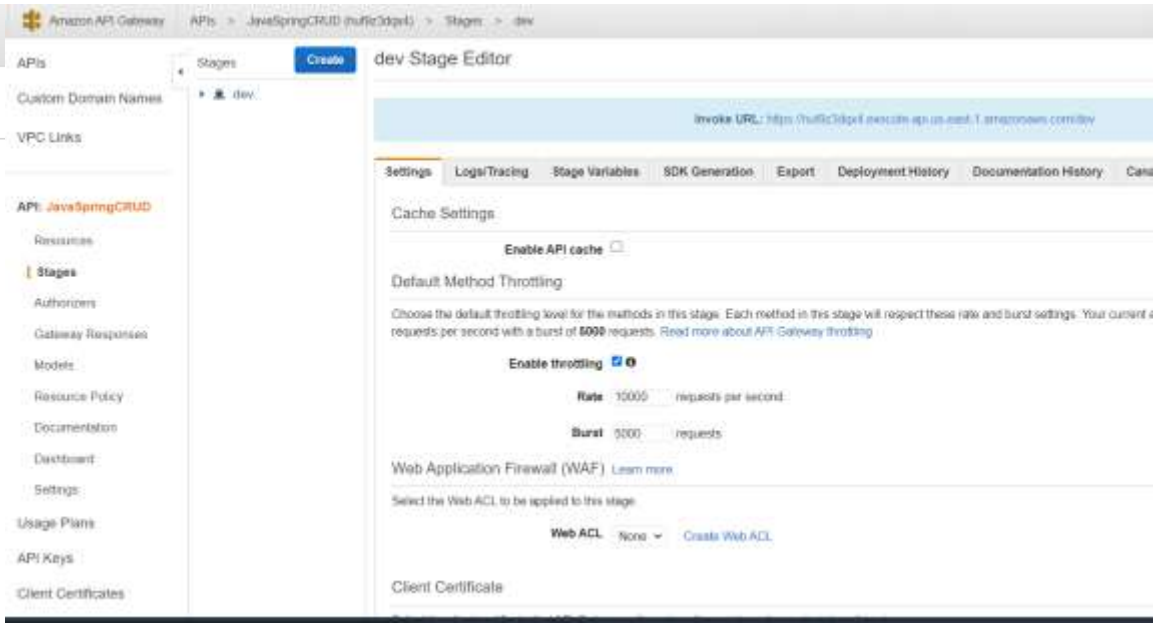




Choose jar file found in target folder

Set API gateway for Elastic Bean Stalk in order to get HTTPS





Until now

- You have already deployed backend + database + Hardware
- Now deploy middleware and frontend as well as the IoT Rules and destination confirmation

AWS IoT Core set rules and destination

AWS IoT > Message routing > Rules

Rules (2) Info

Rules allow your things to interact with other services. Rules are analyzed and perform specific actions based on messages published by your devices.

Activate | Deactivate | Edit | Delete | **Create rule**

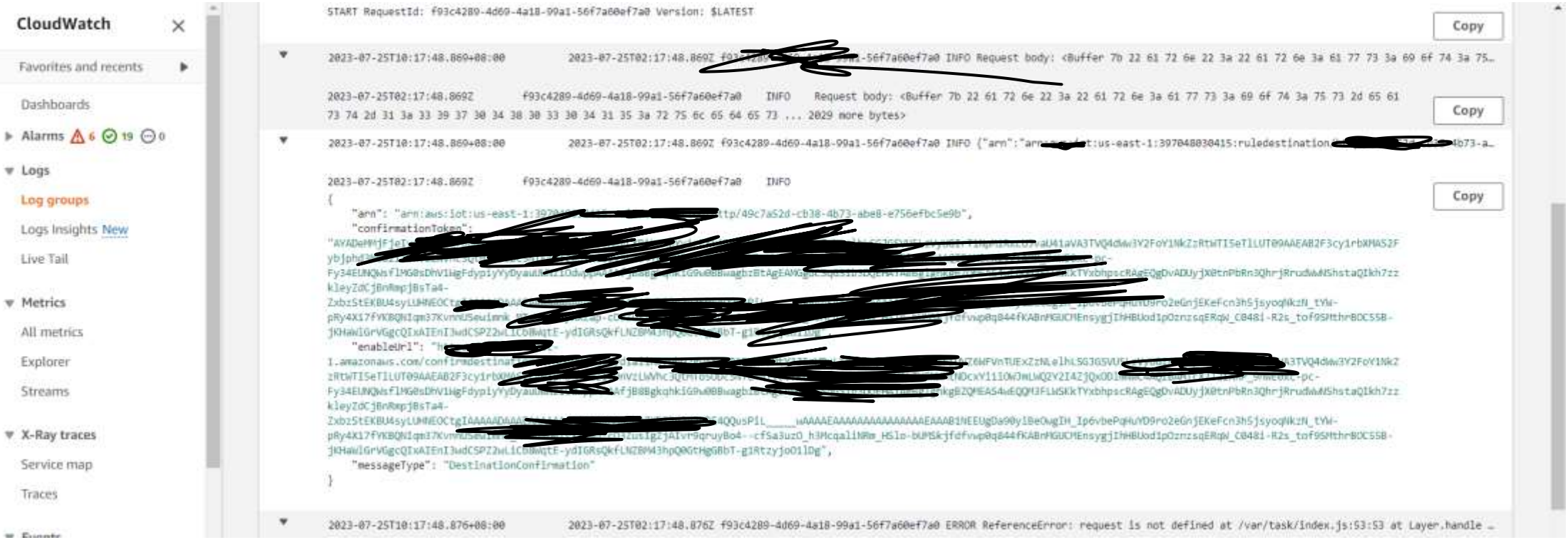
Find rules

1

Name	Status	Rule topic	Created date
CRUD	Active	sdk/test/python	July 24, 2023, 14:56:14 (UTC+08:00)
HTTPSCRUD	Active	sdk/test/js	July 25, 2023, 10:01:40 (UTC+08:00)

If your destination is not confirmed

- You are required to check the log to get the confirmation token
- By using cloudwatch



- IoT AWS Core from Raspberry Pi will send to sdk/test/python
- Create rules to redirect iot topic into sdk/test/js
- Then create rules for sdk/test/js send HTTPS request to Lambda API gateway

AWS Lambda Serverless

- npm install -g serverless@3.31.0

```
PS D:\Project\Serveless> serverless

Creating a new serverless project

? What do you want to make? (Use arrow keys)
> AWS - Node.js - Starter
  AWS - Node.js - HTTP API
  AWS - Node.js - Scheduled Task
  AWS - Node.js - SQS Worker
  AWS - Node.js - Express API
  AWS - Node.js - Express API with DynamoDB
  AWS - Python - Starter
  AWS - Python - HTTP API
  AWS - Python - Scheduled Task
  AWS - Python - SQS Worker
  AWS - Python - Flask API
  AWS - Python - Flask API with DynamoDB
  Other
```

```
npm i axios
```

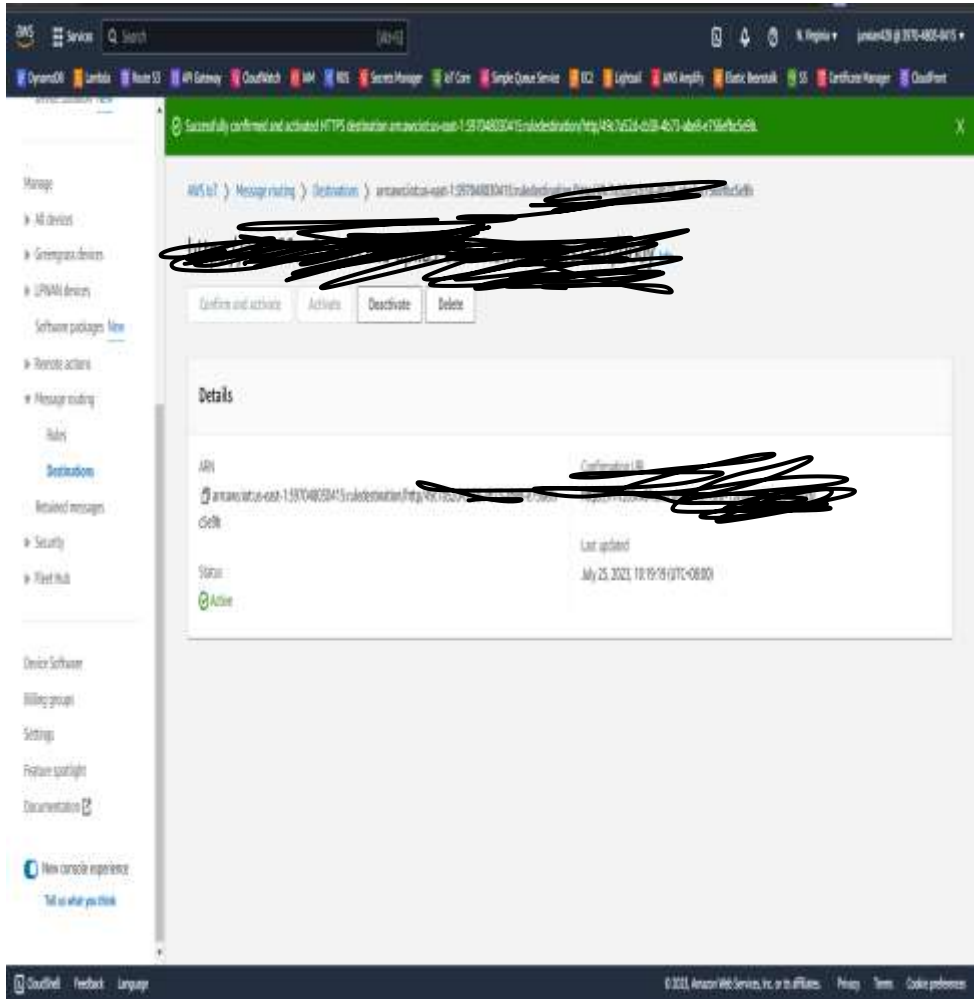
```
onst axios = require("axios");
```

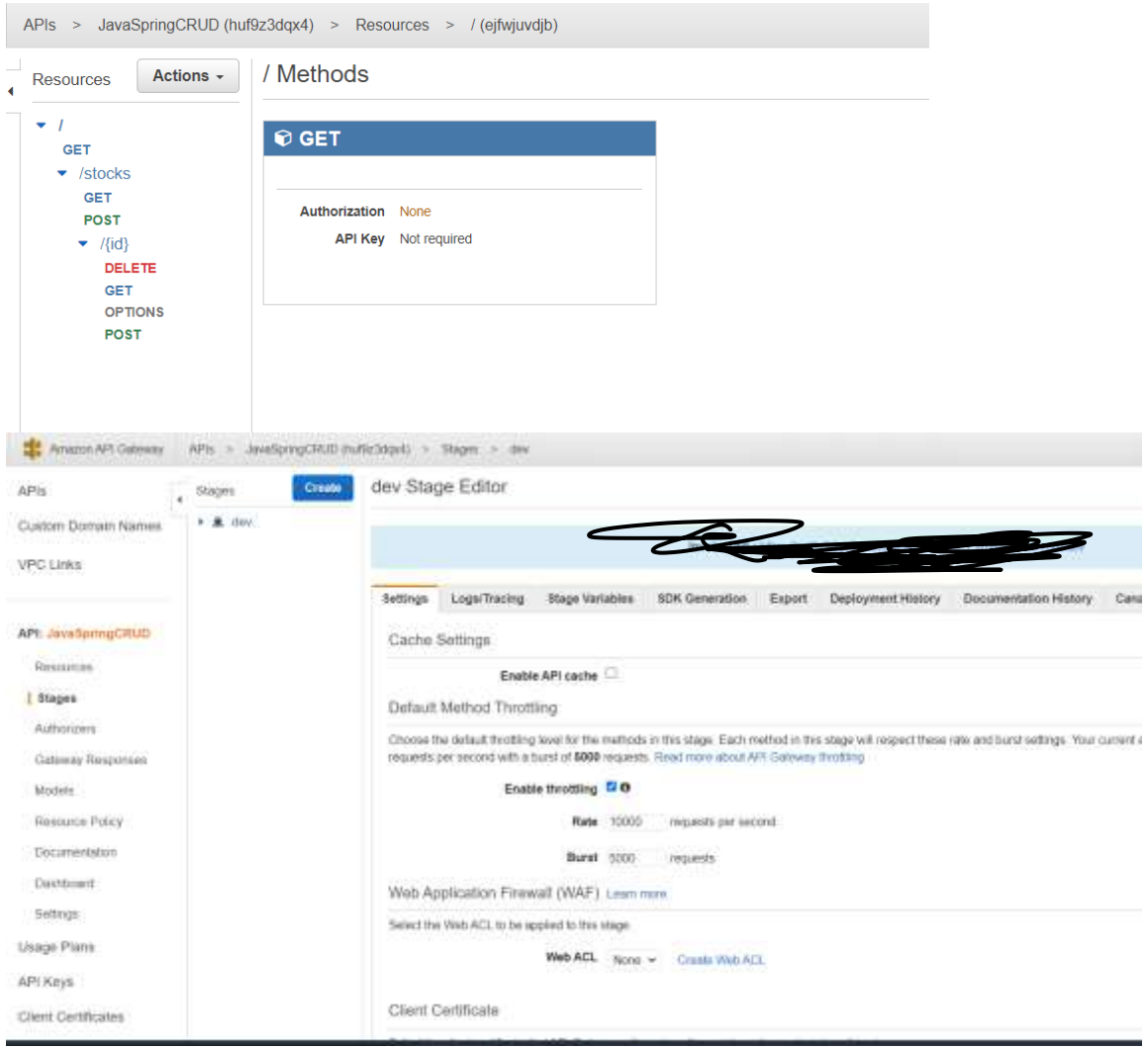
```
app.get("/", (req, res, next) => {

  // Replace the following URL with the API you want to fetch data from
const apiUrl = '';

// Making a GET request using Axios
axios.get(apiUrl)
  .then(response => {
    // The data from the API will be available in the 'response.data' property
    const responseData = response.data;
    console.log('Response data:', responseData);
    return res.status(200).json({
      message: responseData,
    });

  })
  .catch(error => {
    console.error('Error fetching data:', error);
    return res.status(400).json({
      message: error,
    });
  });
});
```

This is help your application deployed into AWS Lambda and API gateway -serverless deploy

AWS Lambda Serverless

Your AWS Lambda Serverless endpoint should
Set the AWS API gateway for Java Spring Elastic BeanStalk





Frontend Last Methodology

- Create vuejs
- Vue create frontendproject

Choose router & vuex

Frontend Deploy (AWS Amplify)