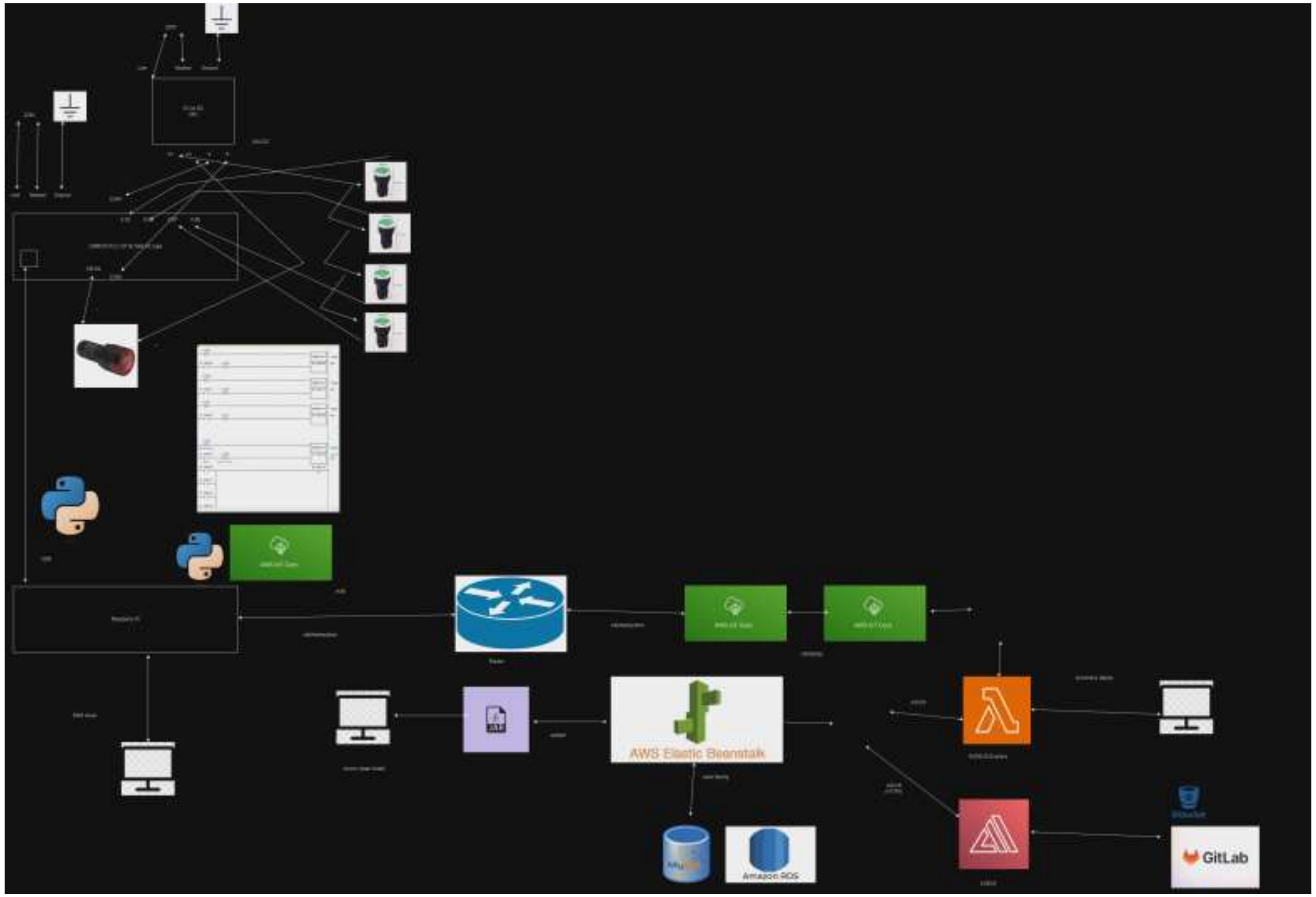
Overview SDLC

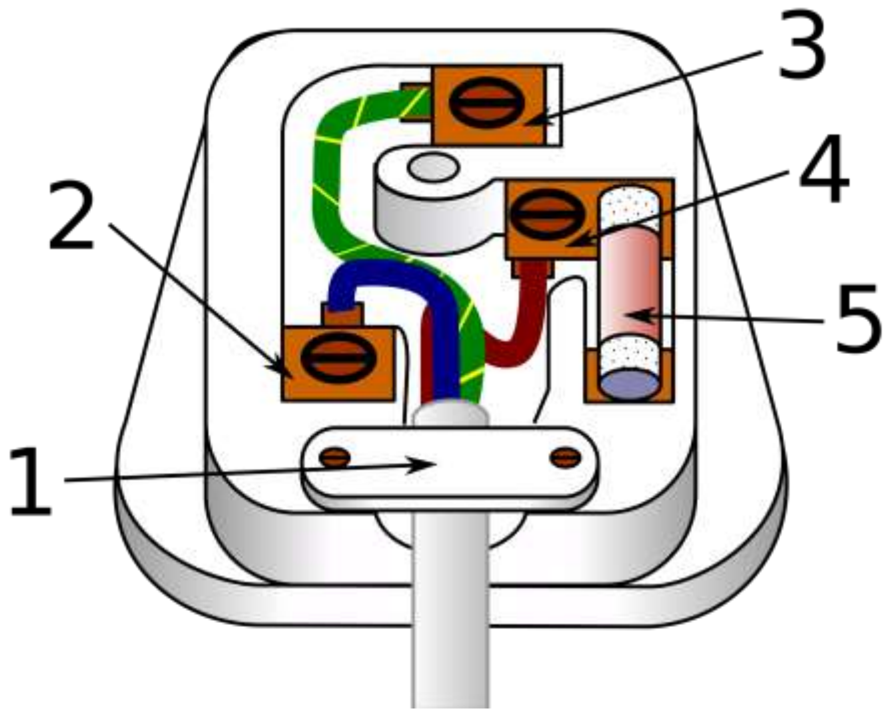
written by junxian428
25/7/2023
Include

- Requirement
- Planning
- Implementation
- Testing
- Deploy

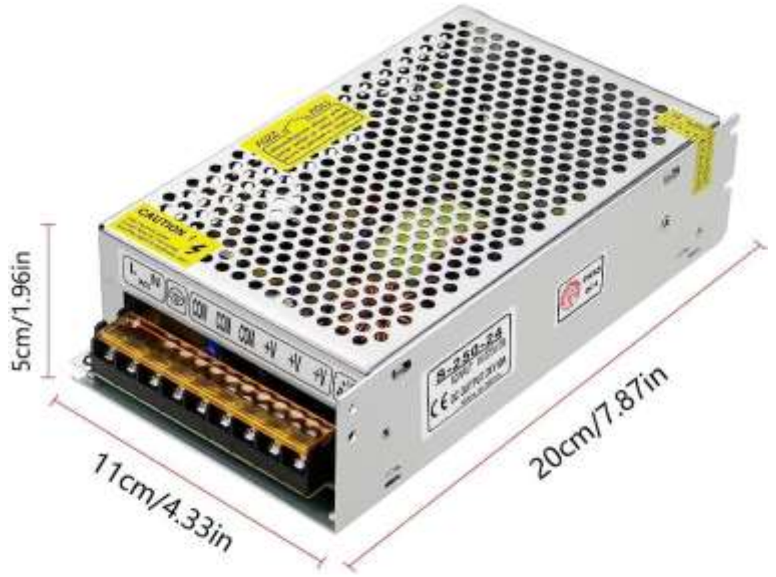
FE/BE architecture and
Event-driven architecture
Serverless
*Microservices
* Java Spring, Python, Qt, VueJS, MySQL
* PLC, Raspberry Pi
*AWS

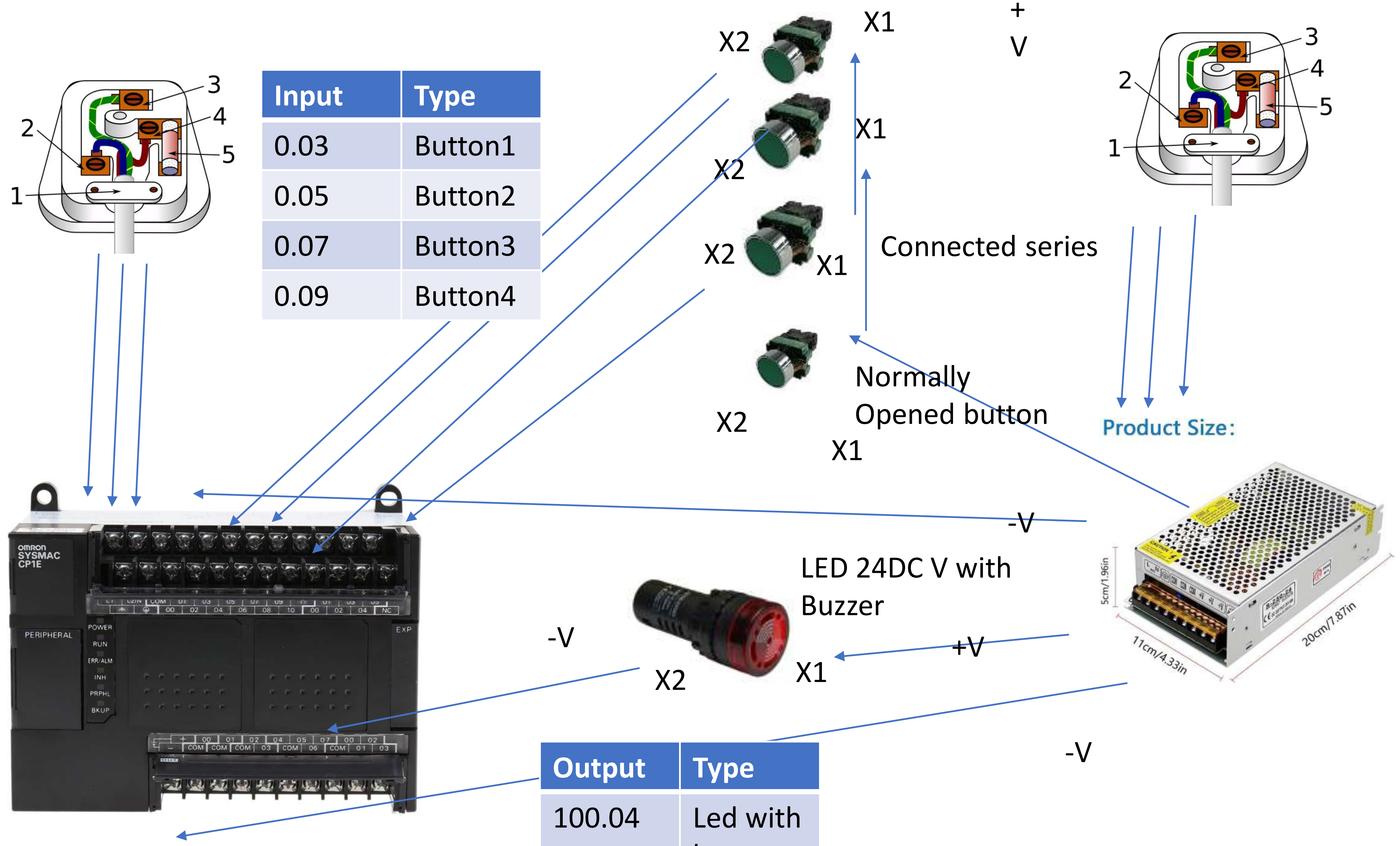


Wiring



Product Size:

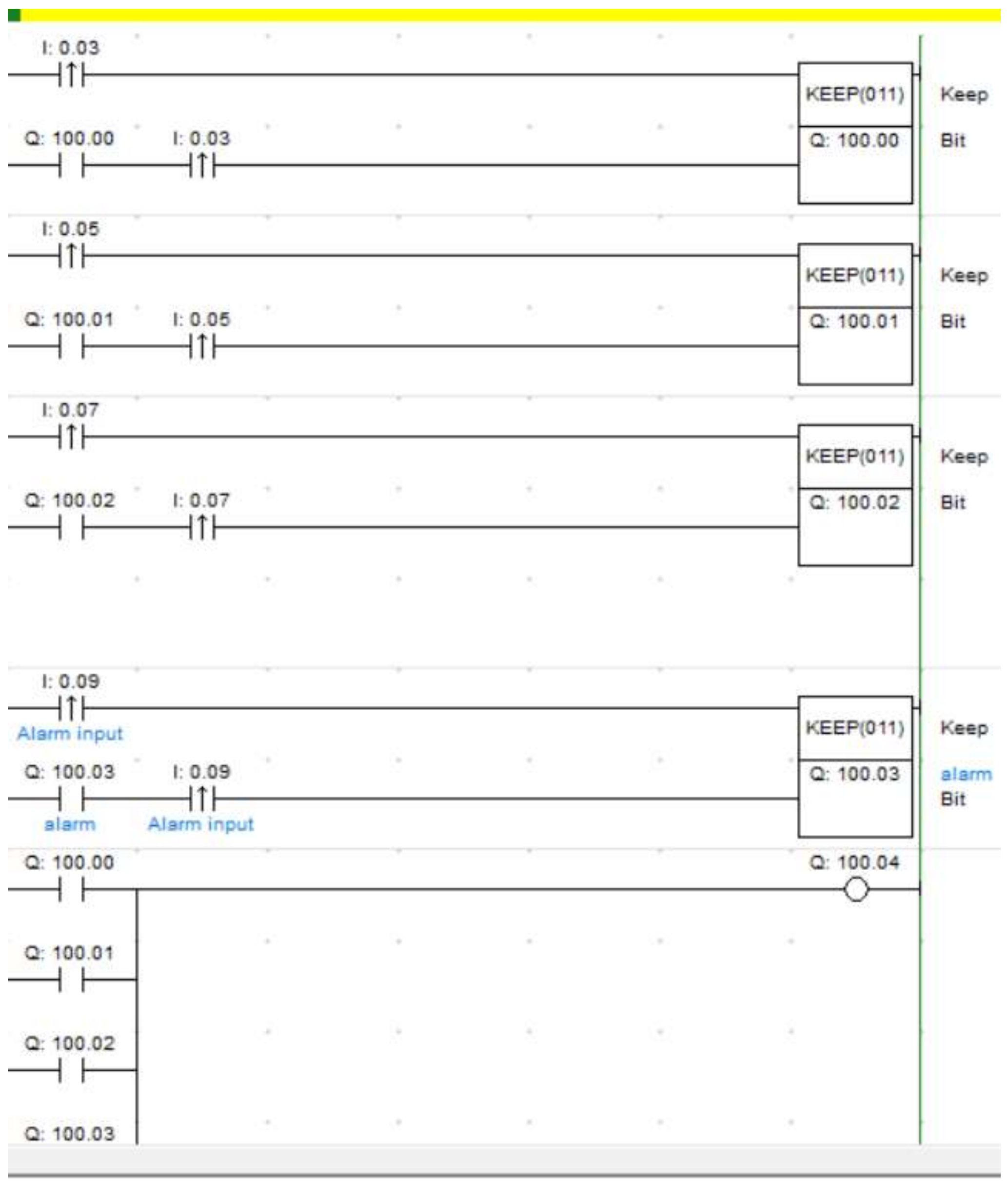




Input	Type
0.03	Button1
0.05	Button2
0.07	Button3
0.09	Button4

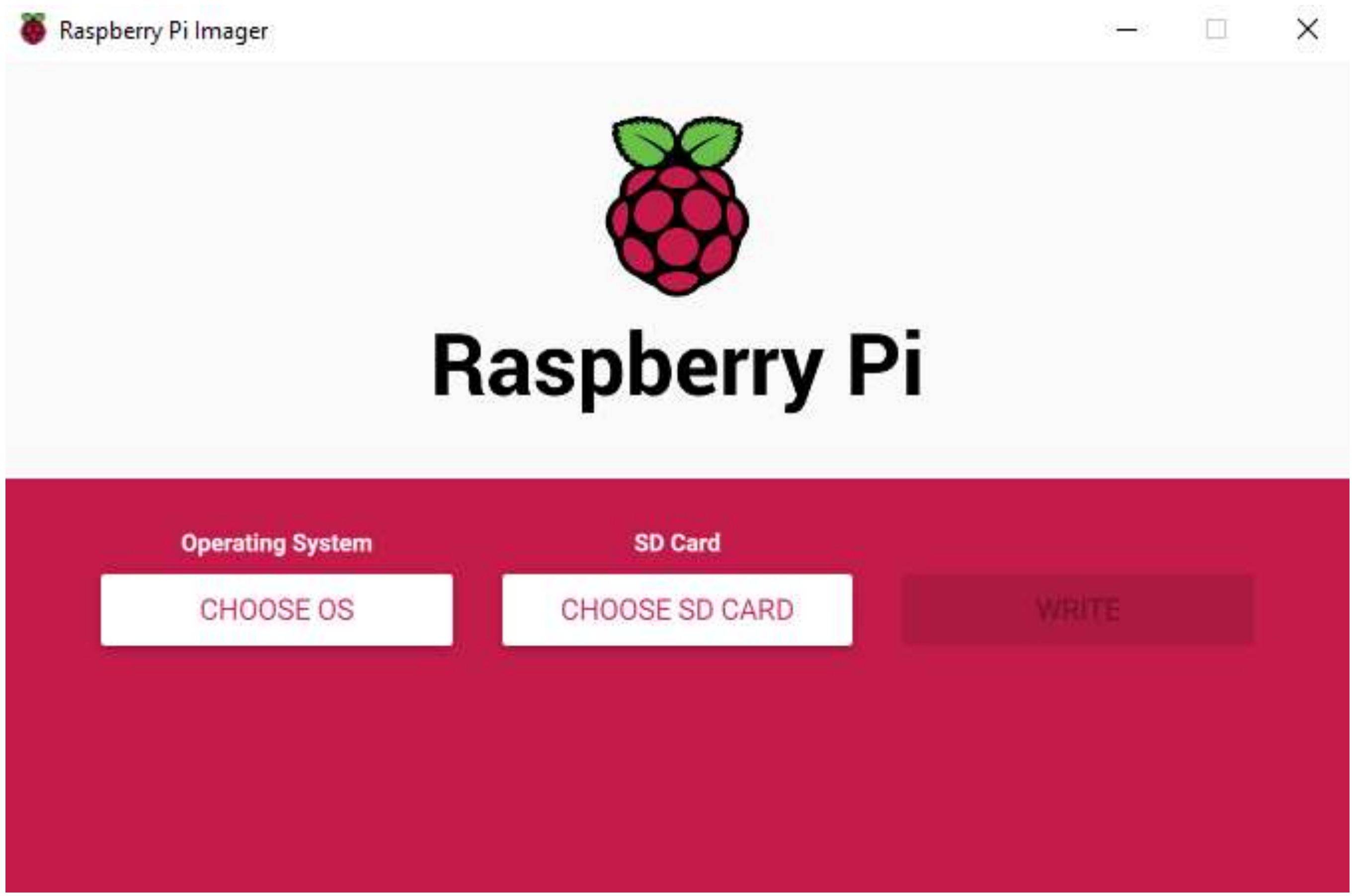
Output	Type
100.04	Led with buzzer

PLC & Ladder Diagram

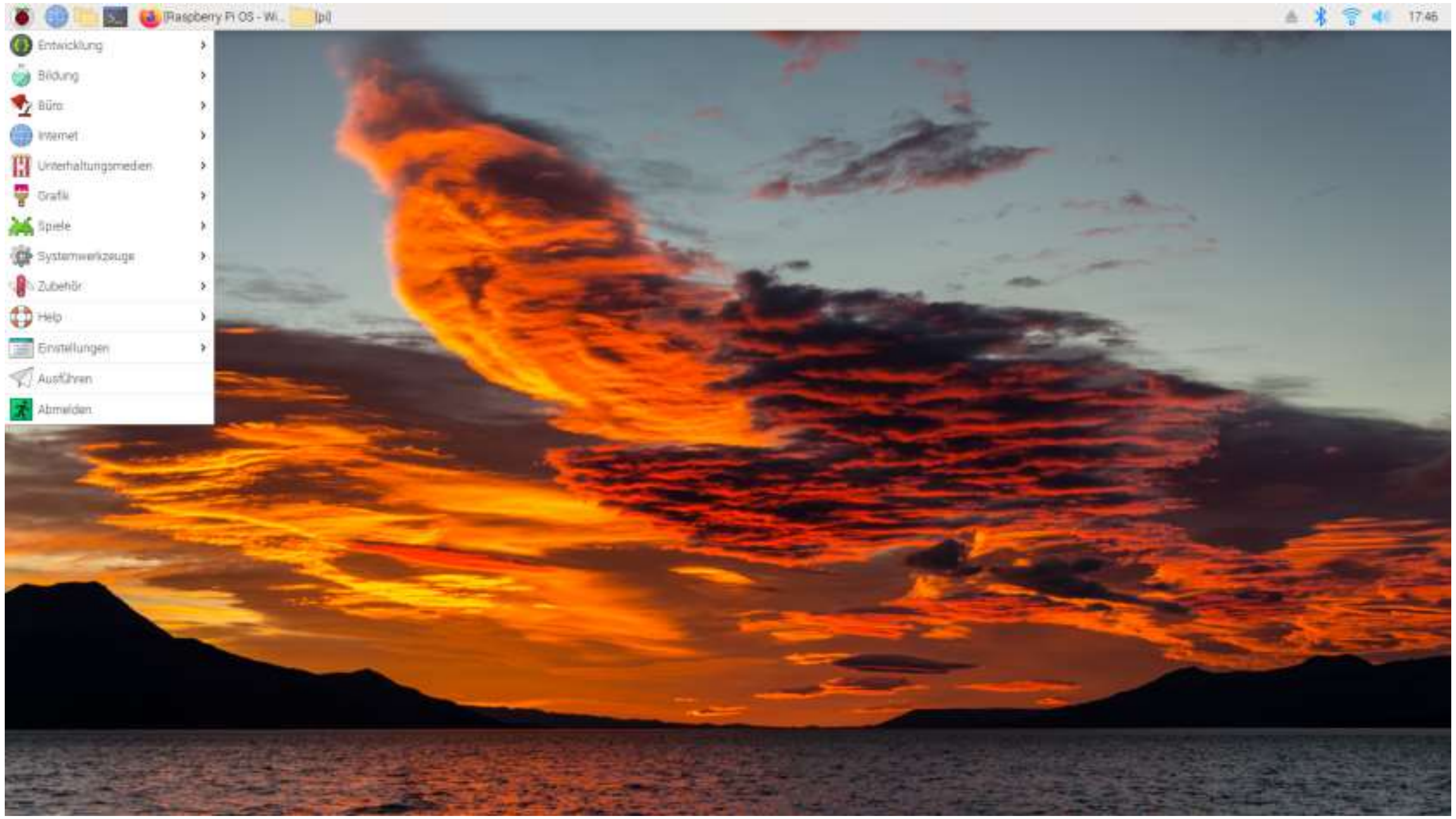


Raspberry Pi

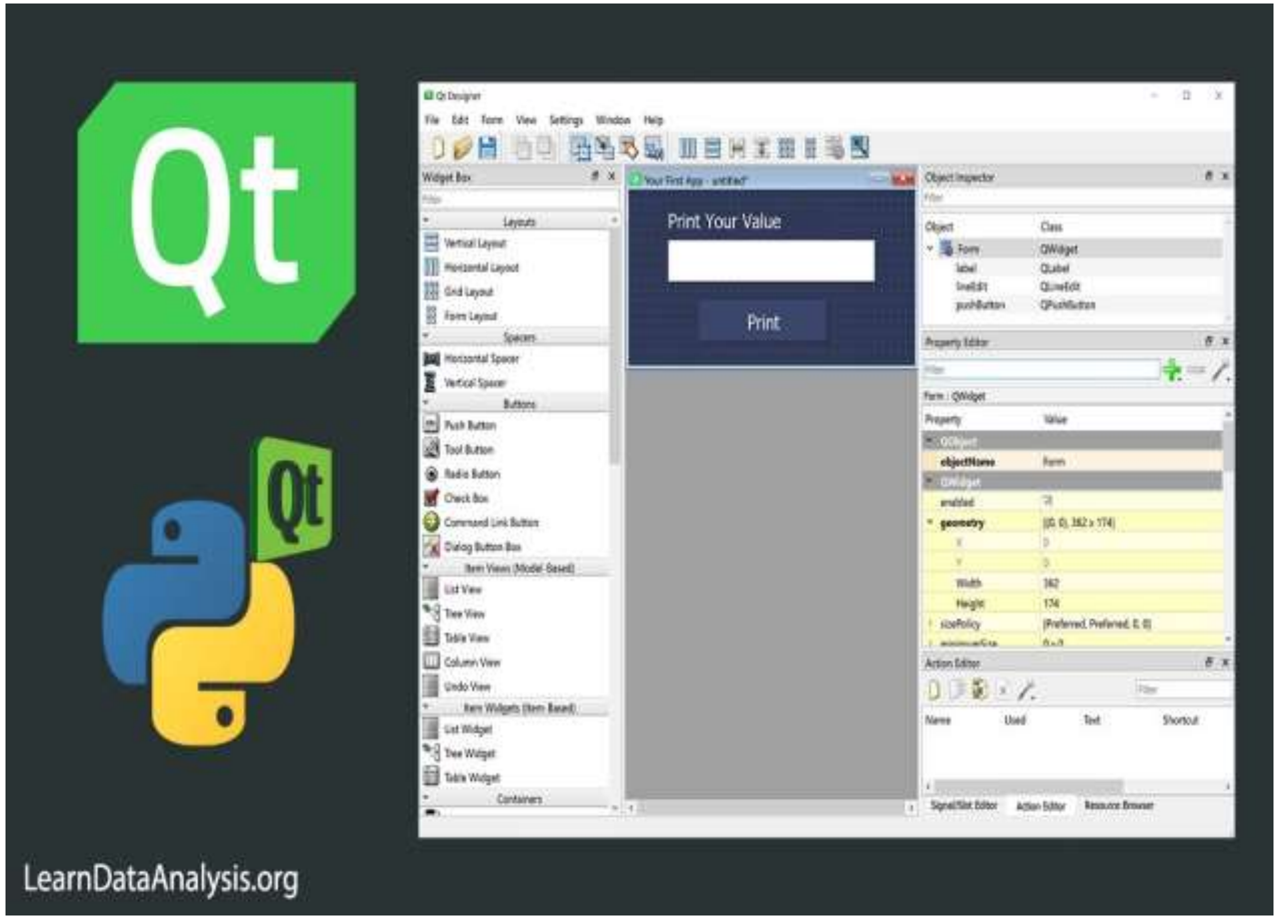
Burn OS into SD card



Raspberry Pi OS



Qt Frontend Python



Object	Class
Form	QWidget
label	QLabel
lineEdit	QLineEdit
pushButton	QPushButton

Property	Value
QObject	
objectName	Form
QWidget	
enabled	✓
geometry	[(0, 0), 362 x 174]
X	0
Y	0
Width	362
Height	174
sizePolicy	[Preferred, Preferred, 0, 0]

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

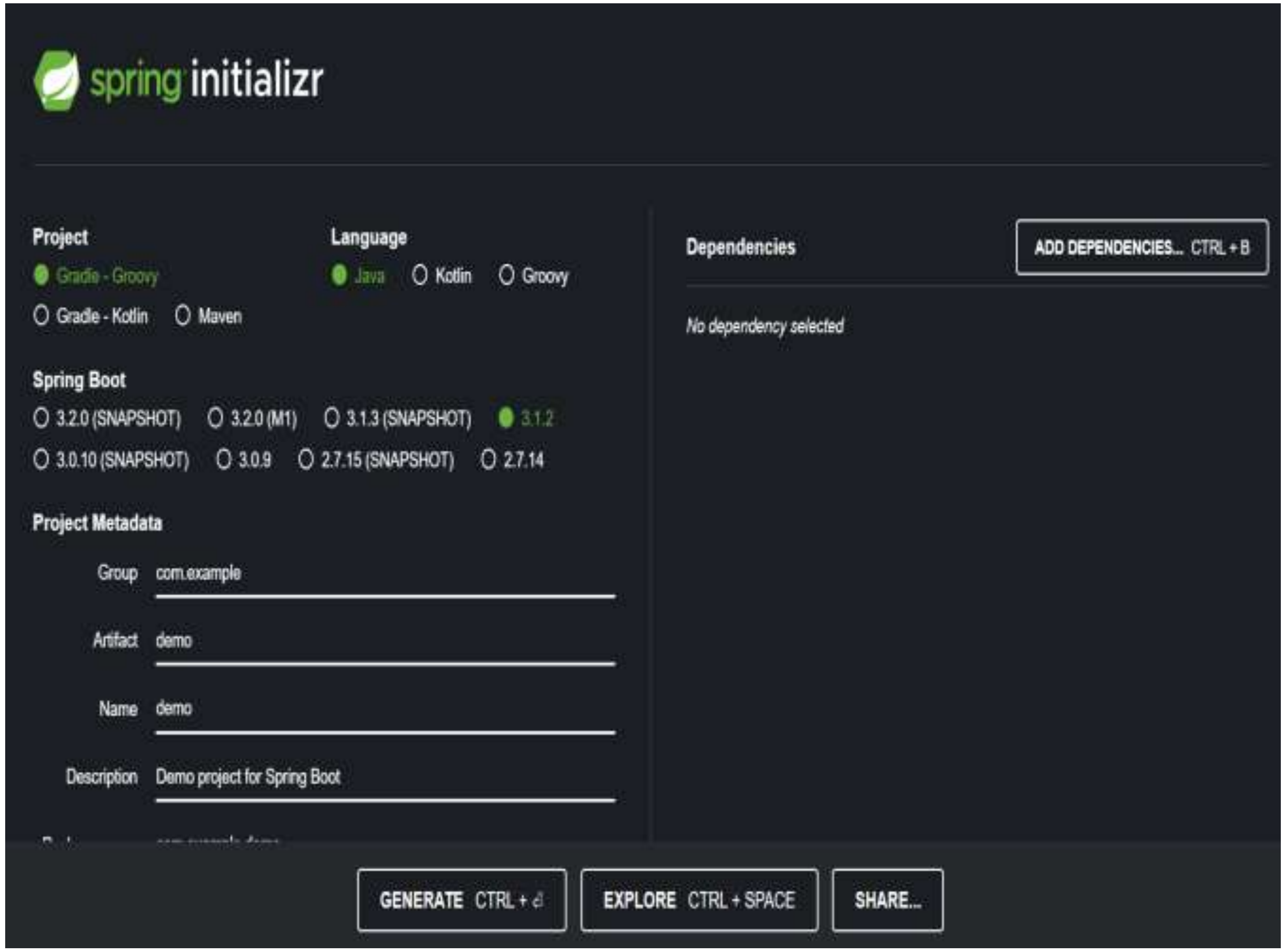
Message = “your message”

- In order to run,

./start.sh

AWS IoT Core

Backend First Methodology



spring initializr

Project

● Gradle - Groovy ○ Gradle - Kotlin ○ Maven

Language

● Java ○ Kotlin ○ Groovy

Spring Boot

○ 3.2.0 (SNAPSHOT) ○ 3.2.0 (M1) ○ 3.1.3 (SNAPSHOT) ● 3.1.2
○ 3.0.10 (SNAPSHOT) ○ 3.0.9 ○ 2.7.15 (SNAPSHOT) ○ 2.7.14

Project Metadata

Group	com.example
Artifact	demo
Name	demo
Description	Demo project for Spring Boot

Dependencies

No dependency selected

https://start.spring.io/

JWT

https://github.com/junxian428/Java_Spring_JWT

CRUD

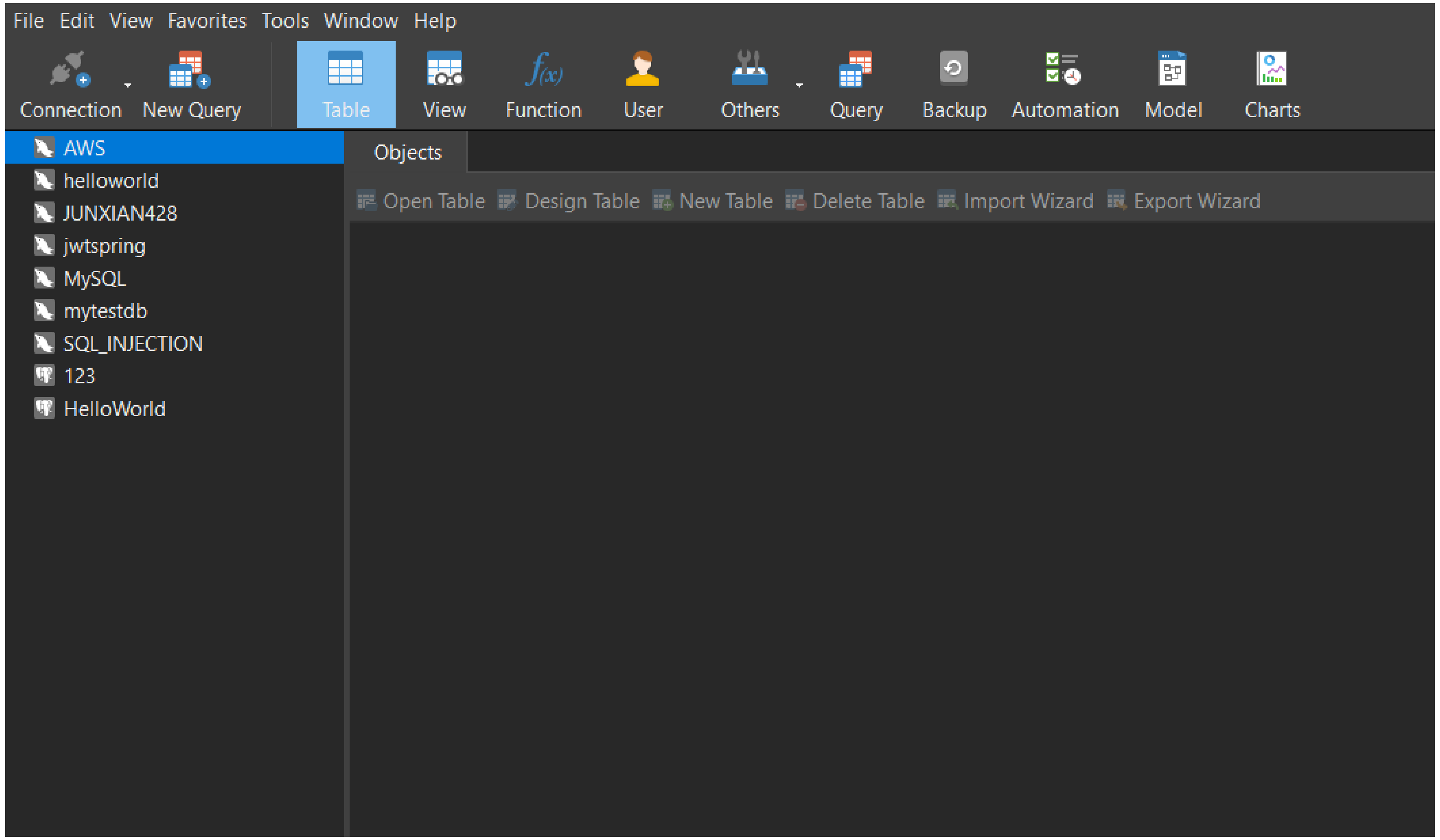
https://github.com/junxian428/Spring_RESTAPI_CRUD_STOCK

Database RDS (MySQL)

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

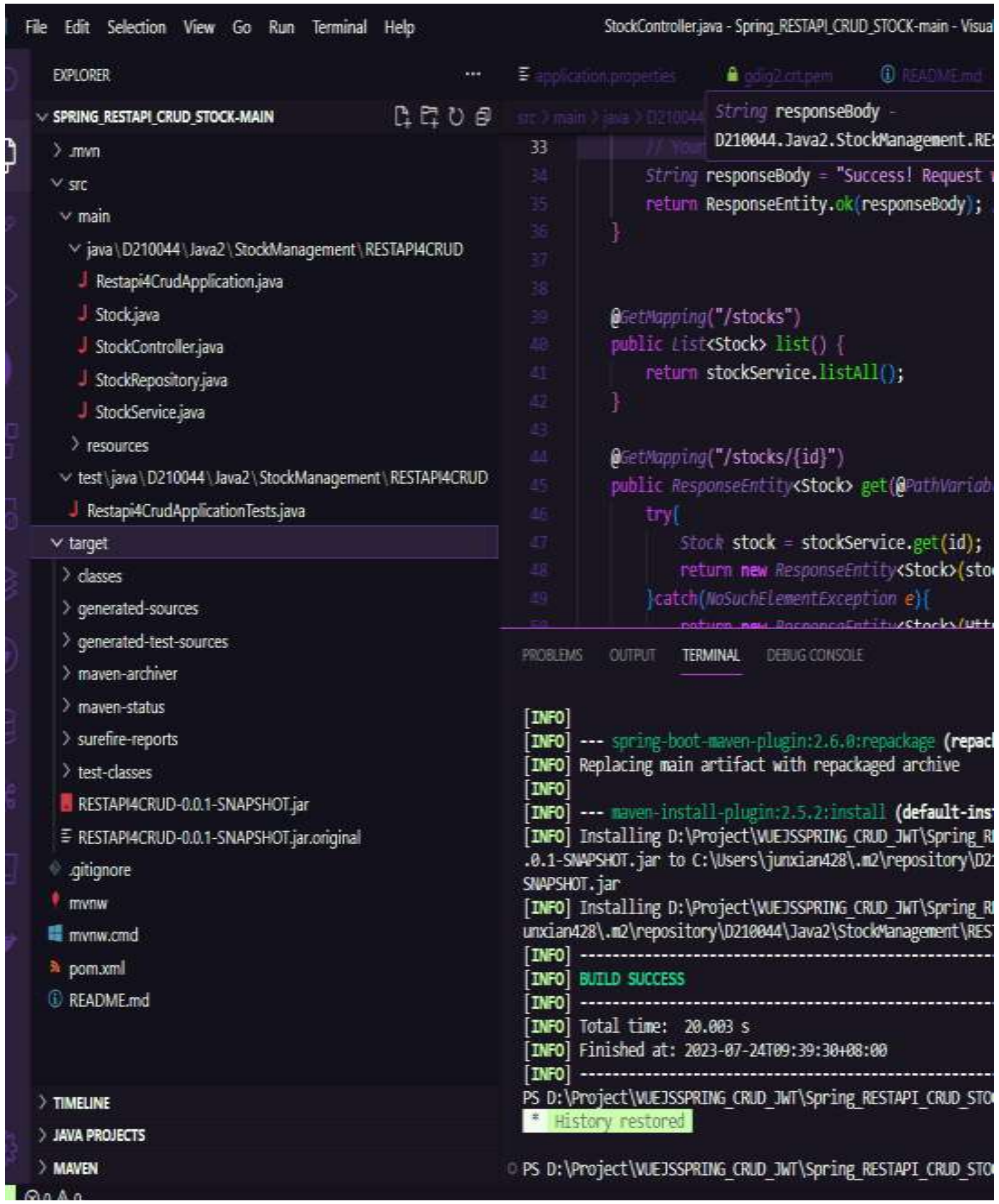
* Change URL to AWS

Navicat connects database



AWS Elastic BeanStalk

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

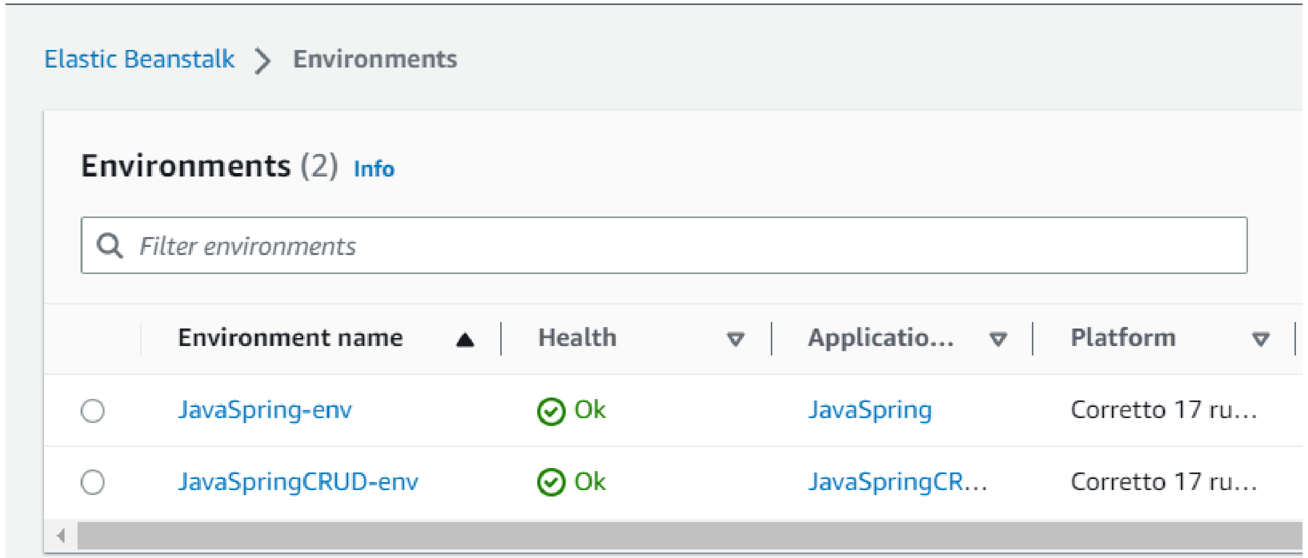


File Edit Selection View Go Run Terminal Help
StockController.java - Spring_RESTAPI_CRUD_STOCK-main - Visua
EXPLORER
SPRING_RESTAPI_CRUD_STOCK-MAIN
.mvn
src
main
java \ D210044 \ Java2 \ StockManagement \ RESTAPI4CRUD
Restapi4CrudApplication.java
Stock.java
StockController.java
StockRepository.java
StockService.java
resources
test \ java \ D210044 \ Java2 \ StockManagement \ RESTAPI4CRUD
Restapi4CrudApplicationTests.java
target
classes
generated-sources
generated-test-sources
maven-archiver
maven-status
surefire-reports
test-classes
RESTAPI4CRUD-0.0.1-SNAPSHOT.jar
RESTAPI4CRUD-0.0.1-SNAPSHOT.jar.original
.gitignore
mvnw
mvnw.cmd
pom.xml
README.md
TIMELINE
JAVA PROJECTS
MAVEN
application.properties
gdig2.crt.pem
README.md
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE

```
String responseBody -
D210044.Java2.StockManagement.RE
```

```java
        String responseBody = "Success! Request
        return ResponseEntity.ok(responseBody);
    }


    @GetMapping("/stocks")
    public List<Stock> list() {
        return stockService.listAll();
    }

    @GetMapping("/stocks/{id}")
    public ResponseEntity<Stock> get(@PathVariab
        try{
            Stock stock = stockService.get(id);
            return new ResponseEntity<Stock>(sto
        }catch(NoSuchElementException e){
```

```
[INFO]
[INFO] --- spring-boot-maven-plugin:2.6.0:repackage (repac
[INFO] Replacing main artifact with repackaged archive
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-ins
[INFO] Installing D:\Project\VUEJSSPRING_CRUD_JWT\Spring_R
.0.1-SNAPSHOT.jar to C:\Users\junxian428\.m2\repository\D2
SNAPSHOT.jar
[INFO] Installing D:\Project\VUEJSSPRING_CRUD_JWT\Spring_R
unxian428\.m2\repository\D210044\Java2\StockManagement\RES
[INFO] ------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------
[INFO] Total time:  20.003 s
[INFO] Finished at: 2023-07-24T09:39:30+08:00
[INFO] ------------------------------------------------------
PS D:\Project\VUEJSSPRING_CRUD_JWT\Spring_RESTAPI_CRUD_STO
 *  History restored

PS D:\Project\VUEJSSPRING_CRUD_JWT\Spring_RESTAPI_CRUD_STO
```

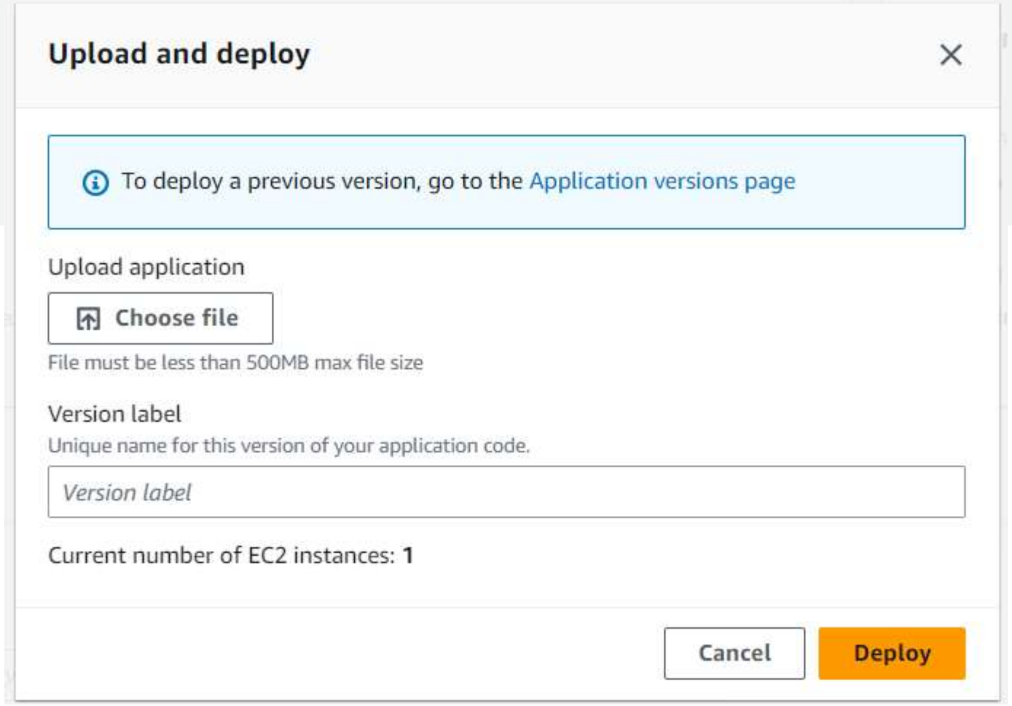


Elastic Beanstalk > Environments

Environments (2) Info

Filter environments

Environment name	Health	Applicatio...	Platform
JavaSpring-env	Ok	JavaSpring	Corretto 17 ru...
JavaSpringCRUD-env	Ok	JavaSpringCR...	Corretto 17 ru...



Upload and deploy

To deploy a previous version, go to the Application versions page

Upload application

Choose file

File must be less than 500MB max file size

Version label

Unique name for this version of your application code.

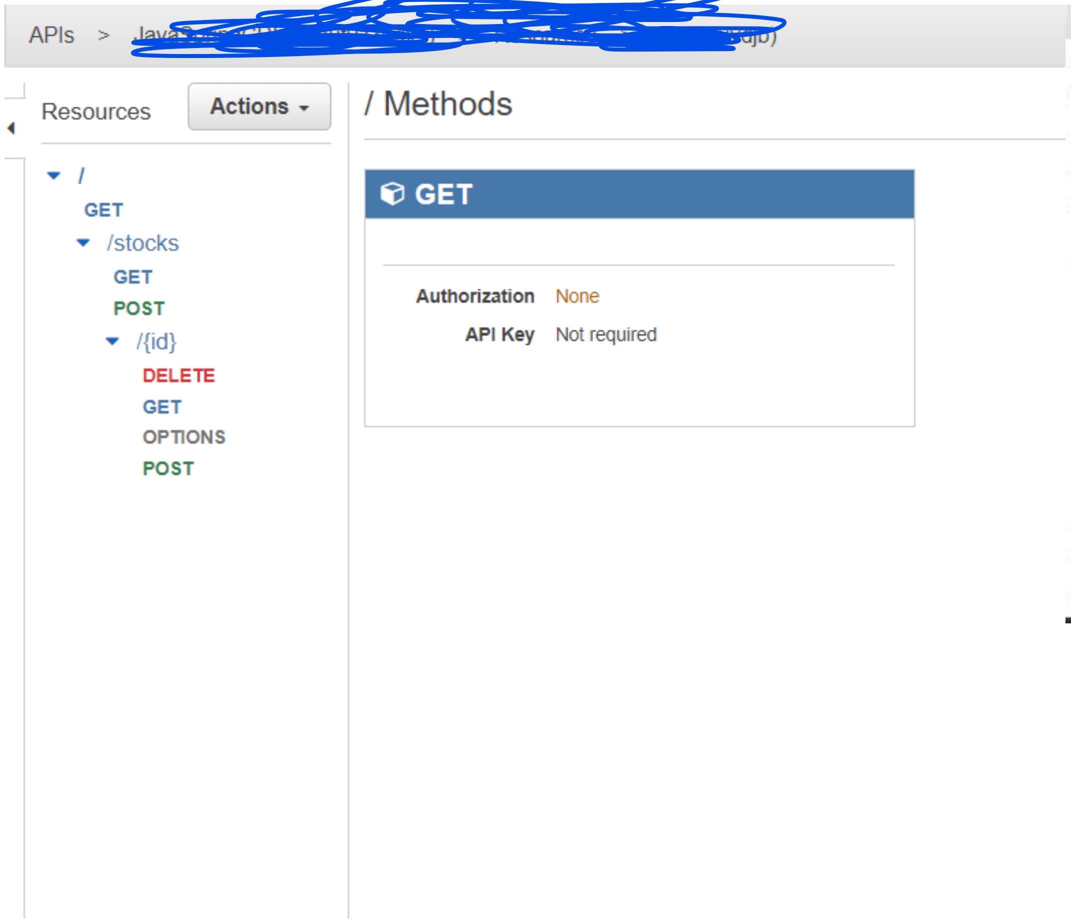
Version label

Current number of EC2 instances: 1

Cancel Deploy

Choose jar file found in target folder

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



APIs > Java[redacted]

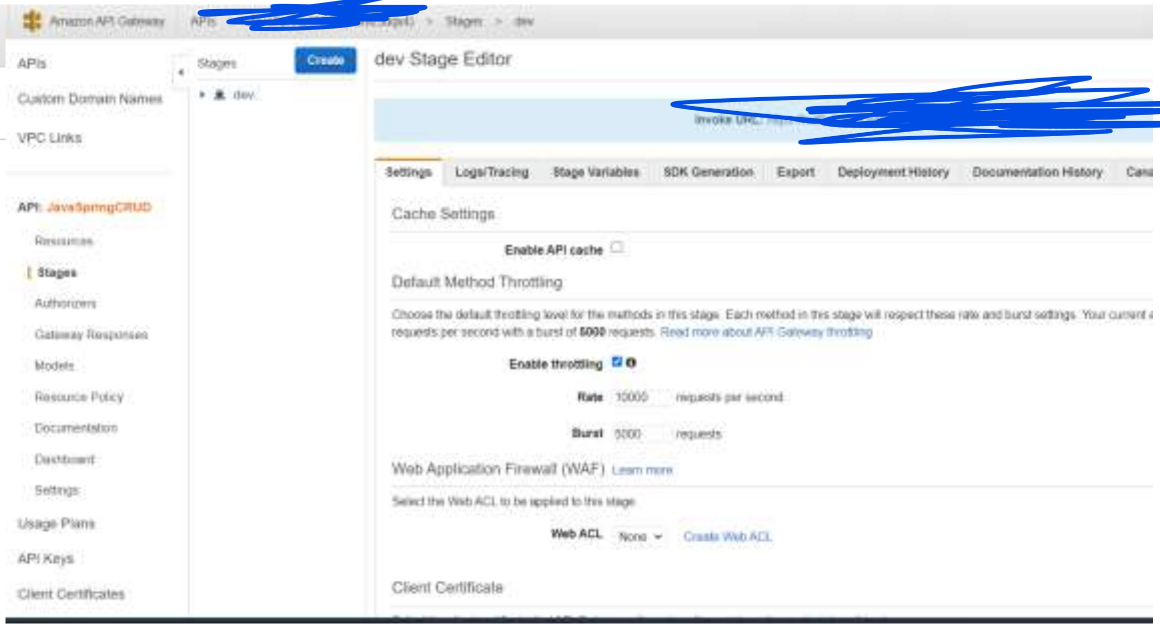
Resources | Actions

/ Methods

- /
  - GET
  - /stocks
    - GET
    - POST
    - /{id}
      - DELETE
      - GET
      - OPTIONS
      - POST

GET

Authorization	None
API Key	Not required



Amazon API Gateway | APIs [redacted] > Stages > dev

APIs
Custom Domain Names
VPC Links

API: JavaSpringCRUD
- Resources
- **Stages**
- Authorizers
- Gateway Responses
- Models
- Resource Policy
- Documentation
- Dashboard
- Settings

Usage Plans
API Keys
Client Certificates

Stages | Create
- dev

dev Stage Editor

Invoke URL: [redacted]

Settings | Logs/Tracing | Stage Variables | SDK Generation | Export | Deployment History | Documentation History | Cana

Cache Settings

Enable API cache ☐

Default Method Throttling

Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current a
requests per second with a burst of **5000** requests. Read more about API Gateway throttling

Enable throttling ☑

Rate 10000 requests per second

Burst 5000 requests

Web Application Firewall (WAF) Learn more

Select the Web ACL to be applied to this stage

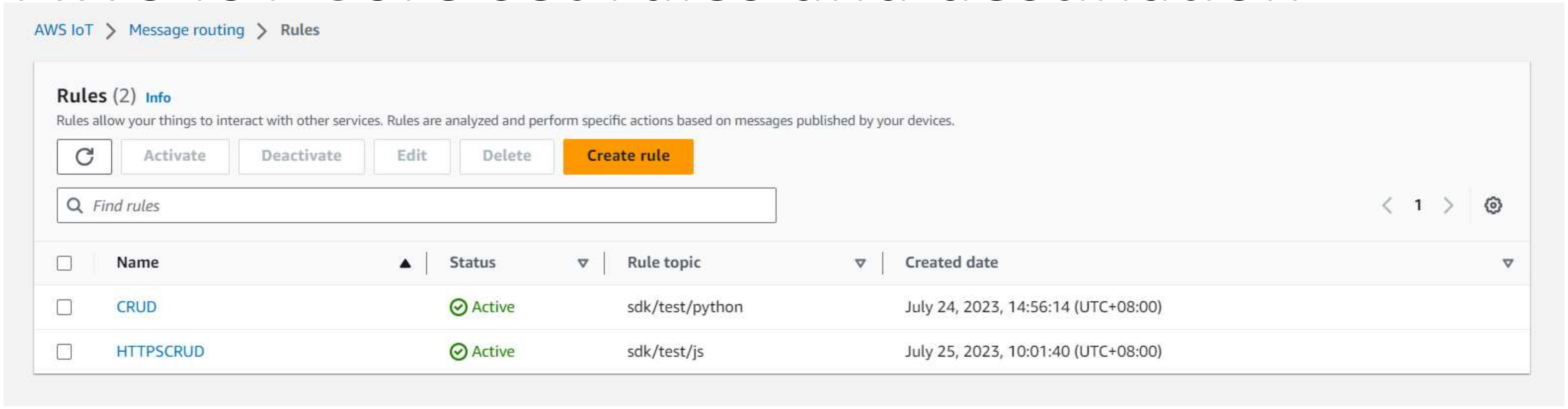
Web ACL None | Create Web ACL

Client Certificate

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

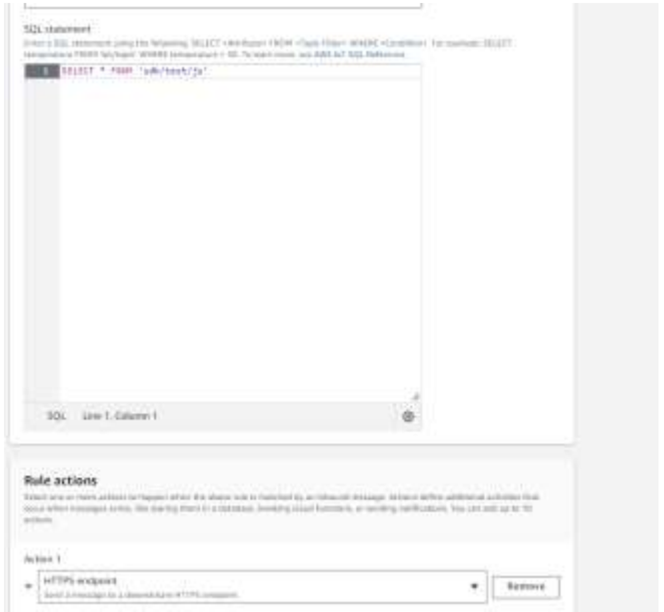


AWS IoT > Message routing > Rules

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.

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)



SQL statement

[illegible]

```
SELECT * FROM 'sdk/test/js'
```

Rule actions

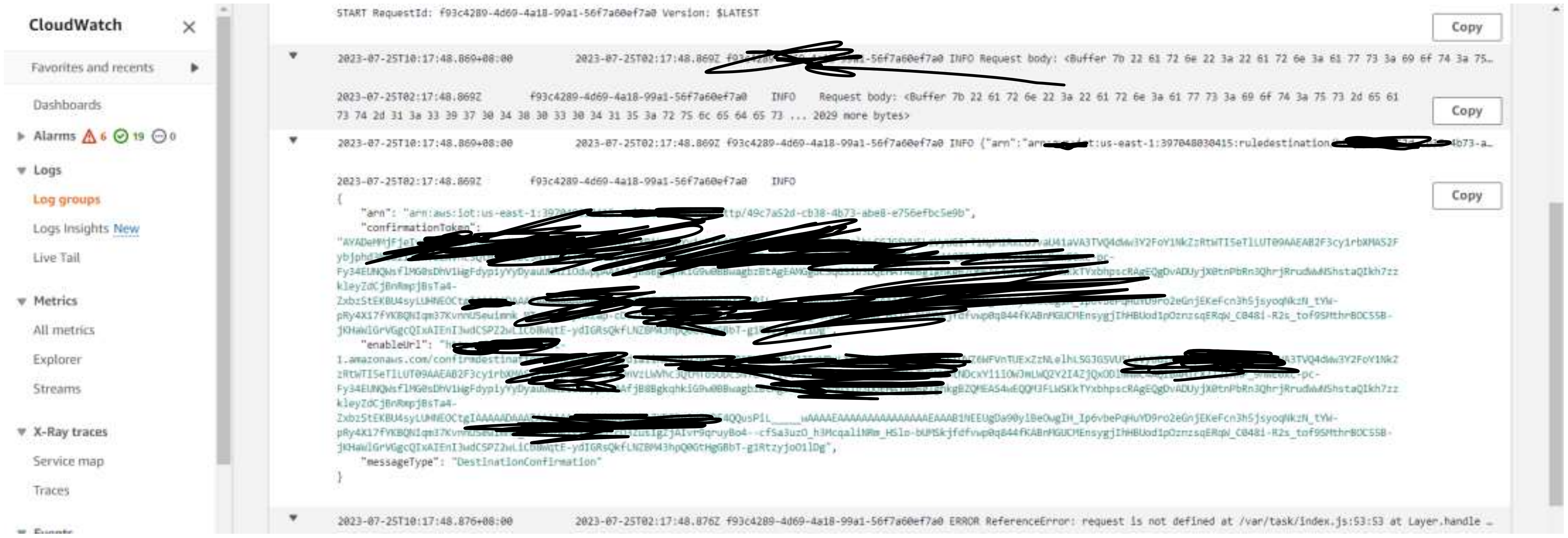
[illegible]

Action 1

HTTPS endpoint

If your destination is not confirmed

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



```
START RequestId: f93c4289-4d69-4a18-99a1-56f7a60ef7a0 Version: $LATEST

2023-07-25T10:17:48.869+08:00    2023-07-25T02:17:48.869Z f93[redacted]-56f7a60ef7a0 INFO Request body: <Buffer 7b 22 61 72 6e 22 3a 22 61 72 6e 3a 61 77 73 3a 69 6f 74 3a 75…

2023-07-25T02:17:48.869Z    f93c4289-4d69-4a18-99a1-56f7a60ef7a0    INFO    Request body: <Buffer 7b 22 61 72 6e 22 3a 22 61 72 6e 3a 61 77 73 3a 69 6f 74 3a 75 73 2d 65 61
73 74 2d 31 3a 33 39 37 30 34 38 30 33 30 34 31 35 3a 72 75 6c 65 64 65 73 ... 2029 more bytes>

2023-07-25T10:17:48.869+08:00    2023-07-25T02:17:48.869Z f93c4289-4d69-4a18-99a1-56f7a60ef7a0 INFO {"arn":"arn[redacted]t:us-east-1:397048030415:ruledestination[redacted]4b73-a…

2023-07-25T02:17:48.869Z    f93c4289-4d69-4a18-99a1-56f7a60ef7a0    INFO
{
    "arn": "arn:aws:iot:us-east-1:397[redacted]ttp/49c7a52d-cb38-4b73-abe8-e756efbc5e9b",
    "confirmationToken":
"AYADeMMjFjeI[redacted]aU41aVA3TVQ4dWw3Y2FoY1NkZzRtWTI5eTlLUT09AAEAB2F3cy1rbXMAS2F
ybjphd[redacted]pc-
Fy34EUNQWsflMG0sDhV1HgFdyp1yYyDyau[redacted]kTYxbhpscRAgEQgDvADUyjX0tnPbRn3QhrjRrudWWShstaQIkh7zz
kleyZdCjBnRmpjBsTa4-
ZxbzStEKBU4syLUHNEOCt[redacted]Ro2eGnjEKeFcn3hSjsyoqNkzN_tYW-
pRy4X17fYKBQNIqm37KvnnUSewimnk[redacted]fdfvwp0q844fKABnMGUCMEnsygjIhHBUod1pOznzsqERqW_C048i-R2s_tof9SMthrBOC55B-
jKHaWlGrVGgcQIxAIEnI3wdCSPZ2wLiCbWqtE-ydIGRsQkfLNZBM43hpQ[redacted]",
    "enableUrl": "h[redacted]
1.amazonaws.com/confirmdestina[redacted]Z6WFVnTUExZzNLelhLSGJGSVU[redacted]A3TVQ4dWw3Y2FoY1NkZ
zRtWTI5eTlLUT09AAEAB2F3cy1rbX[redacted]NDcxY11IOWJmLWQ2Y2I4ZjQxOD[redacted]-pc-
Fy34EUNQWsflMG0sDhV1HgFdyp1yYyDyau[redacted]AfjB8BgkqhkiG9w0BBwagb[redacted]kgBZQMEAS4wEQQM3FLWSKkTYxbhpscRAgEQgDvADUyjX0tnPbRn3QhrjRrudWWShstaQIkh7zz
kleyZdCjBnRmpjBsTa4-
ZxbzStEKBU4syLUHNEOCtgIAAAAADAAA[redacted]F4QQusPiL____wAAAAEAAAAAAAAAAAAAAAEAAAB1NEEUgDa90y1BeOwgIH_Ip6vbePqHuYD9ro2eGnjEKeFcn3hSjsyoqNkzN_tYW-
pRy4X17fYKBQNIqm37Kvn[redacted]Igzjalvr9qruyBo4--cfSa3uzO_h3McqaliNRm_HSlo-bUMSkjfdfvwp0q844fKABnMGUCMEnsygjIhHBUod1pOznzsqERqW_C048i-R2s_tof9SMthrBOC55B-
jKHaWlGrVGgcQIxAIEnI3wdCSPZ2wLiCbWqtE-ydIGRsQkfLNZBM43hpQ0GtHgGBbT-g1RtzyjoO1lDg",
    "messageType": "DestinationConfirmation"
}

2023-07-25T10:17:48.876+08:00    2023-07-25T02:17:48.876Z f93c4289-4d69-4a18-99a1-56f7a60ef7a0 ERROR ReferenceError: request is not defined at /var/task/index.js:53:53 at Layer.handle …
```

- 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
```

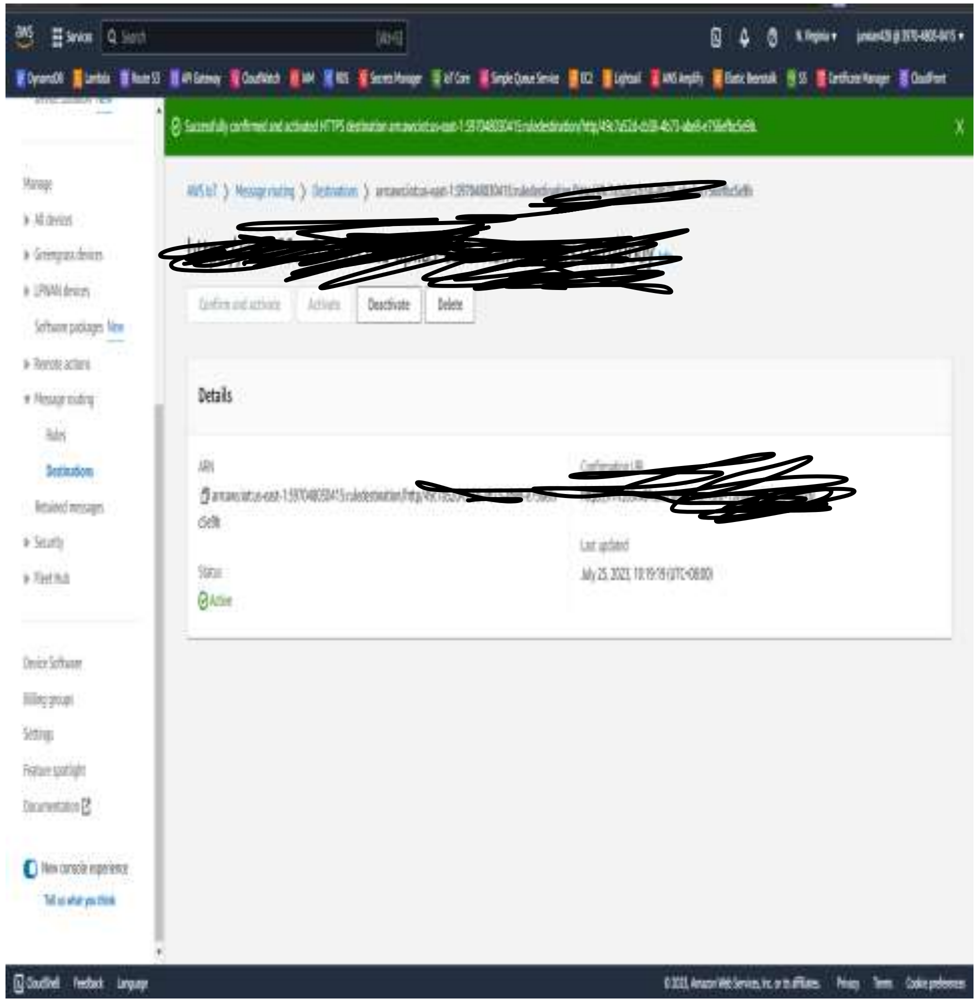
```
onst axios = require("axios");
```

```
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


Successfully confirmed and activated HTTPS destination arn:aws:iot:us-east-1:597048030415:ruledestination/http/49c7a52d-cb59-4b73-abe6-e756efbc5e9b

AWS IoT > Message routing > Destinations > arn:aws:iot:us-east-1:597048030415:ruledestination/[redacted]

[redacted]

Confirm and activate | Activate | Deactivate | Delete

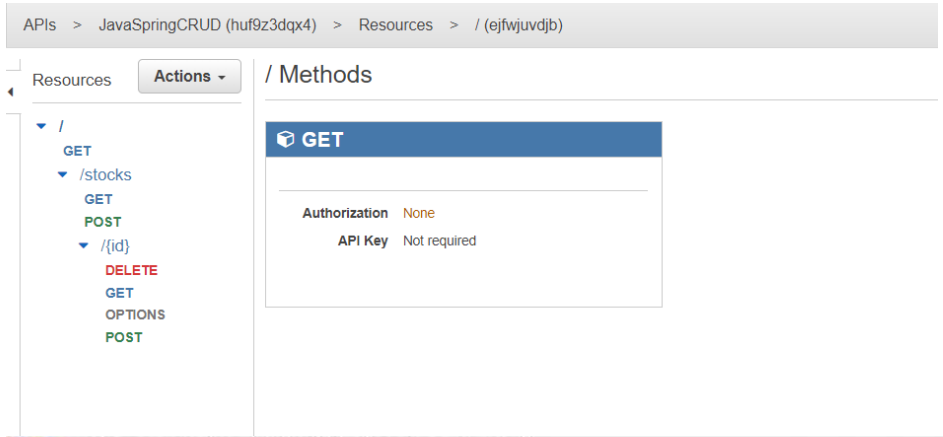
Details

ARN
arn:aws:iot:us-east-1:597048030415:ruledestination/http/[redacted]
c5e9b

Confirmation URL
[redacted]

Status
Active

Last updated
July 25, 2023, 10:19:18 (UTC-08:00)

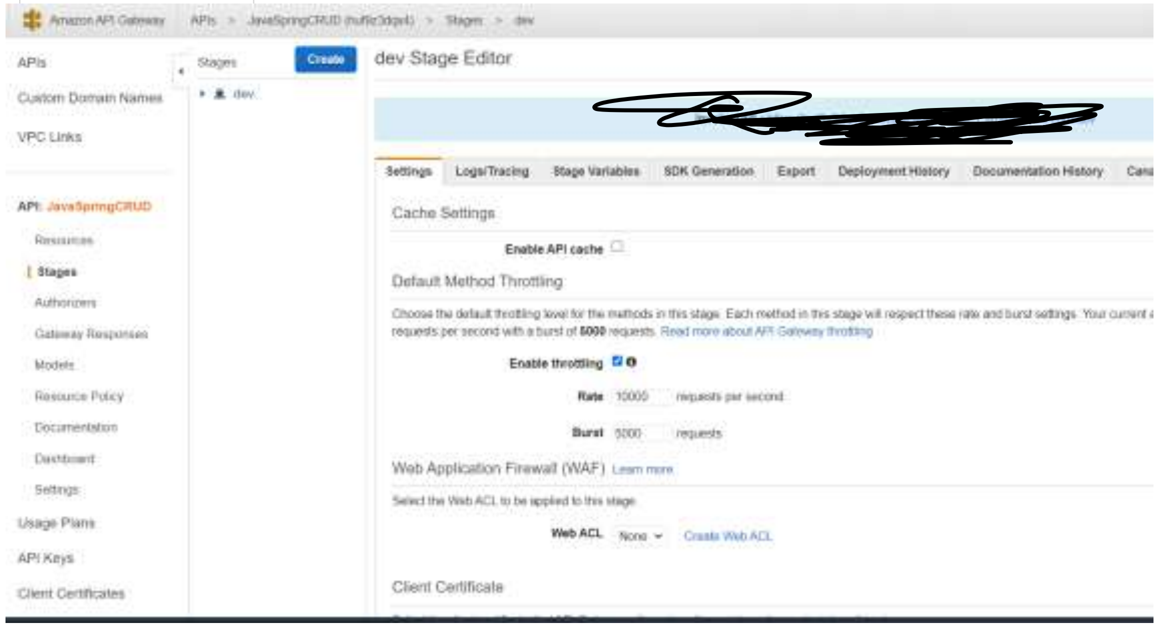

APIs > JavaSpringCRUD (huf9z3dqx4) > Resources > / (ejfwjuvdjb)

Resources | Actions | / Methods

```
/
  GET
  /stocks
    GET
    POST
    /{id}
      DELETE
      GET
      OPTIONS
      POST
```

GET

Authorization	None
API Key	Not required


Amazon API Gateway > APIs > JavaSpringCRUD (huf9z3dqx4) > Stages > dev

dev Stage Editor

[redacted]

Settings | Logs/Tracing | Stage Variables | SDK Generation | Export | Deployment History | Documentation History

Cache Settings

Enable API cache ☐

Default Method Throttling

Choose the default throttling level for the methods in this stage. Each method in this stage will respect these rate and burst settings. Your current requests per second with a burst of 5000 requests. Read more about API Gateway throttling

Enable throttling ☑

Rate 10000 requests per second

Burst 5000 requests

Web Application Firewall (WAF) Learn more

Select the Web ACL to be applied to this stage

Web ACL None Create Web ACL

Client Certificate

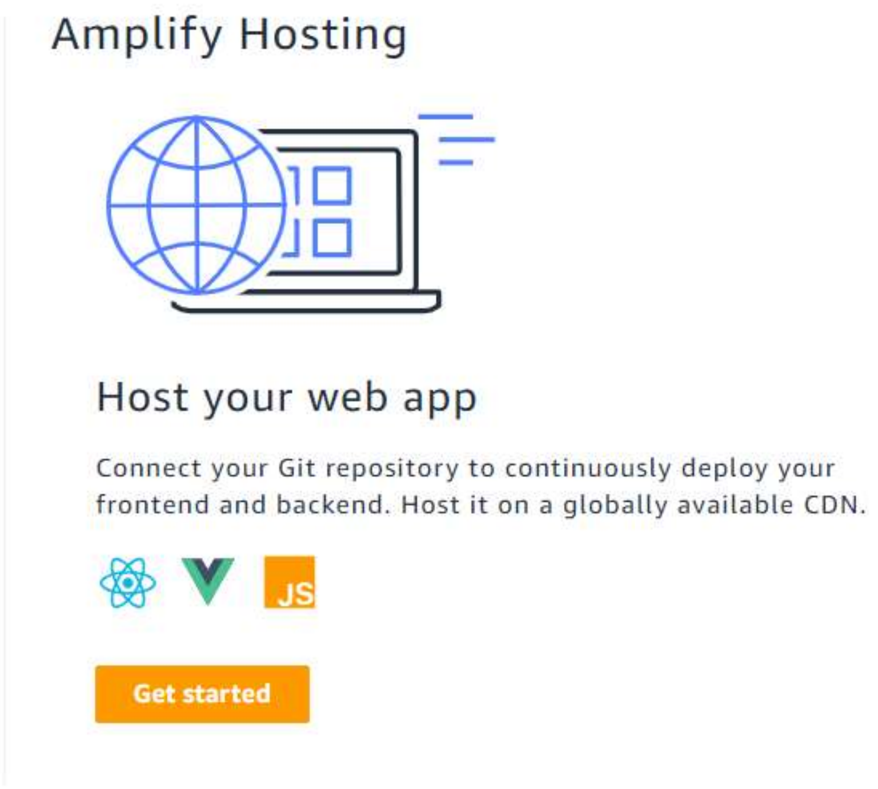
Frontend Last Methodology

- Create vuejs
- Vue create frontendproject

Choose router & vuex

Frontend Deploy (AWS Amplify)

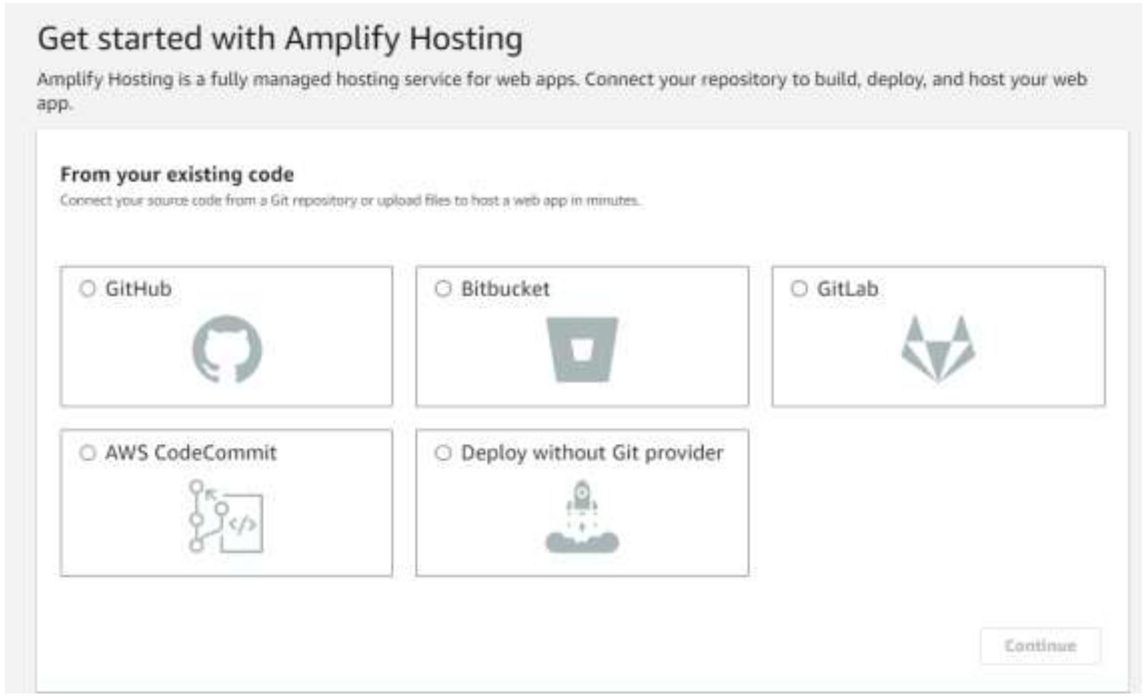
AWS Amplify



Amplify Hosting

Host your web app

Connect your Git repository to continuously deploy your frontend and backend. Host it on a globally available CDN.



Get started with Amplify Hosting

Amplify Hosting is a fully managed hosting service for web apps. Connect your repository to build, deploy, and host your web app.

From your existing code

Connect your source code from a Git repository or upload files to host a web app in minutes.