

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
Javascript
Spring Boot
Selenium
Sonarqube
Advanced Testing (Theory)
Cucumber
MongoDB
Express
NodeJS
React
Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
<div><div></div>Virtual Private Cloud (VPC)</div>
<div><div></div>EC2 VPC Security Groups</div>
<div><div></div>EC2 VPC Subnets</div>

AWS API Gateway

Contents

- [Overview](#)
- [Pre-requisite](#)
- [Tutorial](#)
- [Resource](#)
- [Exercises](#)

Overview

Amazon API Gateway is a fully managed service which makes it easier to create, monitor, secure and maintain and API's at any scale. This service acts as a "front door" to access data, business logic, or functionality from your backend services.

This service is capable of handling thousands of concurrent API calls, including traffic management, throttling, monitoring and more. The pricing model is flexible, so you only pay for what you use, no upfront cost.

Pre-requisite

Before you begin, you must deploy the following application from github onto an EC2 instance and run the application.

GitHub Repo:

- https://gitlab.com/qacdevops/api_gateway_example.git

EC2 Instance:

- At least t2.small

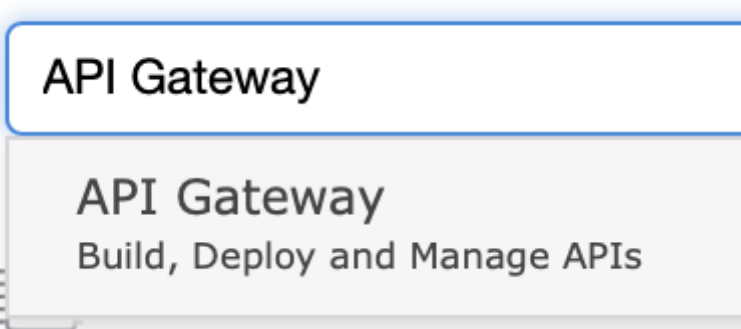
Command:

- mvn clean install

-mvn spring-boot:run

Tutorial

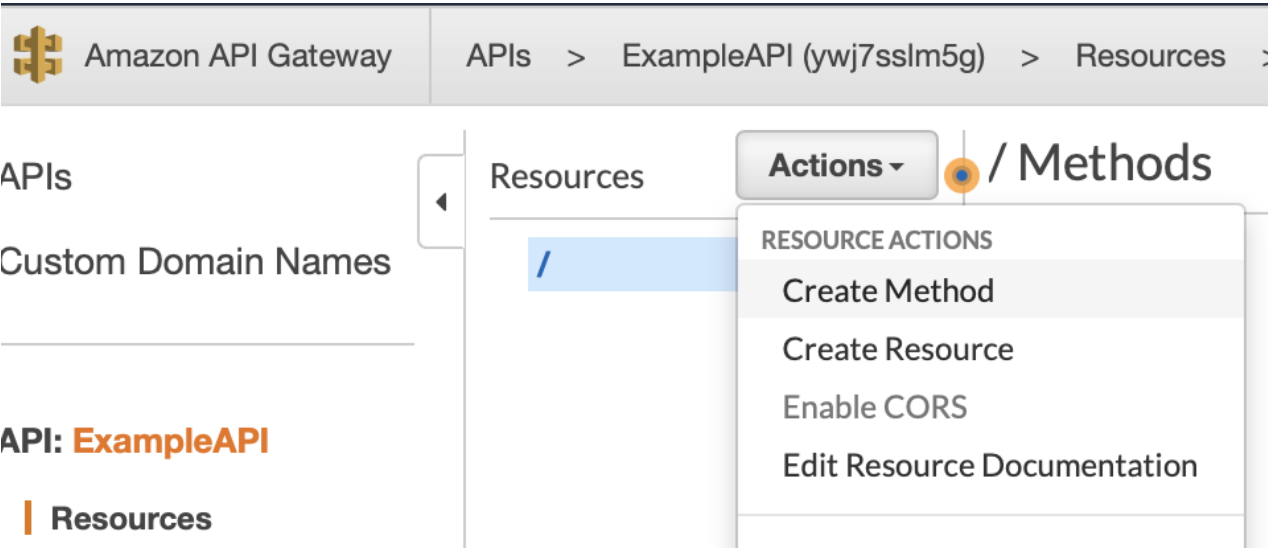
- Navigate to the AWS Console and sign in [here](#)
- Search for API Gateway under the services drop-down menu, and click on API Gateway



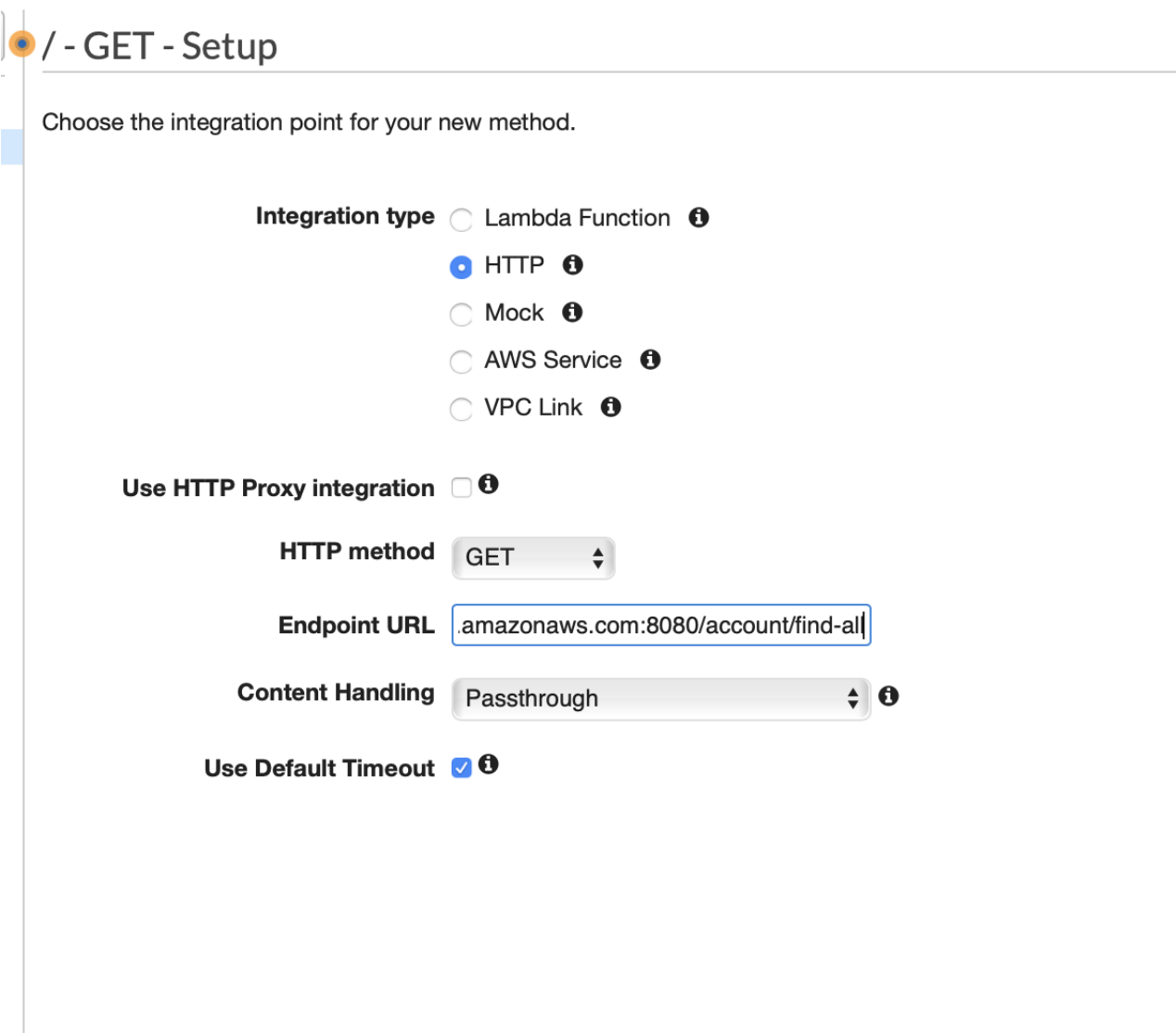
- You will be redirected to the **API Gateway** homepage. This will only show your current API's which you have created. Click on **Create API**
- We are going to build a **REST API**. Click **Build** under this section to get started.

<input checked="" type="radio"/>	EC2 VPC Internet Gateways
<input checked="" type="radio"/>	AWS Route Tables
<input checked="" type="radio"/>	AWS Network Address Translation (NAT) Gateway
<input checked="" type="radio"/>	AWS Network Access Control Lists (NACLs) CLI
<input checked="" type="radio"/>	AWS Java SDK
<input checked="" type="radio"/>	AWS DynamoDB
<input checked="" type="radio"/>	AWS Lambda Functions
<input checked="" type="radio"/>	AWS API Gateway
<input type="radio"/>	SQS Introduction
<input type="radio"/>	AWS Serverless CRUD Solution
<input type="radio"/>	AWS Serverless Solution with DynamoDB
<input type="radio"/>	CloudWatch CLI
<input type="radio"/>	CloudTrail
Linux	
DevOps	
Jenkins Introduction	
Jenkins Pipeline	
Markdown	
IDE Cheatsheet	

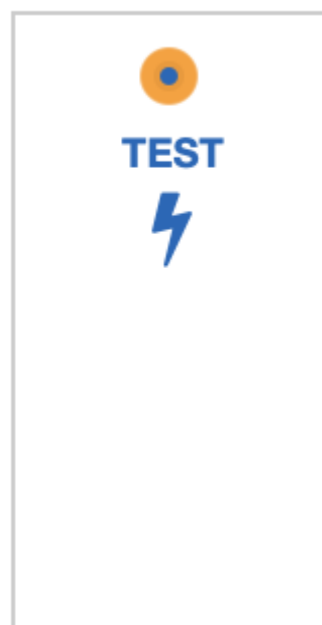
5. Leave everything as default and only provide a name for the API you are creating.
Continue to **Create**
6. Click on action and **create method**.
This will then give you an option of the type of REST API method you want to make.



7. Select **GET** as we will be making a simple method to get information.
8. Fill in the section as follows:

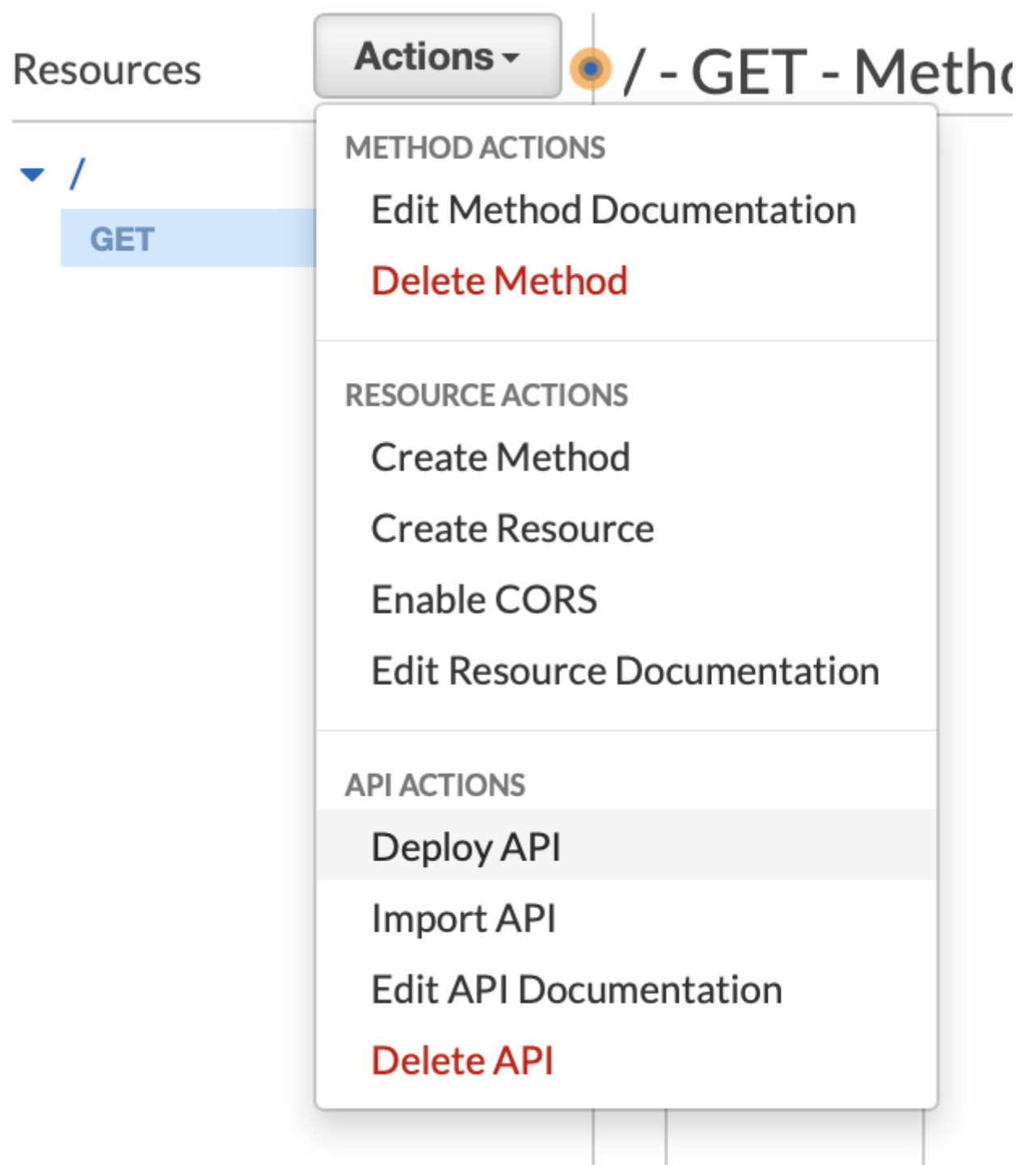


9. The **Endpoint** should be the public DNS of your EC2 instance which is running the application provided.
- Add **http://** at the beginning of your DNS, since this is suppose to be a HTTP endpoint.
 - Add **:8080/account/find-all** to the end the DNS when providing the endpoint.
10. Continue to **save**
11. You can only test this. Click on **Test** and if it returns **200** then it was a success.



12. In order for **API Gateway** to be properly utilised, you will need to deploy this API.

Click on **Action** and you have the option to **Deploy API**.



13. You will be prompted to select an existing stage or create a new stage for your API.

Choose a stage where your API will be deployed. For example, a test version of your API could be deployed to a stage named beta.

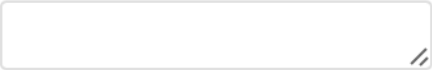
Deployment stage

[New Stage] 

Stage name*

test|

Stage description



Deployment description



Cancel

Deploy

14. Continue to **Deploy**

15. The next window will show you the stage which you have just created.

And at the top you are given the option **Invoke URL**.

When the URL is called it will be as if you called the API for your application, which you created previously when creating your **GET** Method.

Resource

- EC2
- API Gateway

Exercises

Create an API Method which uses the **POST** method. The application provided has all the code in, so you will need to create the Method and the stage and test the API Gateway.