Build, Secure and Manage Serverless Applications at Scale on AWS lab

In this lab, we will explore creating a serverless application using an API-first design and the AWS Serverless Application Model (AWS SAM) where the user can create, update, delete, and view orders.

Prerequisites

- Have access to an AWS account, ensure you are following the remaining workshop steps as an IAM user with *AdministratorAccess* to the AWS account
- You will need to create a Cloud9 Workspace. Follow the setup link here: Cloud9
- You will need to download Postman. Follow this link to download Postman

To do the Build, Secure and Manage Serverless Applications at Scale on AWS lab

In this task you will cover the following 6 labs; 1) Setup, 2) Serverless CRUD API, 3) Order Poller, 4) Order Management, 5) Order Authentication, and 6) Advanced Authorization in Order API. Each lab below is dependent on the previous one.

The total duration of the following 6 labs: **4 hours**

1. Lab 0: Setup

Lab link: Setup

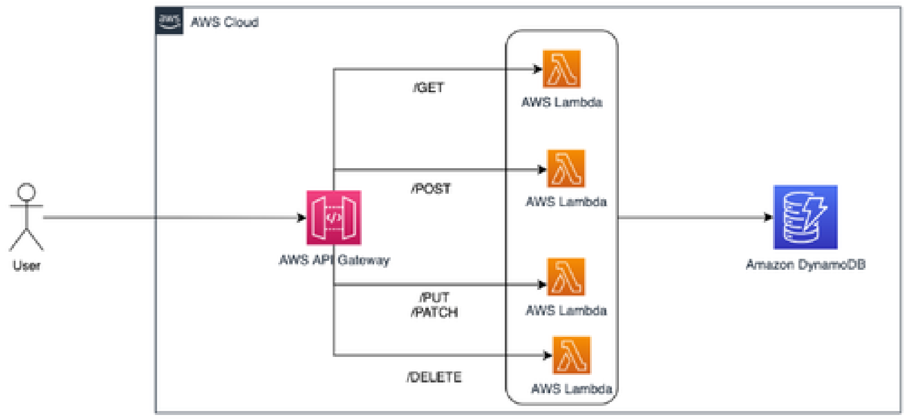
Duration: 45 Minutes

Overview: In this lab, we will use the AWS SAM CLI to create an initial skeleton of a *hello world* application.

NOTE:

- Under the **next steps** section, you should manually delete the `hello-world` Lambda function in the `Resources` section and `Outputs` section from our `template.yaml` file.

2. Lab 1: Serverless CRUD API



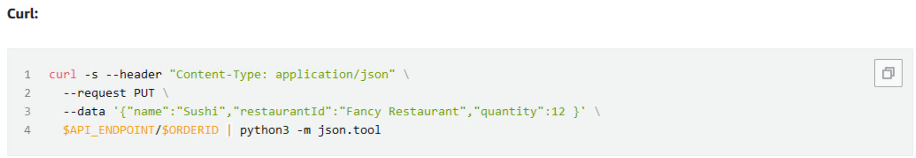
Lab link: Serverless CRUD API

Duration: 45 Minutes

Overview: In this lab, we will be building an Order API by using the AWS Serverless Stack (**ApiGateway** - **Lambda** - **DynamoDB**) and the synchronous request-response pattern with AWS SAM. As we go through the labs the architecture provided in Figure 1 will get upgraded.

NOTE:

- To get the URL for Postman, on your terminal in Cloud9 run this command `echo {$API_ENDPOINT}` .
- In the ***Update Order*** section, number **5** Testing. Below you will see two code snippets the only thing that is different between the two is the **data** provided. Figure 2 is a screenshot of the code that should be replaced. Copy and use the code snippet that follows Figure 2, instead of the one provided in the lab in number **5** it does not give the desired results.



Curl:

```
curl -s --header "Content-Type: application/json" \
  --request PUT \
  --data '{"name":"Sushi","restaurantId":"Fancy Restaurant","quantity":12 }' \
  $API_ENDPOINT/$ORDERID | python3 -m json.tool
```

```
curl -s --header "Content-Type: application/json" \
  --request PUT \
  --data '{"name":“Burger","restaurantId":"Restaurant 2","quantity":3 }' \
  $API_ENDPOINT/$ORDERID | python3 -m json.tool
```

3. Lab 2: Order Poller

Lab link: Order Poller

Duration: 45 Minutes

Overview: In this lab, the API Gateway integration pushes a message to Amazon SQS whenever a new order is created. That way we effectively change the process from a synchronous one to an asynchronous one.

4. Lab 3: Order Management

Lab link: Order Management

Duration: 45 Minutes

Overview: In this lab, we will be using AWS Step functions to process and show payment /restaurant checking/ and sending notifications to users about the process.

5. Lab 4: Order Authentication

Lab link: Order Authentication

Duration: 45 Minutes

Overview: In this lab, we will be adding Authentication to our API to create/fetch/update/delete orders and access our APIs securely using Amazon Cognito.

6. Lab 5: Advanced Authorization in Order API(Optional)

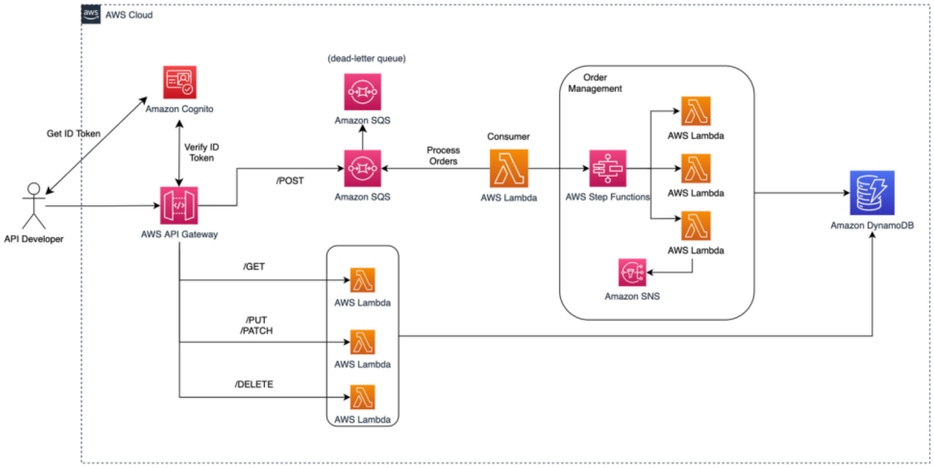
Lab link: Advanced Authorization in Order API

Duration: 45 Minutes

Overview: In this lab, we will be using a more Advanced Authorization mechanism with **OAuth 2.0 authorization framework** to access API.

Task results

The serverless application looks like the architecture in Figure 3, where users of your application can list their orders, create new orders, show order details, update order status and cancel the order. For each lab, the result i.e. the lab's end-state architecture, is shown as you go through the labs.



Cleanup

Lab link: Lab cleanup

Duration: 2 Minutes

Overview: This will cover how to clean up each lab that was setup

Additional resources

- [Concept] AWS ramp-up guide
- https://catalog.us-east-1.prod.workshops.aws/workshops/b34eab03-4ebe-46c1-bc63-cd2d975d8ad4/en-US/1-introduction
- https://auth0.com/docs/authenticate/protocols/oauth?&amp
- https://docs.aws.amazon.com/cognito/latest/developerguide/what-is-amazon-cognito.html
- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/welcome.html
- https://docs.aws.amazon.com/serverless-application-model/latest/developerguide/serverless-getting-started.html
- https://docs.aws.amazon.com/step-functions/latest/dg/welcome.html
- https://aws.amazon.com/what-is/api/
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_job-functions.html#jf_administrator