13

Aim:- Implementing a serverless Architecture on Aws

Description:- This lab demonstrates how to trigger a lambda function when a file is uploaded to Amazon Simple storage service (Amazon S3). The file will be loaded into a Amazon DynamoDB table. The data will be available for you to view on a dashboard page that retrieves the data directly from DynamoDB.

Task 1:-

* on the services menu, Choose Lambda
* Choose Create function
* Configure the following setting:
  - function name: Load-Inventory
  - Runtime: Python 3.7
  - Expand Choose or Create an execution role.
  - Execution role: Use an existing role
  - Existing role: Lambda-load-Inventory-Role
* Choose Create function.
* In the code source editor, copy + paste the following code:
* Choose Deploy to save your changes

Task 2: Configuring an Amazon S3 event

* on the services menu, choose S3
* Choose create bucket
* for Bucket name enter: inventory<number>
* Choose create bucket
* Choose the name of your inventory-bucket

* choose the properties tab
* scroll down to Event notifications
* click [Create event notification] then configure these settings:
  - Name: Load-Inventory
  - Event types: All object create events
  - Destination: Lambda function
  - Lambda function: Load-Inventory
  - Choose save Changes

Task 3: Testing the loading process

* Download the inventory files by opening the context menu for these links
  inventory-berlin.csv
  inventory-calcutta.csv
  inventory-karchi.csv
* choose objects tab Choose Upload
* choose [Add files]
* choose Upload.
* From the Credentials window, copy the Dashboard URL
* on the services menu, choose DynamoDB
* choose Tables
* choose the Inventory table
* choose the Items tab.

Task 4: Configuring Notifications

* on the services menu, choose Simple Notification service
* enter: Nostock
* choose create topic
* - Protocol: Email
  - endpoint: Enter your Email address
  - Choose Create Subscription

Tasks:- Creating a lambda function to send notifications

* on the services menu, Choose lambda
* Choose Create function
  - function name: check-stock
  - Runtime: Python 3.7
  - Expand Choose or create an execution role
  - Execution role: use an existing role
  - Choose Create function
* Choose Deploy to save your code changes
* scroll to the Designer section
* Choose Add trigger
  - select a trigger: DynamoDB
  - DynamoDB Table: Inventory
  - Choose Add

Task 6: Testing the System

* on the services menu, Choose S3
* Choose the name of your inventory-bucket
* Choose Upload
* Return to the Inventory System Dashboard & refresh the page.

submitting your work

* Choose submit and then Click on End Lab