

Name:- Aditya Adinath Kirtane -
PIN:- D15C, Roll No. 26.

Page No.	OB
Date	05/01/2024

Assignment - 1.

Adv - DevOps.

use S3 bucket and host bucket streaming
the Step by Step procedure is as follows:-

Step 1). Create S3 bucket.

- (1). Log into your AWS management console.
- (2). Navigate to S3 under Storage section.
- (3). Click Create Bucket and provide a unique bucket name.
- (4). Configure permissions.
- (5). Complete the bucket creation process.

Step 2). Upload video files on S3.

- (1). Click on your bucket name.
- (2). Use the Upload button to add your video file.
- (3). Set the access control list (ACL) to allow public read access.

Step 3). Enable static website hosting.

- (1). Inside the S3 bucket, go inside the properties tab.
- (2). Scroll the static website hashing section and enable it.
- (3). Set the index document and provide an optional error document.

16/10/2024

Step 4) Configure bucket permissions
to allow public access.

Step 5) 1. Create a index.html file containing a video player using <video> tag.

2. Upload this index.html to your S3 bucket

Step 6) Access your hosted video. Your video would be played in embedded HTML page.

Q.2- BMW and Hotstar Case Study up AWS.

→ BMW uses AWS to power its connected system, enabling real-time data processing, over-the-air updates, a personalized driving experience. AWS global infrastructure helps BMW manage large-scale vehicle data, allowing maintenance and enhancing customer experience through features like traffic info, parking availability.

• Hotstar, a leading streaming platform, relies on AWS to manage massive traffic spikes during events like IPL where millions of users connect simultaneously. This ensures smooth service during peak loads, enabling Hotstar to handle over 25 million users.

Why Kubernetes and its advantages and disadvantages. Adidas case study. Kubernetes automates the deployment, scaling and management of containerized application, making it popular choice for orchestrating microservices.

Advantages.

Automated Scaling and Self healing.
~~Supports~~ Supports multicloud and hybrid deployment efficient resource usage through container orchestration.

Disadvantages.

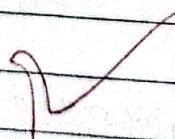
Complexity in Setup and operation. require steep learning curve for configuration.

Adidas uses Kubernetes to manage its e-commerce infrastructure, which involves deploying 100's of microservices. Kubernetes enables Adidas to handle peak traffic during product launches by scaling services automatically and distributing load efficiently.

Q. 4 What is Nagios and how it is used in E-Services?

→ Nagios is an open source monitoring tool used to monitor the health and performance of IT infrastructure. It tracks system metrics, like CPU, memory, disk usage, alert administrators of potential risks.

In E-Services, Nagios ensures the uptime and performance of critical online services by monitoring service health detections, failures, and automatically triggering failure actions, downtime and maintain service availability, making it a crucial tool for maintaining seamless user experience.



Name: Aditya Adinath Kirtane.

Roll no. 26, Division: DISC

Page No.

Date

Advance devops - Assignment No.2.

Creating a REST API using the Serverless framework is a great way to build a Scalable and efficient API with minimal infrastructure management.

Prerequisites:-

AWS account

Node.js installed.

Serverless Framework installed on npm
Install -g Serverless.

Step 1:- install the Serverless framework.
npm install -g serverless.

Step 2:- create a new service.
Create a new service by running
following command

Serverless create --template aws-nodejs
--path rest-api-serverless
This creates a directory called rest-api-serverless with some basic configuration files.

Step 3:- Update Serverless.yml.

The Serverless.yml is the configuration file that defines your service, including function, event and resources.

Service: rest-api-serverless

Provider:-

name: aws.

runtime: nodejs18.x.

stage: dev.

Region: us-east-1.

functions:-

createUser:-

handler: handler.createUser.

events:

- http:

Path: users.

method: Post.

getUser:-

handler: handler.getUser.

events:

- http:

Path: users/{id}.

method: get

~~updateUser~~

handler: updateUser.

events:

- http:

Path: users/{id}.

method: put.

BillingMode: PAY PER REQUEST.

Step 4:- Create the Handler functions
In the handler.js file, implement your lambda functions.

```
const AWS = require('aws-sdk');
const dynamoDb = new AWS.DynamoDB.DocumentClient();
try {
    await dynamoDb.delete({Params}).promise();
    return {
        statusCode: 200,
        body: JSON.stringify({message: 'User deleted successfully'})
    };
} catch(error) {
    return {
        statusCode: 500,
        body: JSON.stringify({message: 'could not delete user'})
    };
}

```

Step 5:- Deploy the Service
Serverless deploy.

Step 6:- Test the API.
After deployment, the Serverless Framework will return the API gateway endpoint.

Step 7: ~~Ans~~ Monitor and manage.

- To view logs - Serverless logs for each service.
- To remove the service when done. Serverless remove.

Q. 2. Case study for Sonarqube.

- Create your own profile in Sonarqube for testing project quality.
- Use Sonarcloud to analyze your GitHub code.
- Install Sonarlint.
- Analyze Python project with Sonarqube.
- Analyze Node JS project with Sonarqube.

Step 1. Create your own profile in Sonarqube for testing project quality.

- (1) Install Sonarqube.
 - Download and install Sonarqube from Sonarqube downloads.
 - Start the Sonarqube Server.
 - Open the dashboard.

(2) Login and Set up profile.

- Log in to Sonarqube.
- Go to Quality Profiles from top menu.
- Click Create to make a profile.

3) Run analysis on your project.
 Install and configure SonarQube Scanner on your local machine
 configure the Sonar project properties

4) Review results-

After the Scan complete, go to the SonarQube dashboard checks the detected issues, bugs, and vulnerabilities based on custom profile.

Step 2. Use SonarCloud to analyze your Github code
Objectives:- analyze your Github repository.

Procedure:-

1. Set up SonarCloud
2. Link a Github repository
3. Integrate SonarCloud
4. Run analysis

Step 3- Install Sonarlint in IntelliJ or Eclipse IDE for Java code

1. install Sonarlint plugin
2. configure Sonarlint
3. analyze code

Step 4:- Analyze a python Project with SonarQube

- (1). Prepare python Project
- (2). Run SonarQube Scanner.
- (3). Review Results

Step 5:- Analyze node.js Project with SonarQube

- (1) Prepare Node.js Project.
- (2) Run SonarQube Scanner.
- (3) Review Results.

Q.3 ~~Ans~~ At a large organization your centralized operation team may get many repetitive infrastructure request.

lets product team manage their own infrastructure independently using terraform modules that codify the standards

Organization, allowing team to efficiently deploy service. ServiceNow to automatically generate new infrastructure requests

Building a Self Service infrastructure model with terraform for large organization.

key concepts

self-service infrastructure

terraform modules

terraform cloud

terraform integration

Step 1:- Designing Terraform module for self service

(1) Create Standard terraform module

(2) Encapsulate standards

(3) Version control for modules

Step 2:- Implementing terraform cloud for collaboration

1. Set up workspaces

2. Set up permission

3. Automate runs

Step 3:-

1. Setup Service Now

2. Service-Now terraform cloud integration

3. Automate request fulfillment

Step 4:- Deploying and managing infrastructure by teams

Step 5:- monitoring and continuous compliance