

Read the following instructions before completing the given task.

Step 1: Click on the Online Task Link

Step 2: Download the document

Step 3: Check the online questions

Step 4: Complete the task and take a screenshot for each and every task and upload

Step 5: Save the document

Step 6: Upload the document in Google form

- **AWS Account is mandatory to complete the tasks.**
- **Include AWS account Name in your screenshots**
- **Resume / Copied and similar tasks will be rejected directly.**

Assignment 1.

Creating an IAM User and Assigning Permissions:

Click on "Users" in the left-hand menu and choose "Add user."

Provide a username (ABCD) and select the access type (programmatic access, AWS Management Console access, or both).

Set permissions by adding the user to one or more IAM groups with predefined policies or by attaching custom policies directly.

Upload the final output Screenshot

Assignment 2

Associate an Elastic IP to an EC2 Instance using Terraform

- Create an ec2 instance using terraform workflow

- associate an elastic IP

Upload the final output Screenshot

Assignment 3.

CI/CD Pipeline:

- Use GitHub or any version control system of your choice for source code management.
- Set up an AWS CodePipeline to automate the CI/CD process.
- Configure AWS CodeBuild as a build stage in your pipeline to build the application artifact.
- Add a deployment stage using AWS CodeDeploy or any other deployment tool to deploy the application to the EC2 instance.

Upload the final output Screenshot

Assignment 4

Creating a Basic Amazon S3 Lifecycle Policy

- Create an S3 Bucket and Upload an Object
- Create a Lifecycle Policy - Create a lifecycle policy that moves objects to Glacier Flexible Retrieval (formerly Amazon Glacier) if they haven't been accessed in the last 30 days.

Upload the final output Screenshot

Assignment 5

Create a VPC with 2 subnets (public and private) in different AZs, attach an internet gateway, and launch EC2 instances. Try to access the private instance through the instance in public subnet (Clue:- using SSH)

Upload the final output Screenshot

Assignment 6

Set up Cross-Region S3 Bucket Replication

- Create an S3 Bucket and Enable Replication
- Test Replication and Observe Results

Upload the final output Screenshot

Assignment 7

Creating a Custom Amazon Machine Image (AMI)

- Launch a New EC2 Instance
- Install http on the new instance, enable the http service to start at boot.
- Create a New AMI from customized instance and name the AMI MicroDegreeWeb
- Launch a New Instance Using the Custom AM
- Verify that http is running.

Upload the final output Screenshot

Assignment 8

Web Application:

- Develop a simple "Hello World" web application (you can use any programming language or framework of your choice).
- Package the application into a deployable artifact (e.g., a JAR file, Docker image).

Upload the final output Screenshot

Assignment 9

Create a Staging branch in GitHub, push code from local repository to remote, and share commands

Upload the final output Screenshot

Assignment 10

Create a sample Jenkins CICD pipeline and share output logs and pipeline script

Upload the final output Screenshot

Assignment 11

Create a docker service with image "tomcat", name "HTTP", port 80

Upload the final output Screenshot

Assignment 12

Exposing an Application with a Service:

Create a YAML file (service.yaml) that defines a Service to expose your application. Specify the Service type (e.g., LoadBalancer, NodePort, ClusterIP), ports, and target selectors.

Apply the Service to the cluster using `kubectl apply -f service.yaml`.

Access the exposed application using the appropriate service endpoint (e.g., LoadBalancer external IP, NodePort, ClusterIP).

Upload the final output Screenshot

Assignment 13

Configure email notifications

Description: Configure Jenkins to send email notifications when a build fails.

Instructions:

Upload the final output Screenshot

Assignment 14

Setting up a Continuous Integration (CI) Pipeline: Configure a CI pipeline using tools like Jenkins, CircleCI, or AWS CodePipeline to automate the building, testing, and deployment of your application code whenever changes are pushed to the repository.

Upload the final output Screenshot

Assignment 15

Scenario:

Your team is working on an application that requires scalability and high availability. How would you architect the application using AWS services and DevOps best practices?

Question: Explain how you would design an architecture that leverages AWS services to ensure scalability and high availability for your application, following DevOps principles.