SOFTWARE ENGINEERING LAB

EXERCISE – 8

TOPIC – 2

PROJECT DEPLOYMENT IN THE AWS CLOUD USING EC2 INSTANCE

In this exercise, we will be:

- Launch a virtual server (EC2 instance) on AWS.
- Install essential tools like Docker, Git, and Nano.
- Create and deploy a simple web application using Docker.
- Access the application online.
- Clean up resources to avoid unnecessary charges.

Note: At every step take screenshots and save in a document

Step 1: Log in to AWS and Go to EC2

In this step, we will log in to our AWS account and access the EC2 service.

1. Log in to your AWS account.
2. On the AWS homepage, click **Services**, then choose **EC2** under **Compute**.

Step 2: Launch an EC2 Instance

Here, we will set up a virtual server to host our web application.

1. Click **Launch Instance**.
2. Configure the settings as follows:
    - **Name**: Enter a name like "MyWebServer" to identify your server.
    - **Application and OS**: Choose **Ubuntu (Free Tier Eligible)**.

- **Instance Type**: Select **t2.micro** (1 CPU, 1 GB RAM).
- **Key Pair**: Create a new key pair, download the `.pem` file, and save it securely.
- **Network**: Enable **Allow HTTP/HTTPS traffic** to make your website accessible.
- **Storage**: Use the default 8 GB.

3. Click **Launch Instance** and wait until the status changes to "Running."

Step 3: Connect to the EC2 Instance

In this step, we will connect to our virtual server.

1. Select your instance, click **Connect**, and copy the **SSH command**.
2. Open **PowerShell** (Windows) or **Terminal** (Mac/Linux) on your computer.
3. Navigate to the folder where your `.pem` file is saved using the `cd` command.
4. Paste the SSH command and press Enter. Type "yes" if prompted.

Step 4: Prepare the Instance

Now, we will prepare the server by installing required tools.

1. **Update the system** to ensure all software is up to date:

```
sudo apt update
```

2. **Install Docker** to package and run our web application:

```
sudo apt-get install docker.io
```

3. **Install Git** to manage and download code:

```
sudo apt install git
```

4. **Install Nano** for editing files directly on the server:

```
sudo apt install nano
```

Step 5: Create Your Web Application

In this step, we will build a simple web page and upload it to GitHub.

1. On your computer, create a file named `index.html` and add the following content:

```
<html>
<head><title>My Webpage</title></head>
<body><h1>Hello from AWS!</h1></body>
</html>
```

2. Initialize Git in the file's folder:

```
git init
git add .
git commit -m "First commit"
```

3. Create a GitHub repository, copy its HTTPS URL, and upload your file:

```
git remote add origin <Your_Repo_URL>
git push -u origin main
```

Step 6: Deploy the Web Application Using Docker

Here, we will deploy the web application to the EC2 instance.

1. On the EC2 instance, clone your GitHub repository:

```
git clone <Your_Repo_URL>
```

2. Create a `Dockerfile` in the project folder using Nano:

```
nano Dockerfile
```

Add the following content:

```
FROM nginx:alpine
COPY . /usr/share/nginx/html
```

Save the file by pressing **Ctrl + O**, then Enter, and exit Nano with **Ctrl + X.**

3. Build and run the Docker container to serve the web application:

```
sudo docker build -t my-web-app .
sudo docker run -d -p 80:80 my-web-app
```

Step 7: Access Your Web Application

In this step, we will view the deployed web page online.

1. Copy the **Public IP Address** of your EC2 instance from the AWS console.
2. Paste it into your browser (e.g., **http://<Public_IP>).**
3. You'll see your web page with the message "Hello from AWS!" displayed.

Step 8: Clean Up

Finally, we will clean up resources to avoid any charges.

1. Stop the running Docker container:

```
sudo docker ps
sudo docker stop <Container_ID>
```

2. Terminate the EC2 instance in the AWS console by selecting it, clicking **Instance State**, and choosing **Terminate Instance**.