Assignment 10

Deploy a project from github to EC2 by creating new security group and user data.

Objective: To **deploy a Node.js project from GitHub** to an EC2 instance using a **custom security group** and **user data script** during launch.

🛡 Part 1: Create a New Security Group

◆ Step 1: Open EC2 Dashboard

- Log in to the AWS Management Console.
- Navigate to **EC2 Dashboard** (under the "Services" menu).

◆ Step 2: Delete Existing (Non-default) Security Groups

- Go to **Network & Security > Security Groups** in the EC2 menu.
- Select any **non-default security groups**.
- Click **Actions > Delete Security Groups** (You cannot delete the default one).

◆ Step 3: Create a New Security Group

- Click on **"Create security group"**.
- Fill in the following:
  - **Security group name:** SnehaSecurityGroup
  - **Description:** A brief description (e.g., Security group for Node.js app)
  - **VPC:** Leave it as the default.

◆ Step 4: Add Inbound Rules

Click **"Add Rule"** and input the following:

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
Custom TCP	TCP	0-65535	0.0.0.0/0

⚠ **Note:** Be cautious using 0.0.0.0/0 as it allows access from anywhere. For production, restrict this.

◆ Step 5: Create Security Group

- Click **"Create security group"** to save it.

Assignment 10

🚀 Part 2: Launch EC2 Instance and Deploy App

◆ Step 1: Launch New Instance

- Go to **EC2 Dashboard > Instances > Launch Instance**.

◆ Step 2: Instance Configuration

- **Name:** myinstance21
- **Application and OS Image (AMI):** Choose **Ubuntu (Free tier eligible)**.
- **Instance type:** t2.micro

◆ Step 3: Key Pair

- Under **Key pair (login)**, choose your existing key pair (snehaa1234) or:
  - Click **Create new key pair**
  - Download the .pem file for SSH access.

◆ Step 4: Network Settings

- Click **Edit** in the Network settings section.
- Choose **"Select existing security group"**
- Select SnehaSecurityGroup created earlier.

◆ Step 5: Configure User Data (Auto-deploy app)

Scroll to **Advanced Details > User data**, and paste the following script:

Replace the GitHub repo path with your actual repository
e.g., https://github.com/itsmesneha/SNEHAREPO

```
#!/bin/bash
apt-get update
apt-get install -y nginx
systemctl start nginx
systemctl enable nginx
apt-get install -y git
curl -sL https://deb.nodesource.com/setup_18.x | sudo -E bash -
apt-get install -y nodejs
git clone https://github.com/itsmesneha/SNEHAREPO
cd SNEHAREPO
npm install
node index.js
```

✅ This script will:

    - Install Nginx and Git
    - Set up Node.js environment
    - Clone your GitHub repo
    - Install dependencies
    - Start the app

◆ Step 6: Launch Instance

- Click **Launch instance** and wait until it is in the **running state**.

🌐 Part 3: Test the Deployment

◆ Step 1: Open Instance Summary

- Go to **Instances**, click on your newly created instance name.

◆ Step 2: Get Public IPv4

- Copy the **Public IPv4 address** from the summary panel.

◆ Step 3: Access App via Browser

- Paste the address into your browser (e.g., `http://<your-ip-address>`)
- If your app runs on a port (e.g., 3000), try `http://<your-ip>:3000`

You should see your deployed application running!

□ Notes:

- Ensure your GitHub repository is **public**:
    - Go to **Settings > Change repository visibility > Public**
- If using a custom port (other than 80), make sure it is open in the **Security Group Inbound Rules**.