This is a classic cloud computing practical. For your POE (Practical Oral Exam), the examiner will look closely at two things: **Security Group configuration** (opening port 80) and **file permissions**.

Here is the step-by-step guide to launching a web app on an AWS EC2 Ubuntu instance using Apache.

Phase 1: Launching the Instance (AWS Console)

1. **Login to AWS Console** and go to **EC2 Dashboard**.
2. Click **Launch Instance**.
3. **Name:** Give it a name (e.g., "MyWebServer").
4. **OS Image:** Select **Ubuntu** (e.g., Ubuntu Server 22.04 LTS).
5. **Instance Type:** Select **t2.micro** (Free Tier eligible).
6. **Key Pair (Login):**
   - Click "Create new key pair".
   - Name it my-web-key.
   - Select .pem format.
   - **Download the file** and keep it safe.
7. **Network Settings (Crucial Step):**
   - Under **Firewall (security groups)**, select "Create security group".
   - Check **Allow SSH traffic from Anywhere**.
   - Check **Allow HTTP traffic from the internet**. (This opens Port 80).
8. Click **Launch Instance**.

Phase 2: Connecting via SSH

1. Open your terminal (Git Bash on Windows or standard Terminal on Mac/Linux).
2. Navigate to the folder where your .pem file is downloaded (e.g., cd Downloads).
3. **Fix Key Permissions** (AWS rejects keys that are too open):

   Bash
   ```
   chmod 400 my-web-key.pem
   ```

4. **Connect to the Instance** (Replace <Public-IP> with your instance's IP from the AWS console):

   Bash
   ```
   ssh -i "my-web-key.pem" ubuntu@<Public-IP>
   ```

(Type yes when asked about fingerprints).

Phase 3: Installing & Configuring Apache

Once you are logged into the Ubuntu terminal:

1. **Update the Repository:**
   Bash
   ```
   sudo apt update
   ```

2. **Install Apache Web Server:**
   Bash
   ```
   sudo apt install apache2 -y
   ```

3. **Start and Enable Apache** (Ensures it runs if the server reboots):
   Bash
   ```
   sudo systemctl start apache2
   sudo systemctl enable apache2
   ```

4. **Verify Status:**
   Bash
   ```
   sudo systemctl status apache2
   ```

   (It should say "active (running)" in green).

Phase 4: Deploying Your Web App

Apache serves files from the /var/www/html directory.

1. **Navigate to the web directory:**
   Bash
   ```
   cd /var/www/html
   ```

2. **Remove the default index page:**
   Bash
   ```
   sudo rm index.html
   ```

3. **Create your own Web App:**
   Bash

```
sudo nano index.html
```

4. **Paste your HTML code:**

```html
<!DOCTYPE html>
<html>
<head>
  <title>My AWS App</title>
  <style>body { font-family: sans-serif; text-align: center; padding: 50px; }</style>
</head>
<body>
  <h1>Hello from AWS EC2!</h1>
  <p>This web app was deployed using Apache on Ubuntu.</p>
</body>
</html>
```

5. **Save and Exit:** Press Ctrl + O, Enter, then Ctrl + X.

Phase 5: Testing

1. Go to your browser.
2. Enter the **Public IP** of your instance (e.g., http://54.123.45.67).
3. You should see your "Hello from AWS EC2!" page.

❓ Possible Viva Questions for This Task

- **Q: "I entered the IP but the site isn't loading. What's wrong?"**
  - **A:** "Most likely, the **Security Group** does not allow inbound traffic on **Port 80 (HTTP)**. By default, AWS only opens Port 22 (SSH)."
- **Q: "Why did you use sudo to edit the file?"**
  - **A:** "The /var/www/html directory is owned by the root user. As the default ubuntu user, I don't have write permissions there, so I need sudo (superuser do) to create or edit files."
- **Q: "What command checks if Apache is running?"**
  - **A:** sudo systemctl status apache2.