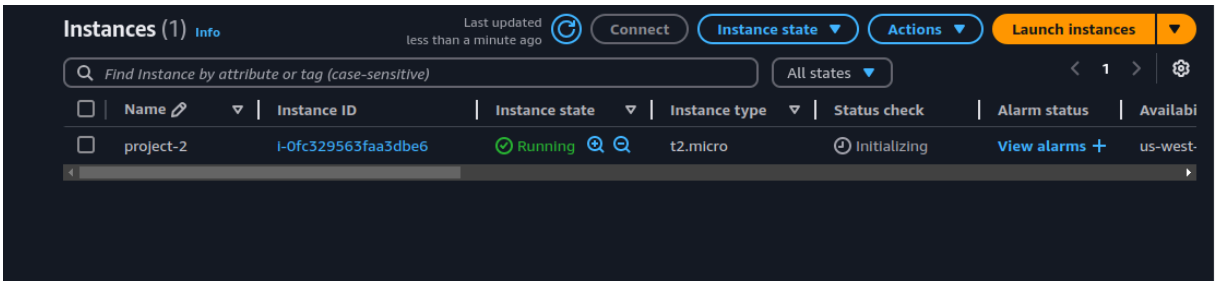
DOCUMENTATION

Run HTML/CSS Application on EC2 with Ubuntu by using Apache and SCP

Step 1: Launch an Ubuntu EC2 Instance

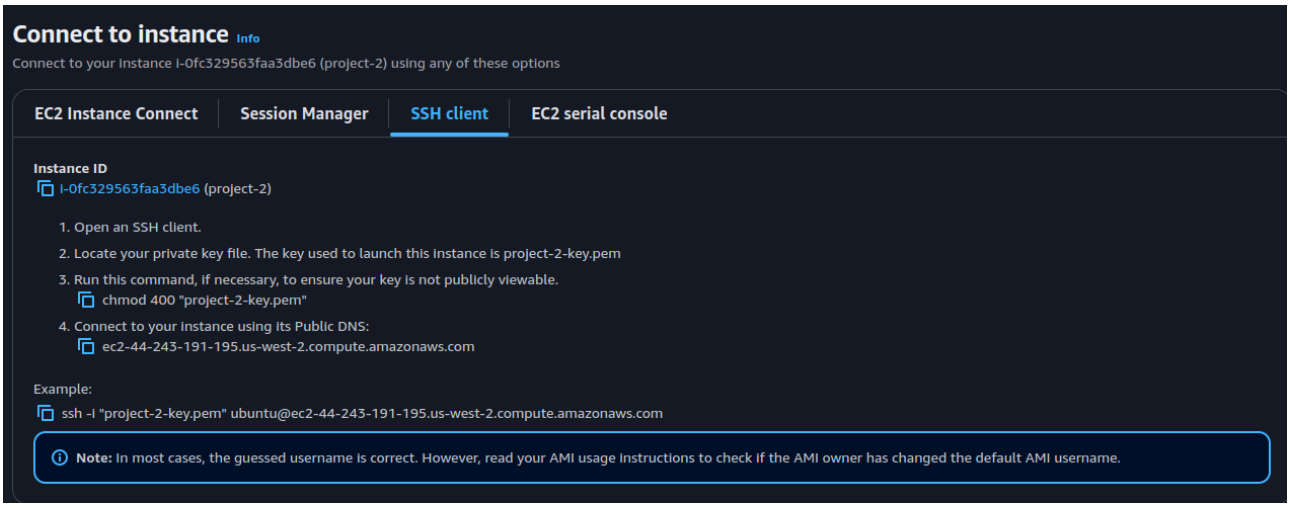
1. **Log in to the AWS Management Console.**
2. **Launch an Instance:**
   - Select **Ubuntu Server** as the AMI (e.g., Ubuntu 24.04).
   - Choose an instance type like `t2.micro` (free tier eligible).
   - Configure a **Security Group** to allow:
     - **HTTP (Port 80)** from 0.0.0.0/0.
     - **SSH (Port 22)** from your IP (or 0.0.0.0/0 for testing, but this is insecure).
   - Assign or create a **key pair** for SSH access.



Step 2: Connect to the Ubuntu Instance

1. **SSH into the instance:**

```
ssh -i "your-key-file.pem" ubuntu@<INSTANCE_PUBLIC_IP>
```




```
s$ sudo ssh -i "project-2-key.pem" ubuntu@ec2-44-243-191-195.us-west-2.compute.amazonaws.com
```

```
  System load:  0.1               Processes:             106
  Usage of /:   22.9% of 6.71GB   Users logged in:       0
  Memory usage: 21%               IPv4 address for enX0: 172.31.7.98
  Swap usage:   0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status


The list of available updates is more than a week old.
To check for new updates run: sudo apt update


The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-7-98:~$
```

2. **Update the system:**

```
sudo apt update
sudo apt upgrade -y
```

```
ubuntu@ip-172-31-7-98:~$     sudo apt update
    sudo apt upgrade -y
```

Step 3: Install Apache Web Server

1. **Install Apache:**

```
sudo apt install apache2 -y
```

```
ubuntu@ip-172-31-7-98:~$     sudo apt install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap
```

2. **Start and Enable Apache:**

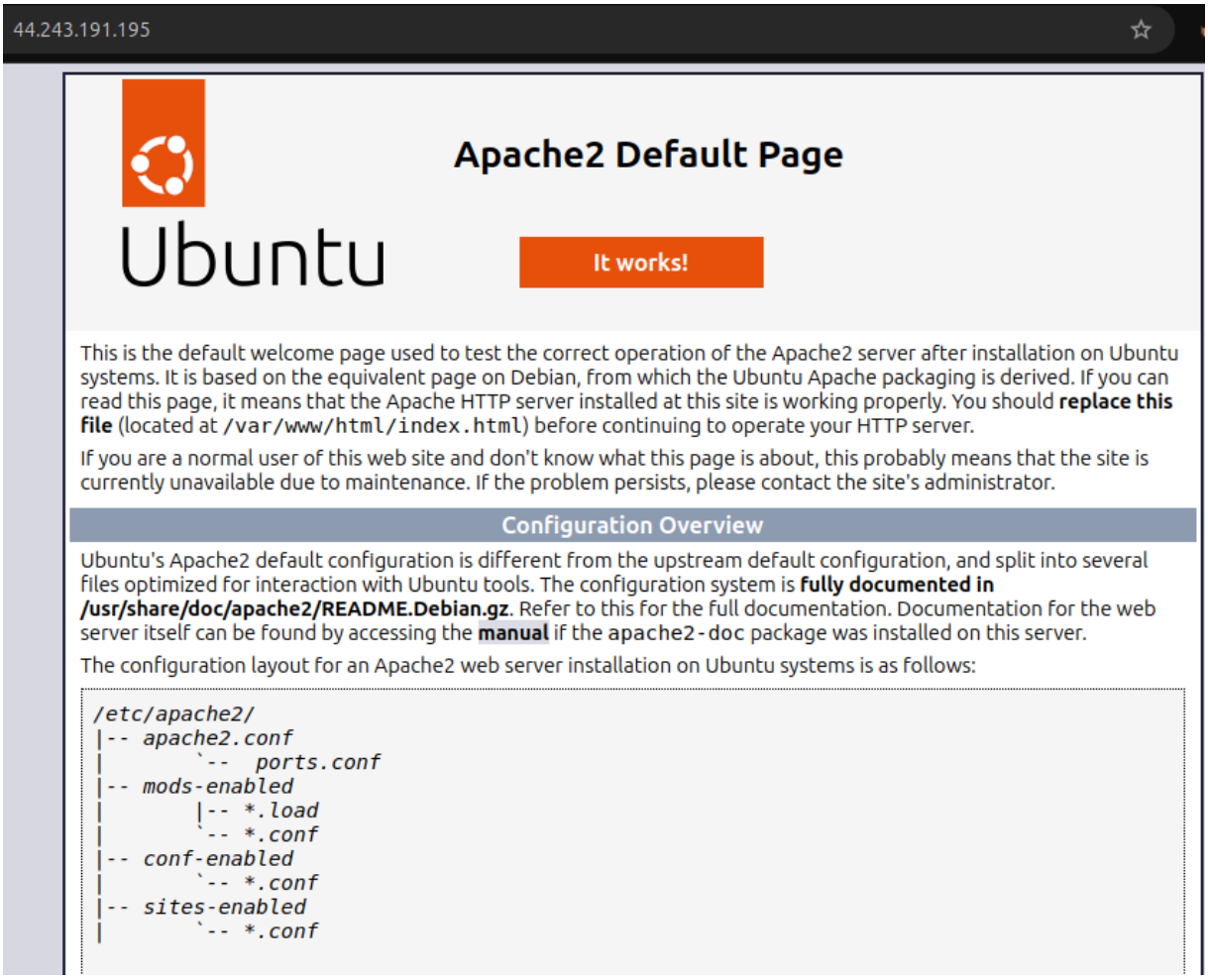
```
sudo systemctl start apache2
sudo systemctl enable apache2
```

```
ubuntu@ip-172-31-7-98:~$ sudo systemctl start apache2
ubuntu@ip-172-31-7-98:~$ sudo systemctl enable apache2
Synchronizing state of apache2.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable apache2
ubuntu@ip-172-31-7-98:~$
```

3. **Verify Apache Installation:**
   - Open your browser and visit:

```
http://<INSTANCE_PUBLIC_IP>
```

- You should see the Apache default "It works!" page.



Step 4: Deploy Your HTML/CSS Application

1. **Upload HTML/CSS Files to the Instance:**
   - Use `scp` to transfer files:

https://github.com/MuzakkirHossainMinhaz/panda-commerce.git

```
    ```bash
scp -i "your-key-file.pem" path/to/your/files/*
ubuntu@<INSTANCE_PUBLIC_IP>:/tmp
    ```
```

```
   ![[9-scp.png]]
```

2. **Move Files to Apache's Web Directory:**

```
- The default web directory for Apache on Ubuntu is `/var/www/html/`. Move your files there:
```

```
sudo mv /tmp/* /var/www/html/
```

```
         ![[10-mv.png]]
```

3. **Set Proper Permissions:**

```
- Ensure Apache has the necessary permissions to serve the files:

    ```shell
    sudo chmod -R 755 /var/www/html/
    ```

     ![[11-chmod.png]]
```

4. **Test Your Application:**

```
- Visit your public IP in the browser:

    ```
    http://<INSTANCE_PUBLIC_IP>
    ```

- Your HTML/CSS application should now load.

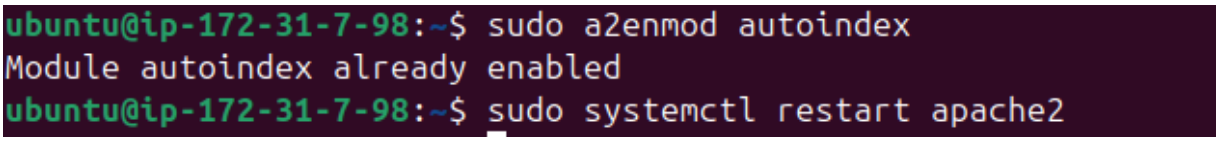
    ![[12-webpage.png]]
```

Step 5: Customize Apache Configuration (Optional)

1. **Enable Directory Indexing (Optional):**

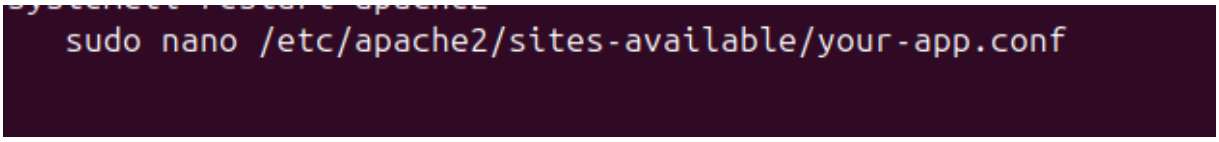
- If your application contains subdirectories and you want directory browsing, enable indexing:

```
sudo a2enmod autoindex
sudo systemctl restart apache2
```



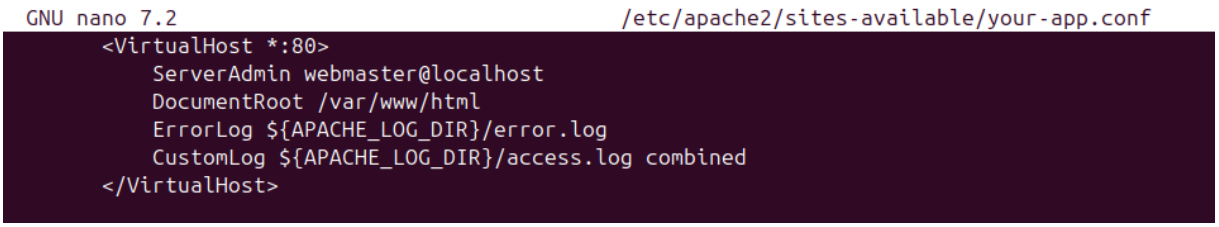
2. **Custom Virtual Host Configuration (Optional):**
   - If you plan to host multiple applications or use a custom domain, create a new virtual host file:

```
sudo nano /etc/apache2/sites-available/your-app.conf
```



   - Add the following configuration:

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```



   - Enable the new configuration:

```
sudo a2ensite your-app.conf
sudo systemctl reload apache2
```

```
ubuntu@ip-172-31-7-98:~$         sudo a2ensite your-app.conf
        sudo systemctl reload apache2
Enabling site your-app.
To activate the new configuration, you need to run:
  systemctl reload apache2
ubuntu@ip-172-31-7-98:~$ sudo systemctl reload apache2
ubuntu@ip-172-31-7-98:~$
```

Step 6: Secure and Maintain the Instance

1. **Limit SSH Access:**
   - Edit the security group to allow SSH only from your IP address.
2. **Apply System Updates Regularly:**

   ```
   sudo apt update && sudo apt upgrade -y
   ```

3. **Monitor Logs for Errors:**
   - Access logs: `/var/log/apache2/access.log`
   - Error logs: `/var/log/apache2/error.log`

Conclusion

By following this guide, you've successfully deployed an HTML/CSS application on an EC2 instance using Ubuntu and Apache. You can now access your web application from the browser using your EC2 instance's public IP.

This **README.md** gives clear instructions for setting up the EC2 instance, installing Apache, uploading files using SCP, and testing the deployment. Feel free to adapt it based on the specifics of your project.
