Placement Empowerment Program

Cloud Computing and DevOps Centre

Host a Static Website on a Cloud VM Install Apache on your cloud VM and host a simple HTML website.

Name: Joan Festina J

Department : ECE

Introduction

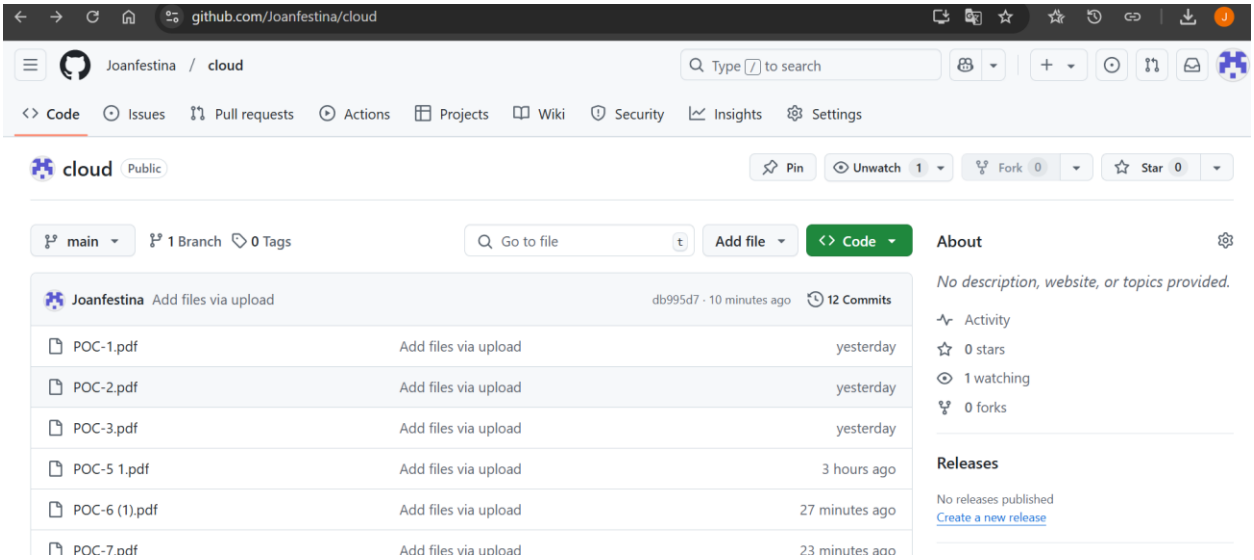
A static website serves pre-written HTML, CSS, and JavaScript files to the end user without requiring server-side processing. Hosting such websites on a cloud-based Virtual Machine (VM) has become a preferred choice for individuals and businesses due to its flexibility, scalability, and cost-effectiveness. By leveraging the cloud, developers can quickly deploy websites accessible from anywhere in the world.

Objectives

- Learning Cloud Computing Fundamentals: Understanding how virtual machines operate in a cloud environment.
- Practical Web Hosting Skills: Gaining hands-on experience in setting up and configuring web servers like Apache or Nginx.
- Website Deployment: Successful

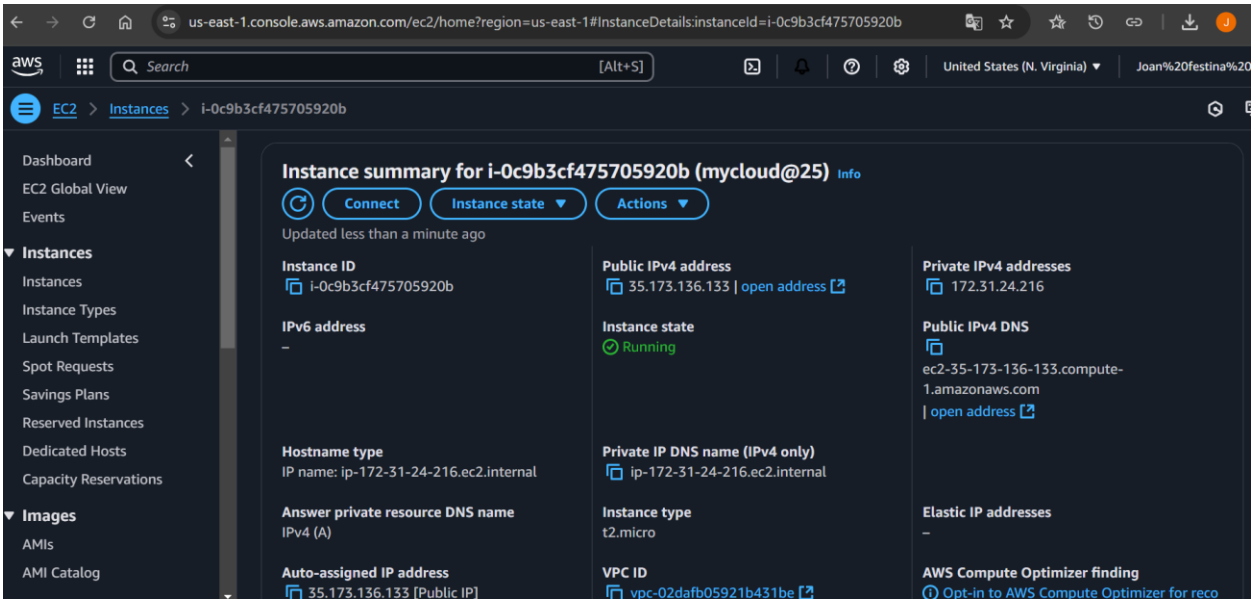
Step by Step Overview

1. Have an HTML file (with any related assets like CSS/JavaScript) that you want to host in your GitHub repository.



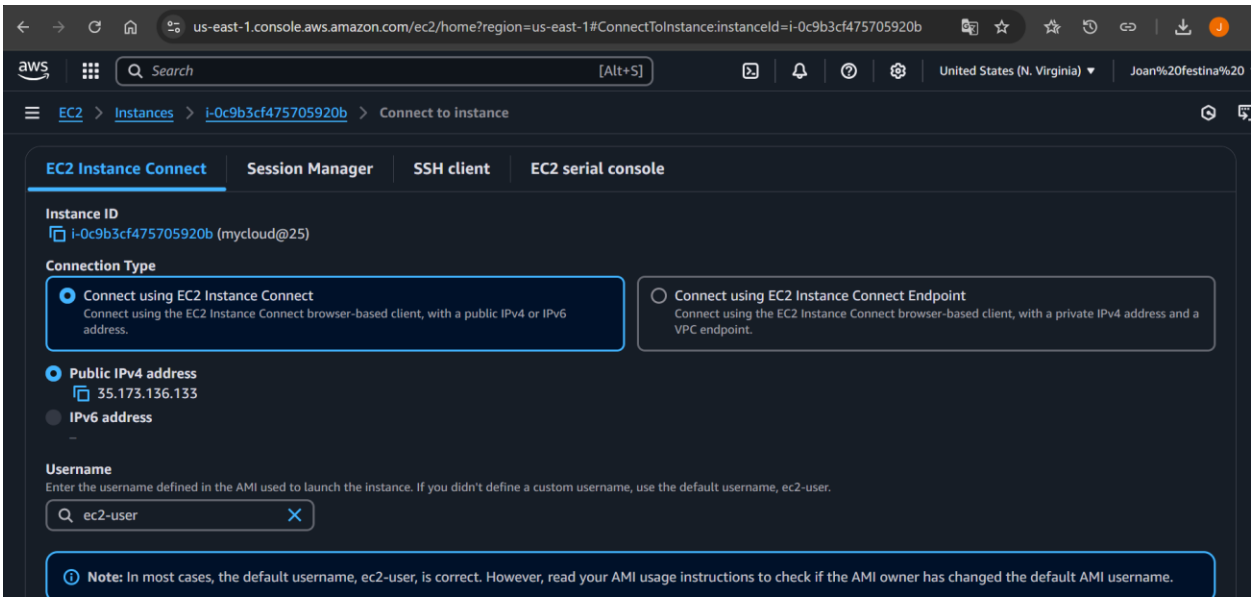
2. Create a EC2 server

Launch an EC2 instance, select Ubuntu as the OS, configure security groups to allow all network traffic, create a key pair (e.g., new.pem), and download it for SSH access.



3. Connect

Click the 'Connect' option on your launched instance, go to the SSH client section, and copy the command provided under the 'Example' section.



4. Open Power Shell

```
PS C:\Users\nandh> cd Downloads
PS C:\Users\nandh\Downloads> ssh -i "nan.pem" ubuntu@ec2-13-126-9-173.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-13-126-9-173.ap-south-1.compute.amazonaws.com (13.126.9.173)' can't be established.
ED25519 key fingerprint is SHA256:953ZhQ5BRv/LUgNlV04BFqLAKXZOi7tc6B3rgV4gTVk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-126-9-173.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 6.8.0-1021-aws x86_64)
```

5. Run the following commands

Run the command **sudo apt update** to update the package list.

```
ubuntu@ip-172-31-3-243:~$  sudo apt update
```

Run the command **sudo apt upgrade**, and press 'Y' to confirm and continue the upgrade process.

```
ubuntu@ip-172-31-3-243:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

Install the Apache server by running the command **sudo apt install apache2**, and press 'Y' to confirm the installation

```
ubuntu@ip-172-31-3-243:~$ sudo apt install apache2
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 libaprutil1t64
  liblua5.4-0 ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapr1t64 libaprutil1-dbd-sqlite3 libaprutil1-ldap libaprutil1t64
  liblua5.4-0 ssl-cert
0 upgraded, 10 newly installed, 0 to remove and 0 not upgraded.
Need to get 2084 kB of archives.
After this operation, 8094 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Insert your files by running the command **git clone** to clone your repository containing the website files.

```
ubuntu@ip-172-31-3-243:~$ git clone https://github.com/vijayanandana25/my-static-website
Cloning into 'my-static-website'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
```

Run the command **cd /var/www/html** to navigate to the web server's root directory, then type ls to verify that your HTML files from the GitHub repository are present.

```
ubuntu@ip-172-31-3-243:/var/www/html$ ls
index.html
```

6. Execute

Copy the **Public IPv4 DNS** from the instance details page in the EC2 console and paste it in URL.

Outcome:

With this PoC, we learn to access your static website live on the web using the EC2 instance's Public IPv4 DNS.