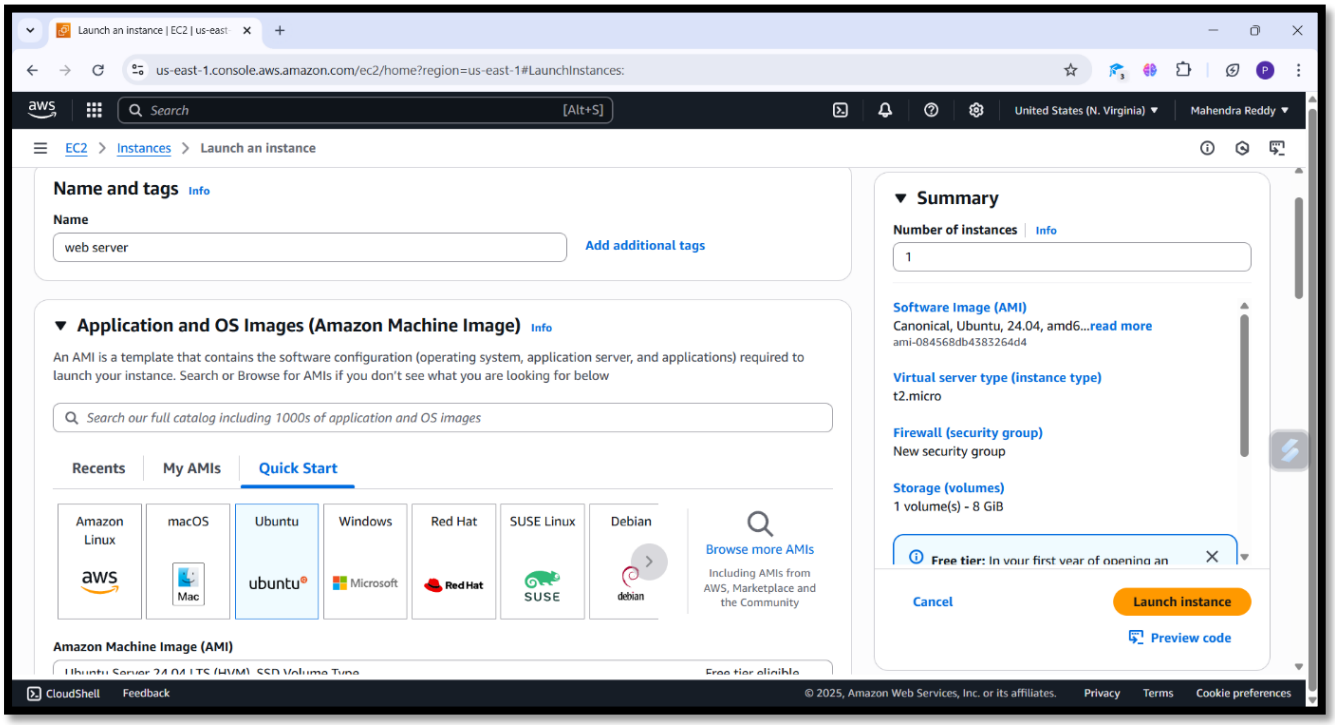
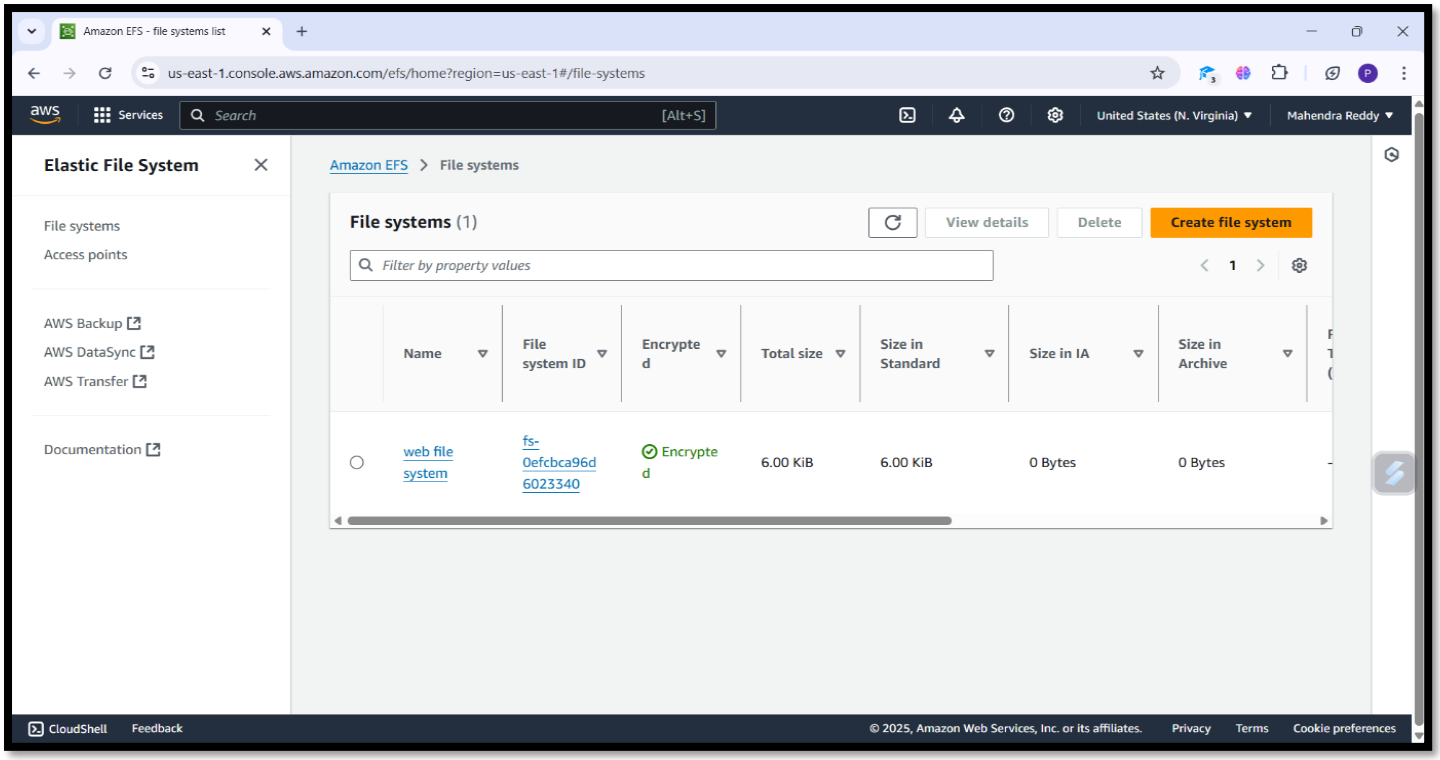
Secure and Scalable Static Website Architecture on AWS

1. Ubuntu:

- **Steps to create an EC2 instance (Ubuntu):**
  - Log in to AWS Console.
  - Navigate to EC2 > Launch Instance.
  - Select Ubuntu Server 20.04 LTS.
  - Choose t2.micro .
  - Configure network settings (VPC, Subnet).
  - Add storage and tags as needed.
  - Add security group with HTTP, HTTPS, and SSH open.
  - Launch with a key pair.



- **Creating the EFS File System:**
  - Go to EFS > Create File System.
  - Choose the same VPC as the EC2 instance.
  - Configure mount targets for availability zones.
  - Finish creation.



- **Installing NFS utilities and mounting EFS:**

  sudo apt update

  sudo apt install -y nfs-common

  sudo mkdir /var/www/html

  sudo mount -t nfs4 -o nfsvers=4.1 <EFS-DNS>:/ /var/www/html(basic Syntax)

- **Attaching EFS via Console:**
  - Select EC2 instance.
  - Use the "Attach" command from EFS console.

- **Screenshot of EFS mount verification:**
  - Use df -Th and take a screenshot showing /var/www/html.

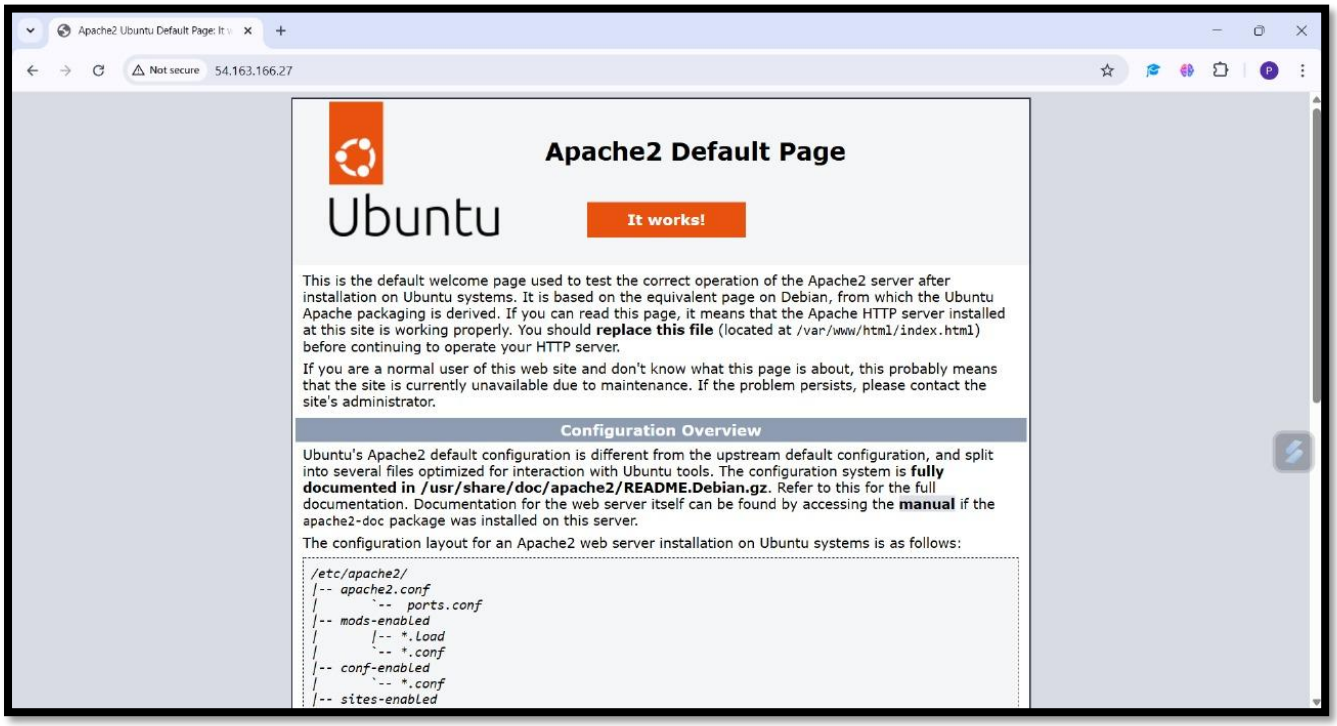
```
root@ip-172-31-29-175: ~
Setting up keyutils (1.6.3-3build1) ...
Setting up nfs-common (1:2.6.4-3ubuntu5.1) ...

Creating config file /etc/idmapd.conf with new version

Creating config file /etc/nfs.conf with new version
info: Selecting UID from range 100 to 999 ...

info: Adding system user `statd' (UID 112) ...
info: Adding new user `statd' (UID 112) with group `nogroup' ...
info: Not creating home directory `/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target → /usr/lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target → /usr/lib/systemd/system/nfs-client.target.
auth-rpcgss-module.service is a disabled or a static unit, not starting it.
nfs-idmapd.service is a disabled or a static unit, not starting it.
nfs-utils.service is a disabled or a static unit, not starting it.
proc-fs-nfsd.mount is a disabled or a static unit, not starting it.
rpc-gssd.service is a disabled or a static unit, not starting it.
rpc-statd-notify.service is a disabled or a static unit, not starting it.
rpc-statd.service is a disabled or a static unit, not starting it.
rpc-svcgssd.service is a disabled or a static unit, not starting it.
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.4) ...
Scanning processes...
Scanning candidates...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

User sessions running outdated binaries:
 ubuntu @ session #8: sshd[1341]

No VM guests are running outdated hypervisor (qemu) binaries on this host.
root@ip-172-31-29-175:~# sudo mount -t nfs4 -o nfsvers=4.1,rsize=1048576,wsize=1048576,hard,timeo=600,retrans=2,noresvport 172.31.82.196:/ /var/www/html
root@ip-172-31-29-175:~# df -Th
Filesystem      Type   Size  Used Avail Use% Mounted on
/dev/root       ext4    19G  2.0G   17G  11% /
tmpfs           tmpfs  479M     0  479M   0% /dev/shm
tmpfs           tmpfs  192M  908K  191M   1% /run
tmpfs           tmpfs  5.0M     0  5.0M   0% /run/lock
/dev/xvda16     ext4   881M   79M  741M  10% /boot
/dev/xvda15     vfat   105M  6.1M   99M   6% /boot/efi
tmpfs           tmpfs   96M   12K   96M   1% /run/user/1000
172.31.82.196:/ nfs4   8.0E     0  8.0E   0% /var/www/html
root@ip-172-31-29-175:~#
```

- **Apache2 server page:**

sudo apt install apache2 -y

sudo systemctl start apache2

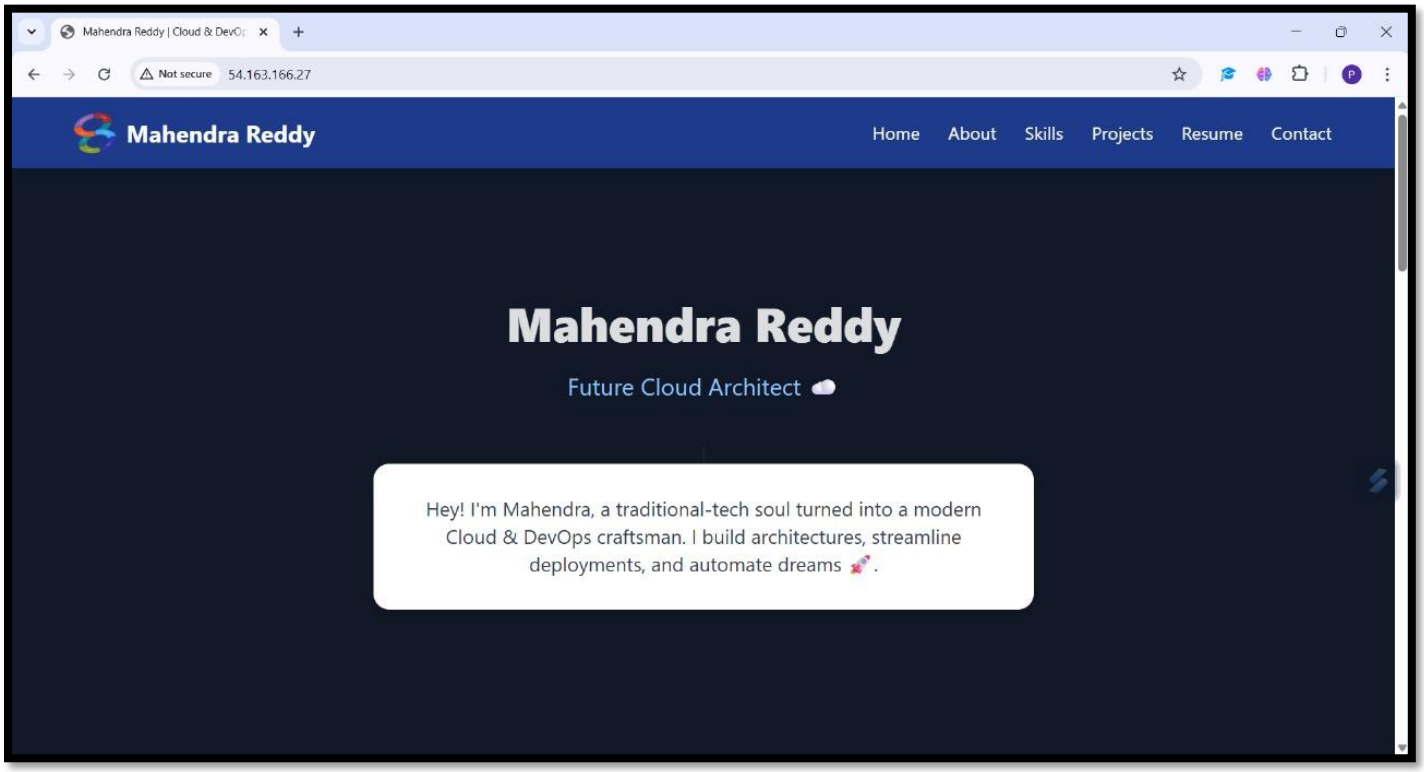
sudo systemctl enable apache2



➢ **Sample Web Page:**

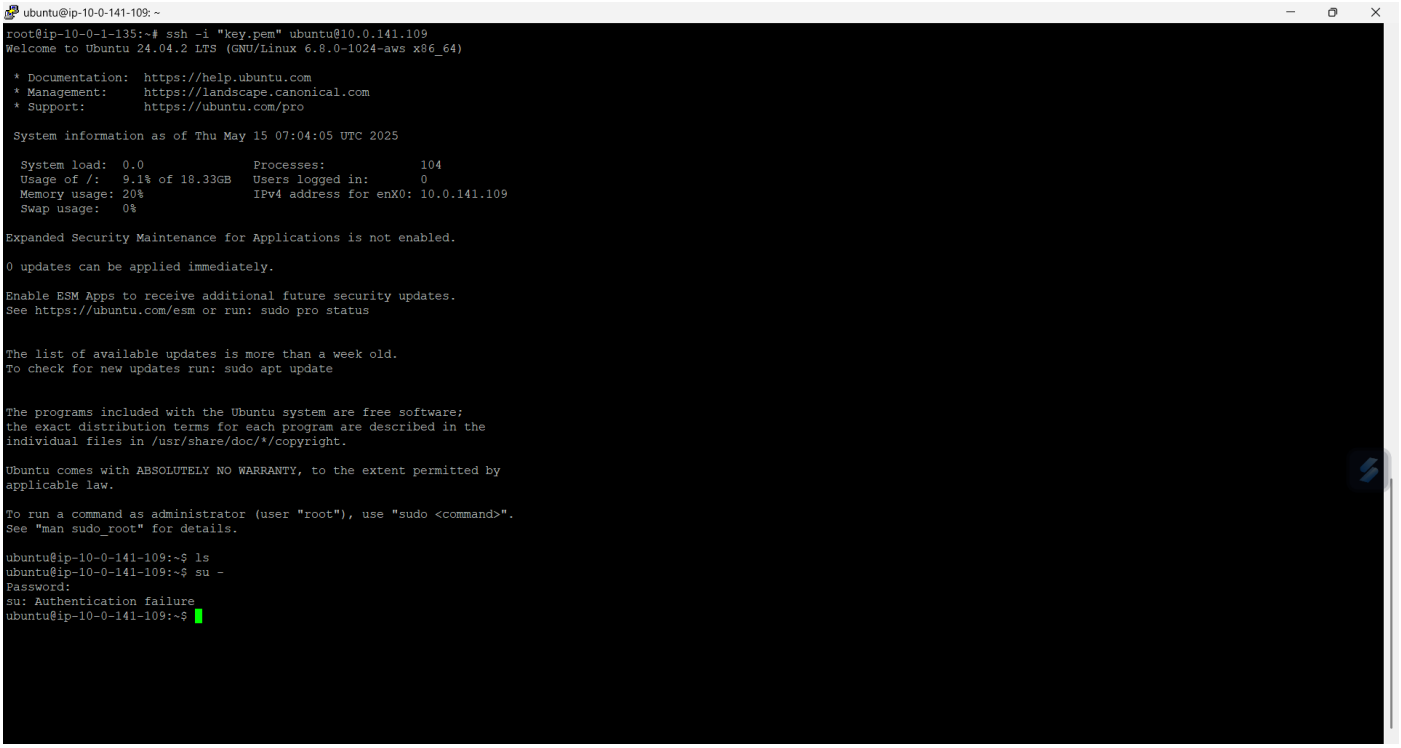
echo "<h1>Welcome to the linux world </h1>" | sudo tee /var/www/html/index.html

- Main website hosted:
- Access via EC2 Public IP.



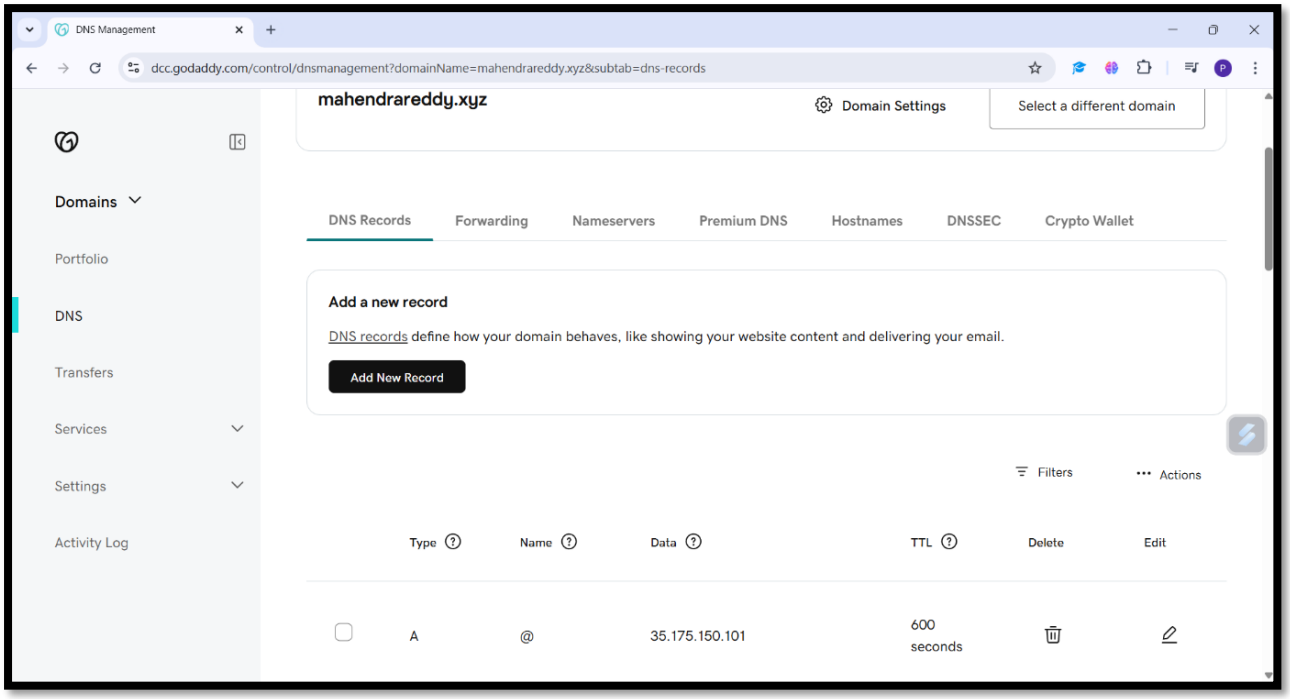
2. Securing Instance Using Bastion Host

- Creating a bastion host in a public subnet.
- The private instance in a private subnet.
- SSH agent forwarding or jump host SSH config (~/.ssh/config).
- Security group configurations.
- Example SSH command:
  ssh -i key.pem -A ubuntu@<private-ip> and then to private instance.

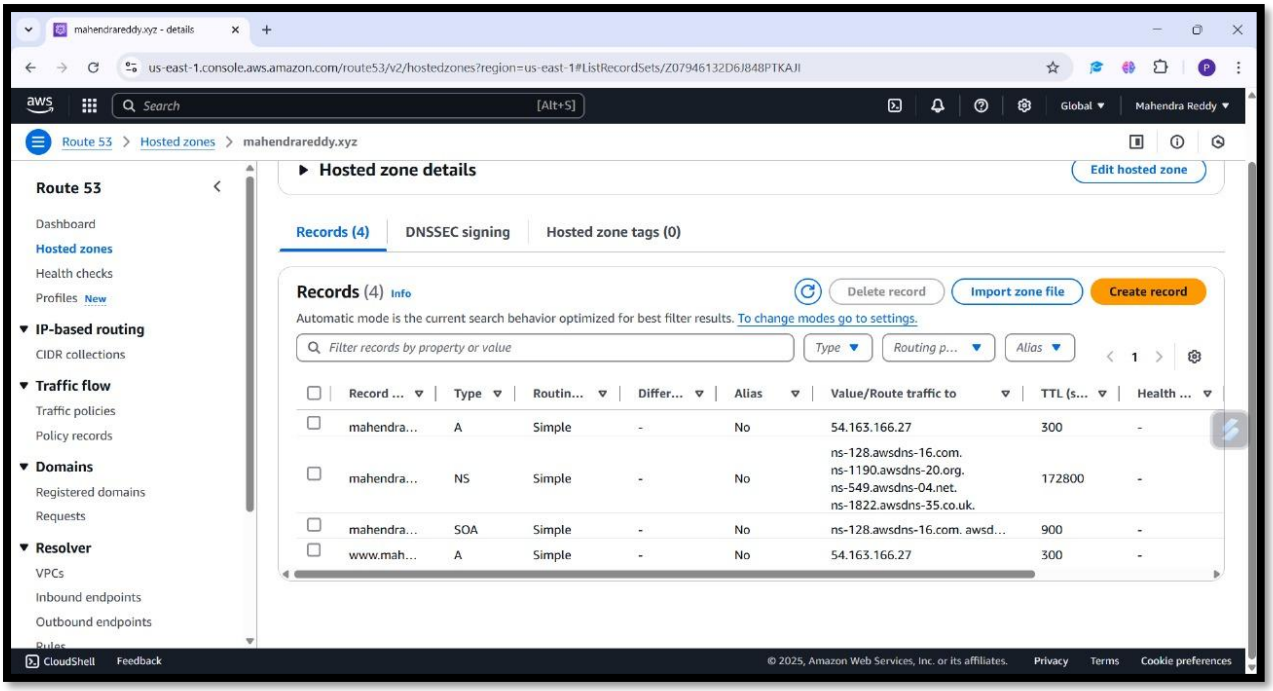


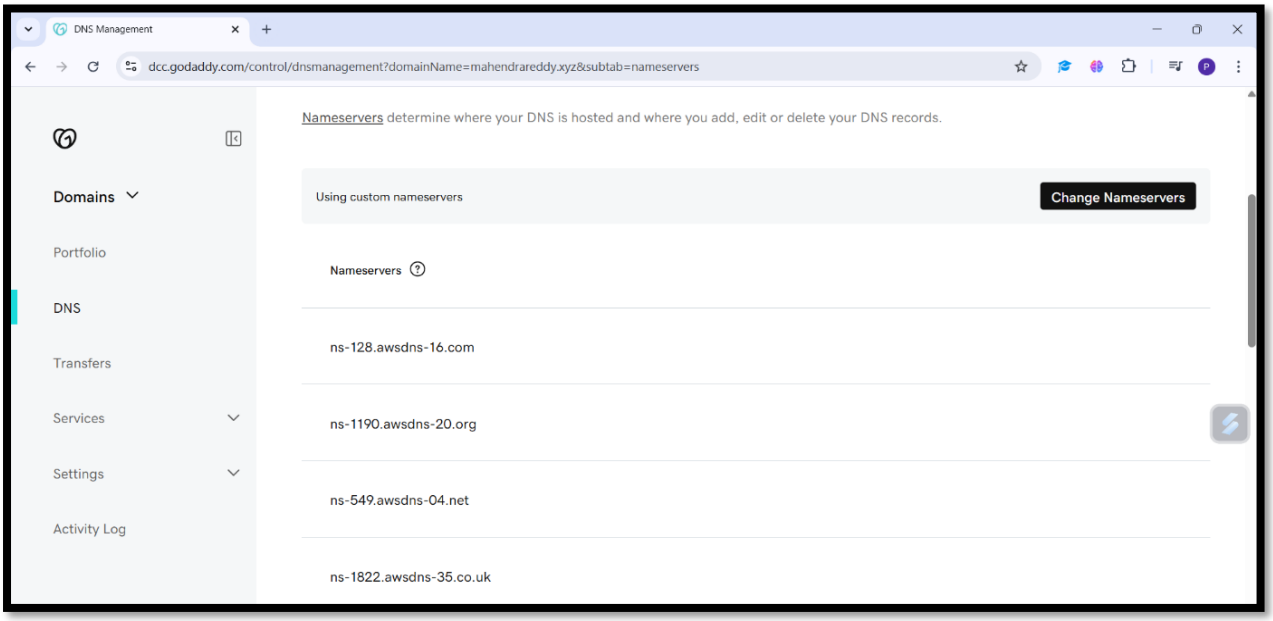
3. Configure Route 53 with Custom Domain

- Domain registration.

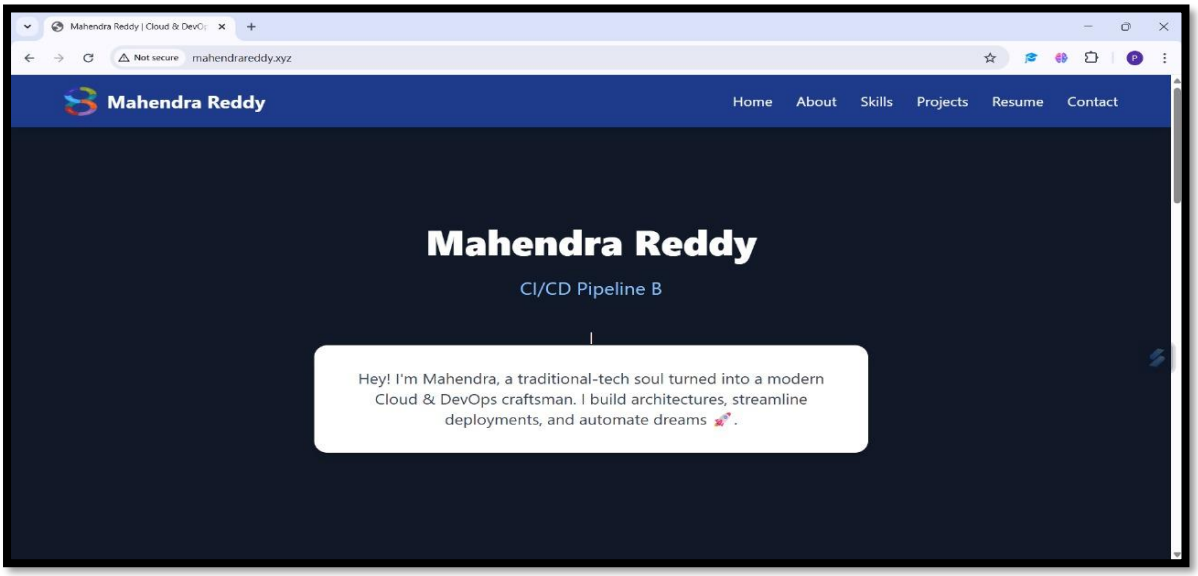


- Hosted zone creation in Route 53.
- Record Set (A) pointing to EC2.
- Screenshot of DNS settings.





- Hosted website using Domain Name:



4. Install SSL/TLS Certificate:-

➢ **Install SSL on Apache Manually (if no Load Balancer is used):**

• Install Certbot:

sudo apt update

sudo apt install certbot python3-certbot-apache -y

• Run Certbot to get and install a certificate:

sudo certbot --apache

• Follow prompts to:

– Choose the domain

– Redirect HTTP to HTTPS

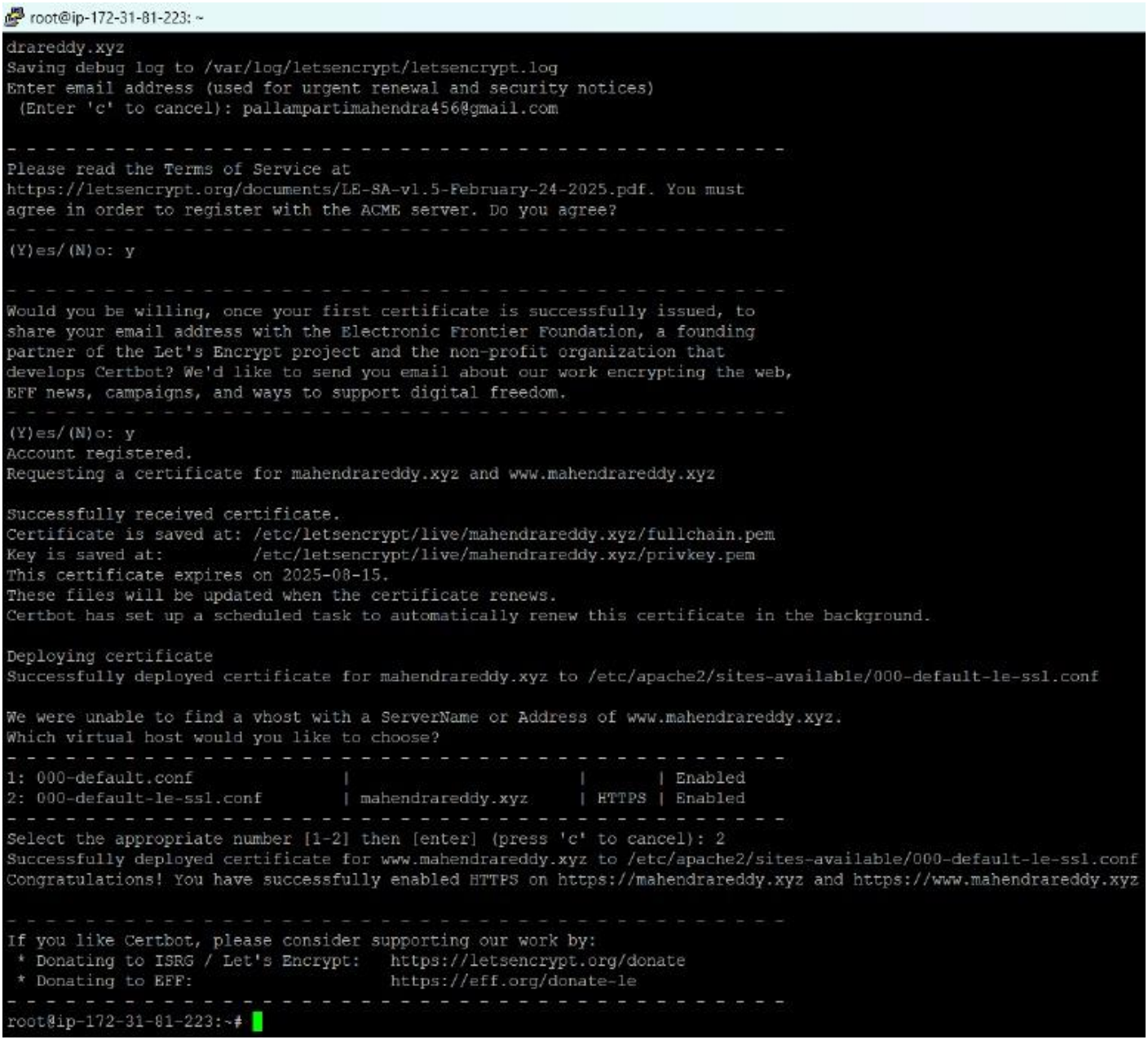
– Install certificate

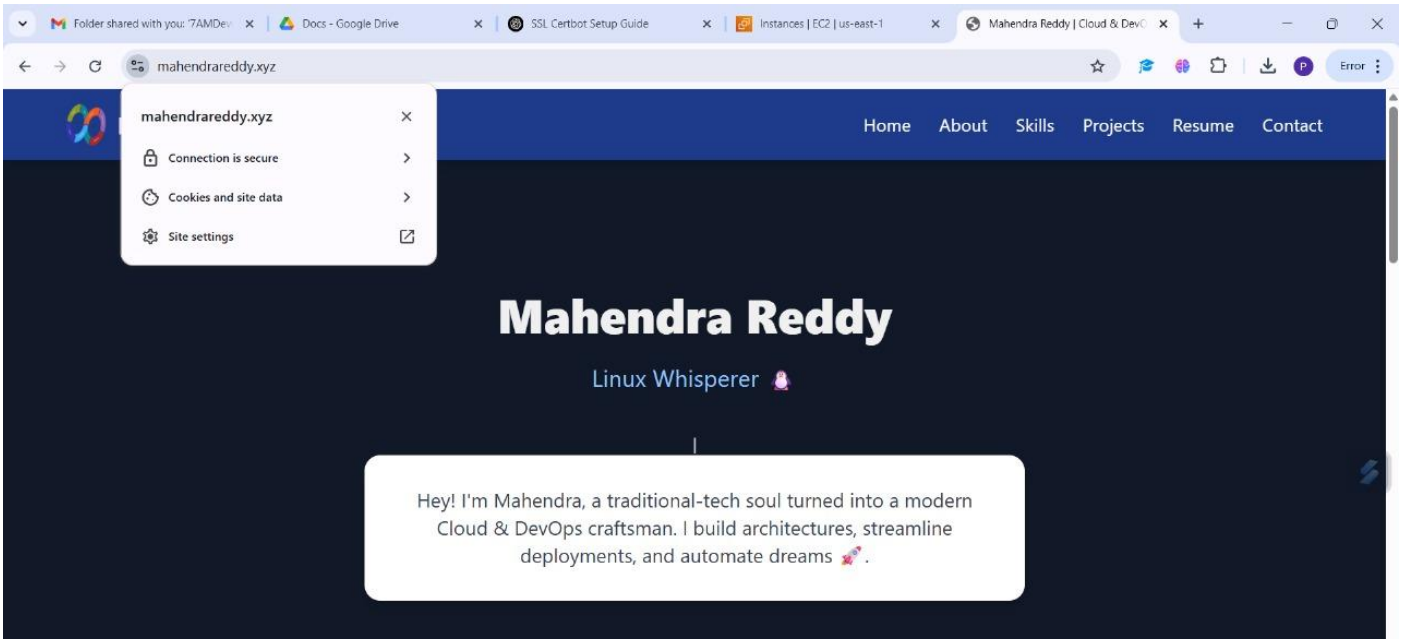
• Verify the certificate with:

sudo certbot certificates

• Set up auto-renewal (already included via systemd timer):

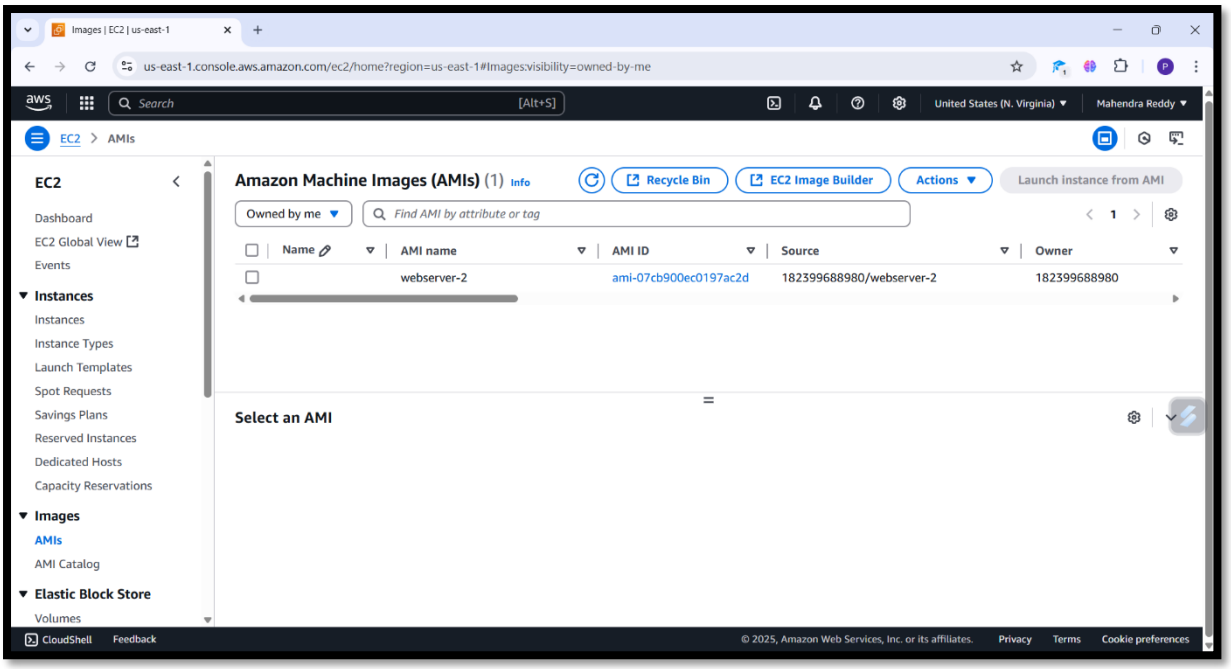
sudo systemctl status certbot.timer





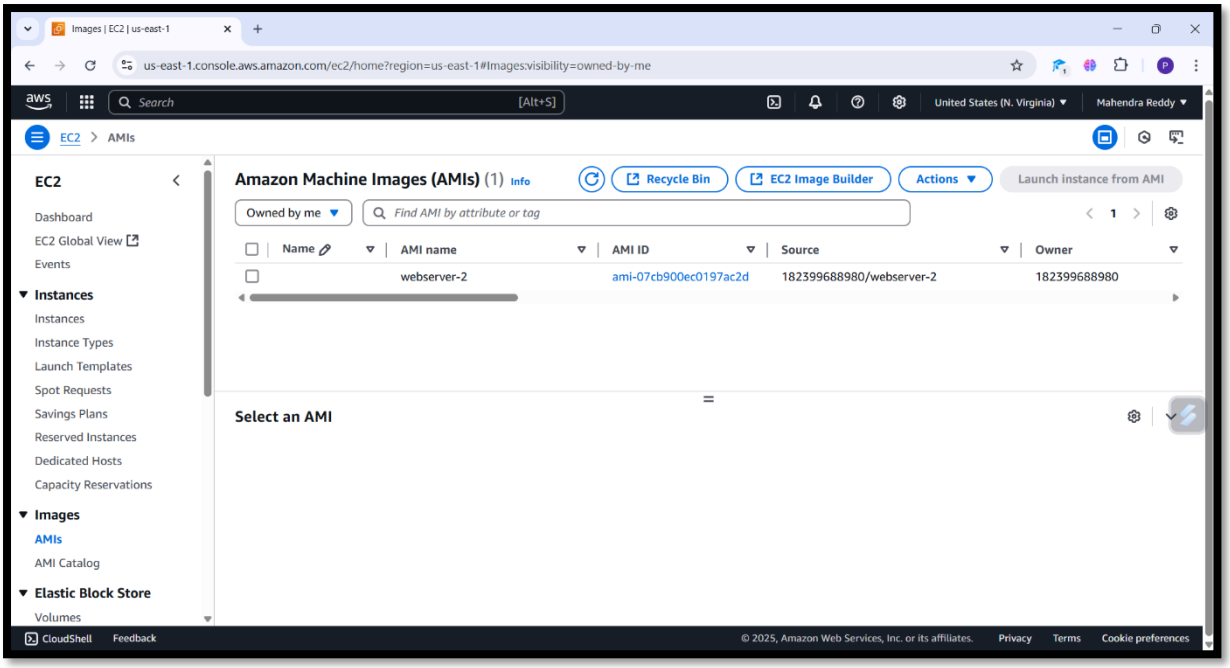
5. Prepare a Custom AMI for Reuse

- create AMI.
- Screenshot of the AMI created under EC2 > AMIs.



6. Prepare a Reusable Launch Template

- Launch template creation with custom AMI.
- Setting instance type, key pair, SGs, user data).
- Screenshots of the template configuration.



- Launch the website from the Template: