St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

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:SAIRAM S

Department: IT

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

A static website delivers pre-written HTML, CSS, and JavaScript files directly to users without requiring server-side processing. Hosting such websites on a cloud-based Virtual Machine (VM) is a popular choice for individuals and businesses due to its flexibility, scalability, and cost-effectiveness. Cloud-based deployment enables developers to launch websites quickly, ensuring global accessibility with minimal effort.

Overview

Hosting a static website on a cloud VM involves the following key steps:

1. **Provisioning a Cloud VM:** Setting up a virtual machine on a cloud provider (like AWS, Azure, or GCP).
2. **Installing a Web Server:** Configuring a web server such as Apache to serve the website's static files.
3. **Uploading Website Files:** Placing HTML, CSS, and JavaScript files in the web server's root directory.
4. **Configuring Network Access:** Ensuring that the web server is accessible via HTTP (port 80) from anywhere.
5. **Testing and Launching:** Verifying the functionality of the website to make it publicly accessible

Objectives

The primary objectives of hosting a static website on a cloud VM include:

1. **Learning Cloud Computing Fundamentals:** Understanding how virtual machines operate in a cloud environment.
2. **Practical Web Hosting Skills:** Gaining hands-on experience in setting up and configuring web servers like Apache or Nginx.
3. **Website Deployment:** Successfully deploying and making a static website live on the internet.
4. **Understanding Networking Basics:** Learning about firewall rules, security groups, and HTTP protocol configurations.
5. **Cost-Effective Hosting:** Exploring affordable methods to host lightweight websites without needing managed services.

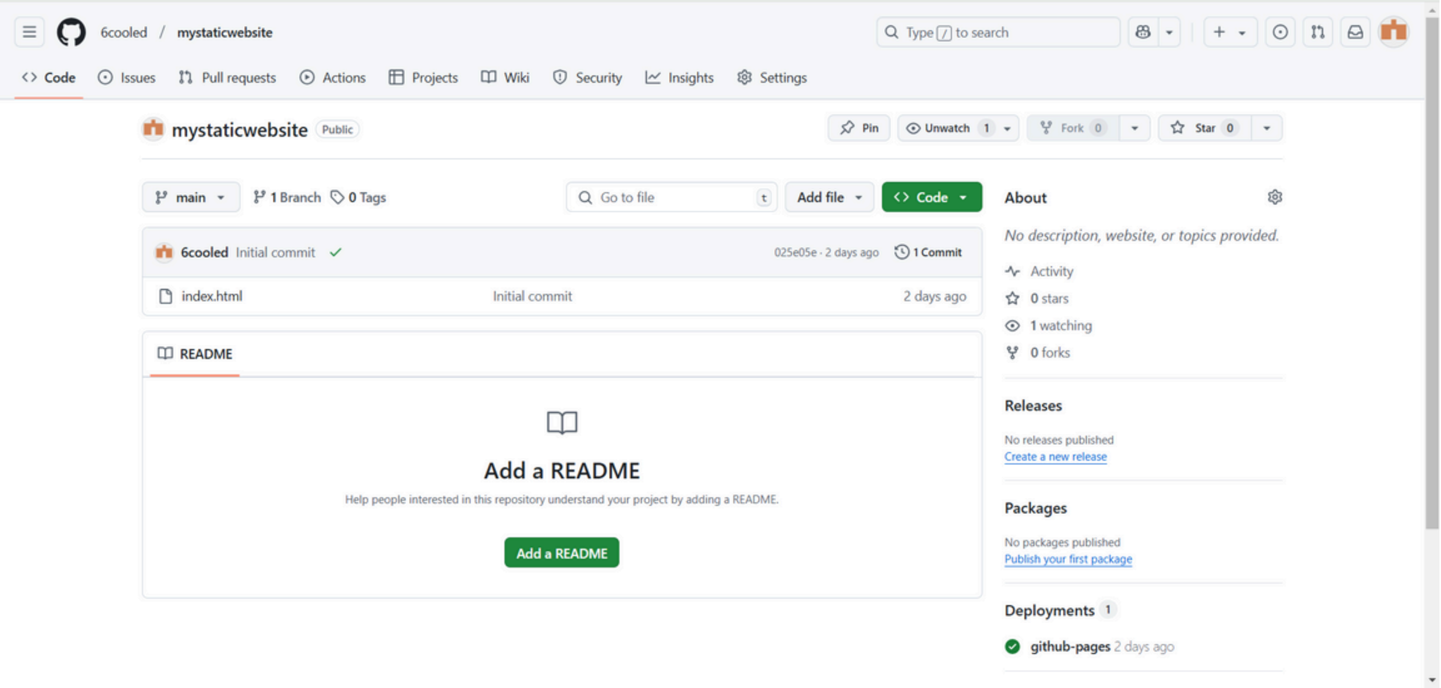
Importance

1. **Hands-On Cloud Experience:** Hosting a static website on a cloud VM is an excellent starting point for understanding the capabilities of cloud platforms and virtual machine operations.
2. **Scalability:** Cloud-based hosting provides flexibility to scale resources up or down as the traffic to the website grows.
3. **Global Accessibility:** By deploying on the cloud, the website becomes accessible from any part of the world with minimal latency.
4. **Customization and Control:** Cloud VMs allow complete control over the hosting environment, enabling advanced configurations and optimizations.
5. **Foundation for Advanced Hosting:** It lays the groundwork for more advanced projects, such as hosting dynamic websites, APIs, or using load balancers.
6. **Professional Development:** Learning to host websites on the cloud adds significant value to your skill set, making you proficient in real-world deployment scenarios.

Step-by-Step Overview

Step 1:

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

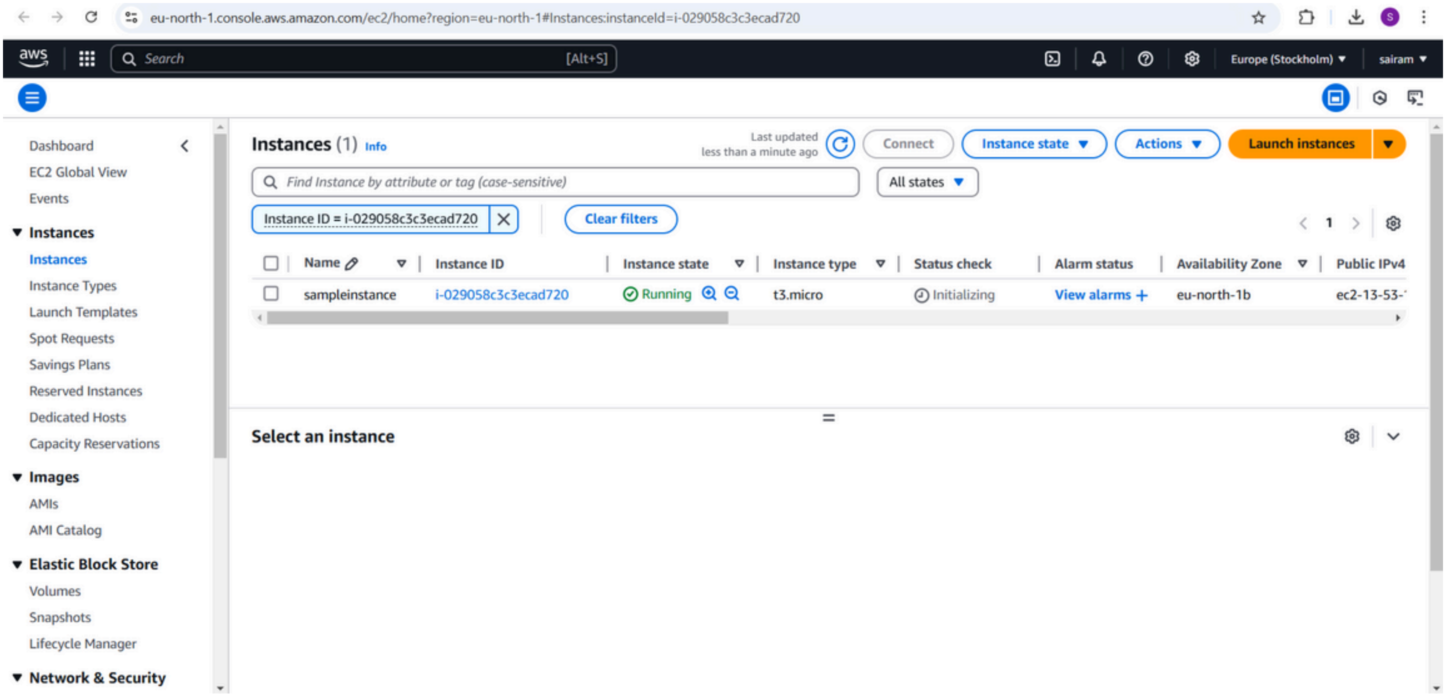


6cooled Initial commit ✓ 025e05e · 2 days ago 1 Commit

index.html	Initial commit	2 days ago

Step 2:

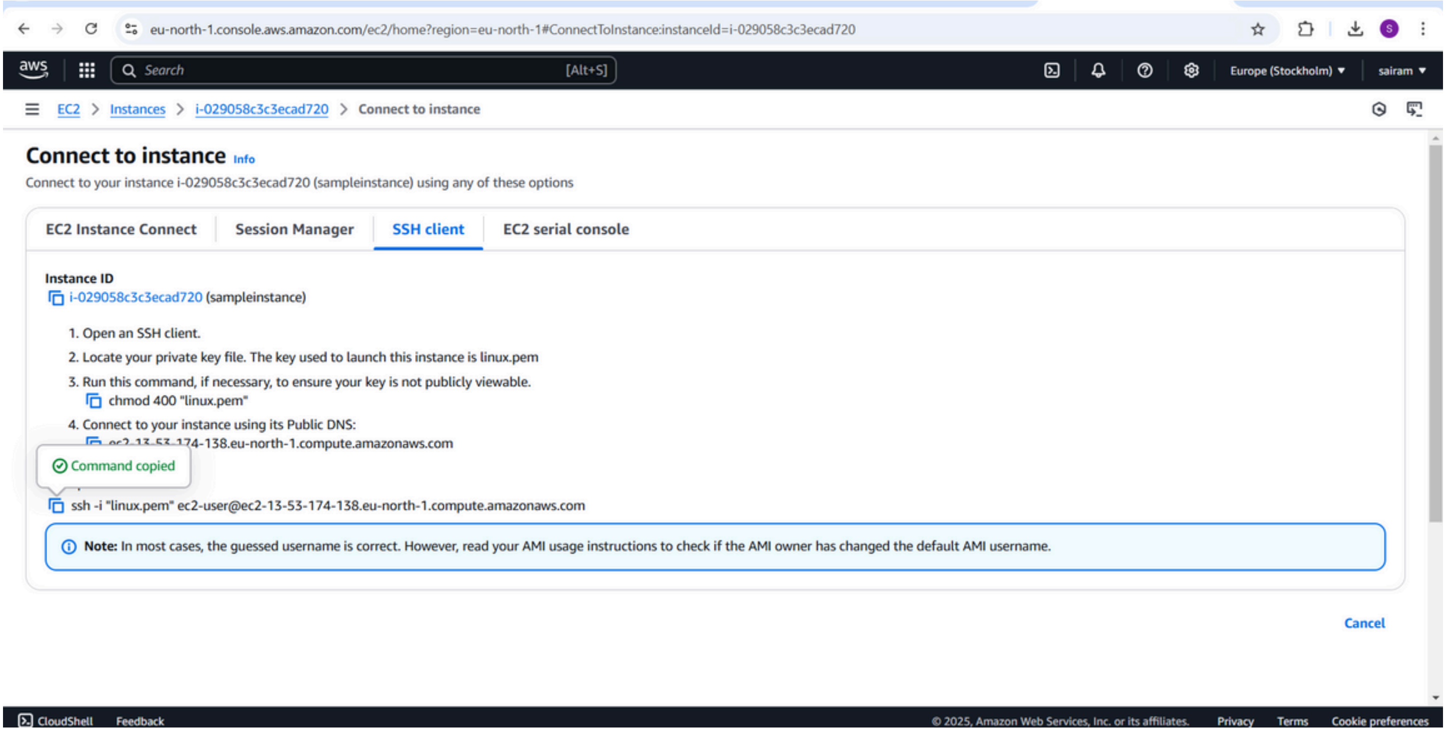
Launch an EC2 instance, select Ubuntu as the OS, configure security groups to allow all network traffic, create a key pair (e.g., new.pem), and download it for SSH access



Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
sampleinstance	i-029058c3c3ecad720	Running	t3.micro	Initializing	View alarms +	eu-north-1b	ec2-13-53-

Step 3:

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



eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#ConnectToInstance:instanceId=i-029058c3c3ecad720

EC2 > Instances > i-029058c3c3ecad720 > Connect to instance

Connect to instance

Connect to your instance i-029058c3c3ecad720 (sampleinstance) using any of these options

EC2 Instance Connect | Session Manager | SSH client | EC2 serial console

Instance ID

i-029058c3c3ecad720 (sampleinstance)

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is linux.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
   chmod 400 "linux.pem"
4. Connect to your instance using its Public DNS:
   ec2-13-53-174-138.eu-north-1.compute.amazonaws.com

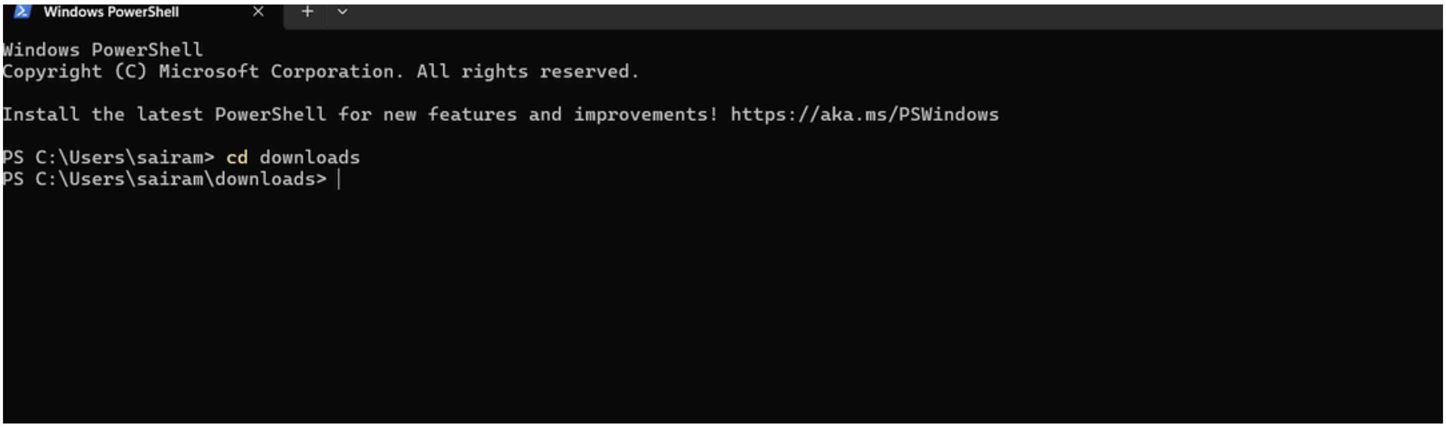
Command copied

ssh -i "linux.pem" ec2-user@ec2-13-53-174-138.eu-north-1.compute.amazonaws.com

Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.

Step 4:

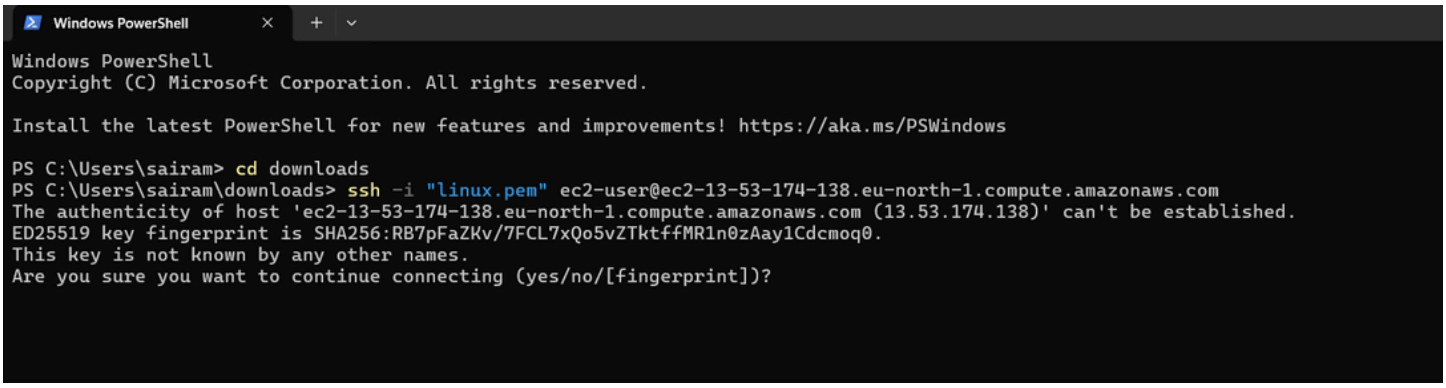
Open PowerShell, navigate to the 'Downloads' directory where the downloaded key pair is located using the **cd Downloads** command



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sairam> cd downloads
PS C:\Users\sairam\downloads> |
```

Step 5:

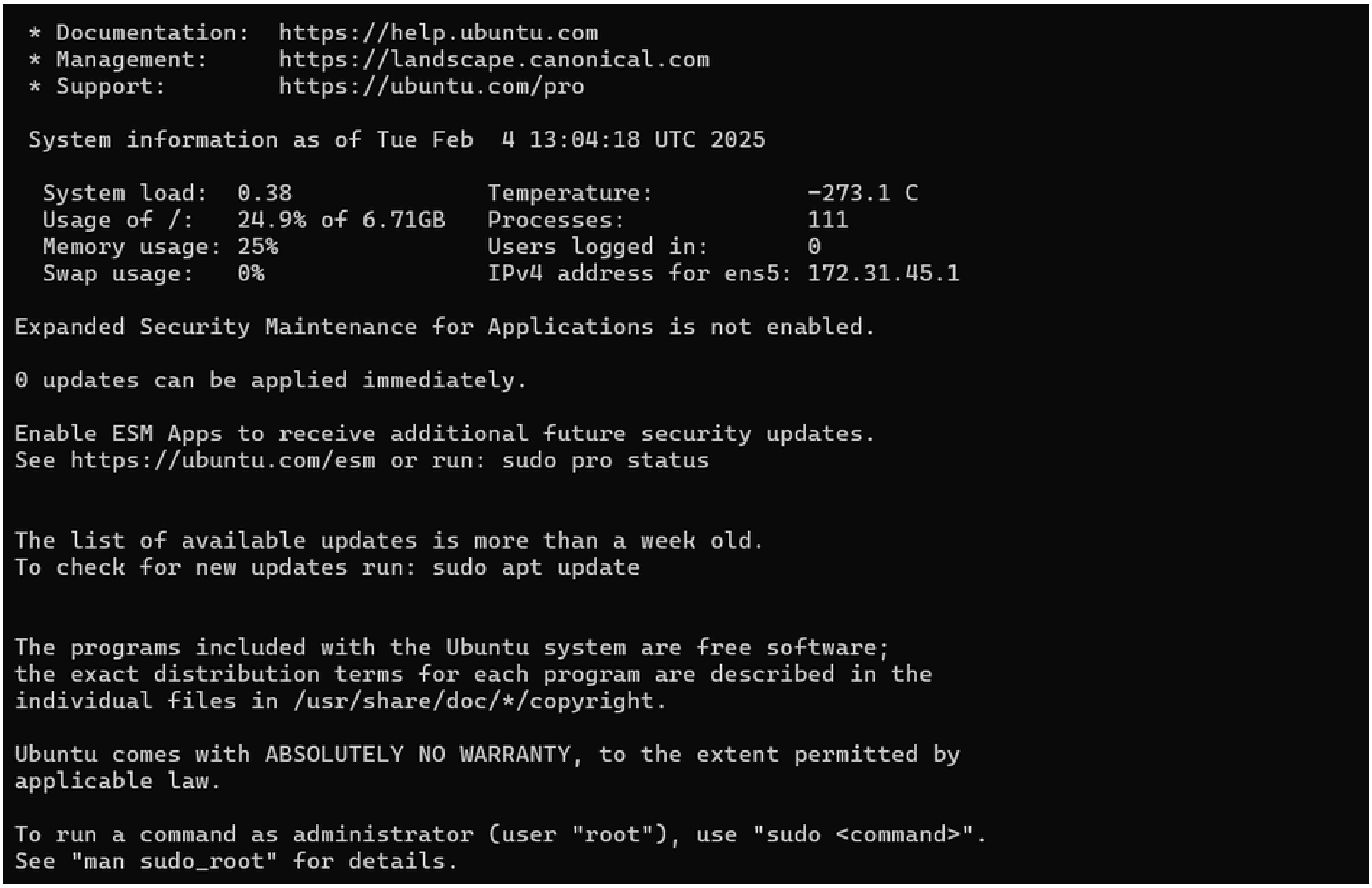
Paste the command copied from the EC2 Connect's SSH client section, replace the key pair name with your downloaded key (e.g., new.pem), press Enter, and type 'yes' when prompted.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sairam> cd downloads
PS C:\Users\sairam\downloads> ssh -i "linux.pem" ec2-user@ec2-13-53-174-138.eu-north-1.compute.amazonaws.com
The authenticity of host 'ec2-13-53-174-138.eu-north-1.compute.amazonaws.com (13.53.174.138)' can't be established.
ED25519 key fingerprint is SHA256:RB7pFaZKv/7FCL7xQo5vZTktffMR1n0zAay1Cdcmoq0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

Step 6:

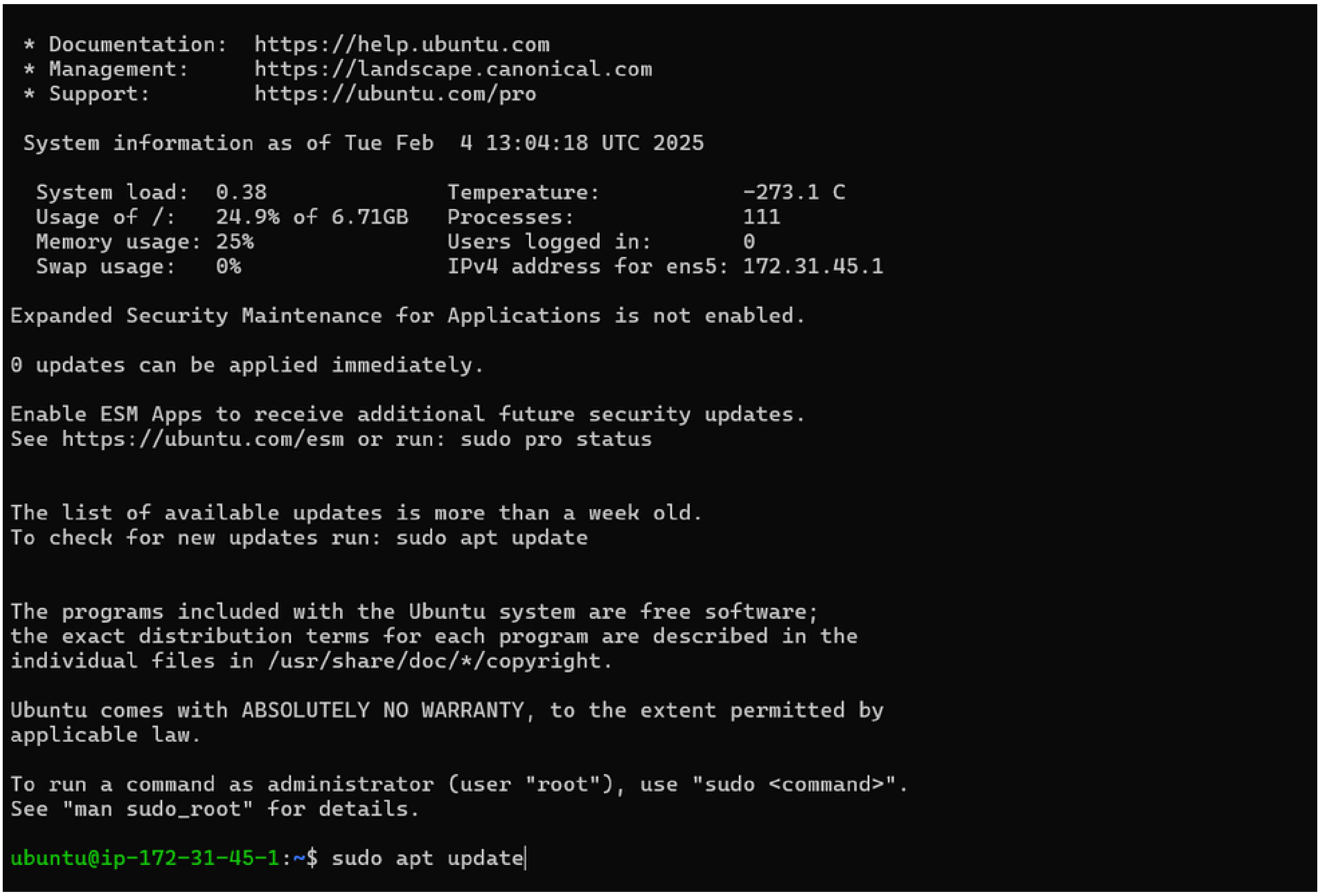
Run the command **sudo apt update** to update the package list.



```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Feb  4 13:04:18 UTC 2025

  System load:  0.38              Temperature:           -273.1 C
  Usage of /:   24.9% of 6.71GB   Processes:             111
  Memory usage: 25%               Users logged in:       0
  Swap usage:   0%                IPv4 address for ens5: 172.31.45.1

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.
```

Step 7:

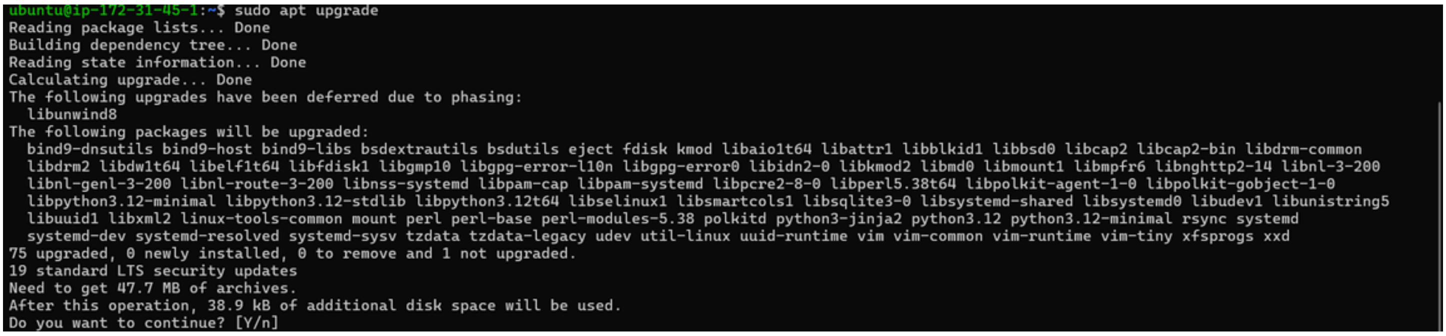
Run the command **sudo apt upgrade**, and press 'Y' to confirm and continue the upgrade process.



```
 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Tue Feb  4 13:04:18 UTC 2025

  System load:  0.38               Temperature:           -273.1 C
  Usage of /:   24.9% of 6.71GB    Processes:             111
  Memory usage: 25%                Users logged in:       0
  Swap usage:   0%                 IPv4 address for ens5: 172.31.45.1

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-172-31-45-1:~$ sudo apt update
```

Step 8:

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

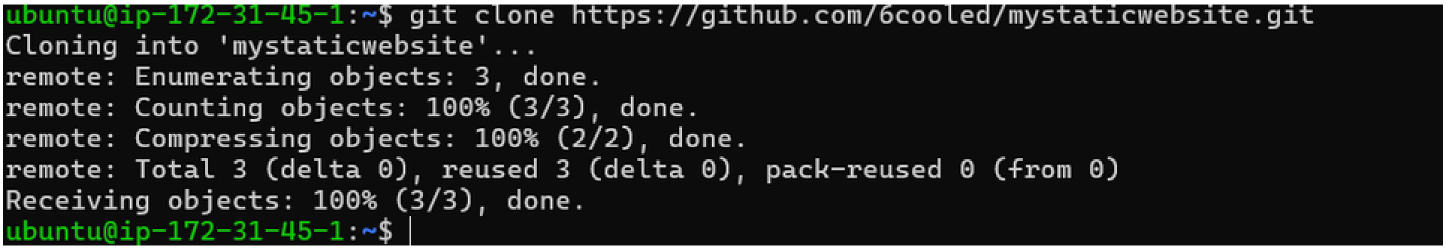


```
ubuntu@ip-172-31-45-1:~$ sudo apt upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following upgrades have been deferred due to phasing:
  libunwind8
The following packages will be upgraded:
  bind9-dnsutils bind9-host bind9-libs bsdextrautils bsdutils eject fdisk kmod libaio1t64 libattr1 libblkid1 libbsd0 libcap2 libcap2-bin libdrm-common
  libdrm2 libdw1t64 libelf1t64 libfdisk1 libgmp10 libgpg-error-l10n libgpg-error0 libidn2-0 libkmod2 libmd0 libmount1 libmpfr6 libnghttp2-14 libnl-3-200
  libnl-genl-3-200 libnl-route-3-200 libnss-systemd libpam-cap libpam-systemd libpcre2-8-0 libperl5.38t64 libpolkit-agent-1-0 libpolkit-gobject-1-0
  libpython3.12-minimal libpython3.12-stdlib libpython3.12t64 libselinux1 libsmartcols1 libsqlite3-0 libsystemd-shared libsystemd0 libudev1 libunistring5
  libuuid1 libxml2 linux-tools-common mount perl perl-base perl-modules-5.38 polkitd python3-jinja2 python3.12 python3.12-minimal rsync systemd
  systemd-dev systemd-resolved systemd-sysv tzdata tzdata-legacy udev util-linux uuid-runtime vim vim-common vim-runtime vim-tiny xfsprogs xxd
75 upgraded, 0 newly installed, 0 to remove and 1 not upgraded.
19 standard LTS security updates
Need to get 47.7 MB of archives.
After this operation, 38.9 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Step 9:

Insertyourfilesbyrunningthecommand**gitclone**

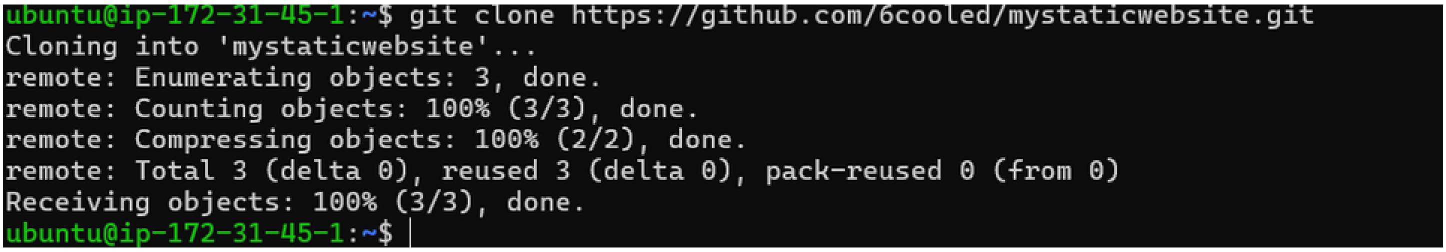
<repository_link> to clone your repository containing the website files



```
ubuntu@ip-172-31-45-1:~$ git clone https://github.com/6cooled/mystaticwebsite.git
Cloning into 'mystaticwebsite'...
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 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-45-1:~$ 
```

Step 10:

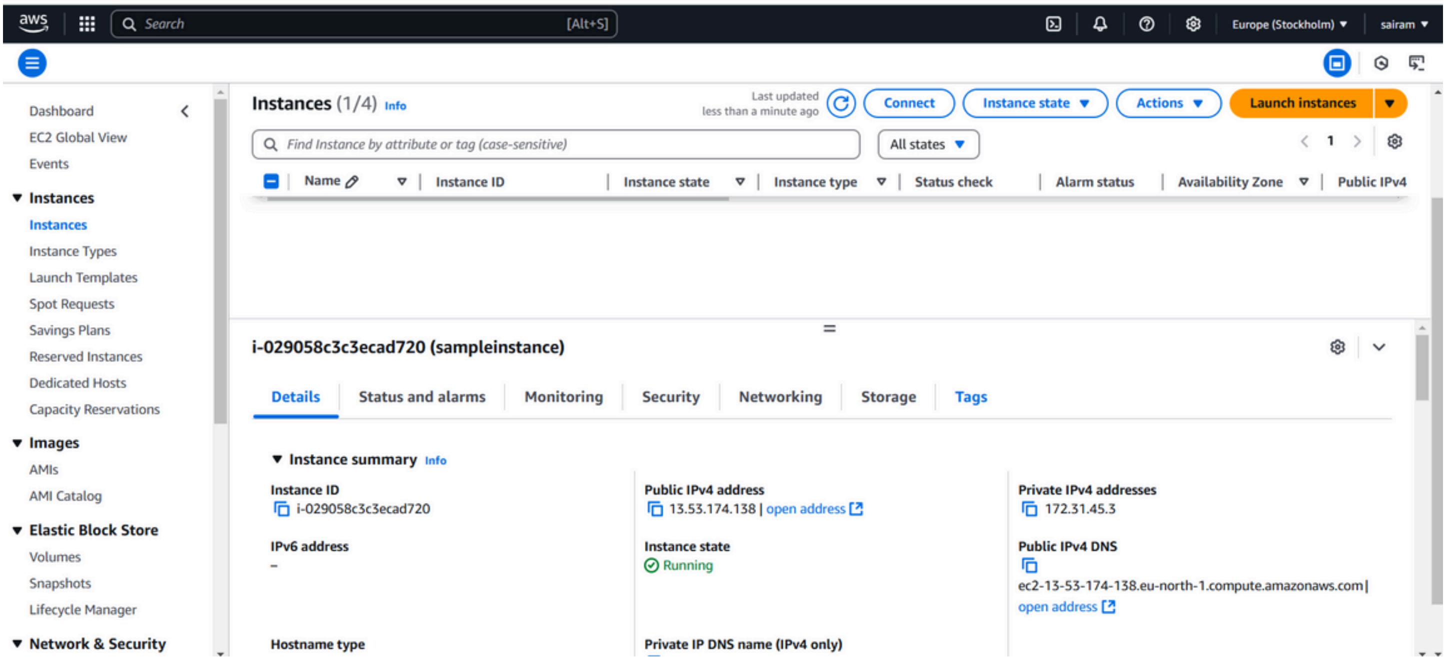
Run the command **cd /var/www/html** to navigate to the web server's root directory, then type ls to verify that your HTML files from the GitHub repository are present.



```
ubuntu@ip-172-31-45-1:~$ git clone https://github.com/6cooled/mystaticwebsite.git
Cloning into 'mystaticwebsite'...
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 3 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (3/3), done.
ubuntu@ip-172-31-45-1:~$ 
```

Step 11:

Copy the Public IPv4 DNS from the instance details page in the EC2 console, as shown in the image below.



i-029058c3c3ecad720 (sampleinstance)

▾ Instance summary Info

Instance ID: i-029058c3c3ecad720

Public IPv4 address: 13.53.174.138 | open address

Private IPv4 addresses: 172.31.45.3

IPv6 address: –

Instance state: Running

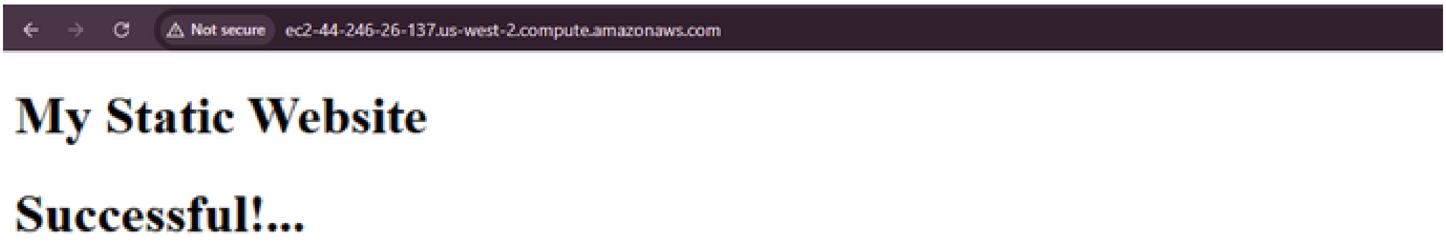
Public IPv4 DNS: ec2-13-53-174-138.eu-north-1.compute.amazonaws.com | open address

Hostname type

Private IP DNS name (IPv4 only)

Step 12:

Open Chrome and paste the copied Public IPv4 DNS in the address bar to view the content of your index.html file.



ec2-44-246-26-137.us-west-2.compute.amazonaws.com

My Static Website

Successful!...

Outcome

By completing this PoC of deploying a static website using an EC2 instance, you will:

1. Launch and configure an EC2 instance with Ubuntu as the OS.

2. Install and configure Apache web server to serve your static website.
3. Clone your GitHub repository containing your static website files (HTML, CSS, JavaScript) onto your EC2 instance.
4. Upload and place the website files in the Apache root directory (/var/www/html).
5. Access your static website live on the web using the EC2 instance's Public IPv4 DNS.