

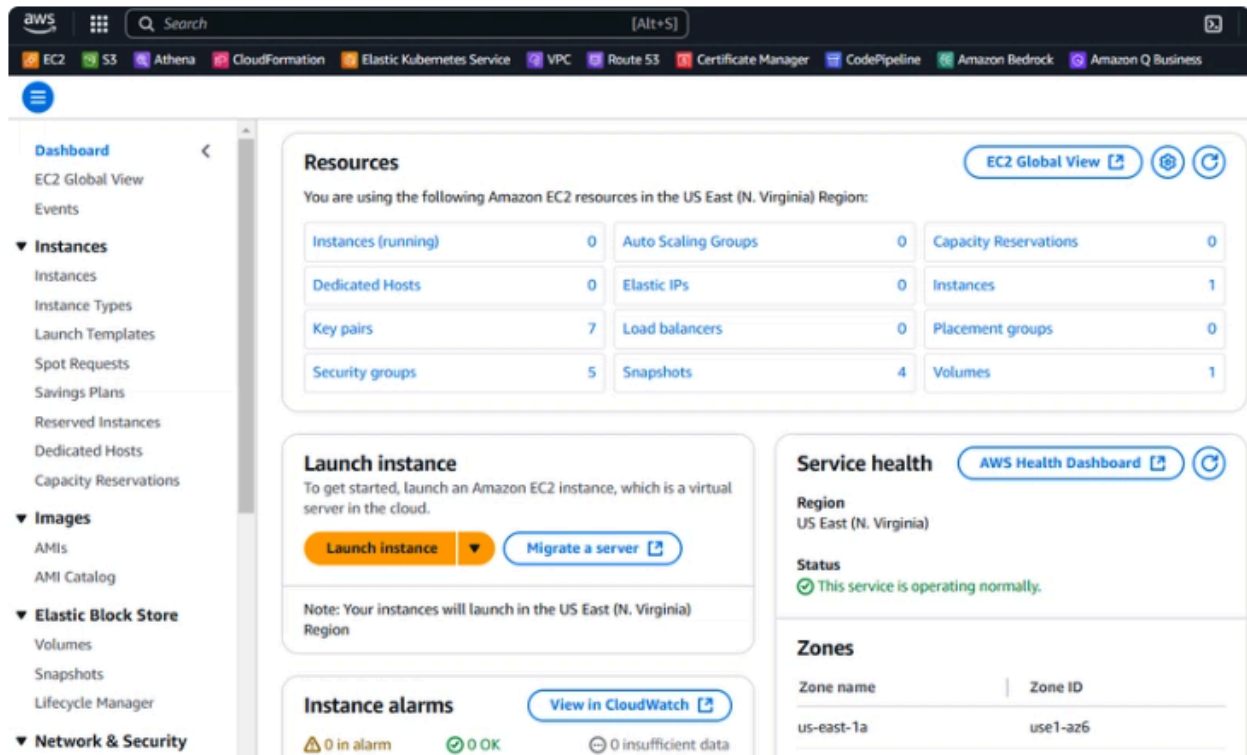
UNIT V – DEPLOYMENT AND CLOUD BASICS [9 hours]

How Software is Deployed in Real Life ,Cloud Platforms Overview (AWS/GCP/Azure), Hosting a Web App on Cloud (AWS EC2/Heroku), Docker Compose Basics, Introduction to Monitoring Tools (like Grafana)

HOSTING A WEB APP ON CLOUD AWS EC2

Step 1: Launch an EC2 Instance

1. Sign in to the AWS Management Console and navigate to EC2 under the Compute section.



2. Click Launch Instance to start a new EC2 instance.

Launch instance

To get started, launch an Amazon EC2 instance, which is a virtual server in the cloud.

[Launch instance](#) ▼ [Migrate a server](#) ↗

Note: Your instances will launch in the US East (N. Virginia) Region

3. Choose an AMI: For this example, we'll use the Amazon Linux 2 AMI. This is a lightweight, secure, and stable Linux server that can serve static content.

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | My AMIs | **Quick Start**

Amazon Linux macOS Ubuntu Windows Red Hat SUSE Linux Debian

Amazon Linux 2023 AMI Free tier eligible

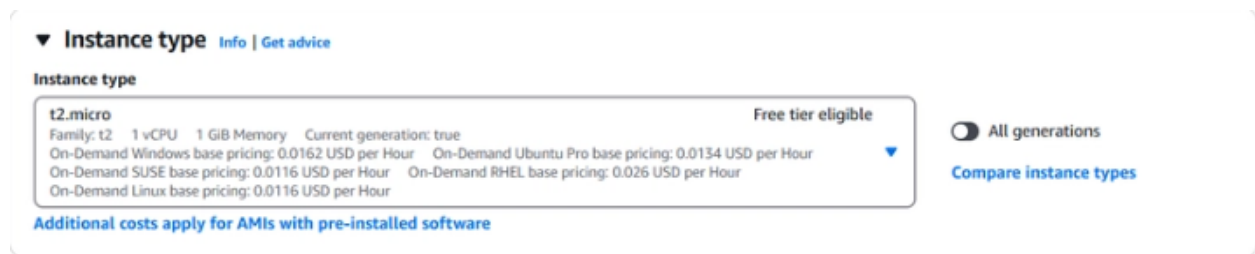
ami-05576a079321f21f8 (64-bit (x86), uefi-preferred) / ami-03ecf97a3bb0705c2 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.6.20250107.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Username ⓘ
64-bit (x86)	uefi-preferred	ami-05576a079321f21f8	ec2-user Verified provider

4. Select Instance Type: Choose t2.micro (free tier eligible) for basic hosting.



The screenshot shows the AWS EC2 console's 'Instance type' selection interface. At the top, there is a dropdown menu for 'Instance type' with a downward arrow, followed by 'Info' and 'Get advice' links. Below this, the selected instance type is 't2.micro', which is marked as 'Free tier eligible'. A detailed box contains the following information: 'Family: t2', '1 vCPU', '1 GiB Memory', and 'Current generation: true'. Below this, pricing information is listed: 'On-Demand Windows base pricing: 0.0162 USD per Hour', 'On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour', 'On-Demand SUSE base pricing: 0.0116 USD per Hour', and 'On-Demand RHEL base pricing: 0.026 USD per Hour'. To the right of this box, there is a radio button labeled 'All generations' and a link 'Compare instance types'. At the bottom of the box, a note states 'Additional costs apply for AMIs with pre-installed software'.

5. Create New Key: Click on "Create New Key Pair": In the EC2 dashboard, navigate to the Key Pairs section and click on Create Key Pair to begin the process of generating a new SSH key pair for your EC2 instance.



The screenshot shows the AWS EC2 console's 'Key pair (login)' creation interface. At the top, there is a dropdown menu for 'Key pair (login)' with a downward arrow, followed by 'Info' and 'Info' links. Below this, a message states: 'You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.' Below the message, there is a label 'Key pair name - required' and a dropdown menu with 'Select' as the current selection. To the right of the dropdown menu, there is a circular refresh icon and a link 'Create new key pair'.

Click "Create": Once you've chosen a name and confirmed the .pem extension, hit the Create button. This will generate your private key and automatically download the .pem file to your computer.

Create key pair ✕

Key pair name
Key pairs allow you to connect to your instance securely.

Security

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA
RSA encrypted private and public key pair

ED25519
ED25519 encrypted private and public key pair

Private key file format

.pem
For use with OpenSSH

.ppk
For use with PuTTY

⚠ When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

[Cancel](#) [Create key pair](#)

6. Configure Instance: Leave the default settings, or configure them based on your needs. Ensure your instance has a public IP so you can access it from the internet.

Network settings [Info](#) Edit

Network [Info](#)
vpc-66e7371b

Subnet [Info](#)
No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)
Enable
Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only. ×

7. Add Storage: You can stick with the default 8GB storage or increase it if necessary.

Configure storage [Info](#) Advanced

1x GiB Root volume 3000 IOPS (Not encrypted)

ⓘ Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage ×

Add new volume

ⓘ Click refresh to view backup information ↻
The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems Edit

8. Configure Security Group: Create a new security group or use an existing one. Allow inbound HTTP (port 80) and SSH (port 22) access.

Firewall (security groups) [Info](#)
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

Allow SSH traffic from Anywhere
Helps you connect to your instance
0.0.0.0/0

Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

9. Configure Security Group: Click on Lunch Instance

▼ Summary

Number of instances | [Info](#)

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...[read more](#)
ami-0df8c184d5f6ae949

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address usage per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots, and 100 GB of bandwidth to the internet. ✕

[Cancel](#) [Launch instance](#)

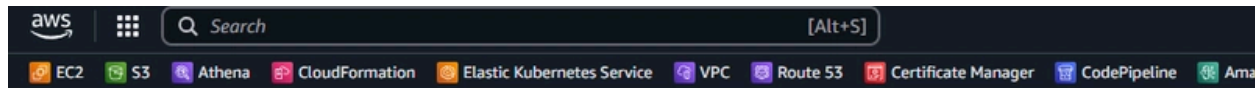
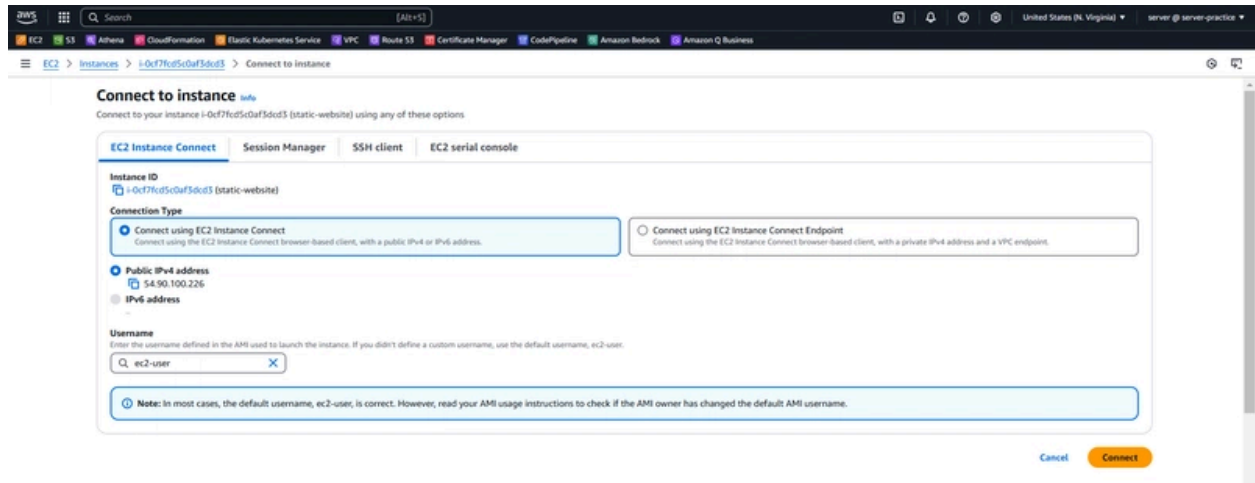
[Preview code](#)

Step 2: Connect to Your EC2 Instance

Once your instance is running, you can SSH into it.

1. From the EC2 dashboard, select your instance and click Connect.

2. From EC2 Instance Connect option click on connect to SSH into the instance using your private key.



Step 3: Install Apache HTTP Server

We'll use Apache HTTP Server to serve the static website.

1. Update your EC2 instance:

```
sudo yum update -y
```

The screenshot shows the AWS console interface at the top with various services like EC2, S3, Athena, CloudFormation, Elastic Kubernetes Service, VPC, Route 53, and Certificate Manager. Below the console is a terminal window for an Amazon Linux 2023 instance. The terminal output shows the user running 'sudo su -' to become root, followed by 'sudo yum update -y'. The system reports that the last metadata expiration check was 0:05:17 ago on Fri Jan 17 14:58:04 2025, dependencies were resolved, and nothing needed to be updated. The process is complete.

```

#
#####
~~~~\#####\
~~~~\###|
~~~~\#/
~~~~V~'-'>
~~~~
~~~~
~~~~/m/'-'
[ec2-user@ip-172-31-16-119 ~]$ sudo su -
[root@ip-172-31-16-119 ~]# sudo yum update -y
Last metadata expiration check: 0:05:17 ago on Fri Jan 17 14:58:04 2025.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-16-119 ~]#
    
```

2. Install Apache:

sudo yum install -y httpd

The screenshot shows the AWS console interface at the top. Below it is a terminal window where the user runs 'sudo yum install -y httpd'. The terminal output shows the installation of httpd and its dependencies. A table lists the packages, their architectures, and versions. The transaction summary indicates that 12 packages will be installed, with a total download size of 2.3 M and an installed size of 6.9 M.

```

[root@ip-172-31-16-119 ~]# sudo yum install -y httpd
Last metadata expiration check: 0:06:32 ago on Fri Jan 17 14:58:04 2025.
Dependencies resolved.

Package                Architecture          Version
-----
Installing:
httpd                   x86_64                 2.4.62-1.amzn2023
Installing dependencies:
apr                     x86_64                 1.7.5-1.amzn2023.0.2
apr-util                x86_64                 1.6.3-1.amzn2023.0.1
generic-logos-httpd    noarch                 18.0.0-12.amzn2023.0.3
httpd-core              x86_64                 2.4.62-1.amzn2023
httpd-filesystem        noarch                 2.4.62-1.amzn2023
httpd-tools             x86_64                 2.4.62-1.amzn2023
libbrotli               x86_64                 1.0.9-4.amzn2023.0.2
mailcap                 noarch                 2.1.49-3.amzn2023.0.3
Installing weak dependencies:
apr-util-openssl        x86_64                 1.6.3-1.amzn2023.0.1
mod_http2                x86_64                 2.0.27-1.amzn2023.0.3
mod_lua                  x86_64                 2.4.62-1.amzn2023

Transaction Summary
-----
Install 12 Packages

Total download size: 2.3 M
Installed size: 6.9 M
Downloading Packages:
(1/12): apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64.rpm
    
```

3. Start Apache:

sudo systemctl start httpd

4. Enable Apache to start on boot:

sudo systemctl enable httpd

```

aws [Alt+S]
EC2 S3 Athena CloudFormation Elastic Kubernetes Service VPC Route 53 Certificate Manager CodePipeline Amazon Bedrock Amazon Q Business

[root@ip-172-31-16-119 ~]# sudo systemctl status httpd
0 httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8).
[root@ip-172-31-16-119 ~]# sudo systemctl start httpd
[root@ip-172-31-16-119 ~]# sudo systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Fri 2025-01-17 15:05:35 UTC; 3s ago
     Docs: man:httpd.service(8).
  Main PID: 26478 (httpd)
   Status: "Started, listening on: port 80"
    Tasks: 177 (limit: 1111)
  Memory: 12.9M
    CPU: 53ms
   CGroup: /system.slice/httpd.service
           └─26478 /usr/sbin/httpd -DFOREGROUND
             └─26479 /usr/sbin/httpd -DFOREGROUND
               └─26480 /usr/sbin/httpd -DFOREGROUND
                 └─26481 /usr/sbin/httpd -DFOREGROUND
                   └─26482 /usr/sbin/httpd -DFOREGROUND

Jan 17 15:05:35 ip-172-31-16-119.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Jan 17 15:05:35 ip-172-31-16-119.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Jan 17 15:05:35 ip-172-31-16-119.ec2.internal httpd[26478]: Server configured, listening on: port 80
[root@ip-172-31-16-119 ~]# sudo systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-16-119 ~]#
    
```

Step 4: Configure Static Website

Once you’ve successfully logged into the EC2 instance, follow these steps to install the necessary files from GitHub and set up your server:

1. Download the Static Website Files from GitHub: Use the git command to clone your repository directly into the instance:

yum install git -y

git clone <https://github.com/simplynadaf/static-website-on-ec2.git>

```

aws [Alt+S] United States (N. Virginia)
EC2 S3 Athena CloudFormation Elastic Kubernetes Service VPC Route 53 Certificate Manager CodePipeline Amazon Bedrock Amazon Q Business

[root@ip-172-31-16-119 ~]# sudo yum install -y httpd
Last metadata expiration check: 0:06:32 ago on Fri Jan 17 14:50:04 2025.
Dependencies resolved.
-----
Package                               Architecture  Version              Repository
-----
Installing:
httpd                                  x86_64       2.4.62-1.amzn2023   amazonlinux
Installing dependencies:
apr                                    x86_64       1.7.5-1.amzn2023.0.2  amazonlinux
apr-util                               x86_64       1.6.3-1.amzn2023.0.1  amazonlinux
generic-logos-httpd                   noarch       18.0.0-12.amzn2023.0.3  amazonlinux
httpd-core                             x86_64       2.4.62-1.amzn2023   amazonlinux
    
```



```

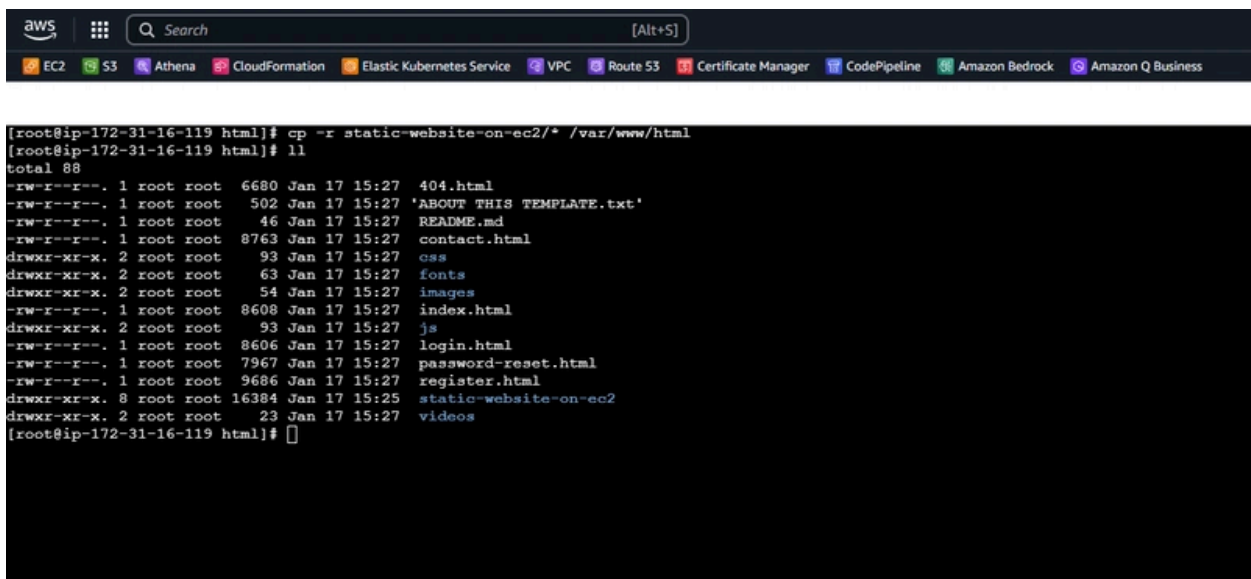
aws [Alt+S]
EC2 S3 Athena CloudFormation Elastic Kubernetes Service VPC Route 53 Certificate Manager CodePipeline Amazon Bedrock Amazon Q Business

[root@ip-172-31-16-119 html]# git clone https://github.com/simplynada/static-website-on-ec2.git
Cloning into 'static-website-on-ec2'...
remote: Enumerating objects: 28, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (20/20), done.
remote: Total 28 (delta 5), reused 25 (delta 5), pack-reused 0 (from 0)
Receiving objects: 100% (28/28), 6.11 MiB | 46.70 MiB/s, done.
Resolving deltas: 100% (5/5), done.
[root@ip-172-31-16-119 html]# ll
total 16
drwxr-xr-x. 8 root root 16384 Jan 17 15:25 static-website-on-ec2
[root@ip-172-31-16-119 html]#

```

2. Move the Files to the Web Root: Copy the contents of the repository to the web root directory:

```
cp -r static-website-on-ec2/* /var/www/html
```



```

aws [Alt+S]
EC2 S3 Athena CloudFormation Elastic Kubernetes Service VPC Route 53 Certificate Manager CodePipeline Amazon Bedrock Amazon Q Business

[root@ip-172-31-16-119 html]# cp -r static-website-on-ec2/* /var/www/html
[root@ip-172-31-16-119 html]# ll
total 88
-rw-r--r--. 1 root root 6680 Jan 17 15:27 404.html
-rw-r--r--. 1 root root 502 Jan 17 15:27 'ABOUT THIS TEMPLATE.txt'
-rw-r--r--. 1 root root 46 Jan 17 15:27 README.md
-rw-r--r--. 1 root root 8763 Jan 17 15:27 contact.html
drwxr-xr-x. 2 root root 93 Jan 17 15:27 css
drwxr-xr-x. 2 root root 63 Jan 17 15:27 fonts
drwxr-xr-x. 2 root root 54 Jan 17 15:27 images
-rw-r--r--. 1 root root 8608 Jan 17 15:27 index.html
drwxr-xr-x. 2 root root 93 Jan 17 15:27 js
-rw-r--r--. 1 root root 8606 Jan 17 15:27 login.html
-rw-r--r--. 1 root root 7967 Jan 17 15:27 password-reset.html
-rw-r--r--. 1 root root 9686 Jan 17 15:27 register.html
drwxr-xr-x. 8 root root 16384 Jan 17 15:25 static-website-on-ec2
drwxr-xr-x. 2 root root 23 Jan 17 15:27 videos
[root@ip-172-31-16-119 html]#

```

3. Clean Up: Remove the cloned repository folder if it's no longer needed:

```
rm -rf static-website-on-ec2
```

Step 5: Test Your Website

Now that everything is set up, open a web browser and visit your EC2 instance's public IP (e.g., <http://your-public-ip>). If everything is configured correctly, your static website should appear.

