

Project: Hosting a Website using AWS EC2 Service

❖ Objective:

The objective of this project is to successfully **deploy and host a static or dynamic website** on an **AWS EC2 instance** running a **Linux-based server** (such as Amazon Linux or Ubuntu). This project demonstrates the use of cloud services to ensure high availability, scalability, and secure delivery of web content.

❖ Project Scope:

- ❖ Launching and configuring a Virtual Private Server (EC2 Instance) on AWS.
- ❖ Installing and setting up a web server (Apache or Nginx) on a Linux machine.
- ❖ Uploading and configuring website files (HTML, CSS, JavaScript, backend if needed).
- ❖ Accessing the website through a public IP address.
- ❖ (Optional) Configuring a custom domain and securing the server using HTTPS/SSL certificates.

❖ Technologies Used:

- ❖ **Cloud Platform:** Amazon Web Services (AWS) EC2
- ❖ **Operating System:** Linux (Amazon Linux 2 / Ubuntu)
- ❖ **Web Server:** Apache HTTP Server / Nginx
- ❖ **Website Files:** HTML, CSS, JavaScript (or Node.js backend for dynamic sites)
- ❖ **Security:** SSH (for secure remote access), Security Groups, HTTPS (optional)

❖ Key Steps Performed:

1. EC2 Instance Setup:

- Created a new EC2 instance using the AWS Management Console.
- Selected an Amazon Linux 2/Ubuntu AMI and a t2.micro instance (free tier).
- Created and downloaded a PEM key pair for SSH access.
- Configured Security Groups to allow SSH (port 22), HTTP (port 80), and HTTPS (port 443).

2. Server Connection:

- Connected to the instance via SSH using terminal (Linux/Mac) or MobaXterm/Putty (Windows).

3. Web Server Installation:

- Updated system packages.
- Installed Apache HTTP server (or Nginx).
- Started and enabled the web server service to auto-start on boot.

4. Website Deployment:

- Uploaded website files (HTML, CSS, JS) to the /var/www/html/ directory.
- Set appropriate file and folder permissions.
- Verified that the website was accessible via the public IP address.

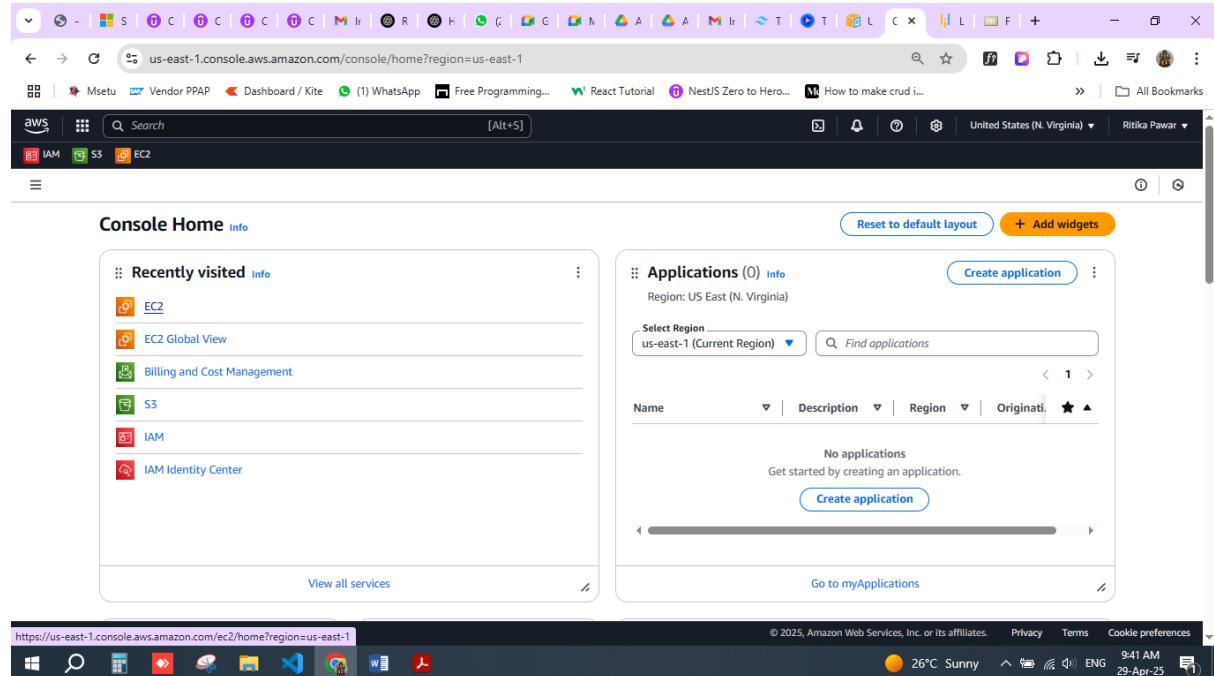
5. (Optional) Domain and SSL Setup:

- Configured DNS settings to point a custom domain to the EC2 public IP.
- Installed SSL certificates using Let's Encrypt Certbot for secure HTTPS access.

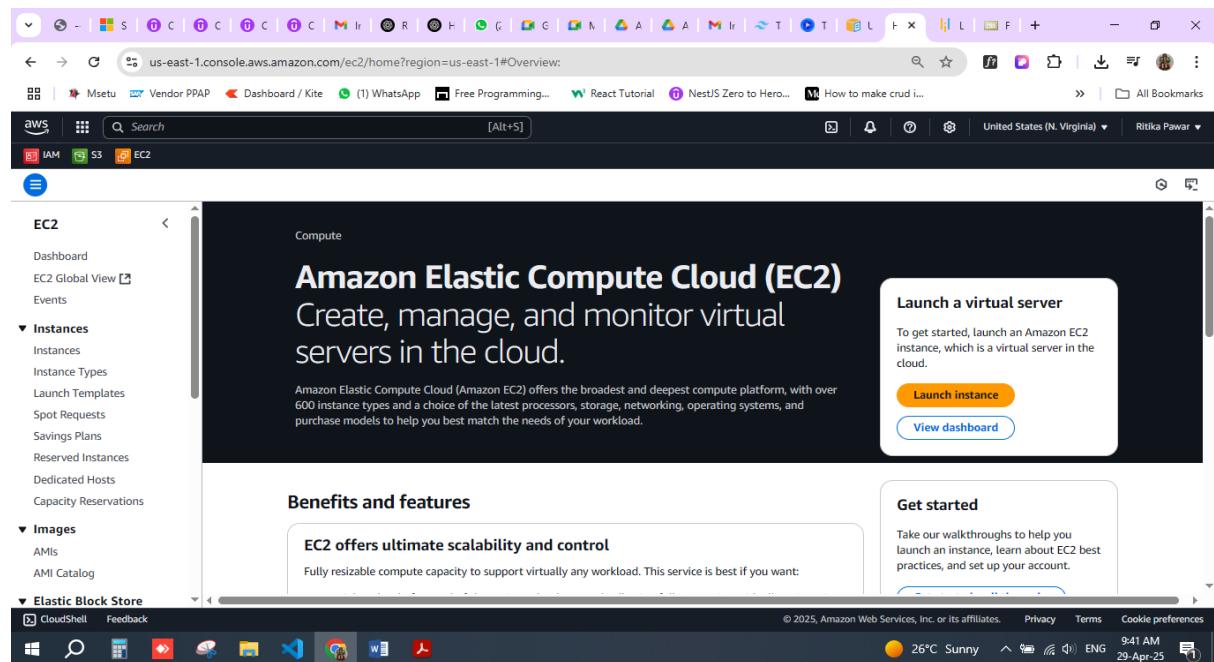
Step-by-Step Guide:

Step 1: create ec2 instances

1. Go to AWS console and open ec2 service



2. Click on “Launch Instances”



3. Give instance name, select OS image (AMI), Instance type, select/ create Key pair (login), Network settings, configure storage, review the summary

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

Do not show me this message again [Take a walkthrough](#)

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name Add additional tags

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

Recents | Quick Start

Summary

Number of instances [Info](#)

1

Software Image (AMI)

Virtual server type (instance type)

t2.micro

Firewall (security group)

Storage (volumes)

1 volume(s) - 8 GiB

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 26°C Sunny 9:41 AM 29-Apr-25

4. Then lunch instances

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:

Name my-demo-instance

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

Recents | **Quick Start**

Amazon Machine Image (AMI)

Amazon Linux 2023 AMI [read more](#)

Free tier eligible

Browse more AMIs Including AMIs from AWS Marketplace and the Community

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when

Cancel **Launch instance** **Preview code**

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Sunny 9:42 AM 29-Apr-25

The screenshot shows a browser window with the AWS logo in the top left corner. The address bar displays the URL: us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances. The main content area is titled "Success" and states "Successfully initiated launch of instance (i-0ea630f998ee8880)". Below this, there is a "Launch log" section which is currently empty. A "Next Steps" section follows, containing six items:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. Includes a "Create billing alerts" button.
- Connect to your instance**: Once your instance is running, log into it from your local computer. Includes a "Connect to instance" button and a "Learn more" link.
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. Includes a "Connect an RDS database" button and a "Create a new RDS database" link.
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. Includes a "Create EBS snapshot policy" button.
- Manage detailed monitoring**
- Create Load Balancer**
- Create AWS budget**
- Manage CloudWatch alarms**

At the bottom of the page, there is a navigation bar with links for CloudShell, Feedback, and various AWS services like Lambda, S3, and CloudWatch. On the right side, there are links for Privacy, Terms, and Cookie preferences, along with system status information: Construction on NH6..., 9:43 AM, 29-Apr-25, ENG.

Step 2: check Instance

1. Instance state and status

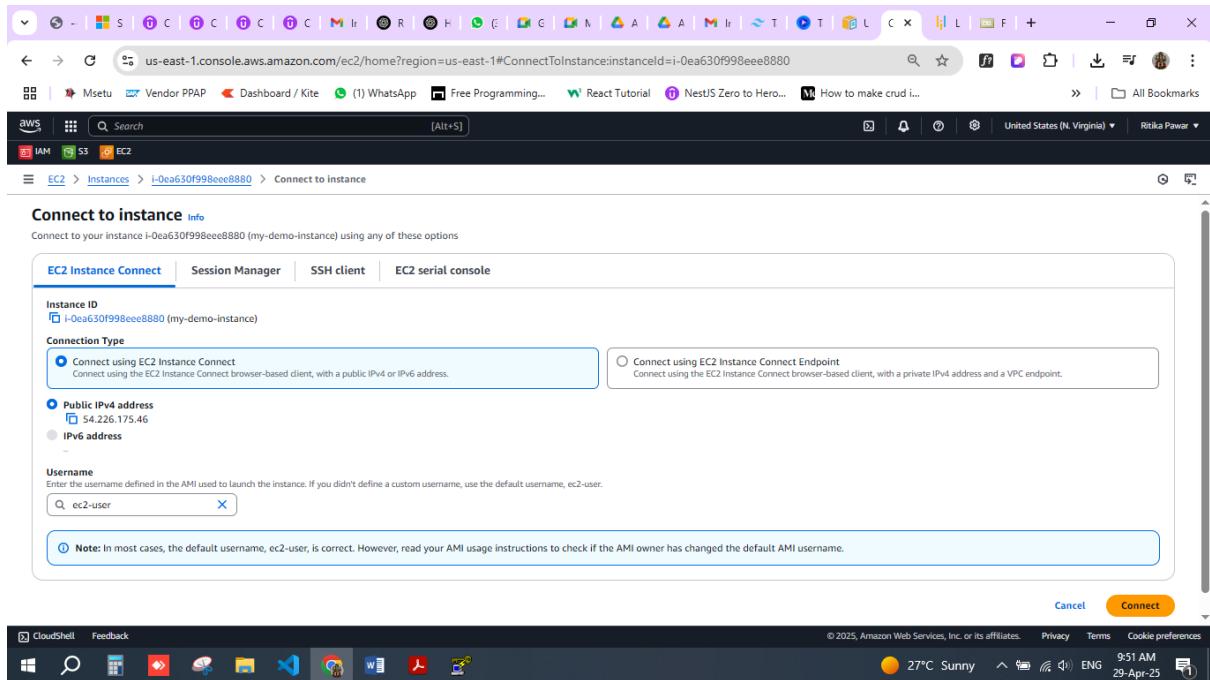
The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table titled 'Instances (1/1) Info'. A single row is selected for the instance 'my-demo-instance' with the ID 'i-0ea630f998eee8880'. The instance is shown as 'Running' with the type 't2.micro'. The 'Details' tab is selected, showing the instance summary. Key details include:

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0ea630f998eee8880	54.226.175.46 [open address]	172.31.19.140
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-54-226-175-46.compute-1.amazonaws.com

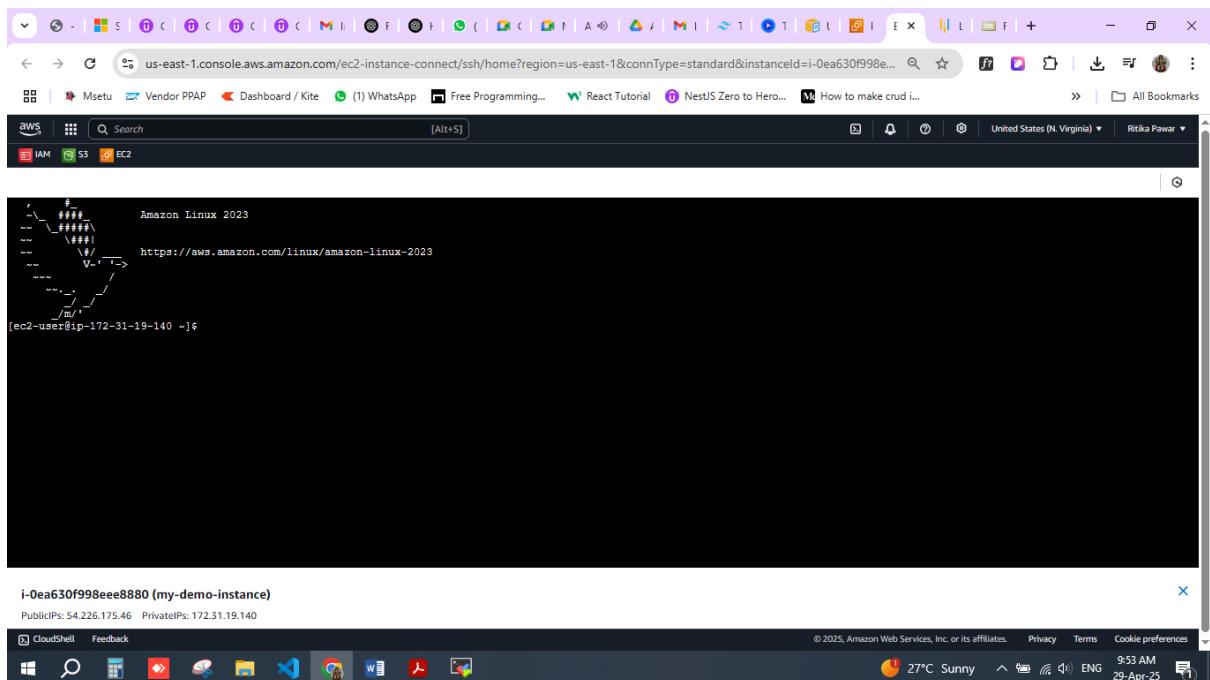
The status bar at the bottom indicates the instance was last updated less than a minute ago.

This screenshot is identical to the one above, showing the AWS EC2 Instances page with the same instance details. The difference is in the status message displayed below the instance summary. It now says '2/2 checks passed' instead of 'Initializing'.

2. Click on connect button , give username than connect

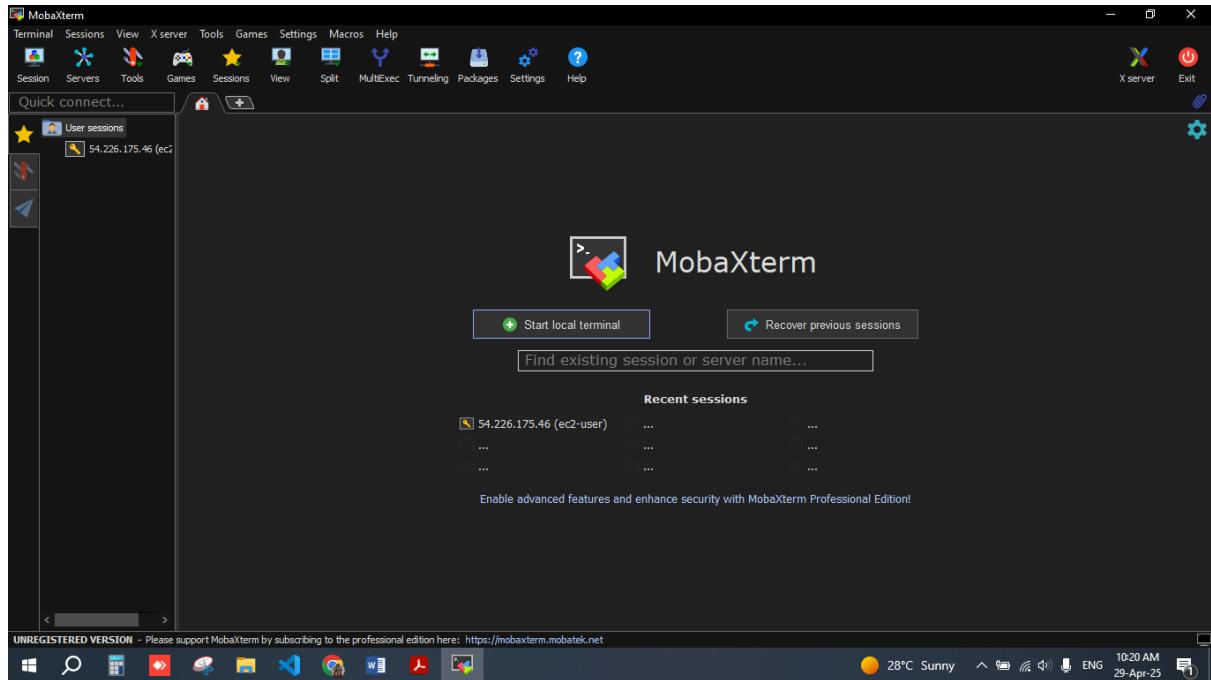


3. After connecting its launch new tab of amazon Linux console

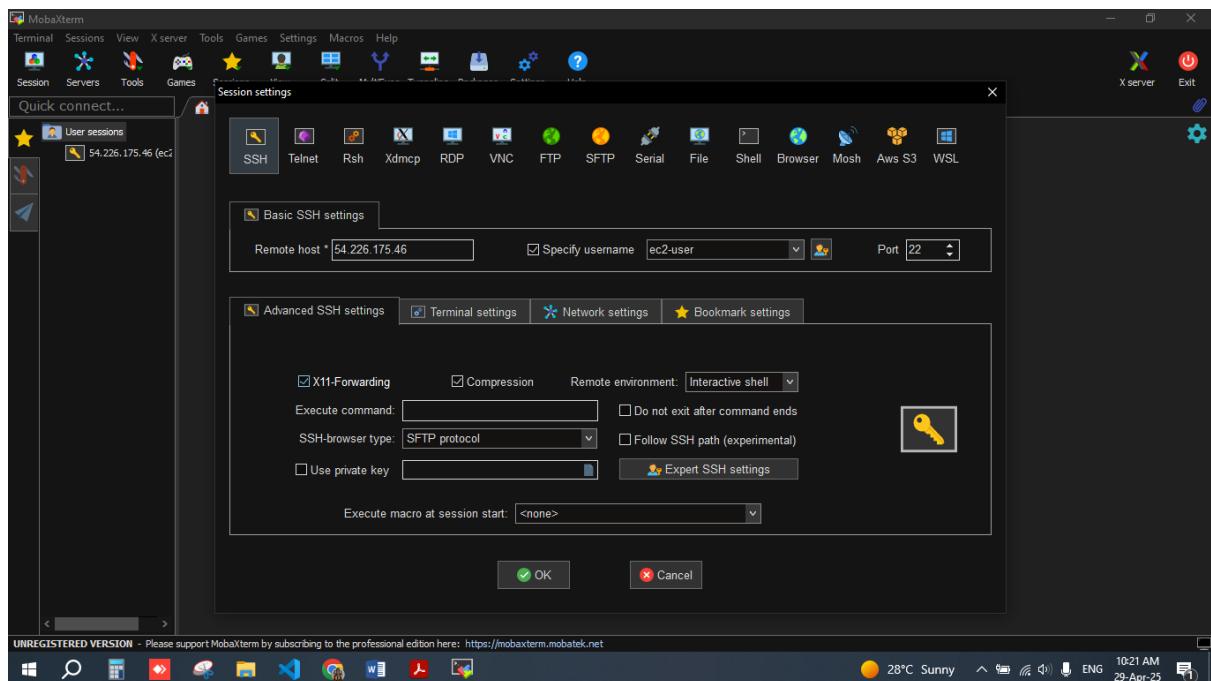


Step 3: setting for MobaXterm

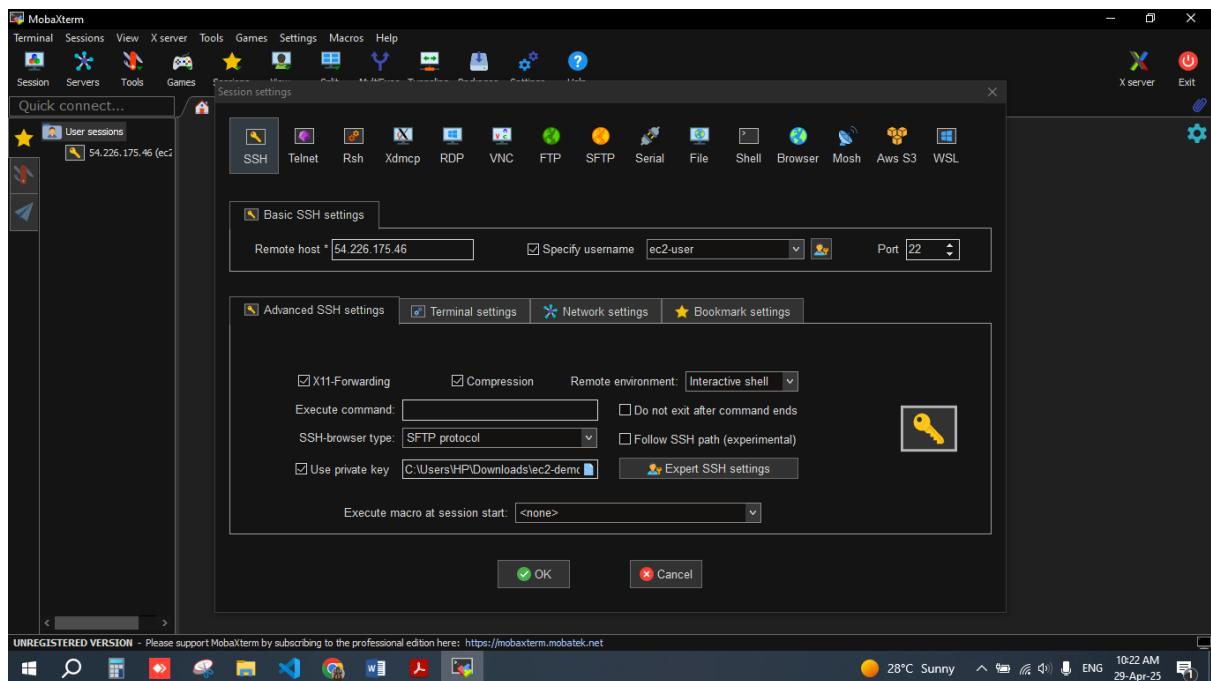
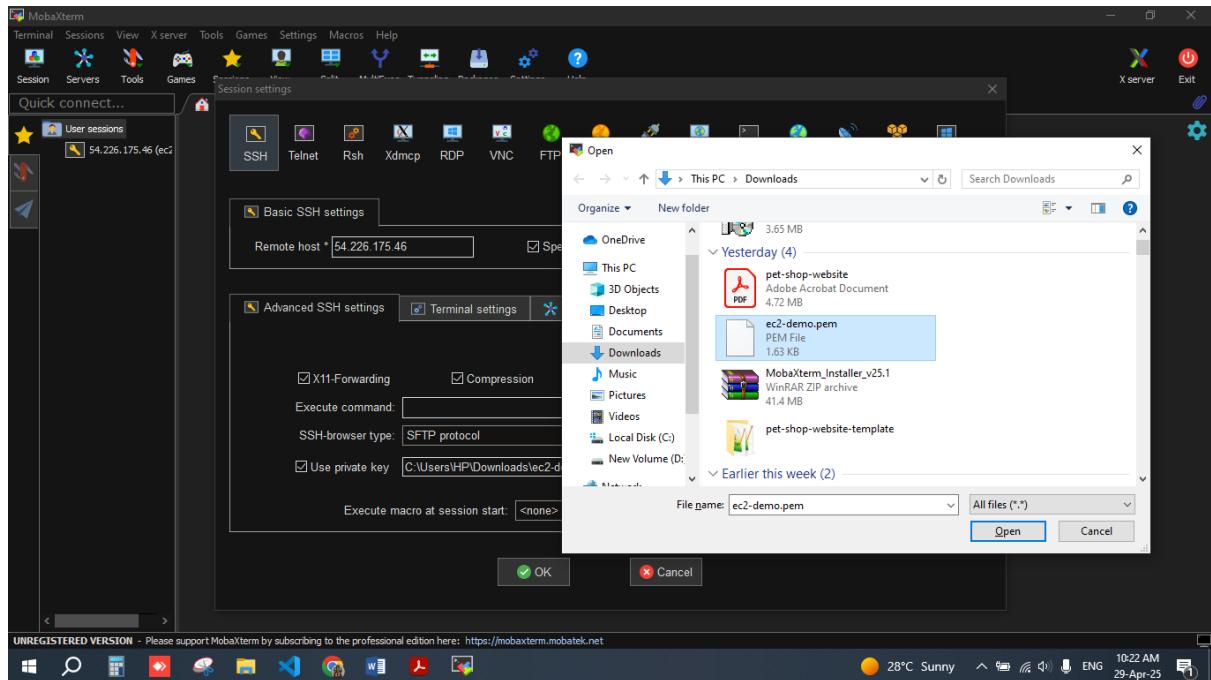
1. Open MobaXterm



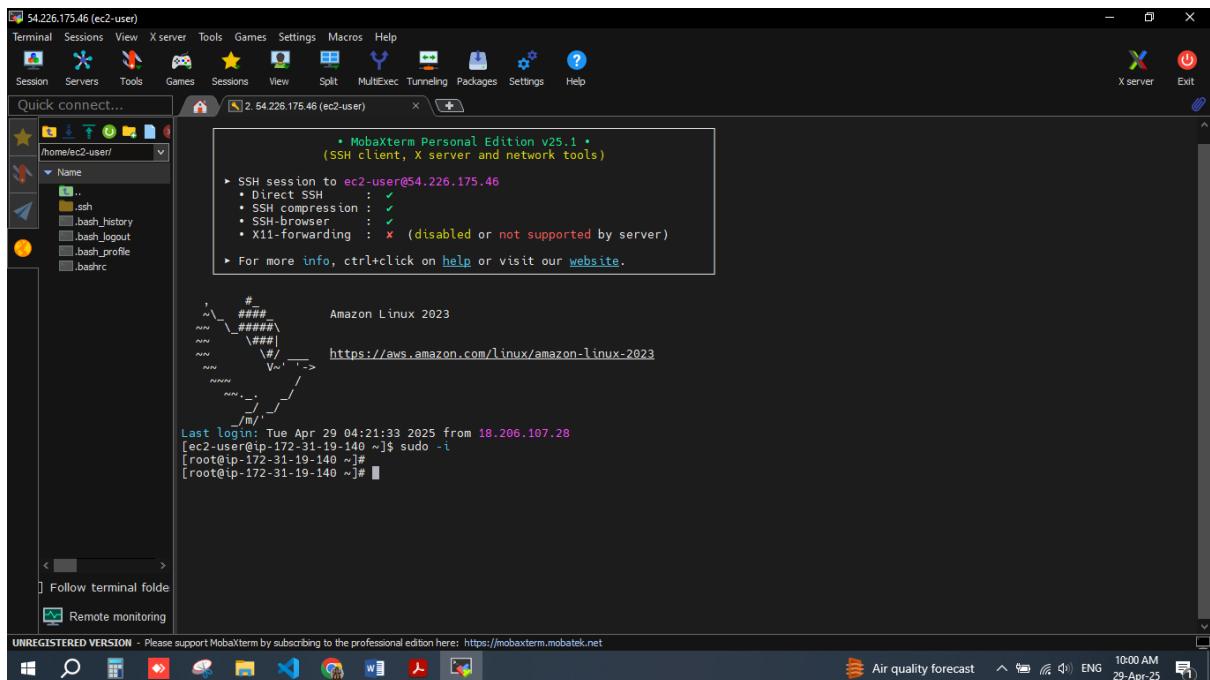
2. Open session tab, select SSH settings Enter public IP as remote host , username of your ec2 instance



3. Select Use private key upload the key which we had created at the time of instance

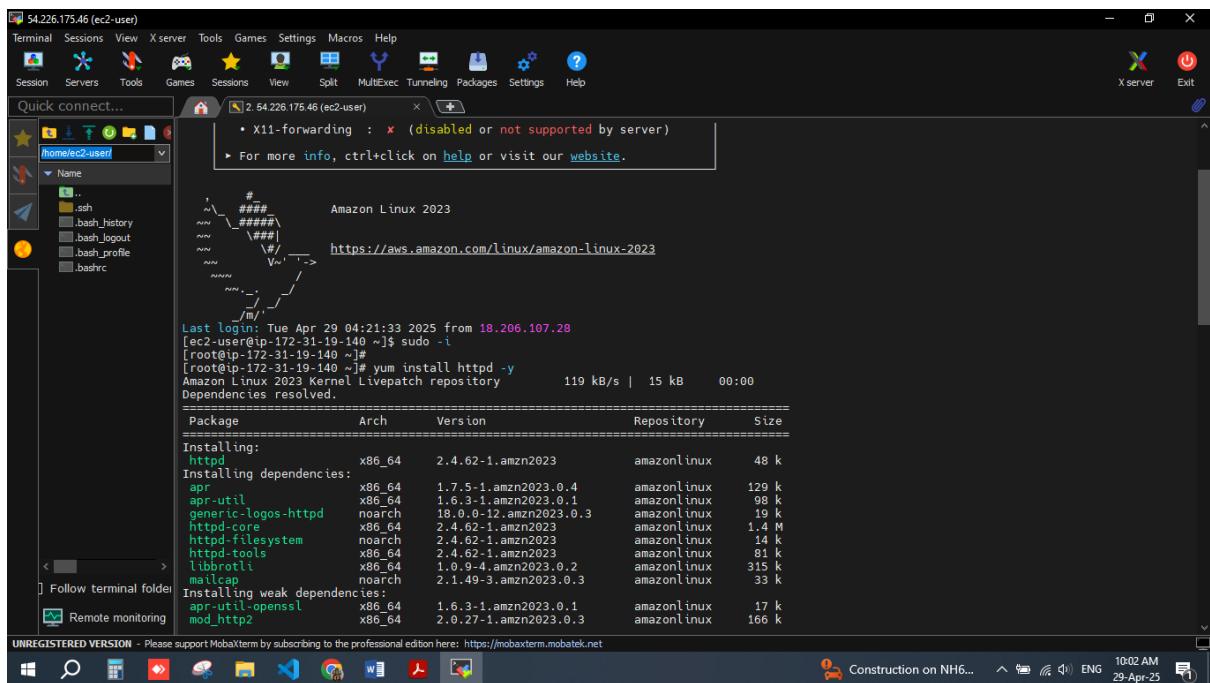


4. Launch user session on MobaXterm



Step 4: commands for enable httpd

1. Switch to root user # sudo –i
2. Install httpd # yum install httpd -y



3. Download zip file of project # curl -O url of zip file

The screenshot shows a MobaXterm window titled '2.54.226.175.46 (ec2-user)'. The terminal window displays the output of a curl command:

```
Installing : mod_lua-2.4.62-1.amzn2023.x86_64 10/12
Installing : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch 11/12
Installing : httpd-2.4.62-1.amzn2023.x86_64 12/12
Running script: httpd-2.4.62-1.amzn2023.x86_64 12/12
Verifying : apr-1.7.5-1.amzn2023.0.4.x86_64 1/12
Verifying : apr-util-1.6.3-1.amzn2023.0.1.x86_64 2/12
Verifying : apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64 3/12
Verifying : generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch 4/12
Verifying : httpd-2.4.62-1.amzn2023.x86_64 5/12
Verifying : httpd-core-2.4.62-1.amzn2023.x86_64 6/12
Verifying : httpd-filesystem-2.4.62-1.amzn2023.noarch 7/12
Verifying : httpd-tools-2.4.62-1.amzn2023.x86_64 8/12
Verifying : libbrotli-1.0.9-4.amzn2023.0.2.x86_64 9/12
Verifying : mailcap-2.1.49-3.amzn2023.0.3.noarch 10/12
Verifying : mod_http2-2.0.27-1.amzn2023.0.3.x86_64 11/12
Verifying : mod_lua-2.4.62-1.amzn2023.x86_64 12/12

Installed:
apr-1.7.5-1.amzn2023.0.4.x86_64
apr-util-1.6.3-1.amzn2023.0.1.x86_64
apr-util-openssl-1.6.3-1.amzn2023.0.1.x86_64
generic-logos-httdp-18.0.0-12.amzn2023.0.3.noarch
httpd-2.4.62-1.amzn2023.x86_64
httpd-core-2.4.62-1.amzn2023.x86_64
httpd-filesystem-2.4.62-1.amzn2023.noarch
httpd-tools-2.4.62-1.amzn2023.x86_64
libbrotli-1.0.9-4.amzn2023.0.2.x86_64
mailcap-2.1.49-3.amzn2023.0.3.noarch
mod_http2-2.0.27-1.amzn2023.0.3.x86_64
mod_lua-2.4.62-1.amzn2023.x86_64

Complete!
[root@ip-172-31-19-140 ~]# curl -O https://www.free-css.com/assets/files/free-css-temp
lates/download/page293/fonicy.zip
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
100 893k 100 893k 0 0 723k 0 0:00:01 0:00:01 ---:--- 723k
[root@ip-172-31-19-140 ~]#
```

At the bottom of the terminal window, there is a message: 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.

4. Unzip file # unzip filename

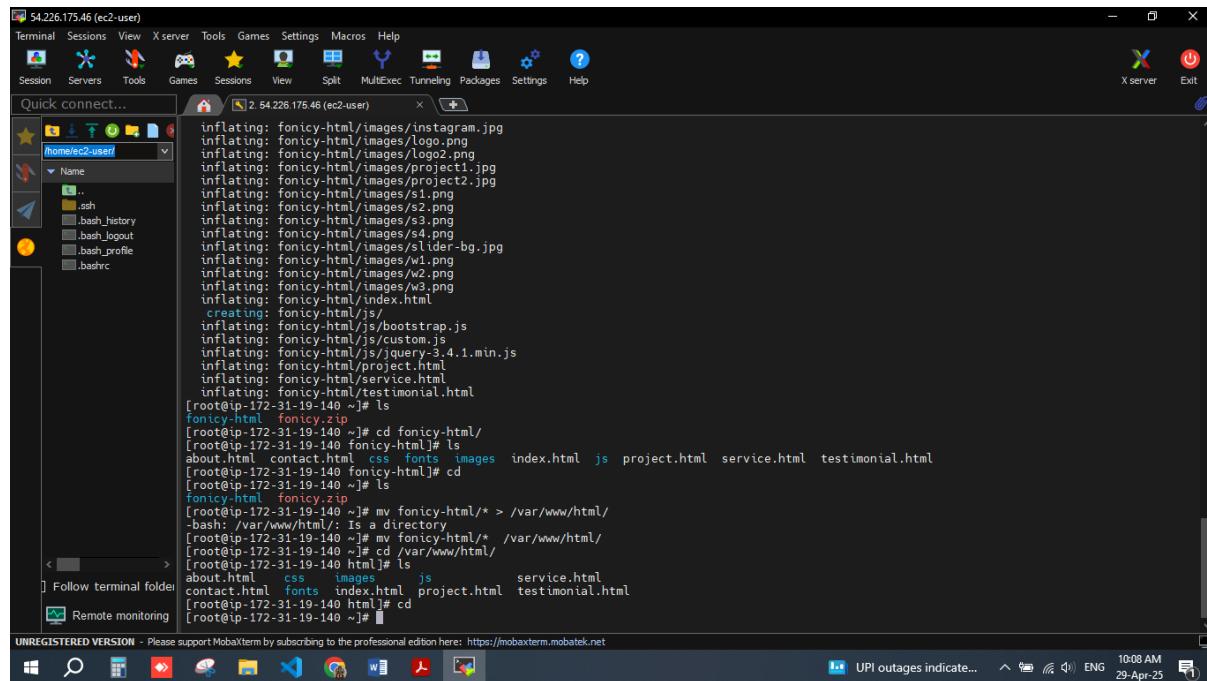
The screenshot shows a MobaXterm window titled '2.54.226.175.46 (ec2-user)'. The terminal window displays the output of an unzip command:

```
[root@ip-172-31-19-140 ~]# ls
fonicy.zip
[root@ip-172-31-19-140 ~]# unzip fonicy.zip
Archive: fonicy.zip
  creating: fonicy.html/
  inflating: fonicy.html/about.html
  inflating: fonicy.html/contact.html
  creating: fonicy.html/css/
  inflating: fonicy.html/css/bootstrap.css
  inflating: fonicy.html/css/font-awesome.min.css
  inflating: fonicy.html/css/responsive.css
  inflating: fonicy.html/css/style.css
  inflating: fonicy.html/css/style.css.map
  inflating: fonicy.html/css/style.scss
  creating: fonicy.html/fonts/
  inflating: fonicy.html/fonts/fontawesome-webfont.ttf
  inflating: fonicy.html/fonts/fontawesome-webfont.woff
  inflating: fonicy.html/fonts/fontawesome-webfont.woff2
  creating: fonicy.html/images/
  inflating: fonicy.html/images/about-img.jpg
  inflating: fonicy.html/images/client-bg.jpg
  inflating: fonicy.html/images/client.jpg
  inflating: fonicy.html/images/flag-france.png
  inflating: fonicy.html/images/flag-uk.png
  inflating: fonicy.html/images/instagram.jpg
  inflating: fonicy.html/images/logo.png
  inflating: fonicy.html/images/logo2.png
  inflating: fonicy.html/images/project1.jpg
  inflating: fonicy.html/images/project2.jpg
  inflating: fonicy.html/images/s1.png
  inflating: fonicy.html/images/s2.png
  inflating: fonicy.html/images/s3.png
  inflating: fonicy.html/images/s4.png
  inflating: fonicy.html/images/slider-bg.jpg
```

At the bottom of the terminal window, there is a message: 'UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: <https://mobaxterm.mobatek.net>'.

5. Move folder to /var/www/html/ directory

```
# mv foldername/* /var/www/html/
```

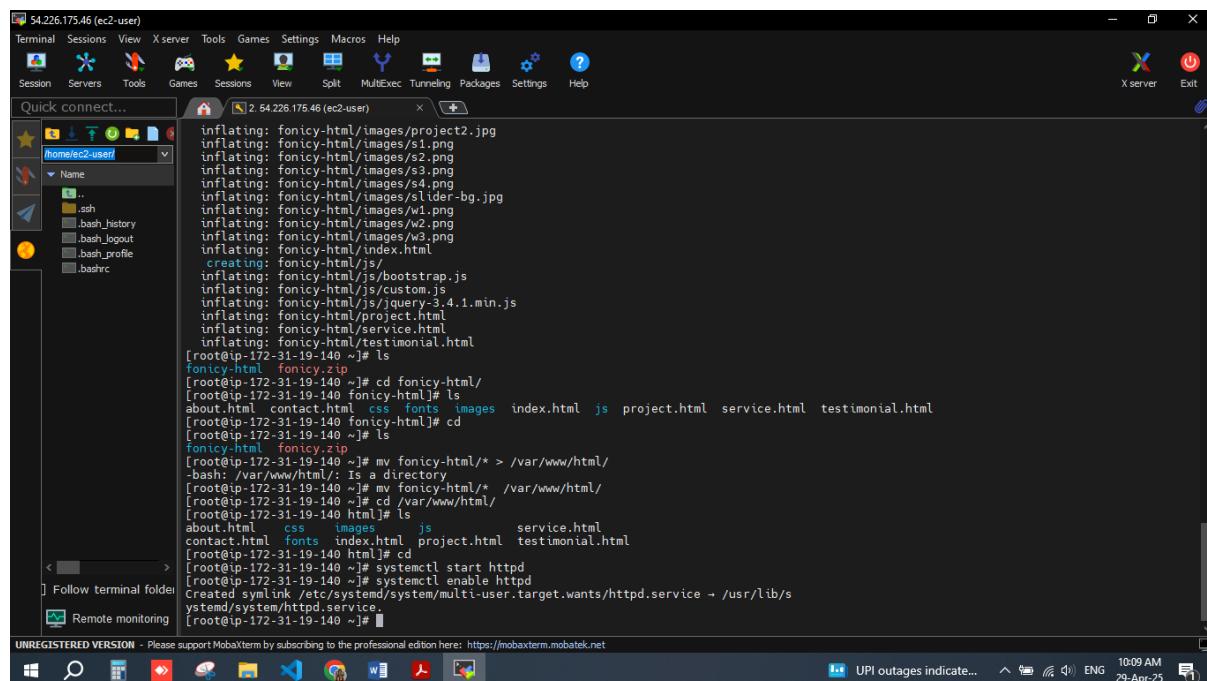


The screenshot shows a MobaXterm window titled "54.226.175.46 (ec2-user)". The terminal session is connected to a remote host at IP 54.226.175.46. The user is root. The terminal window displays the command `mv foldername/* /var/www/html/` being entered and its execution. The output shows the files being moved from the current directory to the /var/www/html directory. The terminal window has a dark theme with various icons in the toolbar and status bar.

```
inflating: fonicy-html/images/instagram.jpg
inflating: fonicy-html/images/logo.png
inflating: fonicy-html/images/logo2.png
inflating: fonicy-html/images/project1.jpg
inflating: fonicy-html/images/project2.jpg
inflating: fonicy-html/images/s1.png
inflating: fonicy-html/images/s2.png
inflating: fonicy-html/images/s3.png
inflating: fonicy-html/images/s4.png
inflating: fonicy-html/images/slider-bg.jpg
inflating: fonicy-html/images/w1.png
inflating: fonicy-html/images/w2.png
inflating: fonicy-html/images/w3.png
inflating: fonicy-html/index.html
inflating: fonicy-html/js/
inflating: fonicy-html/js/bootstrap.js
inflating: fonicy-html/js/custom.js
inflating: fonicy-html/js/jquery-3.4.1.min.js
inflating: fonicy-html/project.html
inflating: fonicy-html/service.html
inflating: fonicy-html/testimonial.html
[root@ip-172-31-19-140 ~]# ls
fonicy-html fonicy.zip
[root@ip-172-31-19-140 ~]# cd fonicy-html/
[root@ip-172-31-19-140 fonicy-html]# ls
about.html contact.htm css fonts images index.html js project.html service.html testimonial.html
[root@ip-172-31-19-140 fonicy-html]# cd ..
[root@ip-172-31-19-140 ~]# ls
fonicy-html fonicy.zip
[root@ip-172-31-19-140 ~]# mv fonicy-html/* > /var/www/html/
-bash: /var/www/html/: Is a directory
[root@ip-172-31-19-140 ~]# mv fonicy-html/* /var/www/html/
[root@ip-172-31-19-140 ~]# cd /var/www/html/
[root@ip-172-31-19-140 html]# ls
about.html contact.htm css fonts images index.html js project.html service.html testimonial.html
[root@ip-172-31-19-140 html]# cd ..
[root@ip-172-31-19-140 ~]# ls
fonicy-html fonicy.zip
[root@ip-172-31-19-140 ~]# mv fonicy-html/* > /var/www/html/
-bash: /var/www/html/: Is a directory
[root@ip-172-31-19-140 ~]# mv fonicy-html/* /var/www/html/
[root@ip-172-31-19-140 ~]# cd /var/www/html/
[root@ip-172-31-19-140 html]# ls
about.html contact.htm css fonts images index.html js project.html service.html testimonial.html
[root@ip-172-31-19-140 html]#
```

6. Start http service

```
# systemctl start httpd
# systemctl enable httpd
```



The screenshot shows a MobaXterm window titled "54.226.175.46 (ec2-user)". The terminal session is connected to a remote host at IP 54.226.175.46. The user is root. The terminal window displays the commands `systemctl start httpd` and `systemctl enable httpd` being entered and their execution. The output shows the creation of a symbolic link in the /etc/systemd/system directory. The terminal window has a dark theme with various icons in the toolbar and status bar.

```
inflating: fonicy-html/images/project2.jpg
inflating: fonicy-html/images/s1.png
inflating: fonicy-html/images/s2.png
inflating: fonicy-html/images/s3.png
inflating: fonicy-html/images/s4.png
inflating: fonicy-html/images/slider-bg.jpg
inflating: fonicy-html/images/w1.png
inflating: fonicy-html/images/w2.png
inflating: fonicy-html/images/w3.png
inflating: fonicy-html/index.html
inflating: fonicy-html/js/
inflating: fonicy-html/js/bootstrap.js
inflating: fonicy-html/js/custom.js
inflating: fonicy-html/js/jquery-3.4.1.min.js
inflating: fonicy-html/project.html
inflating: fonicy-html/service.html
inflating: fonicy-html/testimonial.html
[root@ip-172-31-19-140 ~]# ls
fonicy-html fonicy.zip
[root@ip-172-31-19-140 ~]# cd fonicy-html/
[root@ip-172-31-19-140 fonicy-html]# ls
about.html contact.htm css fonts images index.html js project.html service.html testimonial.html
[root@ip-172-31-19-140 fonicy-html]# cd ..
[root@ip-172-31-19-140 ~]# ls
fonicy-html fonicy.zip
[root@ip-172-31-19-140 ~]# mv fonicy-html/* > /var/www/html/
-bash: /var/www/html/: Is a directory
[root@ip-172-31-19-140 ~]# mv fonicy-html/* /var/www/html/
[root@ip-172-31-19-140 ~]# cd /var/www/html/
[root@ip-172-31-19-140 html]# ls
about.html contact.htm css fonts images index.html js project.html service.html testimonial.html
[root@ip-172-31-19-140 html]# cd ..
[root@ip-172-31-19-140 ~]# ls
fonicy-html fonicy.zip
[root@ip-172-31-19-140 ~]# mv fonicy-html/* > /var/www/html/
-bash: /var/www/html/: Is a directory
[root@ip-172-31-19-140 ~]# mv fonicy-html/* /var/www/html/
[root@ip-172-31-19-140 ~]# systemctl start httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-19-140 ~]#
```

Step 6: set inbound rule

1. Goto instances security tab then click on link under security groups

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with 'EC2' selected. Under 'Instances', 'Inbound rules' is expanded, showing a table with one row:

Name	Security group rule ID	Port range	Protocol	Source	Security groups
sgr-0fa065fc7a7488d1d	sg-07b6add085737d743	22	TCP	0.0.0.0/0	sg-07b6add085737d743 (launch-wizard-3)

2. edit inbound rule

The screenshot shows the AWS Security Groups page. On the left, there's a navigation sidebar with 'EC2' selected. Under 'Security Groups', a specific group named 'sg-07b6add085737d743 - launch-wizard-3' is selected. The 'Inbound rules' tab is active, displaying the same single rule as shown in the previous screenshot.

3. add new rule for http

The screenshot shows the 'Edit inbound rules' step of the EC2 Security Group wizard. It lists two rules:

- SSH (TCP 22) - Source: Custom (0.0.0.0/0)
- HTTP (TCP 80) - Source: Anywhere... (0.0.0.0/0)

A note at the bottom states: "⚠️ Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." At the bottom right are 'Cancel', 'Preview changes', and 'Save rules' buttons.The screenshot shows the 'Details' page for the security group 'sg-07b6add085737d743'. A green success message says: "Inbound security group rules successfully modified on security group (sg-07b6add085737d743 | launch-wizard-3) > Details". The page displays the following details:

Security group name	launch-wizard-3	Security group ID	sg-07b6add085737d743	Description	sg-07b6add085737d743 created 2025-04-29T04:11:49.567Z	VPC ID	vpc-02090bc4d4a645c0d
Owner	322471216536	Inbound rules count	2 Permission entries	Outbound rules count	1 Permission entry		

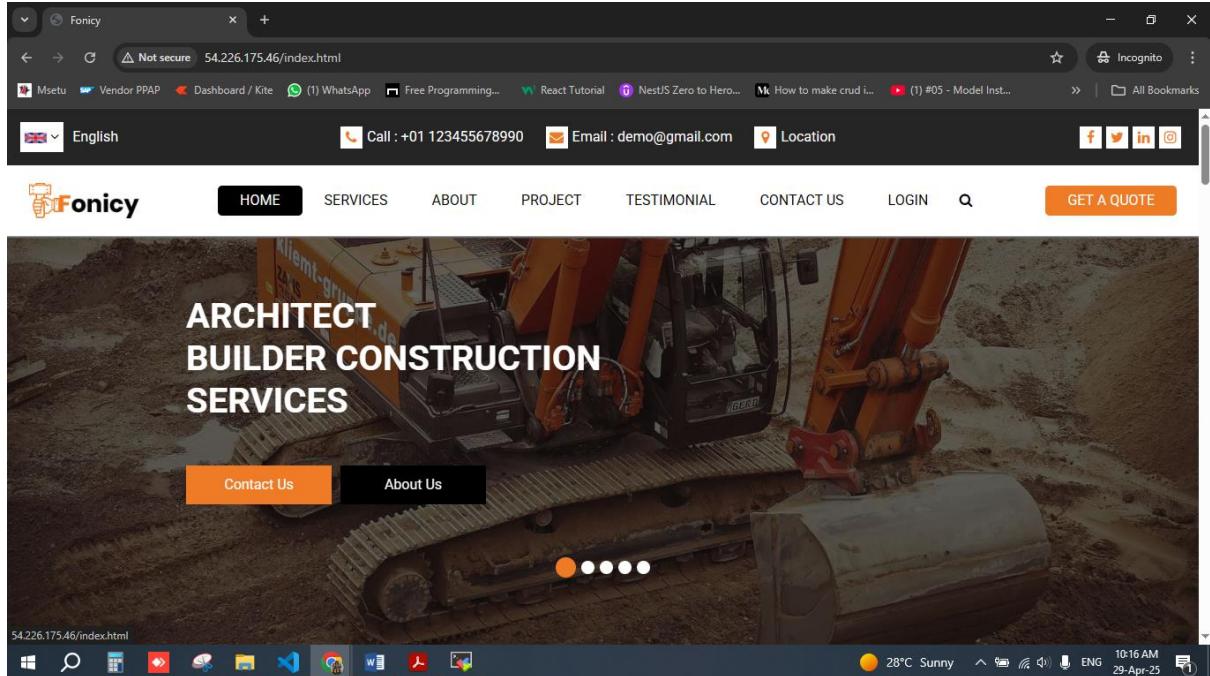
The 'Inbound rules' tab is selected, showing a table with two entries:

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
-	sgr-0fa065fcca7488d1d	IPv4	SSH	TCP	22	0.0.0.0/0
-	sgr-0d39bd09868104c65e	IPv4	HTTP	TCP	80	0.0.0.0/0

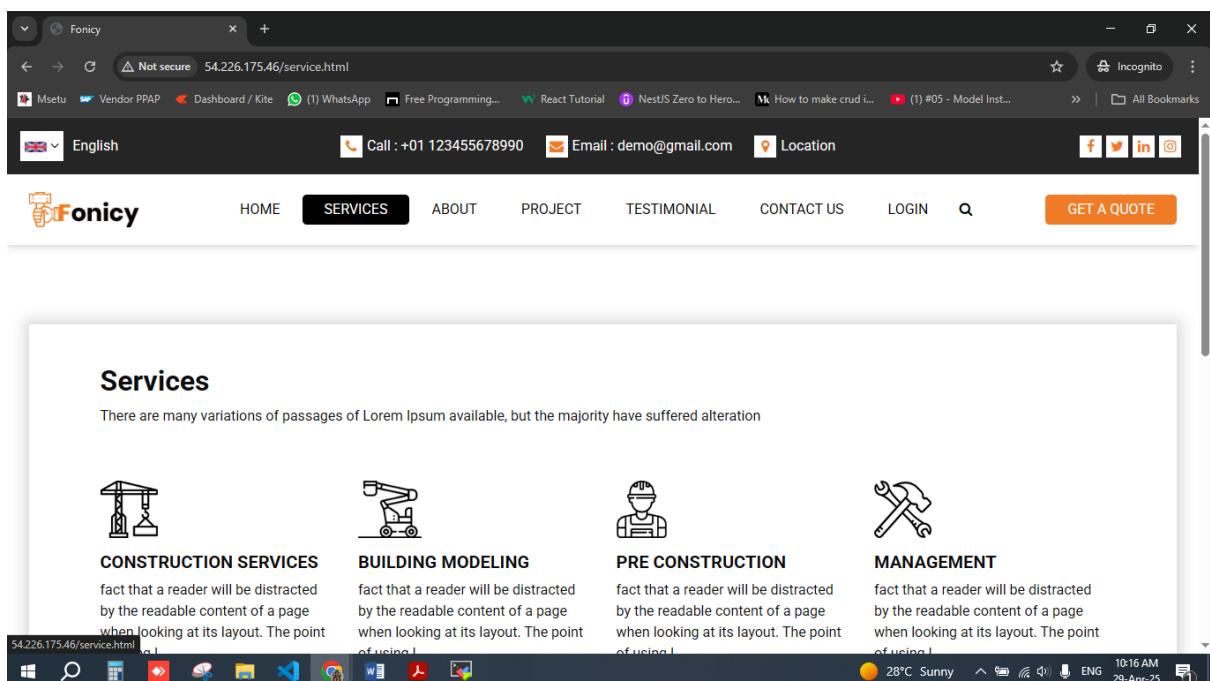
4. hit public IP

Web Pages:

1. Home Page



2. Service Page



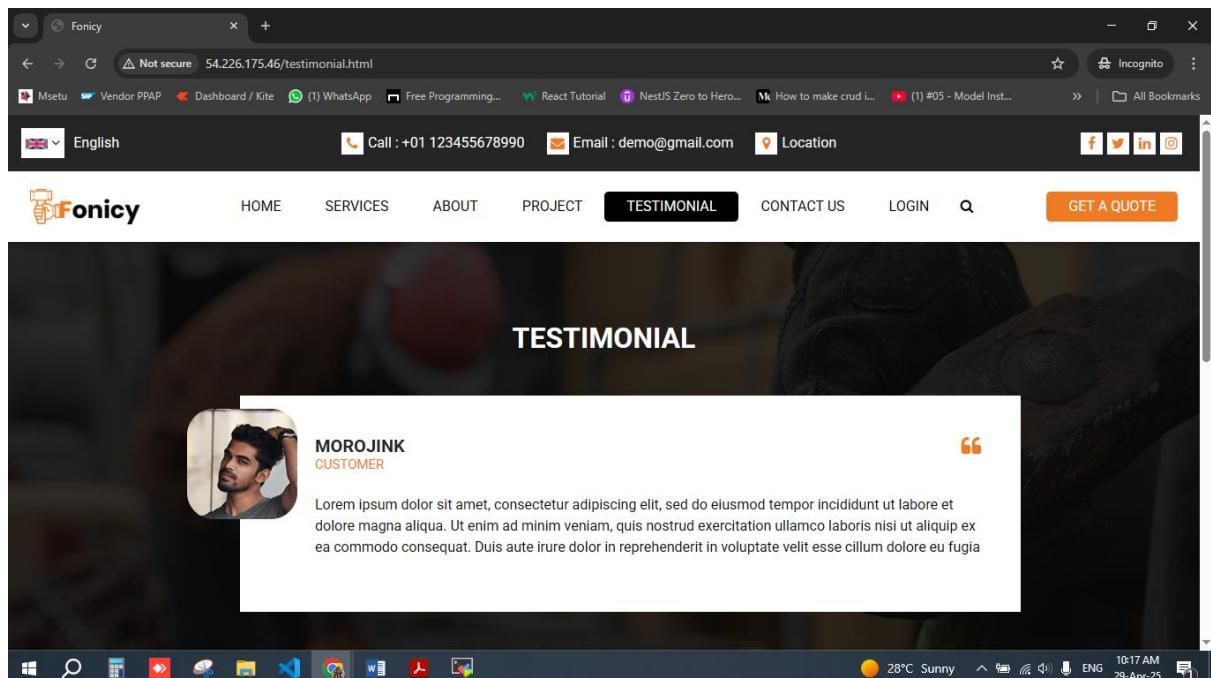
3. About Page

The screenshot shows a web browser window for the Fonicy website at the URL 54.226.175.46/about.html. The page has a dark header bar with the Fonicy logo, a language dropdown set to English, and a navigation menu with links for HOME, SERVICES, ABOUT (which is highlighted in black), PROJECT, TESTIMONIAL, CONTACT US, LOGIN, and a search icon. Below the menu is a large orange "GET A QUOTE" button. The main content area features a heading "About Us" and a paragraph of placeholder text (Lorem Ipsum). A "Read More" button is visible. To the right of the text is a photograph of a person wearing a red hard hat and safety glasses, working with a power tool on a piece of metal. The browser's address bar shows "Not secure". The taskbar at the bottom of the screen displays various application icons and the system clock.

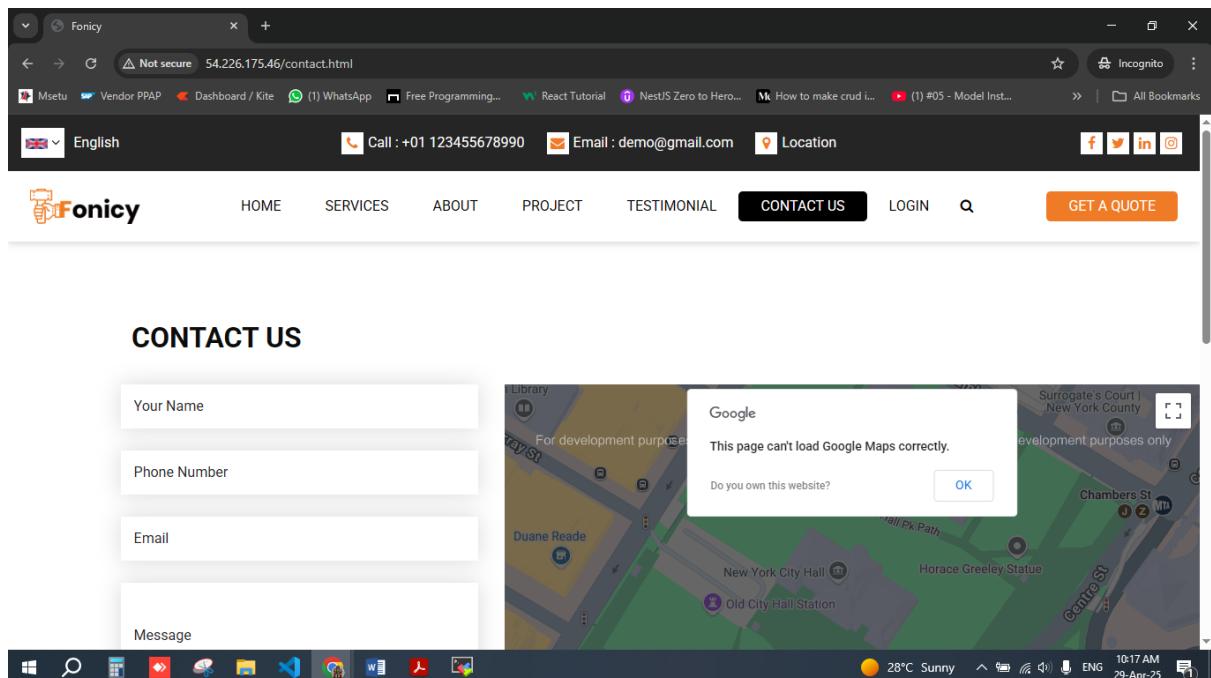
4. Projects Page

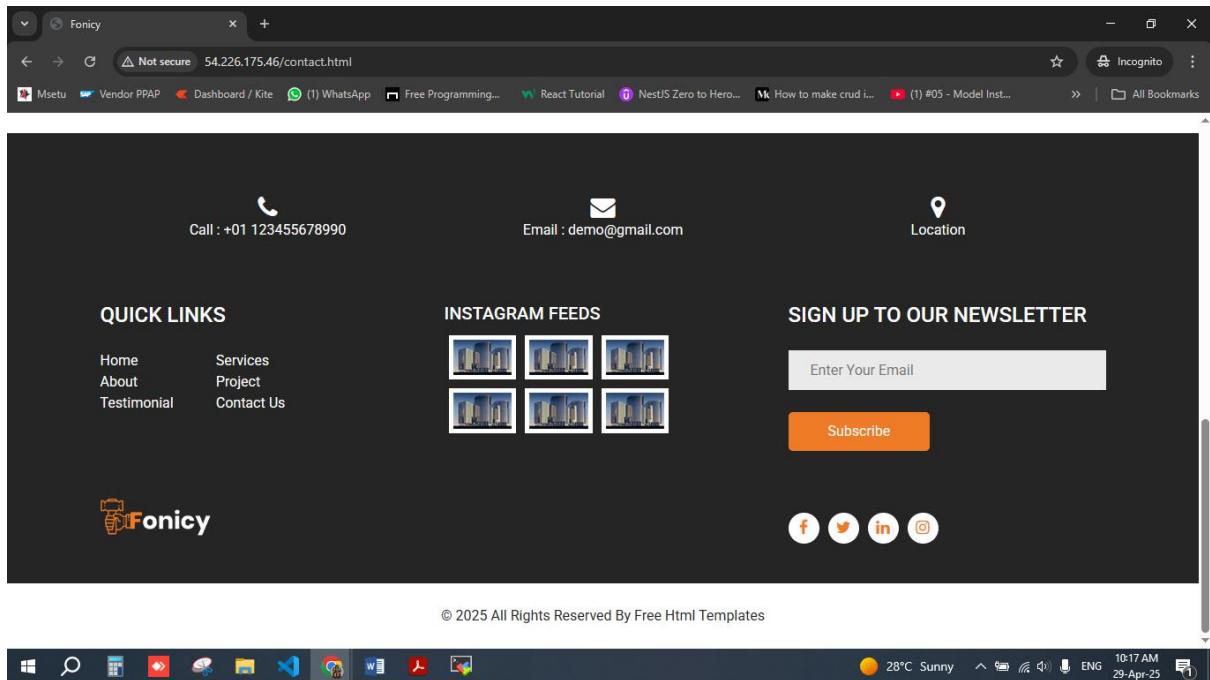
The screenshot shows a web browser window for the Fonicy website at the URL 54.226.175.46/project.html. The layout is similar to the About page, with a dark header bar, English language selection, and a navigation menu. The PROJECT link is highlighted in black. The main content area features a heading "PROJECTS" and a "CATEGORY FILTER" section with options: ALL, PAINTING, RECONSTRUCTION, REPAIRS, RESIDENTIAL, and STYLING. Below this are three images: a modern interior room with a TV and bookshelves, and two images of a city skyline with skyscrapers. The browser's address bar shows "Not secure". The taskbar at the bottom of the screen displays various application icons and the system clock.

5. Testimonial Page



6. Contact Page





❖ **Outcome:**

- Successfully deployed a fully functional website accessible via a web browser.
- Gained practical experience in setting up Linux servers, working with AWS cloud services, and configuring security settings.
- Understood server management, basic networking (DNS, ports), and cloud hosting fundamentals.

❖ **Future Improvements:**

- Implement automatic backups and server monitoring.
- Set up a Load Balancer and Auto Scaling Groups for high availability.
- Deploy HTTPS using a reverse proxy with Nginx.
- Automate deployments using CI/CD pipelines.