

Name: Hrishikesh Kumbhar

Div: D15A

Roll no: 32

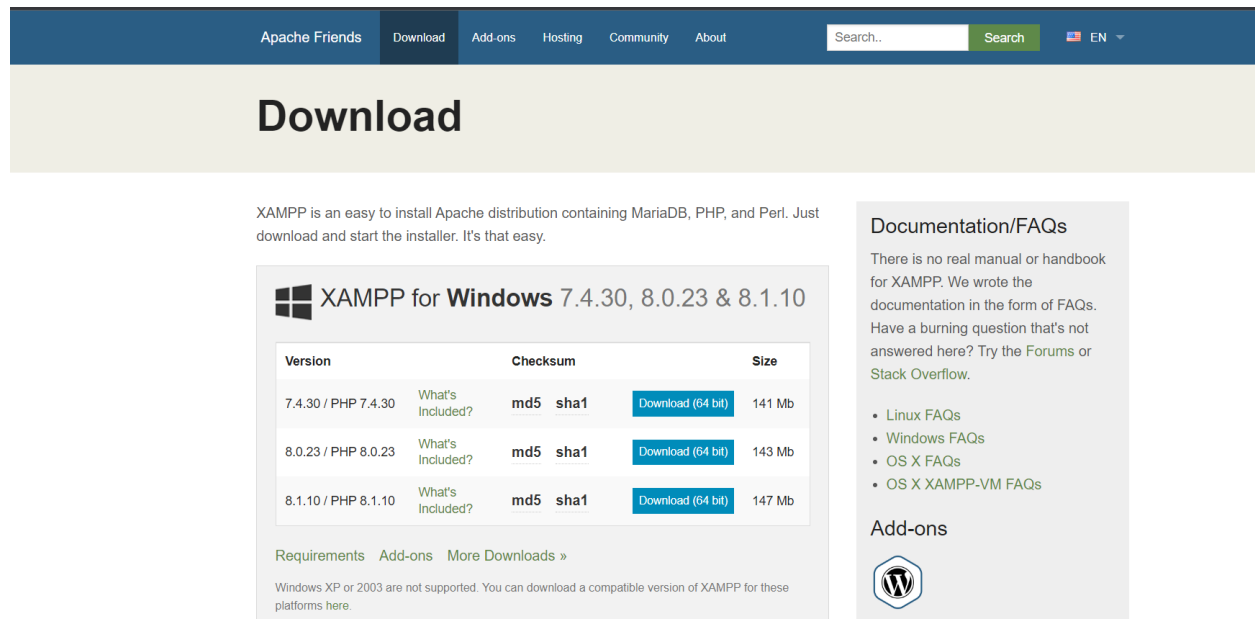
Sub: Advanced DevOps

Assignment No: 01

Implementation: Part 1: Hosting a website on localhost

Step 1: Downloading XAMPP. We need to install XAMPP, which creates an apache server for us on our local machine, on which we can host our website.

You can also use applications like WAMP or MAMP depending on your Operating System. Download link: <https://www.apachefriends.org/download.html>



The screenshot shows the Apache Friends website's download page. The header includes navigation links: Apache Friends, Download, Add-ons, Hosting, Community, and About. A search bar and language selector (EN) are also present. The main heading is "Download". Below this, a text block states: "XAMPP is an easy to install Apache distribution containing MariaDB, PHP, and Perl. Just download and start the installer. It's that easy." The central part of the page features a table for XAMPP for Windows versions 7.4.30, 8.0.23, and 8.1.10. Each row includes the version, a link to "What's Included?", checksums (md5, sha1), a "Download (64 bit)" button, and the file size. To the right, a "Documentation/FAQs" section explains that there is no manual, but provides links to Linux, Windows, OS X, and OS X XAMPP-VM FAQs. Below that, an "Add-ons" section features the WordPress logo.

Version	Checksum	Size
7.4.30 / PHP 7.4.30	What's Included? md5 sha1	Download (64 bit) 141 Mb
8.0.23 / PHP 8.0.23	What's Included? md5 sha1	Download (64 bit) 143 Mb
8.1.10 / PHP 8.1.10	What's Included? md5 sha1	Download (64 bit) 147 Mb

Requirements Add-ons More Downloads »


Windows XP or 2003 are not supported. You can download a compatible version of XAMPP for these platforms [here](#).

Documentation/FAQs

There is no real manual or handbook for XAMPP. We wrote the documentation in the form of FAQs. Have a burning question that's not answered here? Try the Forums or Stack Overflow.

- Linux FAQs
- Windows FAQs
- OS X FAQs
- OS X XAMPP-VM FAQs

Add-ons



****Note:** Make sure you select Apache when you install XAMPP, so that you can use it with XAMPP later.**

Step 2: Dropping the Code folder in the htdocs folder. Go to the xampp root folder, then locate the htdocs folder, such that you are now in xampp/htdocs. Now, open the Terminal here and use the git clone command to clone your code folder in this directory.

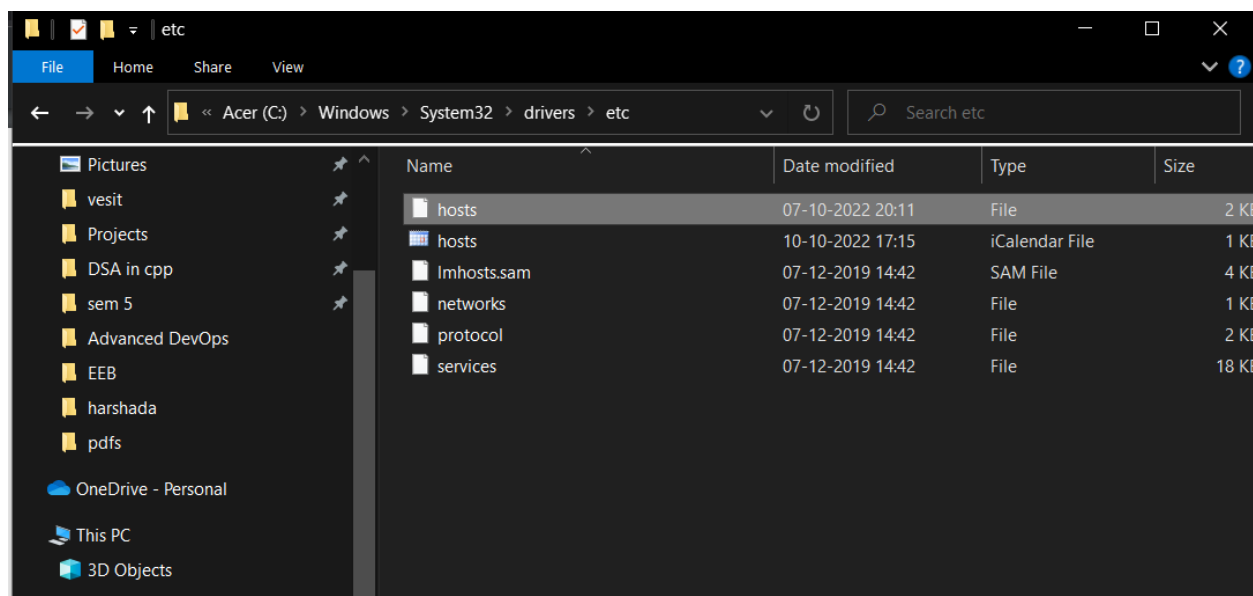
```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

D:\XAMPP\htdocs>git clone https://github.com/Hrishikesh156/Profile-Page.git
Cloning into 'Profile-Page'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (19/19), done.
Receiving objects: 100% (24/24), 9.58 MiB | 3.41 MiB/s, done.

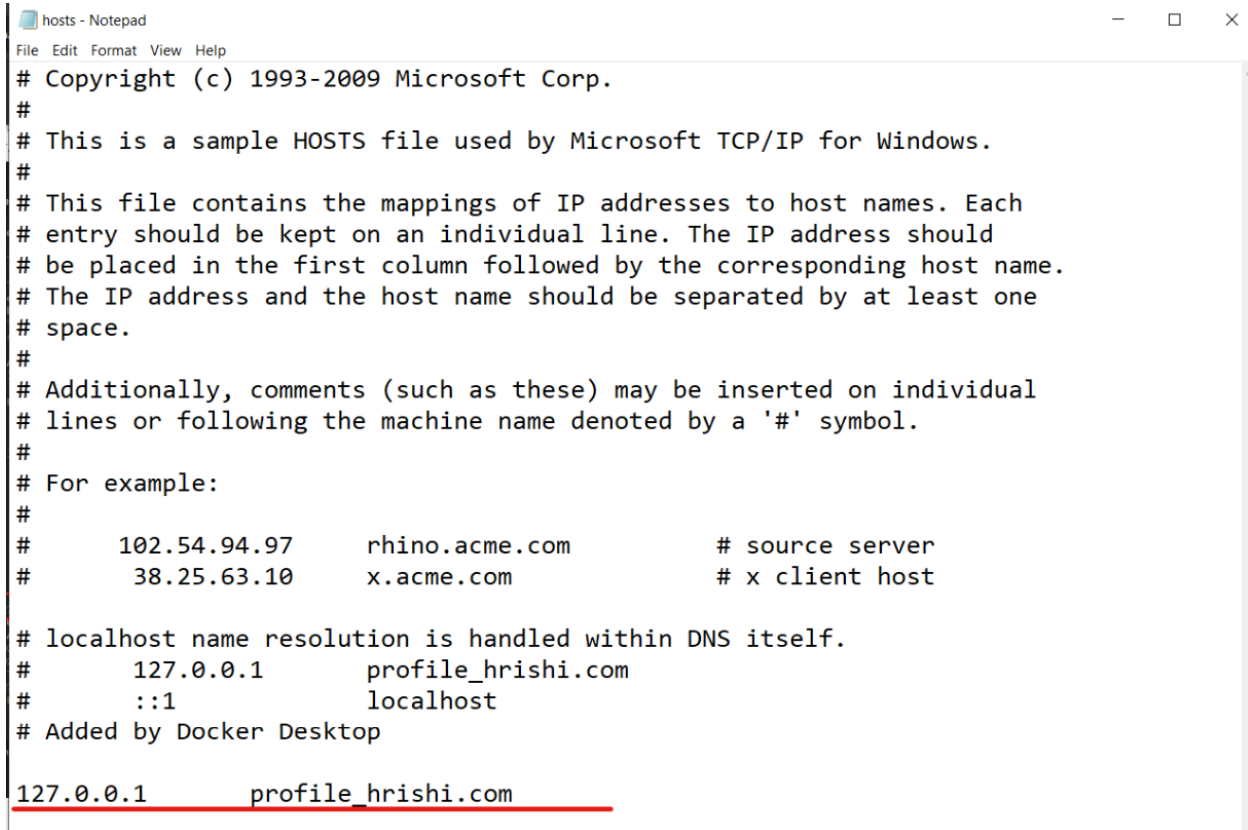
Resolving deltas: 100% (4/4), done.

D:\XAMPP\htdocs>
```

Step 3: Update the hosts file to serve localhost as your domain name (will work only on your local machine). Open the notepad as administrator and open the hosts file in Windows/System32/drivers/etc. Change the filter to All Files to find the hosts file.

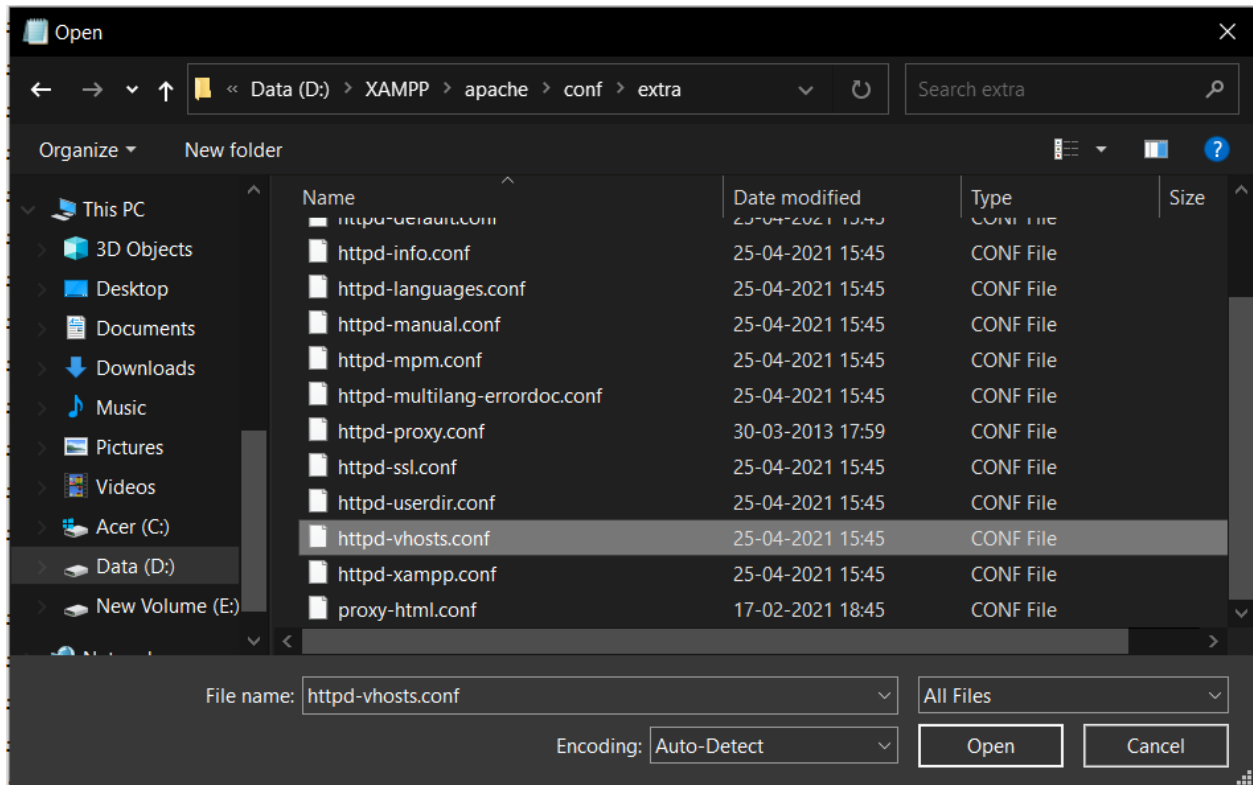


Then, on a new line, enter localhost and map it with your desired domain name, in my case, it is profile_hrishi.com, as shown.



```
hosts - Notepad
File Edit Format View Help
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97       rhino.acme.com           # source server
#       38.25.63.10       x.acme.com               # x client host
#
# localhost name resolution is handled within DNS itself.
#       127.0.0.1        profile_hrishi.com
#       ::1              localhost
# Added by Docker Desktop
127.0.0.1        profile_hrishi.com
```

Step 4: Update the httpd-vhosts.conf file Locate the httpd-vhosts.conf file in xampp/apache/conf/extra. Open this file with your desired text editor, in my case, notepad.



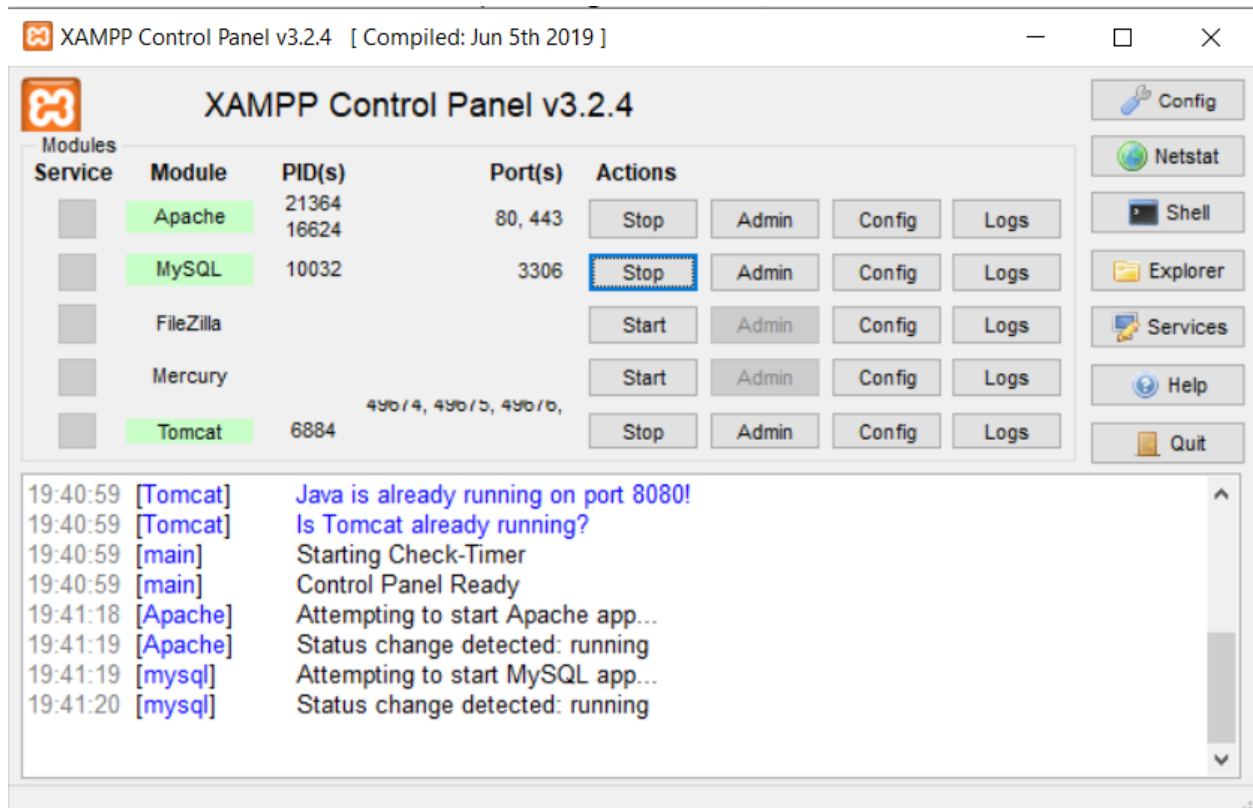
All the code in this file is commented out. Copy the last VirtualHost set which is commented out and paste it out, like so. Inside this tag, make the Document root mapped to the website folder, like so. Change the server name to the mapped domain name, as you did in the hosts folder.

```
#
# Use name-based virtual hosting.
#
##NameVirtualHost *:80
#
# VirtualHost example:
# Almost any Apache directive may go into a VirtualHost container.
# The first VirtualHost section is used for all requests that do not
# match a ##ServerName or ##ServerAlias in any <VirtualHost> block.
#
##<VirtualHost *:80>
    ##ServerAdmin webmaster@dummy-host.example.com
    ##DocumentRoot "D:/XAMPP/htdocs/dummy-host.example.com"
    ##ServerName dummy-host.example.com
    ##ServerAlias www.dummy-host.example.com
    ##ErrorLog "logs/dummy-host.example.com-error.log"
    ##CustomLog "logs/dummy-host.example.com-access.log" common
##</VirtualHost>

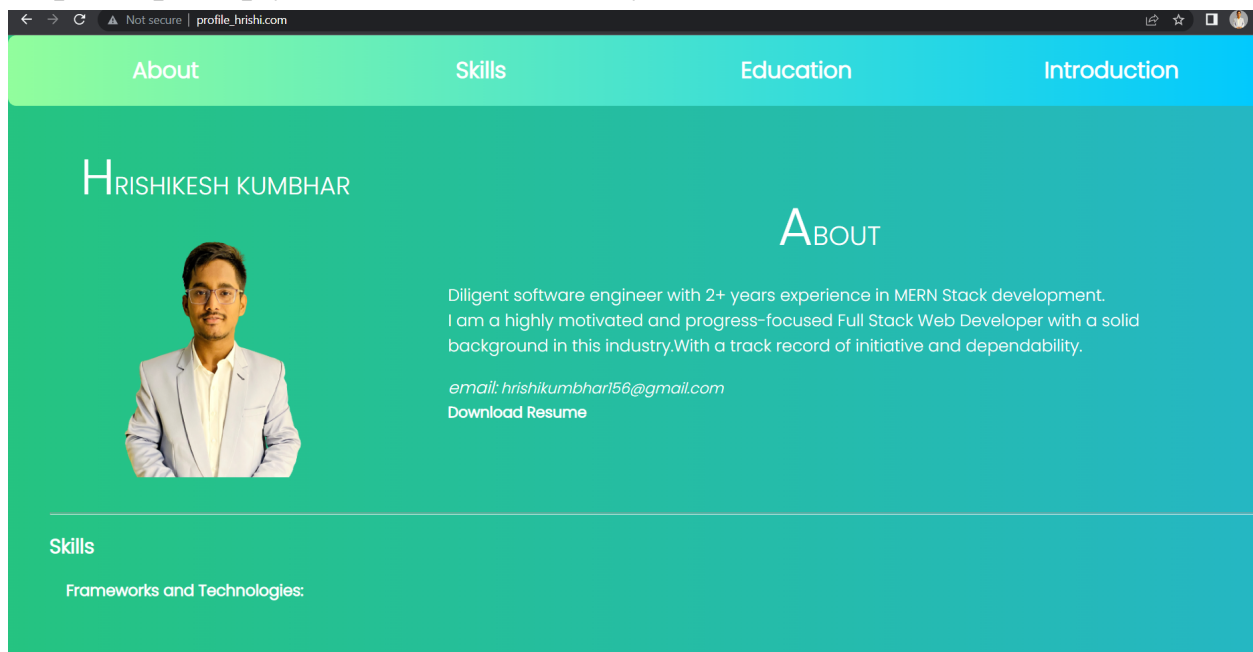
<VirtualHost *:80>
    DocumentRoot "D:/XAMPP/htdocs/Profile-Page"
    ServerAdmin profile_hrishi.com

</VirtualHost>
```

Step 5: Open the XAMPP control panel and start the apache server



Step 6: Open up your domain name on your browser



Part 2: Hosting a website on a Cloud VM (AWS EC2 Instance) To host our website on Cloud, we need to set up a Virtual Machine, in AWS EC2, in this case.

You'd need an AWS free tier account to proceed.

Step 1: Open up EC2 Console and Launch a new Instance

Choose the Linux 2 AMI , t2.micro for configuration, which is the free-tier-eligible one.

Name

web_hosting [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

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

Quick Start

Amazon Linux aws	macOS Mac	Ubuntu ubuntu	Windows Microsoft	Red Hat Red Hat	S
----------------------------	--------------	------------------	----------------------	--------------------	---

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type ami-026b57f3c383c2eec (64-bit (x86)) / ami-0636eac5d73e0e5d7 (64-bit (Arm)) Virtualization: hvm ENA enabled: true Root device type: ebs	Free tier eligible ▼
--	-----------------------------

Proceed with everything default, in Security groups, add these rules so that you can visit the website from anywhere.

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-6**' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance

Anywhere
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. [X](#)

Review the changes and launch the instance

[EC2](#) > [Instances](#) > Launch an instance

Launching instance

Please wait while we launch your instance.

Do not close your browser while this is loading.

Launch initiation

69%

[Details](#)

Step 2: Connect to the instance to access the CLI

```

  _ | _ | _ )
  _ | ( _ - /   Amazon Linux 2 AMI
  _ |\ _ | _ |

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-86-138 ~]$
```

i-05510e97b4316ea86 (web_hosting)
PublicIPs: 54.166.184.44 PrivateIPs: 172.31.86.138

Step 3: Install Git and HTTPD Use the following commands to install git and httpd.

\$yum install git -y

```

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-86-138 ~]$ yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ec2-user@ip-172-31-86-138 ~]$ sudo su
[root@ip-172-31-86-138 ec2-user]# yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.37.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl-Git = 2.37.1-1.amzn2.0.1 for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core-doc = 2.37.1-1.amzn2.0.1 for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: git-core = 2.37.1-1.amzn2.0.1 for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git::I18N) for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Processing Dependency: perl(Git) for package: git-2.37.1-1.amzn2.0.1.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.37.1-1.amzn2.0.1 will be installed
--> Package git-core-doc.noarch 0:2.37.1-1.amzn2.0.1 will be installed
--> Package perl-Git.noarch 0:2.37.1-1.amzn2.0.1 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.37.1-1.amzn2.0.1.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

```

\$yum install httpd -y

```

[ec2-user@ip-172-31-86-138 ~]$ sudo su
[root@ip-172-31-86-138 ec2-user]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Existing lock /var/run/yum.pid: another copy is running as pid 3735.
Another app is currently holding the yum lock; waiting for it to exit...
  The other application is: yum
    Memory : 165 M RSS (384 MB VSZ)
    Started: Mon Oct 10 19:21:40 2022 - 00:05 ago
    State   : Running, pid: 3735
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.54-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem = 2.4.54-1.amzn2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: httpd-filesystem for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: /etc/mime.types for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libaprutil-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Processing Dependency: libapr-1.so.0()(64bit) for package: httpd-2.4.54-1.amzn2.x86_64
--> Running transaction check
--> Package apr.x86_64 0:1.7.0-9.amzn2 will be installed
--> Package apr-util.x86_64 0:1.6.1-5.amzn2.0.2 will be installed
--> Processing Dependency: apr-util-bdb(x86-64) = 1.6.1-5.amzn2.0.2 for package: apr-util-1.6.1-5.amzn2.0.2.x86_64
--> Package generic-logos-httpd.noarch 0:18.0.0-4.amzn2 will be installed

```

Step 4: Set up the GitHub Repository to host the website Go to var/www/html and clone the GitHub Repository.

cd /var/www/html

\$git clone <https://github.com/Hrishikesh156/Profile-Page.git>

```
[root@ip-172-31-86-138 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-86-138 ec2-user]# cd /var/www/html
[root@ip-172-31-86-138 html]# https://github.com/Hrishikesh156/Profile-Page.git
bash: https://github.com/Hrishikesh156/Profile-Page.git: No such file or directory
[root@ip-172-31-86-138 html]# git clone https://github.com/Hrishikesh156/Profile-Page.git
Cloning into 'Profile-Page'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (19/19), done.
remote: Total 24 (delta 4), reused 22 (delta 3), pack-reused 0
Receiving objects: 100% (24/24), 9.58 MiB | 38.15 MiB/s, done.
Resolving deltas: 100% (4/4), done.
[root@ip-172-31-86-138 html]#
```

Type the ls command just to confirm you have all your html files inside the html directory.

```
[root@ip-172-31-86-138 html]# ls
Profile-Page
[root@ip-172-31-86-138 html]#
```

Move the folder outside.

```
[root@ip-172-31-86-138 html]# ls
Profile-Page
[root@ip-172-31-86-138 html]# mv Profile-Page/* .
[root@ip-172-31-86-138 html]# ls
css index.css index.html media Profile-Page
[root@ip-172-31-86-138 html]#
```

Step 5: Start the httpd server

```
[root@ip-172-31-86-138 html]# pwd
/var/www/html
[root@ip-172-31-86-138 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-86-138 html]#
```

Step 6: View your website Create a new tab on your browser and go to the public IP address of your EC2 instance to confirm your website is live. You can find the Public IP address inside the Networking tab of your EC2 console.

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
You can now check network connectivity with Reachability Analyzer. Run Reachability Analyzer ×						
▼ Networking details Info						
Public IPv4 address 54.166.184.44 open address			Private IPv4 addresses 172.31.86.138		VPC ID vpc-0031341b11217a02f	
Public IPv4 DNS ec2-54-166-184-44.compute-1.amazonaws.com open address			Private IP DNS name (IPv4 only) ip-172-31-86-138.ec2.internal			
Subnet ID subnet-017a78f0296f8ad5b			IPv6 addresses -		Secondary private IPv4 addresses -	
Availability zone us-east-1c			Carrier IP addresses (ephemeral) -		Outpost ID -	

Your website is live!

