Name: Hrishikesh Kumbhar

Div: D15A

Roll no: 32

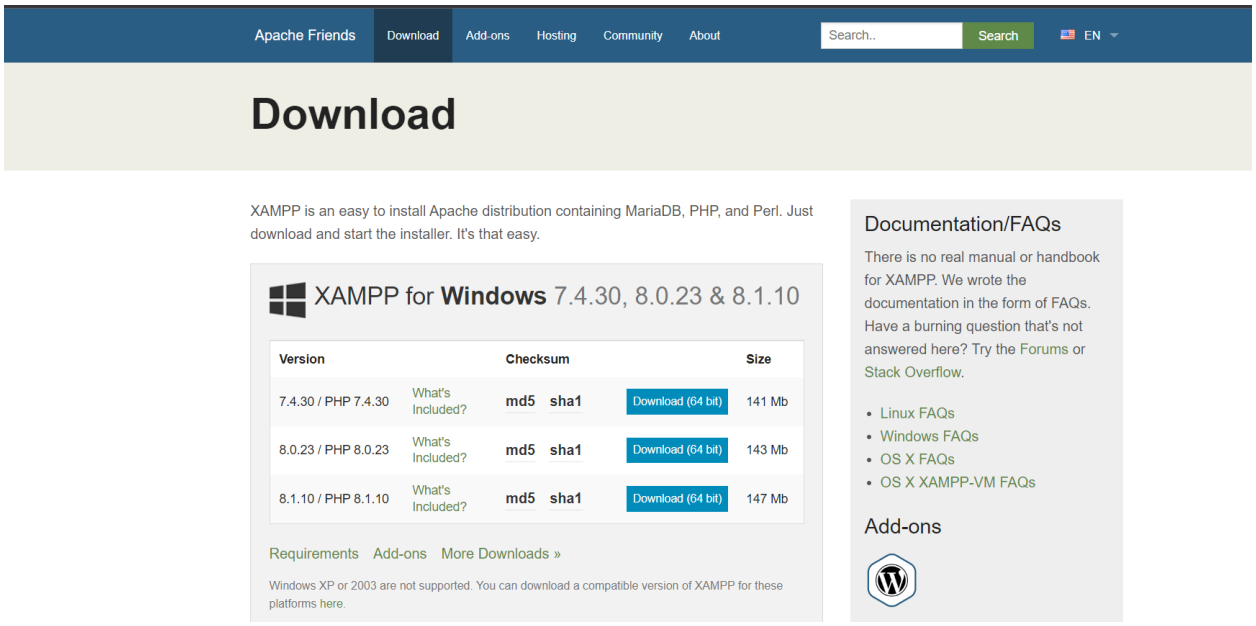
Sub: Advanced DevOps

Assignment No: 01

Implementation: Part 1: Hosting a website on localhost

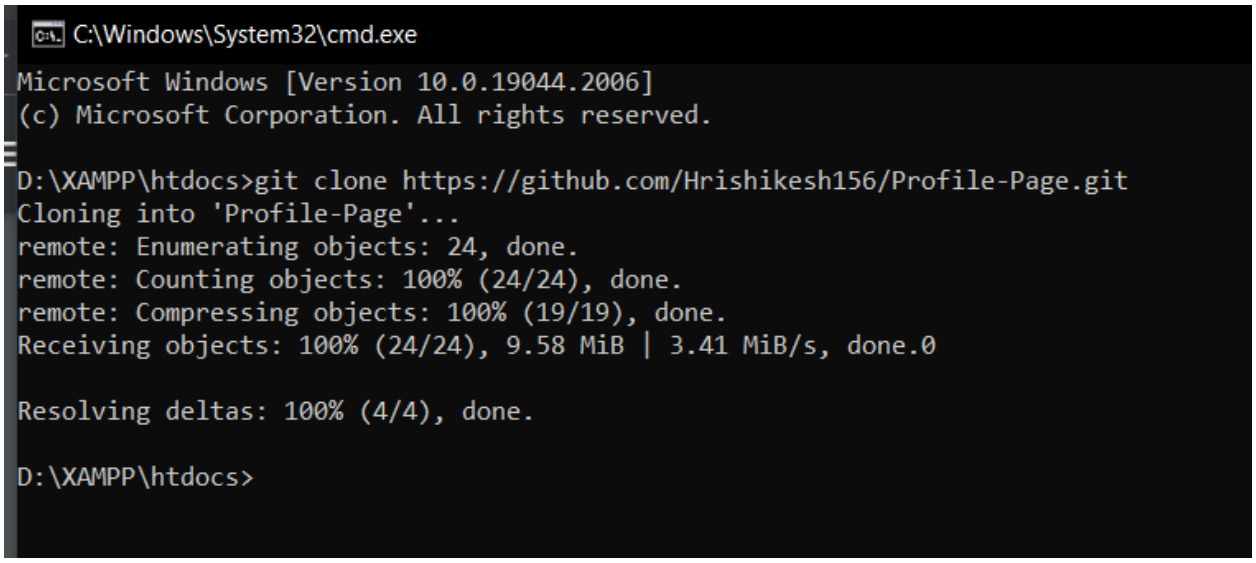
Step 1: Downloading XAMPP. We need to install XAMPP, which creates an apache server for us on our local machine, on which we can host our website.

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

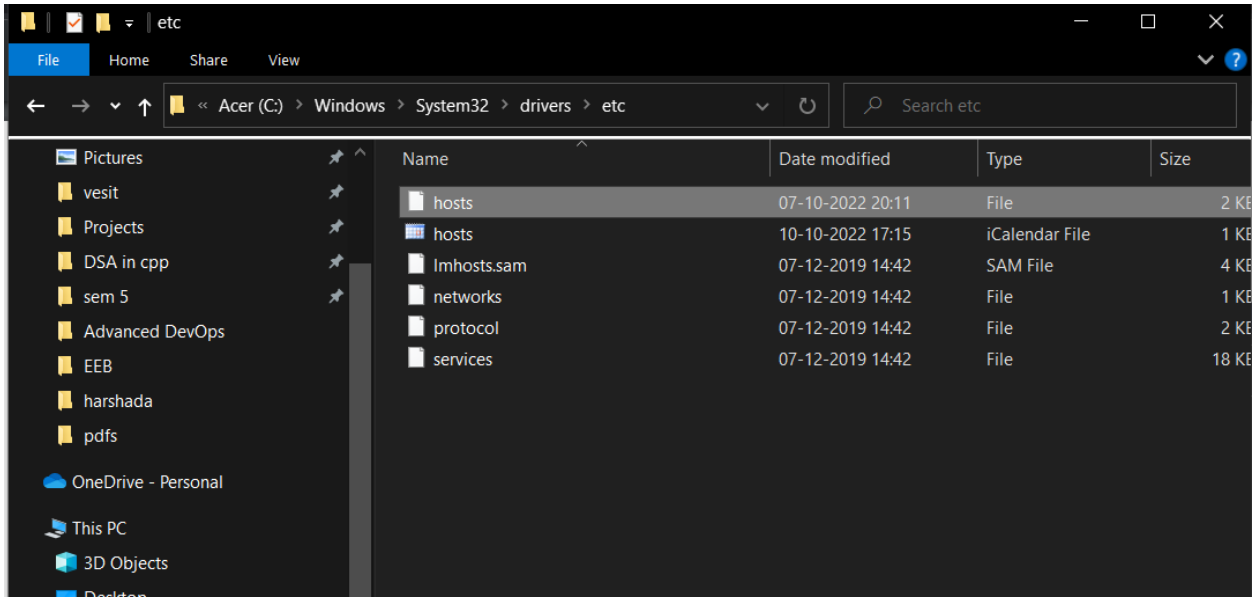


Note: Make sure you select Apache when you install XAMPP, so that you can use it with XAMPP later.

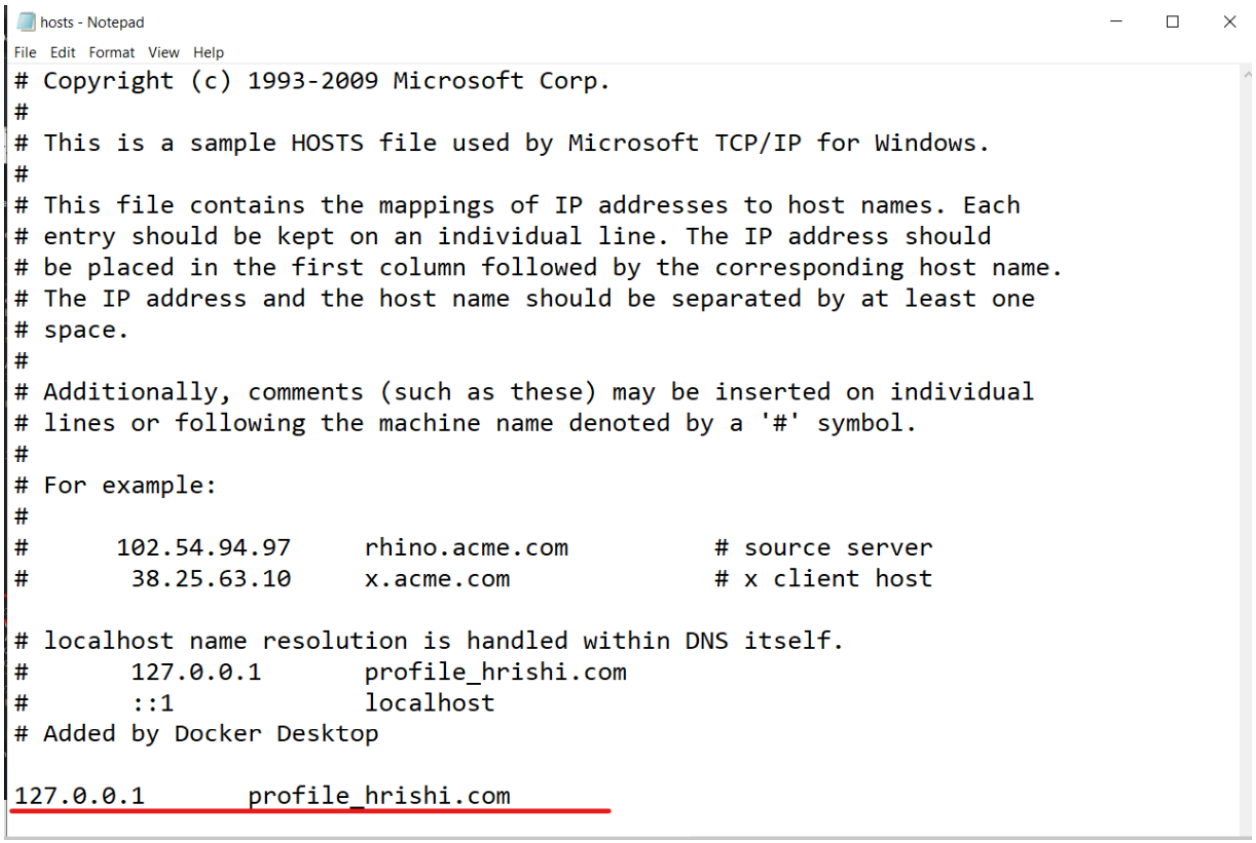
Step 2: Dropping the Code folder in the htdocs folder. Go to the xampp root folder, then locate the htdocs folder, such that you are now in xampp/htdocs. Now, open the Terminal here and use the git clone command to clone your code folder in this directory.



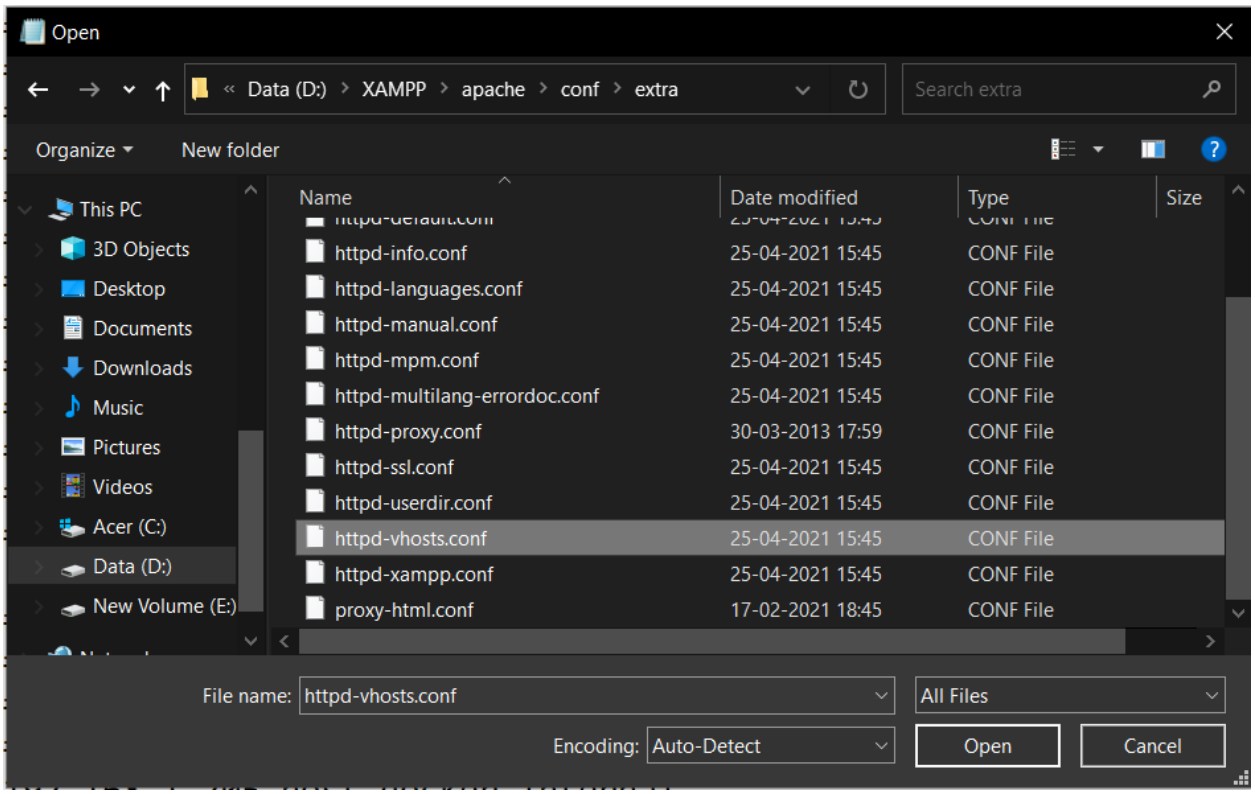
Step 3: Update the hosts file to serve localhost as your domain name (will work only on your local machine). Open the notepad as administrator and open the hosts file in Windows/System32/drivers/etc. Change the filter to All Files to find the hosts file.



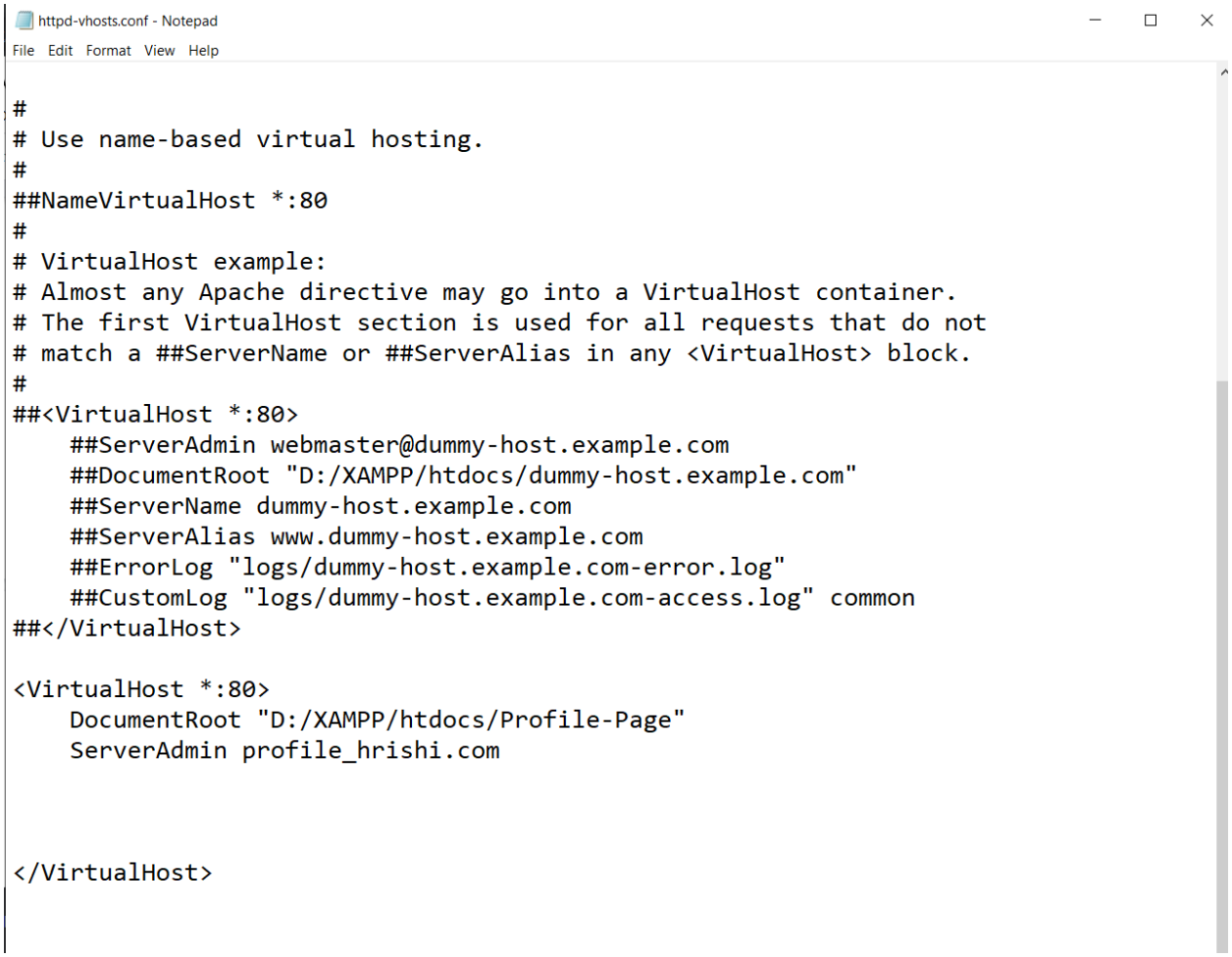
Then, on a new line, enter localhost and map it with your desired domain name, in my case, it is profile_hrishi.com, as shown.



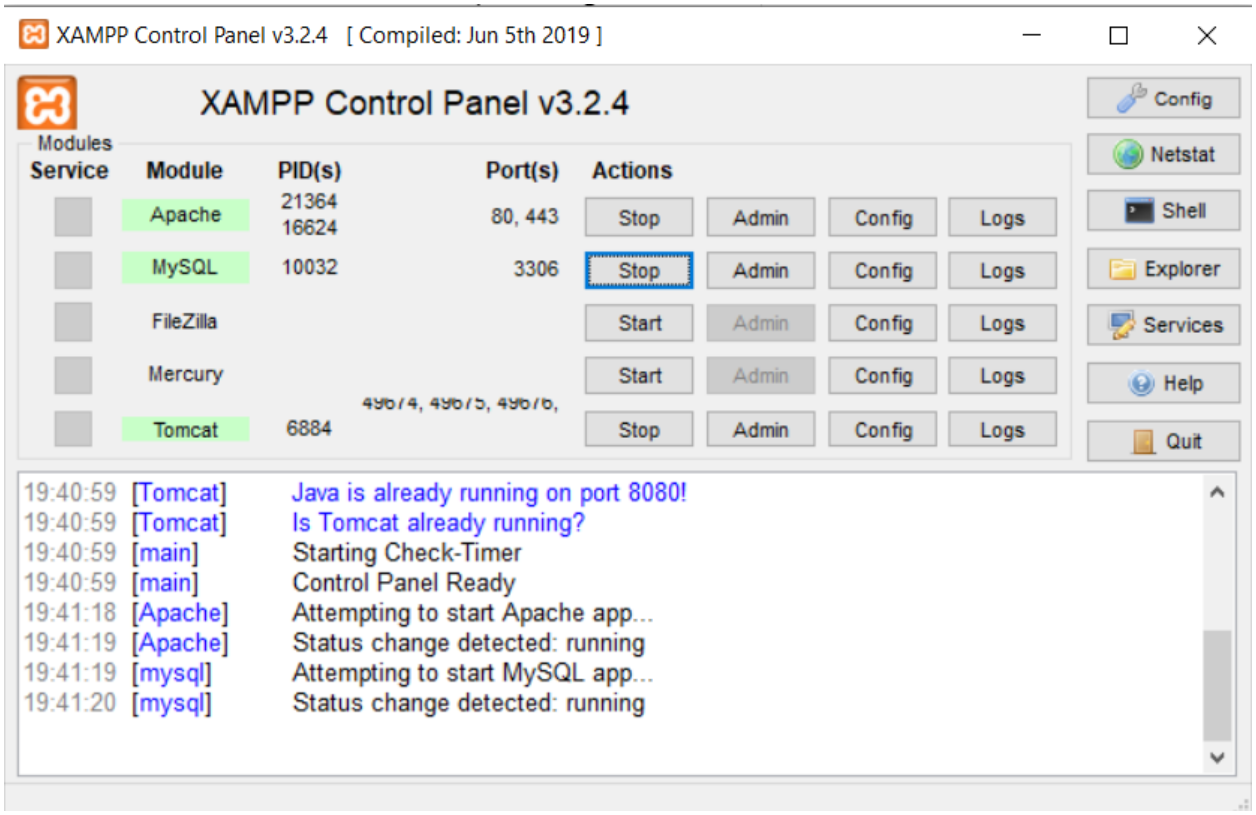
Step 4: Update the httpd-vhosts.conf file Locate the httpd-vhosts.conf file in xampp/apache/conf/extra. Open this file with your desired text editor, in my case, notepad.



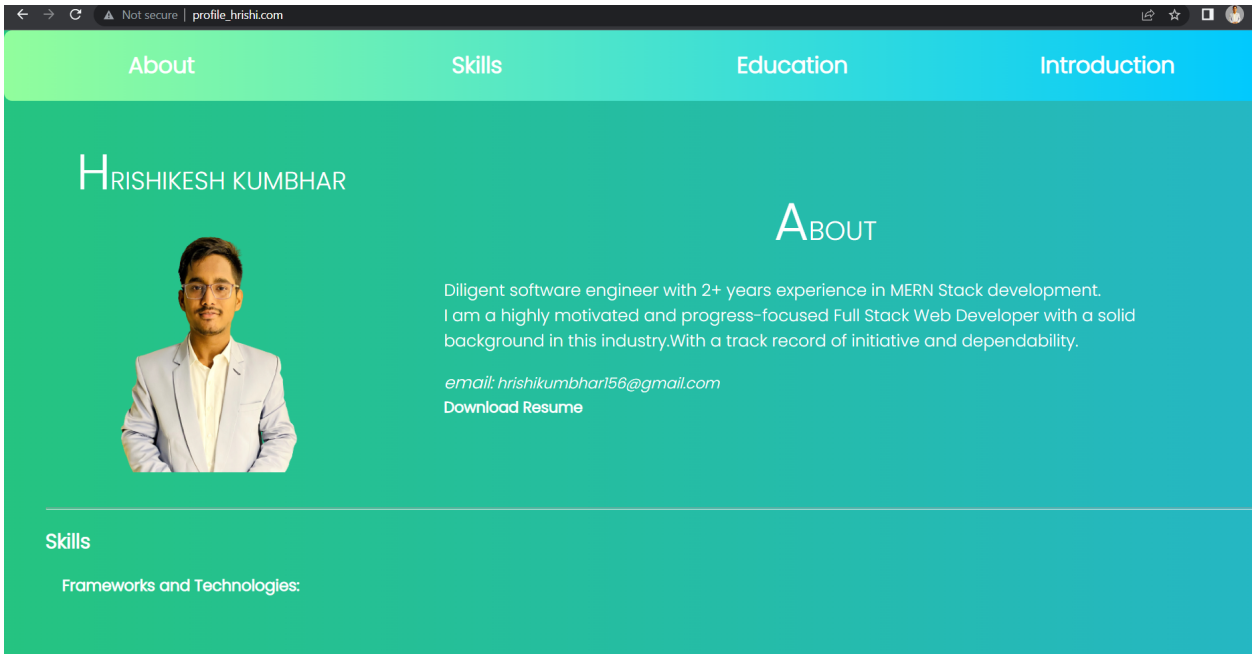
All the code in this file is commented out. Copy the last VirtualHost set which is commented out and paste it out, like so. Inside this tag, make the Document root mapped to the website folder, like so. Change the server name to the mapped domain name, as you did in the hosts folder.


httpd-vhosts.conf - Notepad
File Edit Format View Help
#
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

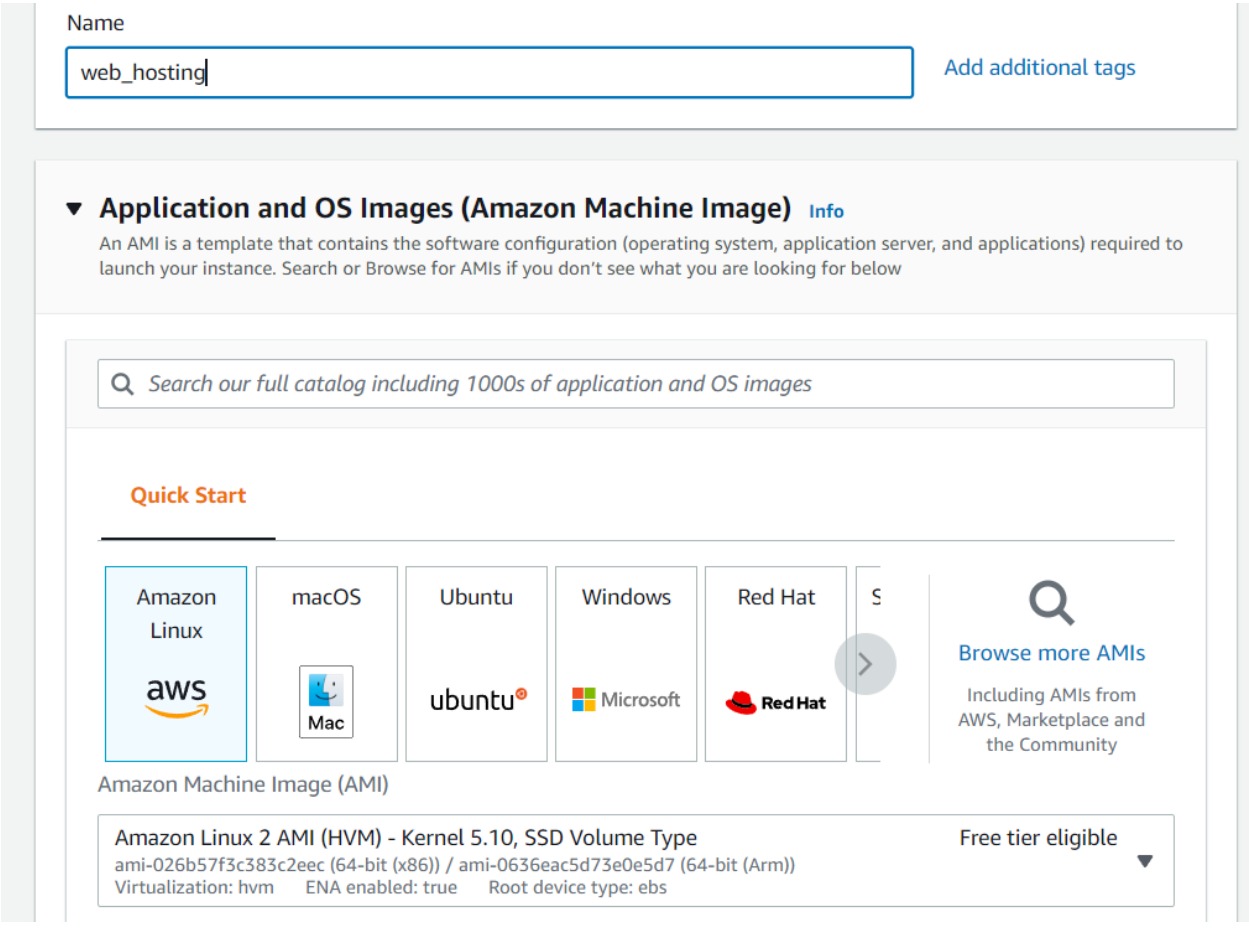


Part 2: Hosting a website on a Cloud VM (AWS EC2 Instance) To host our website on Cloud, we need to set up a Virtual Machine, in AWS EC2, in this case.

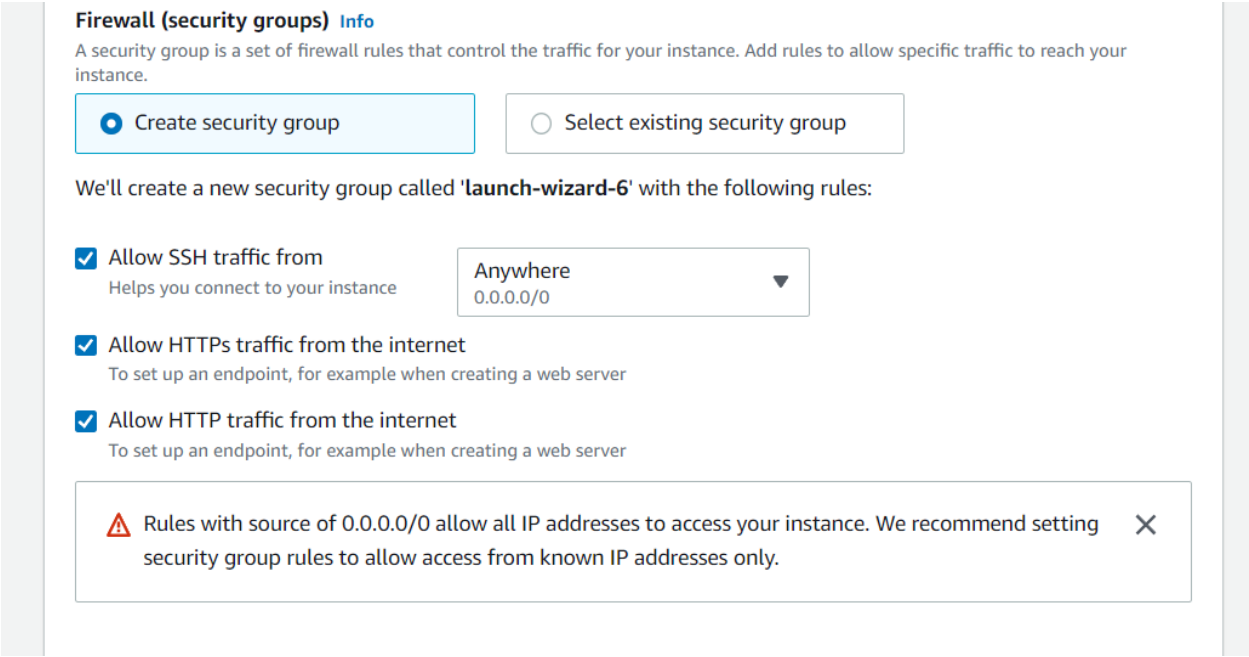
You'd need an AWS free tier account to proceed.

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

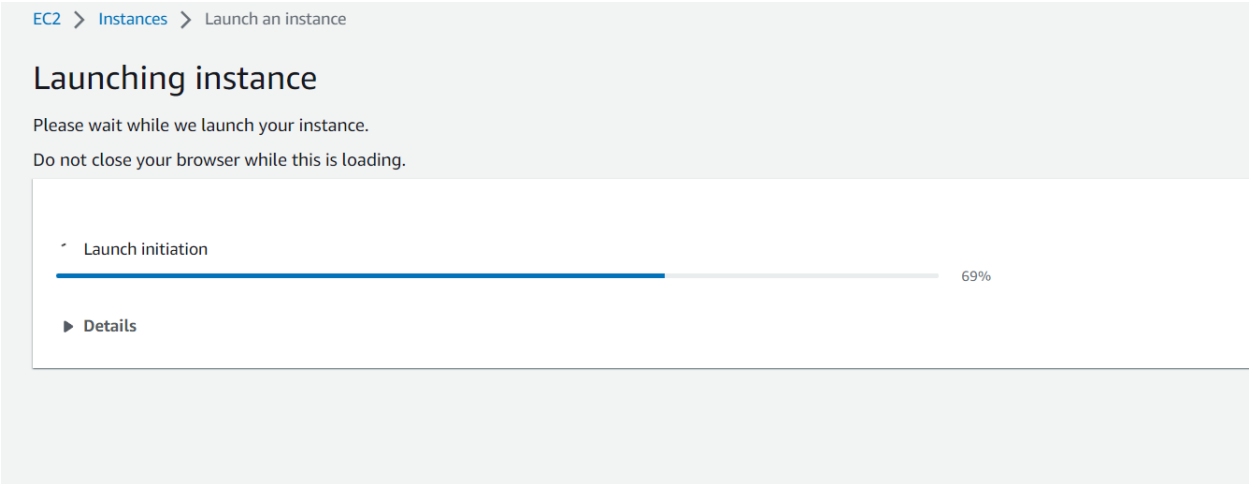
Choose the Linux 2 AMI , t2.micro for configuration, which is the free-tier-eligible one.



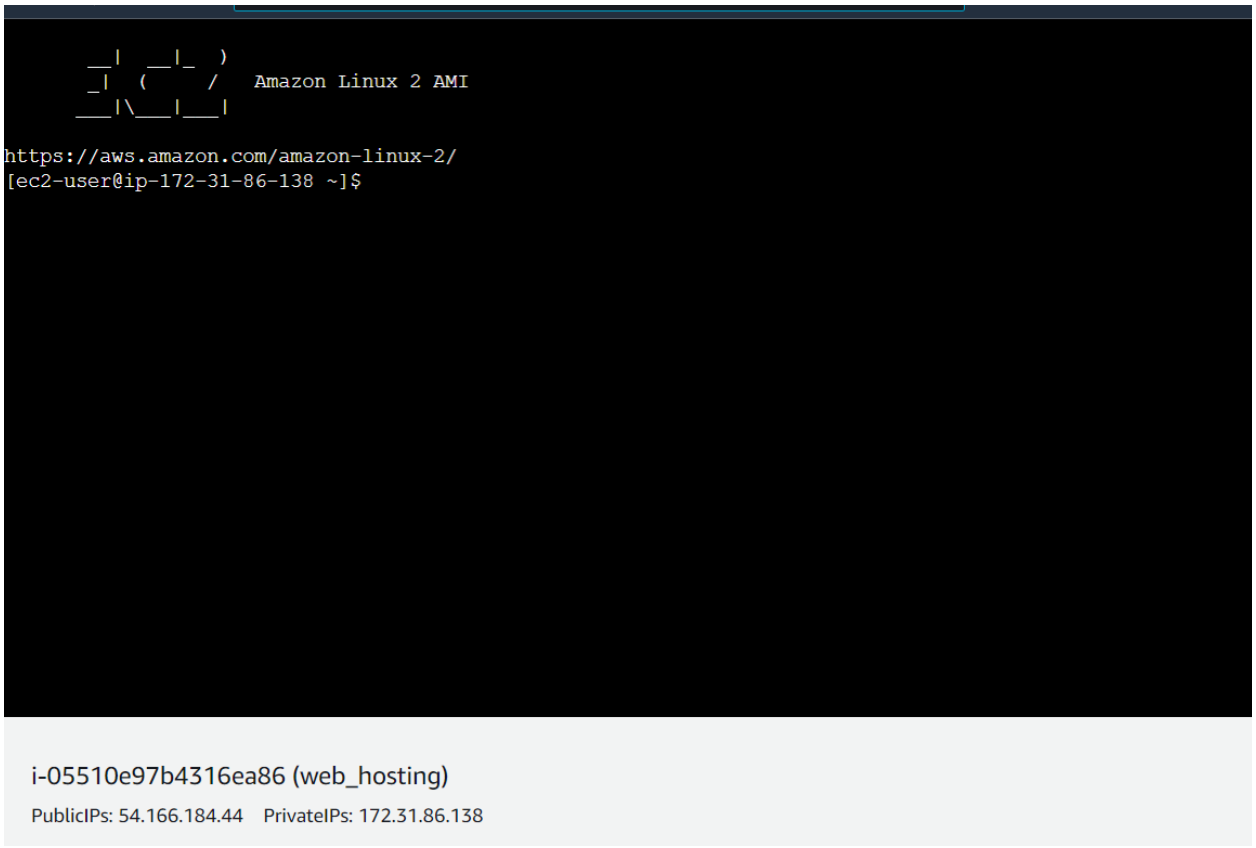
Proceed with everything default,in Security groups, add these rules so that you can visit the website from anywhere.



Review the changes and launch the instance



Step 2: Connect to the instance to access the CLI



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

```
[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]#
```

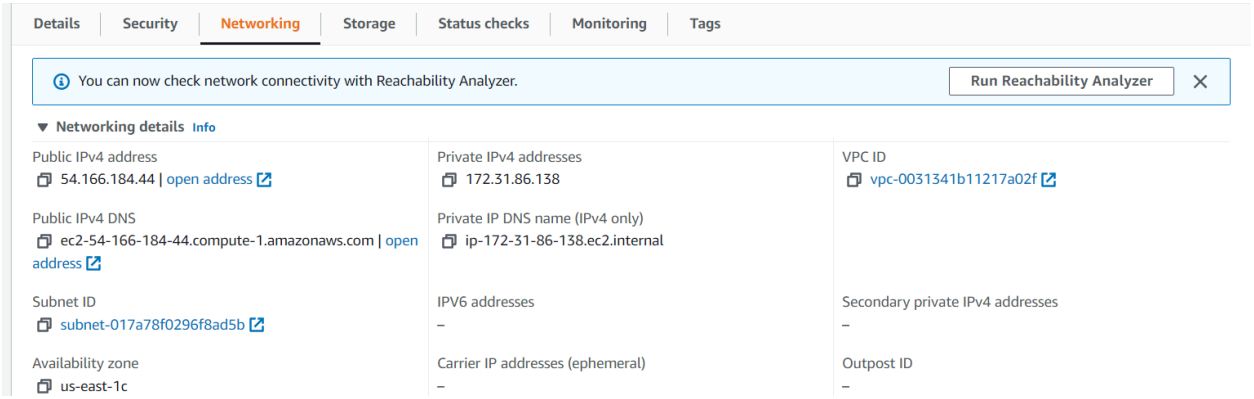
Move the folder outside.

```
[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.



Your website is live!