

# Assignment 01

## Implementation:

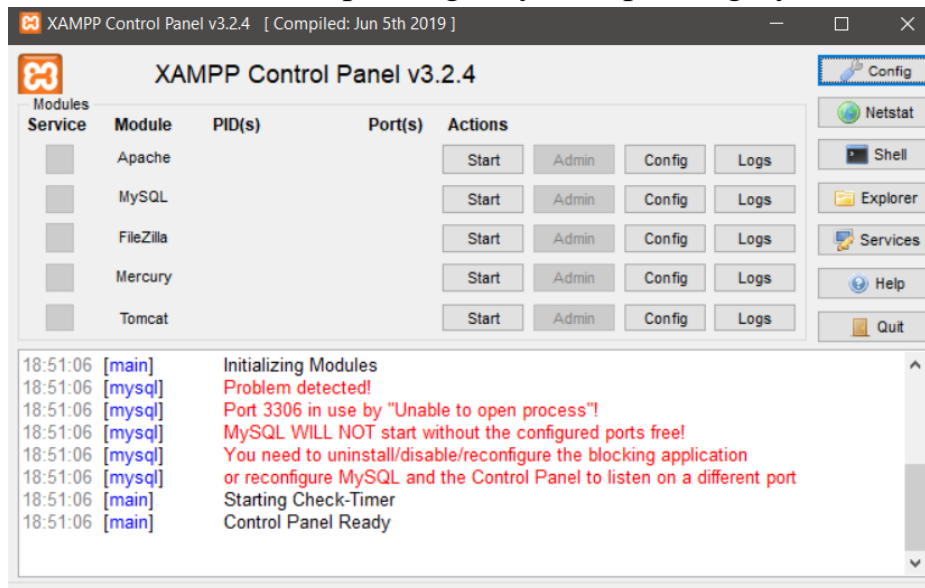
### Part 1: Hosting a website on localhost

#### Step 1: Download XAMPP.

Download link:

<https://www.apachefriends.org/download.html>

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.



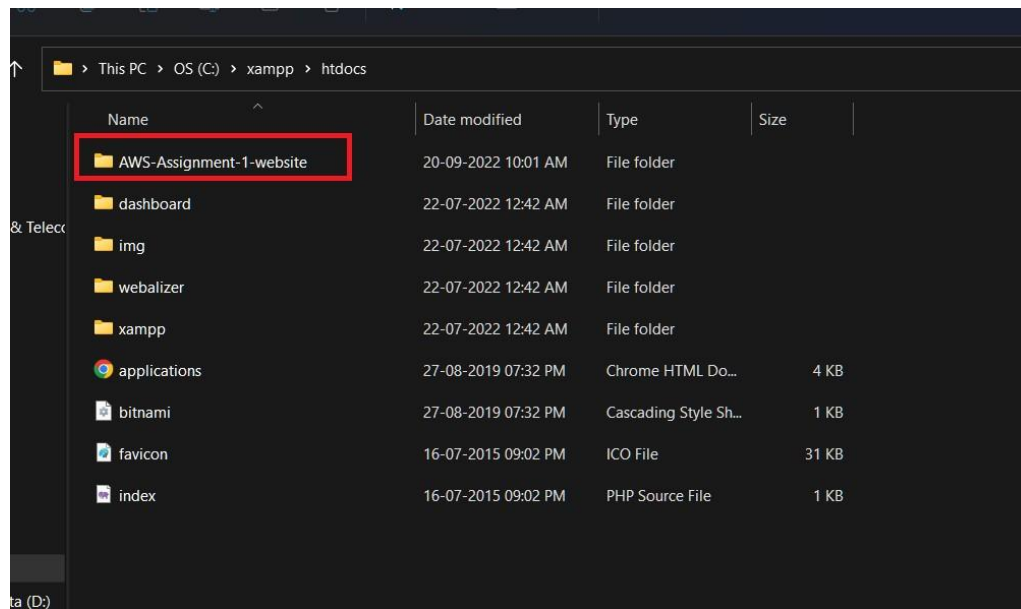
**\*\*Note:** Make sure you select Apache when you install XAMPP, so that you can use it with XAMPP later.\*\*

#### 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.

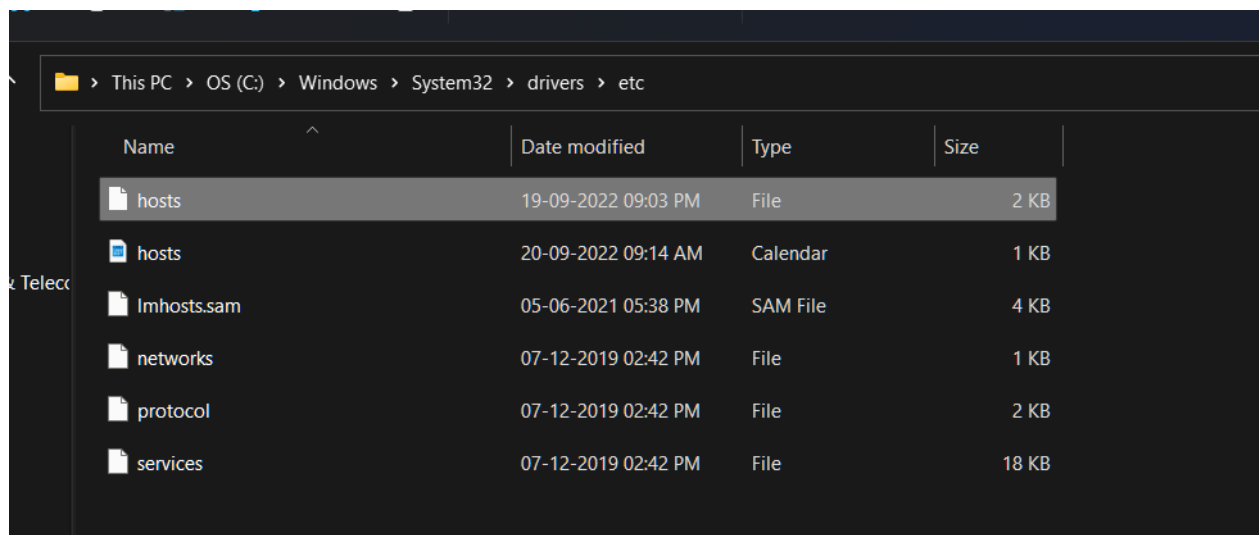
```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\xampp\htdocs> git clone https://github.com/Manavjawrani/AWS-Assignment-1-website.git
Cloning into 'AWS-Assignment-1-website'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 34 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (34/34), 1.49 MiB | 2.04 MiB/s, done.
Resolving deltas: 100% (1/1), done.
PS C:\xampp\htdocs> |
```

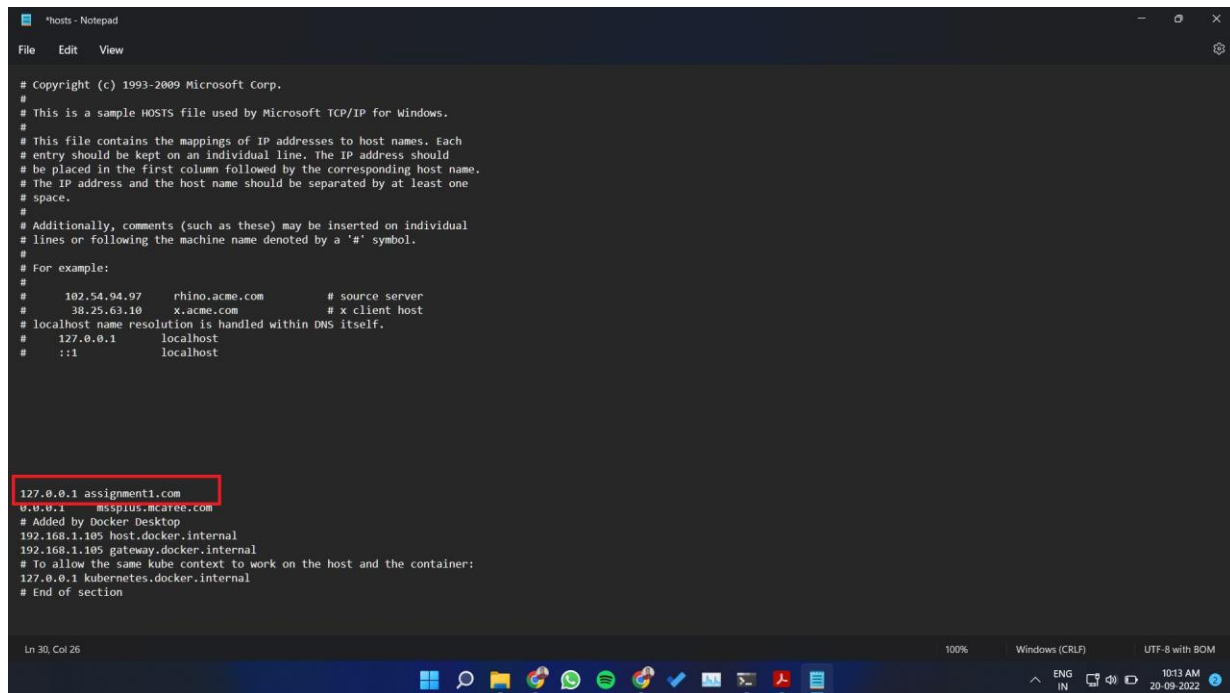


**Step 3:** Update the hosts file to serve localhost as your domain name (will work only on your local machine).

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 assignment1.com, as shown

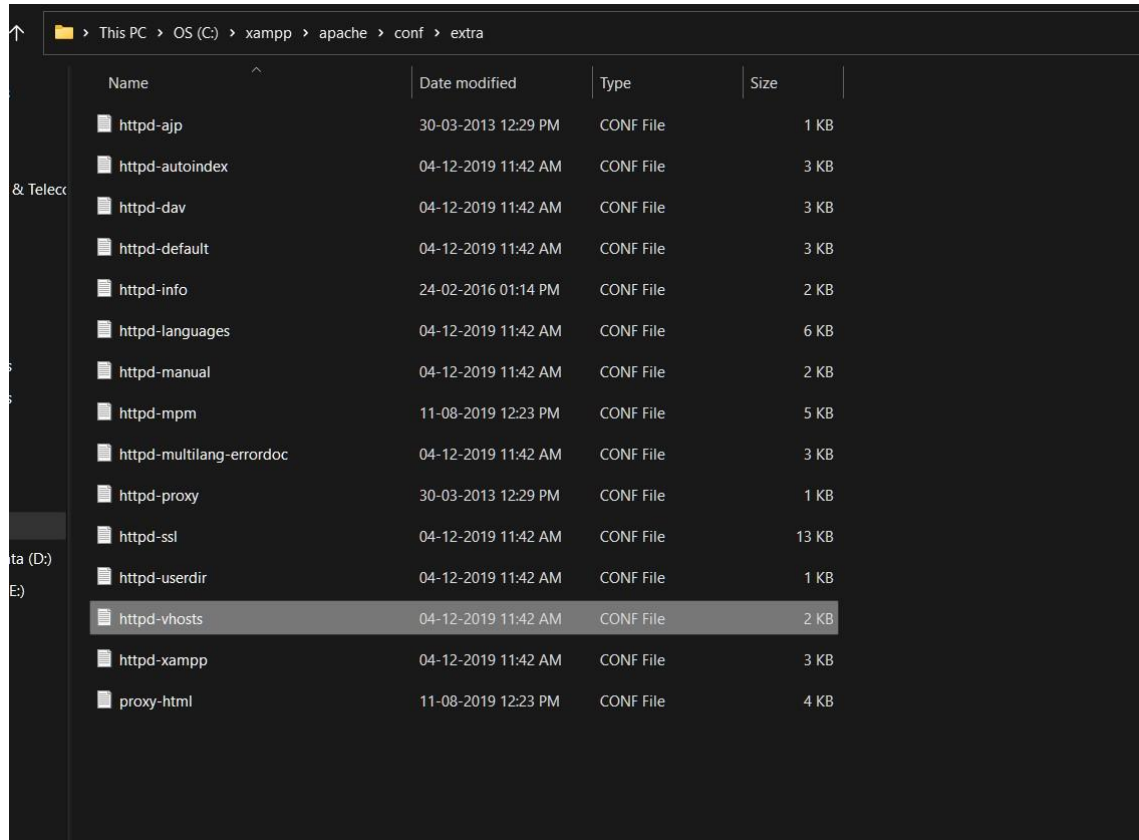


```
# Copyright (c) 1993-2009 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the 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 localhost
# ::1 localhost

127.0.0.1 assignment1.com
0.0.0.0 msspius.mcaree.com
# Added by Docker Desktop
192.168.1.105 host.docker.internal
192.168.1.105 gateway.docker.internal
# To allow the same kube context to work on the host and the container:
127.0.0.1 kubernetes.docker.internal
# End of section
```

#### 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, VSCode



e.

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.

```
##<VirtualHost *:80>
    ##ServerAdmin webmaster@dummy-host2.example.com
    ##DocumentRoot "/xampp/htdocs/dummy-host2.example.com"
    ##ServerName dummy-host2.example.com
    ##ErrorLog "logs/dummy-host2.example.com-error.log"
    ##CustomLog "logs/dummy-host2.example.com-access.log" common
##</VirtualHost>
<VirtualHost *:80>
DocumentRoot "C:\xampp\htdocs\AWS-Assignment-1-website"
ServerAdmin assignment1.com

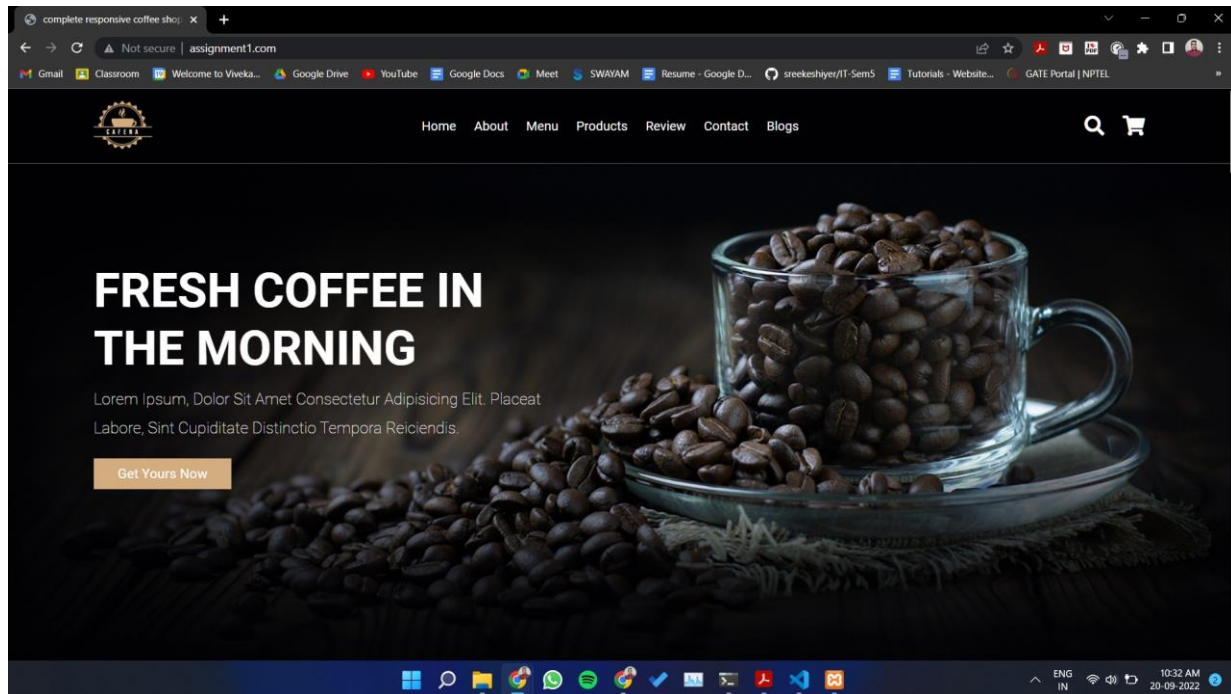
</VirtualHost>
```

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

XAMPP Control Panel v3.3.0 [ Compiled: Apr 6th 2021 ]



## Step 6: Open up your domain name on your browser

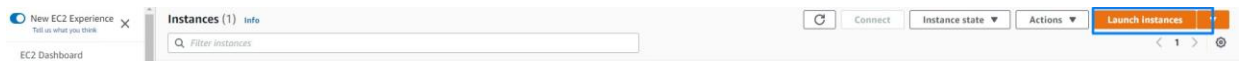


That's it, you have successfully hosted your website locally on your machine.

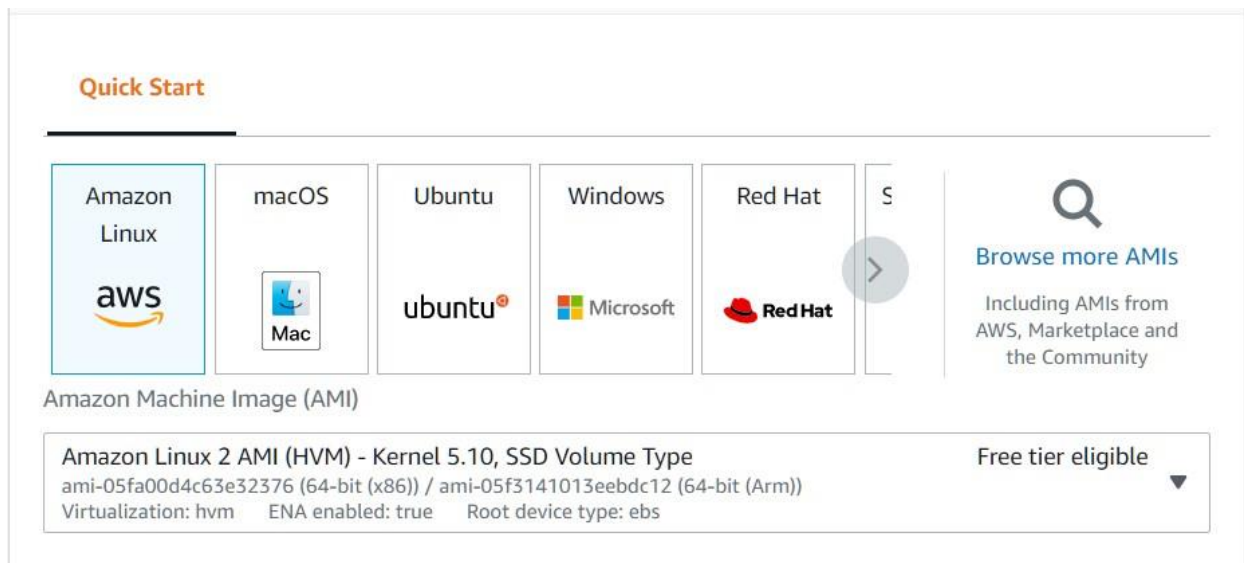
## Part 2: Hosting a website on a Cloud VM (AWS EC2 Instance)

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.

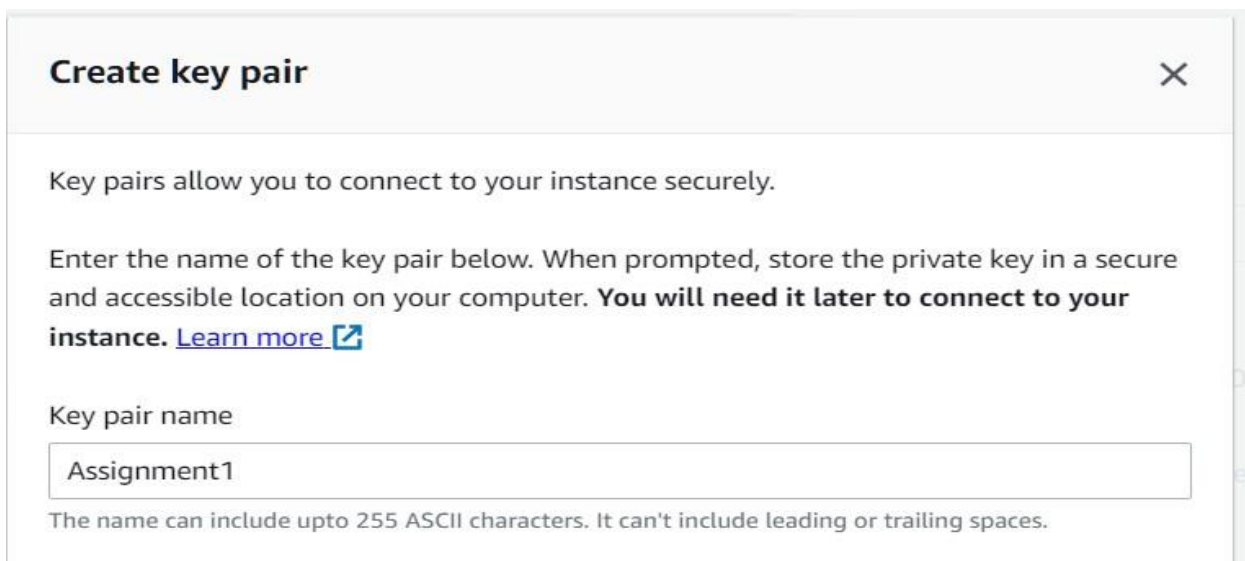
### Step 1: Open up EC2 Console and Launch a new Instance



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



Create and download a Keypair file if asked for.





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-1'** 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.



Review the changes and launch the instance.

[Find out more](#) or [send us feedback](#). Starting October 1, 2022, we will begin decommissioning the previous version.

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

## Launching instance

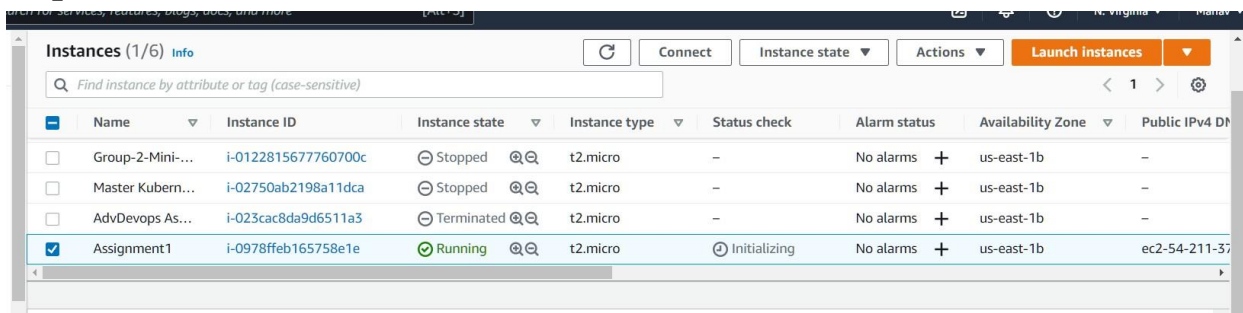
Please wait while we launch your instance.  
Do not close your browser while this is loading.

Launch initiation

69%

[Details](#)

## Step 2: Connect to the instance to access the CLI



Click on the Connect button on the top right.

Use EC2 instance connect to directly access the machine on a CLI, opened on the browser.

```
Installed:
kernel.x86_64 0:5.10.135-122.509.amzn2

Updated:
amazon-ssm-agent.x86_64 0:3.1.1732.0-1.amzn2  chrony.x86_64 0:4.2-5.amzn2.0.2  dhclient.x86_64 12:4.2.5-79.amzn2.1.1  dhcp-common.x86_64 12:4.2.5-79.amzn2.1.1
dhcp-libs.x86_64 12:4.2.5-79.amzn2.1.1  ec2-net-utils.noarch 0:1.7.1-1.amzn2  gnupg2.x86_64 0:2.0.22-5.amzn2.0.5  kernel-tools.x86_64 0:5.10.135-122.509.amzn2
microcode_ctl.x86_64 2:2.1-47.amzn2.0.13  tzdata.noarch 0:2022c-1.amzn2

Complete!
[root@ip-172-31-93-59 ec2-user]#
```

## Step 3: Install Git and HTTPD

Use the following commands to install git and httpd.

\$yum install git -y

```
[root@ip-172-31-93-59 ec2-user]# yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
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-172-31-93-59 ec2-user]# yum install httpd -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
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
```

**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/Manavjawrani/AWS-Assignment-1-website.git`

```
[root@ip-172-31-93-59 ec2-user]# pwd
/home/ec2-user
[root@ip-172-31-93-59 ec2-user]# cd /var/www/html
[root@ip-172-31-93-59 html]# git clone https://github.com/Manavjawrani/AWS-Assignment-1-website.git
Cloning into 'AWS-Assignment-1-website'...
remote: Enumerating objects: 34, done.
remote: Counting objects: 100% (34/34), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 34 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (34/34), 1.49 MiB | 24.54 MiB/s, done.
Resolving deltas: 100% (1/1), done.
[root@ip-172-31-93-59 html]#
```

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

```
[root@ip-172-31-93-59 html]# ls
AWS-Assignment-1-website
[root@ip-172-31-93-59 html]#
```

Move the folder outside.

```
[root@ip-172-31-93-59 html]# mv AWS-Assignment-1-website/* .
[root@ip-172-31-93-59 html]# ls
AWS-Assignment-1-website  css  images  index.html  js  README.md
[root@ip-172-31-93-59 html]#
```

**Step 5:** Start the httpd server

```
[root@ip-172-31-93-59 html]# pwd
/var/www/html
[root@ip-172-31-93-59 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-172-31-93-59 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

