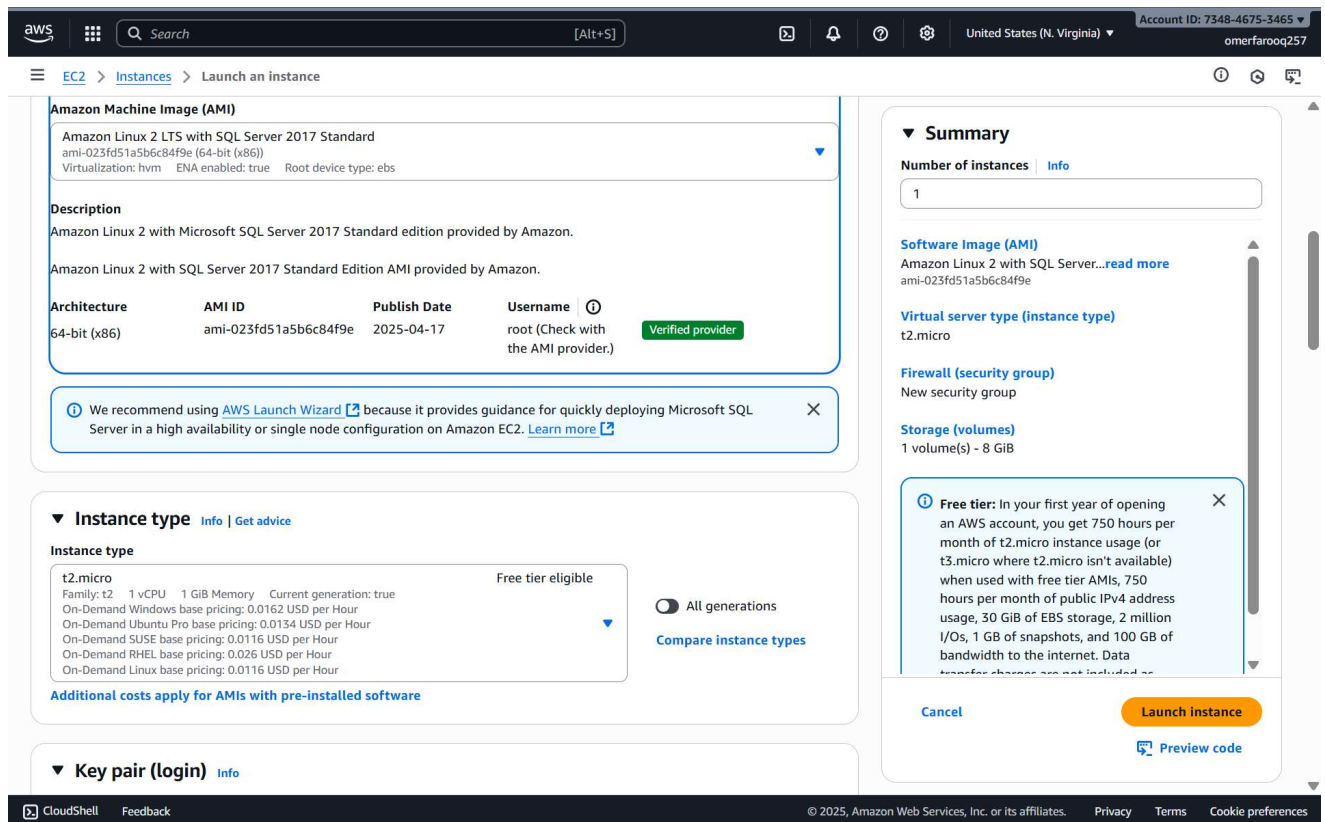


# AWS Challenge

## 1. Launch one EC2 instance using Amazon Linux 2.



### Steps to launch an EC2 instance:

1. Open a browser and go to the AWS website. Log in and click on the EC2 Service.
2. Under EC2 service, you'll find Instances. Click on that.
3. Click on the Launch Instance button. A new window will pop up.
4. Give a name to the instance. Scroll below to find various options and customizations. Choose as per your requirement. For now, we're choosing "Amazon Linux 2".
5. The most important thing while launching the instance is to select a key pair. It is best practice to create a new key pair for every instance launched.
6. Select the appropriate Network Settings. Click on Edit, scroll below, and click on "Add security group rules."
7. For this challenge, we are going to run five services on a single EC2 instance. So we will be selecting the following security rules:
  - i. Three Custom TCP and assigning them port numbers - 81 to run Nginx, 82 to run HTTPD, and 8082 to run Tomcat. For source type, select "anywhere."
  - ii. One HTTP with default port number 80. For source type, select "anywhere."
  - iii. One SSH with default port number 22. For source type, select "anywhere."
8. These settings can be selected later on while we actually perform the task, but it is good to choose network rules at the time of launching the Instance.
9. Upon clicking Launch Instance, the instance will be launched. You can access it via Git Bash or directly on the AWS website.

## 2. Install Docker.

*Steps to Install Docker in Amazon Linux:*

1. To install Docker, first update the system with the latest updates. Upgrade if necessary.

```
# yum update && yum upgrade
```

2. Install Docker using the following command. The Docker will be installed in the system.

```
# yum install docker -y && # docker --version -- To check if installed successfully.
```

```
[root@ip-172-31-26-169 /]# yum install docker -y
Last metadata expiration check: 0:42:09 ago on Wed Sep 17 17:51:14 2025.
Dependencies resolved.
=====
Package                                Architecture    Version                                Repository      Size
=====
Installing:
docker                                x86_64          25.0.8-1.amzn2023.0.5                amazonlinux     46 M
Installing dependencies:
container-selinux                    noarch          3:2.233.0-1.amzn2023.0.1             amazonlinux     55 k
containerd                           x86_64          2.0.5-1.amzn2023.0.2                 amazonlinux     26 M
iptables-libs                        x86_64          1.8.8-3.amzn2023.0.2                 amazonlinux     401 k
iptables-nft                         x86_64          1.8.8-3.amzn2023.0.2                 amazonlinux     183 k
libcgroup                           x86_64          3.0-1.amzn2023.0.1                   amazonlinux     75 k
libnetfilter_conntrack               x86_64          1.0.8-2.amzn2023.0.2                 amazonlinux     58 k
libnftnl                            x86_64          1.0.1-19.amzn2023.0.2                amazonlinux     30 k
libnftnl                             x86_64          1.2.2-2.amzn2023.0.2                 amazonlinux     84 k
runc                                 x86_64          1.2.6-1.amzn2023.0.1                 amazonlinux     3.7 M
Transaction Summary
-----
Installed:
  container-selinux-3:2.233.0-1.amzn2023.0.1.noarch
  docker-25.0.8-1.amzn2023.0.5.x86_64
  iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64
  libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  containerd-2.0.5-1.amzn2023.0.2.x86_64
  iptables-libs-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64
  libnftnl-1.0.1-19.amzn2023.0.2.x86_64
  runc-1.2.6-1.amzn2023.0.1.x86_64
Complete!
[root@ip-172-31-26-169 /]# |

[root@ip-172-31-26-169 /]# docker --version
Docker version 25.0.8, build 0bab007
[root@ip-172-31-26-169 /]#
```

3. Use the following command to start and check the status of Docker:

```
# systemctl start docker
```

```
# systemctl enable docker
```

```
# systemctl status docker
```

```
[root@ip-172-31-26-169 /]# systemctl start docker
[root@ip-172-31-26-169 /]# systemctl enable docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
[root@ip-172-31-26-169 /]# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
   Active: active (running) since Wed 2025-09-17 18:38:40 UTC; 42s ago
 TriggeredBy: ● docker.socket
            Docs: https://docs.docker.com
   Main PID: 9750 (dockerd)
     Tasks: 7
    Memory: 30.9M
       CPU: 278ms
    CGroup: /system.slice/docker.service
            └─9750 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit no
```

4. Docker is successfully installed and running on the instance. Below is a sample test:

```
[root@ip-172-31-26-169 /]# docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
17eec7bbc9d7: Pull complete
Digest: sha256:54e66cc1dd1fcb1c3c58bd8017914dbed8701e2d8c74d9262e26bd9cc1642d31
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.
```

### 3. Install Jenkins.

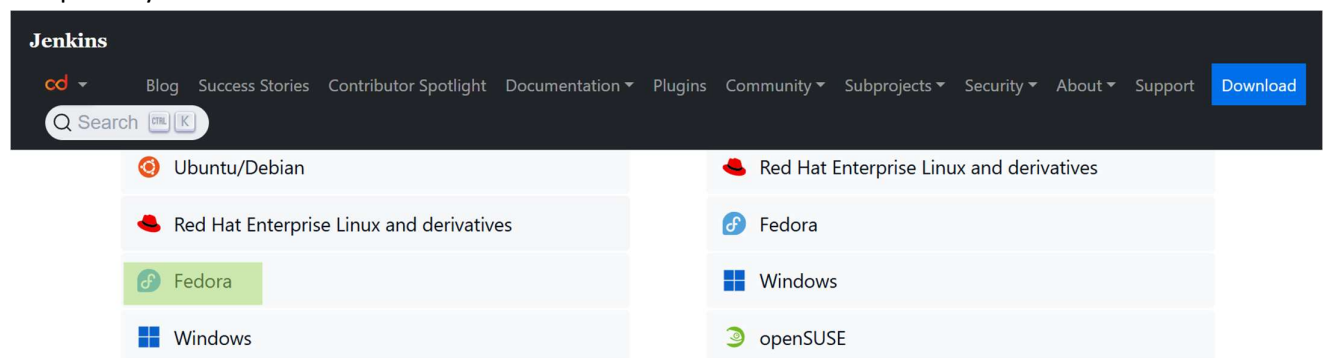
*Steps to Install Jenkins:*

1. Jenkins is installed based on the EC2 operating system. First, we check the OS of the system. To do that, follow the steps below:

# less /etc/os-release – To check the OS installed in the instance.

```
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.8.20250818"
```

2. Open any web browser and search for Jenkins. Go to the downloads section and click on Fedora.



3. Upon clicking on the correct version of OS, the web page gives step-by-step instructions to install Jenkins into the system.

## Long Term Support release

A [LTS \(Long-Term Support\) release](#) is chosen every 12 weeks from the stream of regular releases as the stable release for that time period. It can be installed from the [redhat-stable](#) yum repository.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
sudo dnf upgrade
# Add required dependencies for the jenkins package
sudo dnf install fontconfig java-21-openjdk
sudo dnf install jenkins
sudo systemctl daemon-reload
```

4. Copy all the steps. Create a file to run all the scripts simultaneously, as we learnt in the concepts of Bash Scripting. In my case, I created a file "Jenkins.sh".

5. Execute the file. It won't execute. Because the default permissions are just read and write. Change the permissions to execute as well. Now, execute the file. The next process is as follows:

```

[root@ip-172-31-26-169 home]# vi jenkins.sh
[root@ip-172-31-26-169 home]# ./jenkins.sh
bash: ./jenkins.sh: Permission denied
[root@ip-172-31-26-169 home]# chmod 744 jenkins.sh
[root@ip-172-31-26-169 home]# ./jenkins.sh
--2025-09-17 17:51:13-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.38.133, 2a04:4e42:79::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.38.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

/etc/yum.repos.d/jenkins.re 100%[=====] 85 --.-KB/s in 0s

2025-09-17 17:51:14 (3.72 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

Jenkins-stable 677 kB/s | 32 kB 00:00

Installed:
jenkins-2.516.3-1.1.noarch

Complete!
[root@ip-172-31-26-169 home]# |

```

6. Jenkins is now successfully installed in the system. The result can be seen above.

7. To start and check the status of Jenkins, follow the commands below:

```
# systemctl daemon-reload
```

```
# systemctl enable jenkins
```

```
# systemctl start jenkins
```

```
# systemctl status Jenkins
```

The output looks like the below image:

```

[root@ip-172-31-26-169 /]# systemctl daemon-reload
[root@ip-172-31-26-169 /]# systemctl enable jenkins
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /usr/lib/systemd/system/jenkins.service.
[root@ip-172-31-26-169 /]# systemctl start jenkins
[root@ip-172-31-26-169 /]# systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: disabled)
   Active: active (running) since Wed 2025-09-17 18:02:15 UTC; 9s ago
     Main PID: 6197 (java)
       Tasks: 44 (limit: 1111)
      Memory: 278.8M
         CPU: 9.142s
        CGroup: /system.slice/jenkins.service
                └─6197 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cac

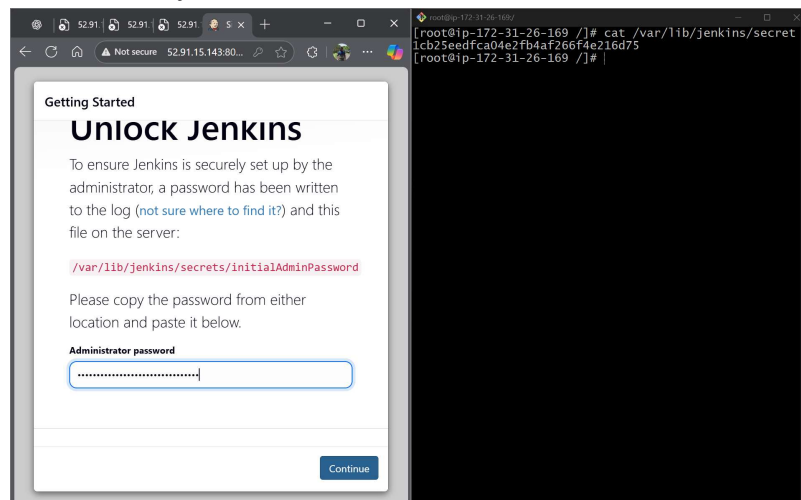
```

8. Jenkins is installed and running. The result can be seen in the above image

9. Open a browser and type the local IP followed by the port number, which is 8080. (We assigned this at the time of launching the EC2 Instance.) A new page will open, asking to log in using a password.

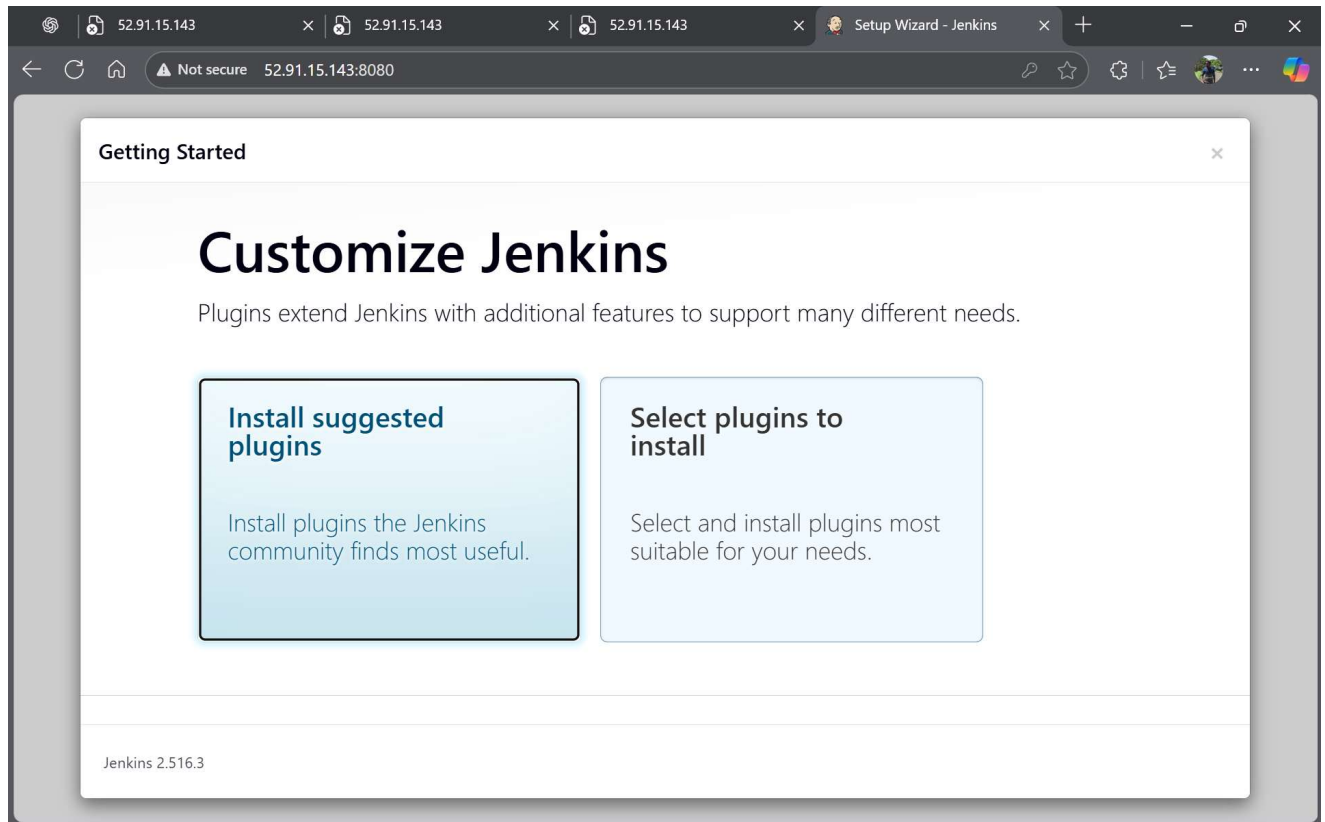
10. To check the initial password, open the password file using:

```
# cat /var/lib/jenkins/secrets/initialAdminPassword
```





10. Copy the password and paste it in the box, and hit continue. Done and ready to use.



#### 4. Install Apache HTTPD.

*Steps to install HTTPD in Amazon Linux 2:*

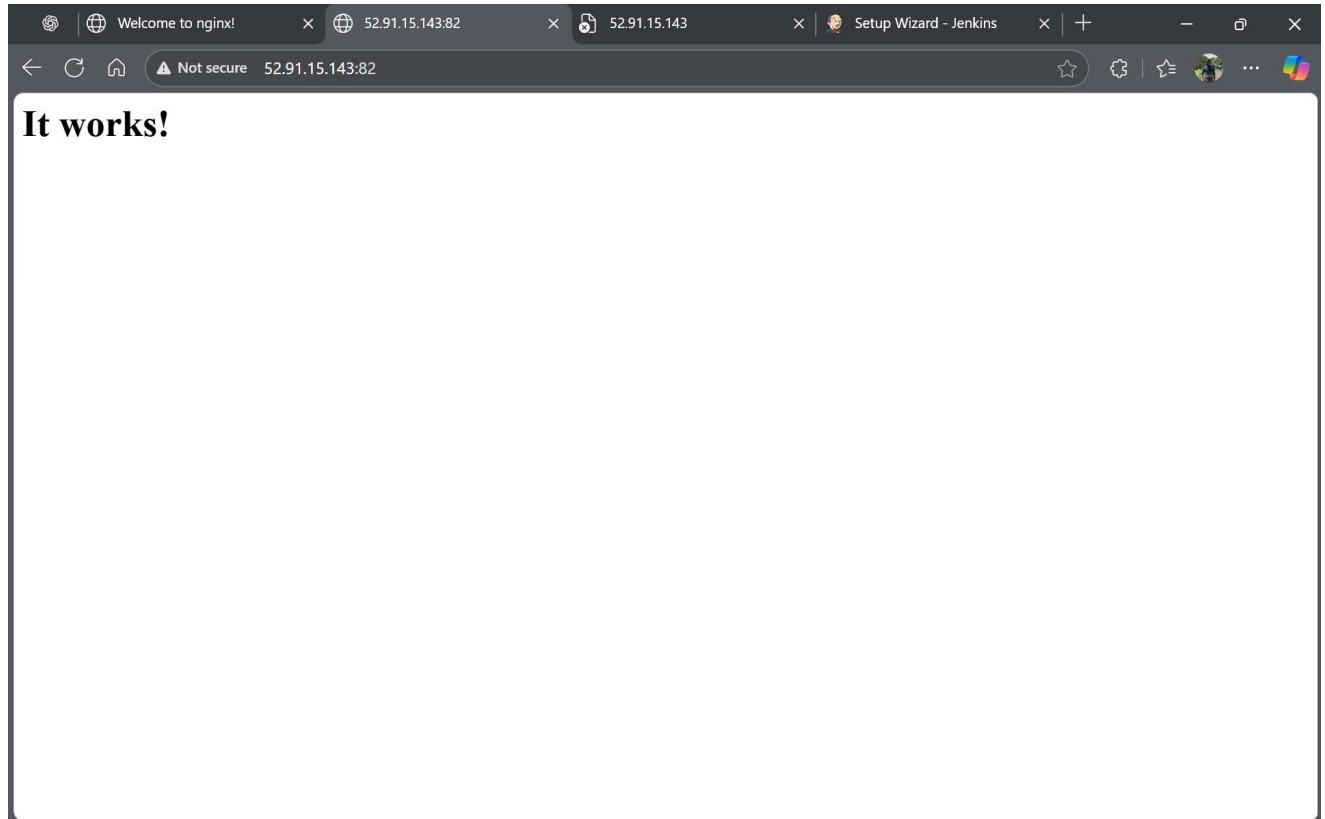
1. Install HTTPD using the following command. It installs Apache HTTPD.  
`# yum install httpd -y`
2. Navigate to `/etc/httpd/conf/` and look for `httpd.conf` file. Make a copy for backup in case the original file gets corrupted.  
`# cd /etc/httpd/conf`  
`# cp httpd.conf httpd.conf.backup`
3. Open the file and change the port from 80 to 82.  
`# sed -i 's/80/82/g' /etc/httpd/conf/httpd.conf`
4. Start HTTPD and check the status  
`# systemctl start httpd`  
`# systemctl status httpd`

Below is the output of the above commands:

```
[root@ip-172-31-26-169 ~]# systemctl start httpd
[root@ip-172-31-26-169 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Wed 2025-09-17 19:07:57 UTC; 11s ago
     Docs: man:httpd.service(8)
  Main PID: 11260 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/sec"
    Tasks: 177 (limit: 1111)
   Memory: 17.7M
      CPU: 70ms
   CGroup: /system.slice/httpd.service
           └─11260 /usr/sbin/httpd -DFOREGROUND
             └─11261 /usr/sbin/httpd -DFOREGROUND
               └─11262 /usr/sbin/httpd -DFOREGROUND
                 └─11263 /usr/sbin/httpd -DFOREGROUND
                   └─11264 /usr/sbin/httpd -DFOREGROUND

Sep 17 19:07:56 ip-172-31-26-169.ec2.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Sep 17 19:07:57 ip-172-31-26-169.ec2.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Sep 17 19:07:57 ip-172-31-26-169.ec2.internal httpd[11260]: Server configured, listening on: port 82
[root@ip-172-31-26-169 ~]#
```

5. Run the HTTPD on a browser. Copy the public IP of the instance and open it with the assigned port number. The result is as follows:



## 5. Install Nginx.

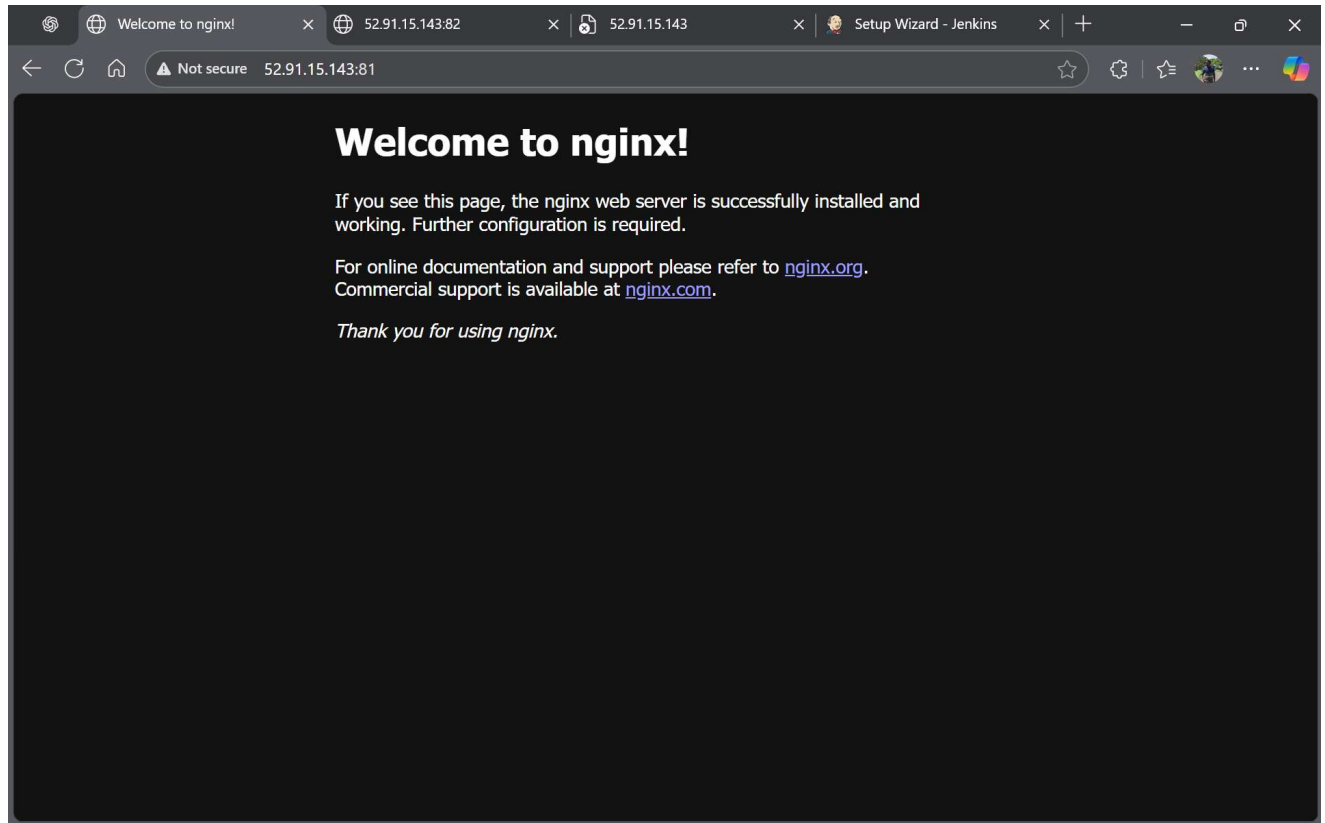
*Steps to install Nginx in Amazon Linux 2:*

1. Install nginx using this command:  
`# yum install nginx -y`
2. Make a copy of the conf file. The conf file is located at `/etc/nginx/nginx.conf`. It is always best practice to make a backup of the file in case the original file gets corrupted.  
`# cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.backup`
3. Edit the "nginx.conf" file and change the port number from 80 to 81 using the following command:  
`# sed -i 's/80/81/g' /etc/nginx/nginx.conf`
4. Start nginx and check the status with the following commands:  
`# systemctl start nginx`  
`# systemctl status nginx`

```
[root@ip-172-31-26-169 /]# systemctl start nginx
[root@ip-172-31-26-169 /]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; disabled; preset: disabled)
   Active: active (running) since Wed 2025-09-17 19:16:22 UTC; 4s ago
     Process: 11959 ExecStartPre=/usr/bin/rm -f /run/nginx.pid (code=exited, status=0/SUCCESS)
     Process: 11960 ExecStartPre=/usr/sbin/nginx -t (code=exited, status=0/SUCCESS)
     Process: 11961 ExecStart=/usr/sbin/nginx (code=exited, status=0/SUCCESS)
    Main PID: 11962 (nginx)
      Tasks: 2 (limit: 1111)
     Memory: 4.9M
        CPU: 47ms
    CGroup: /system.slice/nginx.service
            └─11962 "nginx: master process /usr/sbin/nginx"
              └─11963 "nginx: worker process"

Sep 17 19:16:22 ip-172-31-26-169.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server: (main)
Sep 17 19:16:22 ip-172-31-26-169.ec2.internal nginx[11960]: nginx: the configuration file /etc/nginx/nginx.conf is ok
Sep 17 19:16:22 ip-172-31-26-169.ec2.internal nginx[11960]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Sep 17 19:16:22 ip-172-31-26-169.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server: (main)
```

5. To check if the nginx is running, open a browser and enter the public IP of the instance with the assigned port number. The result can be seen below:



## 6. Install **Apache Tomcat**.

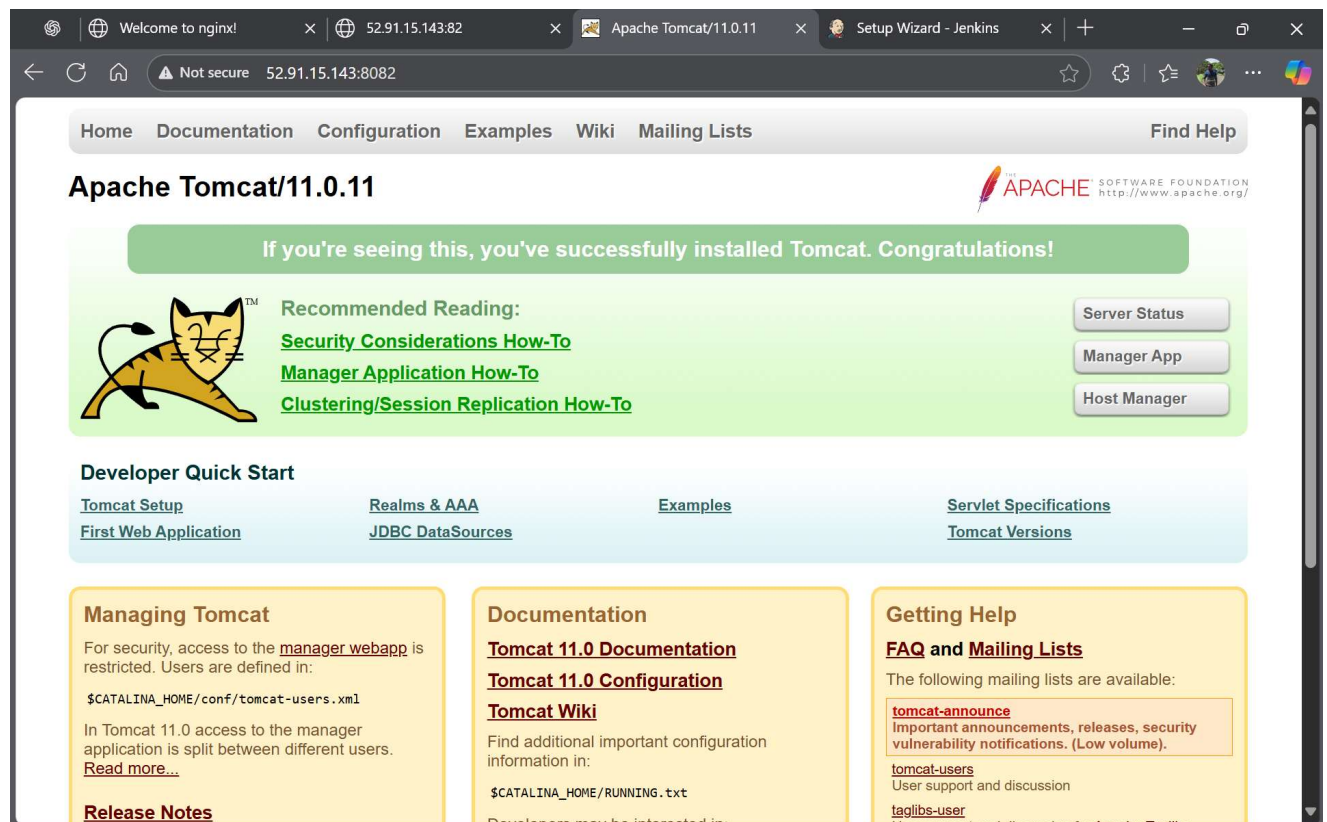
*Steps to install Apache Tomcat in Amazon Linux 2:*

1. Install the appropriate version of Java into the system. Tomcat 11 runs on Java 17 and above. Ensure you install the JDK, not just the JRE.  
`# yum install openjdk-21-amazon-corretto-devel -y`
2. Add a user with the name tomcat and make it a system user with a home directory in /opt/tomcat and give it a /bin/false shell. This is a good practice that helps protect the server even after a cyberattack.  
`# useradd -r -m -U -d /opt/tomcat -s /bin/false tomcat`  
r – system user  
m – to create a home directory  
U – to create a group with the same username  
d – path/to/directory  
s – providing shell, which in this case is /bin/false
3. Download Tomcat from the internet into the “/tmp” directory. Click on the tar file and copy the link. To download, use:  
`# wget <link_address>`
4. Extract the file in the “/opt/tomcat” directory.  
`# tar xvf <file_name> -C /opt/tomcat --strip-components=1.`
5. Change the ownership and permission of the directory “/opt/tomcat”.  
`# chown -R tomcat: /opt/tomcat`  
`# chmod 700 /opt/tomcat`

6. Edit the "server.xml" file. Change the port number from the default 8080 to 8082. Make a backup before editing.  

```
# cp /opt/tomcat/conf/server.xml /opt/tomcat/conf/server.xml.backup  
# sed -i 's/8080/8082/g' /opt/tomcat/conf/server.xml
```
7. Create a "tomcat.service" file in the directory - /etc/systemd/system/tomcat.service. Add all the details. Ensure you type the correct Java version, port number, path of the start and stop files, and user & group information. If the service file is not accurate, the next commands won't run.
8. Run the following commands to start Tomcat:  

```
# systemctl daemon-reload  
# systemctl start tomcat  
# systemctl status tomcat
```
9. Check the output of the Tomcat by opening a browser and checking with the public IP and the assigned port number 8082.



The output of all five services is attached below:

<https://www.playbook.com/s/omerfarooq257/35WNvVhCGG8uNscWckhAuMWO?assetToken=PqXGLimVrKKRoak9gbjNjHKT>