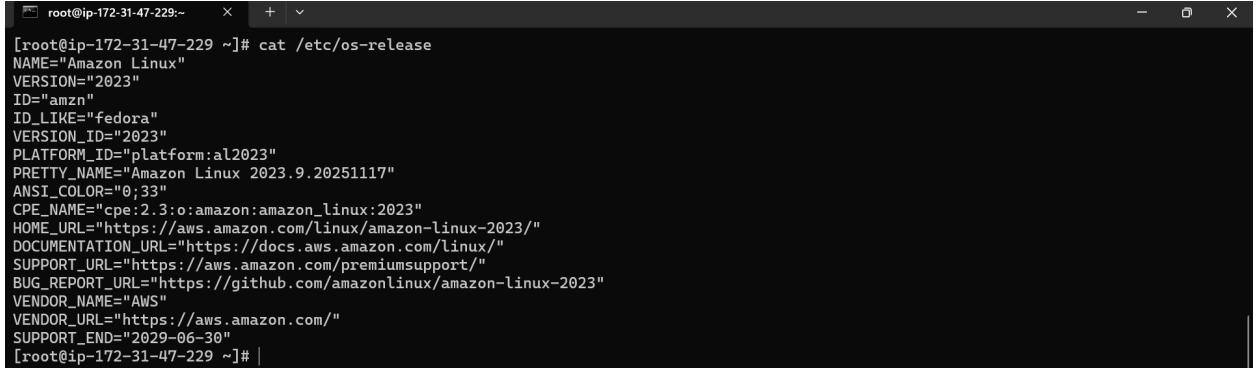


# Web Server Project-1

 Project	 Description
Apache Web Hosting	 Install and configure Apache web server to host a website site.
Virtual Host & SSL Setup	 Configure virtual hosts and enable HTTPS using Open SSL certificates & Redirect all client traffic on https protocol only.

## Step1 . Launch Instance

- AMI → Amazon Linux 2023
- Instance → t2.micro (Free Tier)
- Key pair → create or use existing



```
root@ip-172-31-47-229:~# cat /etc/os-release
NAME="Amazon Linux"
VERSION="2023"
ID="amzn"
ID_LIKE="fedora"
VERSION_ID="2023"
PLATFORM_ID="platform:al2023"
PRETTY_NAME="Amazon Linux 2023.9.20251117"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2023"
HOME_URL="https://aws.amazon.com/linux/amazon-linux-2023/"
DOCUMENTATION_URL="https://docs.aws.amazon.com/linux/"
SUPPORT_URL="https://aws.amazon.com/premiumsupport/"
BUG_REPORT_URL="https://github.com/amazonlinux/amazon-linux-2023"
VENDOR_NAME="AWS"
VENDOR_URL="https://aws.amazon.com/"
SUPPORT_END="2029-06-30"
[root@ip-172-31-47-229 ~]# |
```

- Security Group:
  - Allow 22 (SSH) from your IP
  - Allow 80 (HTTP) from anywhere
  - Allow 443 (HTTPS) from anywhere

Inbound rules (4)						
<input type="checkbox"/>	Name	Security group rule ID	IP version	Type	Protocol	
<input type="checkbox"/>	-	sgr-0b77b33d85ac0cf69	IPv4	SSH	TCP	
<input type="checkbox"/>	-	sgr-0be72830d01773fe3	IPv4	HTTPS	TCP	
<input type="checkbox"/>	-	sgr-0c8c7b9119ab82e0c	IPv4	HTTP	TCP	
<input type="checkbox"/>	-	sgr-0bb5638e82f89df7f	IPv4	All ICMP - IPv4	ICMP	

## Step2 . SSH into EC2

```
ssh -i ec2key_key.pem ec2-user@EC2_PUBLIC_IP
```



The screenshot shows a Windows Command Prompt window titled "Command Prompt - ssh -i off". The window displays the following text:

```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Shabbir Ahmad>cd Downloads

C:\Users\Shabbir Ahmad\Downloads>ssh -i office_key.pem ec2-user@13.126.179.190
The authenticity of host '13.126.179.190 (13.126.179.190)' can't be established.
ED25519 key fingerprint is SHA256:33/n0mI3SApTPiQVjLY+kXrSiHe/23y4OPrxRP29Flg.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? |
```

## Step 3. Install Apache Web Server

### Install Apache

```
sudo dnf install httpd -y
```

### Enable and start service

```
sudo systemctl enable httpd
sudo systemctl start httpd
```

### Check status

```
sudo systemctl status httpd
```

```

root@ip-172-31-47-229:~# systemctl restart httpd
[root@ip-172-31-47-229 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-47-229 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Mon 2025-11-24 09:00:15 UTC; 12s ago
     Docs: man:httpd.service(8)
 Main PID: 26484 (httpd)
   Status: "Total requests: 0; Idle/Busy workers 100/0;Requests/sec: 0; Bytes served/sec: 0 B/sec"
   Tasks: 177 (limit: 1115)
  Memory: 13.2M
    CPU: 67ms
   CGroup: /system.slice/httpd.service
           ├─26484 /usr/sbin/httpd -DFOREGROUND
           ├─26505 /usr/sbin/httpd -DFOREGROUND
           ├─26508 /usr/sbin/httpd -DFOREGROUND
           ├─26509 /usr/sbin/httpd -DFOREGROUND
           └─26547 /usr/sbin/httpd -DFOREGROUND

Nov 24 09:00:15 ip-172-31-47-229.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Nov 24 09:00:15 ip-172-31-47-229.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Nov 24 09:00:15 ip-172-31-47-229.ap-south-1.compute.internal httpd[26484]: Server configured, listening on: port 80
[root@ip-172-31-47-229 ~]#

```

## Step4 . create test webpage

```

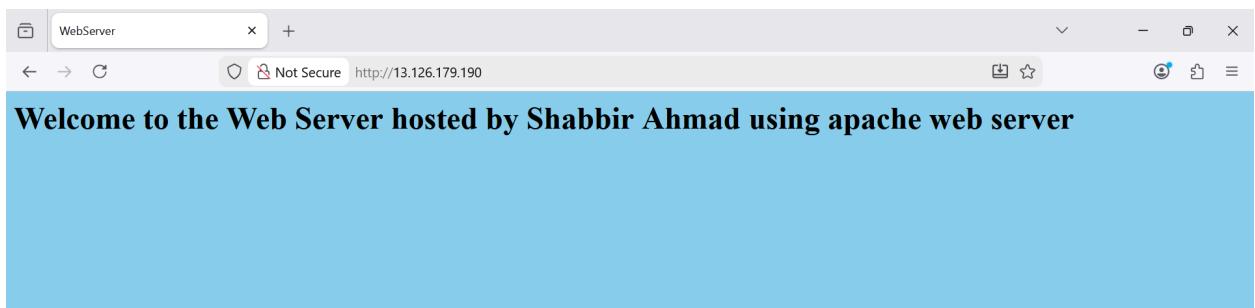
#cd /var/www/html/
#vim index.html
<html>
<head><title>WebServer</title></head>
<body bgcolor=skyblue>
<h1>Welcome to the Web Server hosted by Shabbir Ahmad using apache web server</h1>
</body>
</html>
:wq

```

## Access Web Server

Visit public IP in browser:

👉 <http://13.126.179.190>



## Step 5 .Configure Virtual web hosting

Edit virtual host config

```
vim /etc/httpd/conf.d/vhosts.conf
```

```
<VirtualHost *:80>
ServerName 13.126.179.190 (# Use EC2 Pub-IP Because currently we do not have DNS)
ServerAdmin root@example.com
DocumentRoot /var/www/html/
</VirtualHost>
<Directory /var/www/html/>
<RequireAll>
Require all granted
</RequireAll>
</Directory>
```

```
:wq
```

## Restart Apache

```
sudo systemctl restart httpd
```

## Step 6 .Configure/enable Apache Web hosting with HTTPS using Open SSL certificates .

Install Require package for https

```
#yum install httpd mod_ssl openssl -y
```

### A- create self-signed Certificate and Key for https site

```
#openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /home/server.key -out /home/server.crt
```



A terminal window showing the command to generate a self-signed SSL certificate. The command is: #openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /home/server.key -out /home/server.crt. The output shows the creation of two files: server.key and server.crt in the /home directory.

```
[root@ip-172-31-47-229 ~]# ls /home/
ec2-user  server.crt  server.key
[root@ip-172-31-47-229 ~]#
```

### B- Modify ssl config file

```
#vim /etc/httpd/conf.d/ssl.conf
```

```
SSLCertificateFile /etc/pki/tls/certs/server.crt  
SSLCertificateKeyFile /etc/pki/tls/private/server.key
```

```
:wq
```

**C- Copy generated crt and key on required location**

```
#cp -rvf /home/server.crt /etc/pki/tls/certs/  
#cp -rvf /home/server.key /etc/pki/tls/private/
```

**D- Update configuration file of pune and mumbai**

```
#vim /etc/httpd/conf.d/vhosts.conf
```

```
<VirtualHost *:443>  
SSLEngine on  
SSLCertificateFile /etc/pki/tls/certs/server.crt  
SSLCertificateKeyFile /etc/pki/tls/private/server.key  
ServerName example.com  
ServerAdmin root@example.com  
DocumentRoot /var/www/html/  
</VirtualHost>  
<Directory /var/www/html/>  
<RequireAll>  
Require all granted  
</RequireAll>  
</Directory>
```

```
:wq
```

```
systemctl restart httpd
```

## Access Your Web Server

Visit your public IP in browser:

👉 <https://13.126.179.190>



## Welcome to the Web Server

Hosted by [Shabbir Ahmad](#)

This website is powered by the [Apache HTTP Server](#) running on Amazon Linux.

HTTPS is successfully configured using [OpenSSL self-signed certificates](#), ensuring secure and encrypted communication between your browser and the server.

© 2025 Shabbir Ahmad – Secure Apache Web Hosting

## Step7 . Redirect all client traffic on https protocol

Edit configuration files

```
#vim /etc/httpd/conf.d/vhosts.conf
```

```
<VirtualHost *:443>
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/server.crt
SSLCertificateKeyFile /etc/pki/tls/private/server.key
ServerName 13.126.179.190
ServerAdmin root@example.com
DocumentRoot /var/www/html/
</VirtualHost>
<Directory /var/www/html/>
<RequireAll>
Require all granted
</RequireAll>
</Directory>
<VirtualHost *:80>
ServerName test.com
Redirect / https://13.126.179.190
</VirtualHost>
```

```
:wq
```

```
systemctl restart httpd
```

Visit your public IP in browser:

👉 <http://13.126.179.190>

Note: if we type http:// 13.126.179.190 only site by default redirect us on https

## Troubleshooting (quick)

- Blank page / cannot connect → re-check Security Group inbound rules and EC2 public IP.
- Service failed → check logs: `sudo journalctl -u httpd --since "1 hour ago"` and `/var/log/httpd/error_log`.

✓ Project Completed

Apache Web Server successfully deployed on EC2 as part of DevOps/Linux practice by Shabbir Ahmad.....Thnak You !!