Global IT Providers
Hosting & Server Management

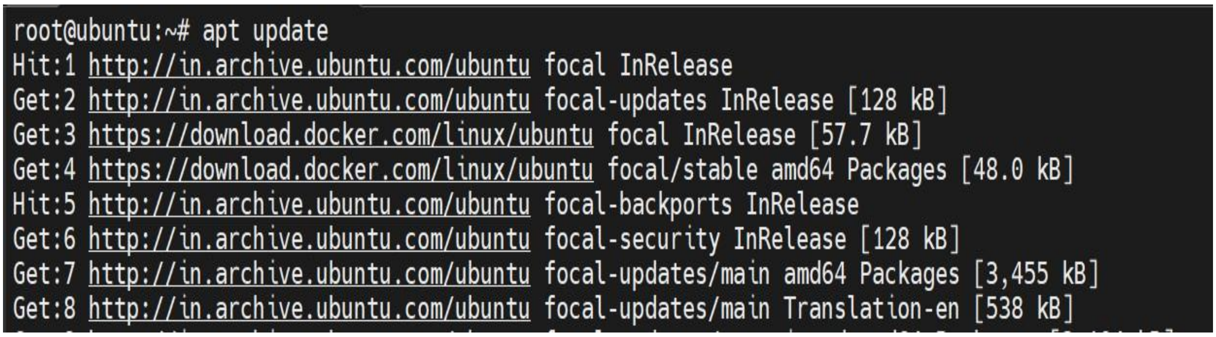
#813818 - Secure Website with SSL Certificate

Objective:
This task involves securing the previously deployed webserver with an SSL certificate, enabling encrypted communication between the web server and clients. You will:
- Obtain an SSL certificate,
- Configure the web server to use the SSL certificate.

Scenario:
To protect sensitive data transmitted between the web server and clients, it's essential to implement SSL encryption. This task focuses on obtaining and configuring an SSL certificate for the web server.

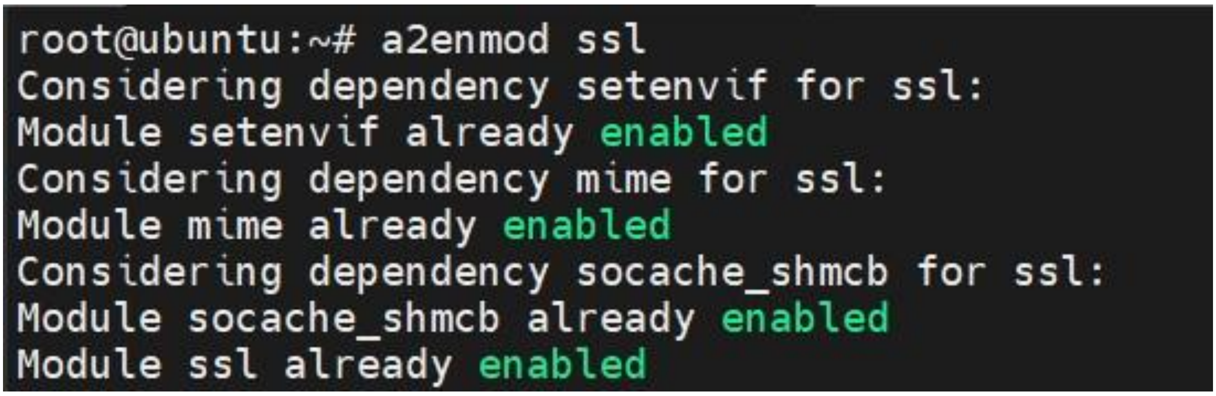
Completion Criteria:
- Apache is configured to use the SSL certificate for secure communication.
- The website is accessible via HTTPS and displays the expected content.

- **Update Package**



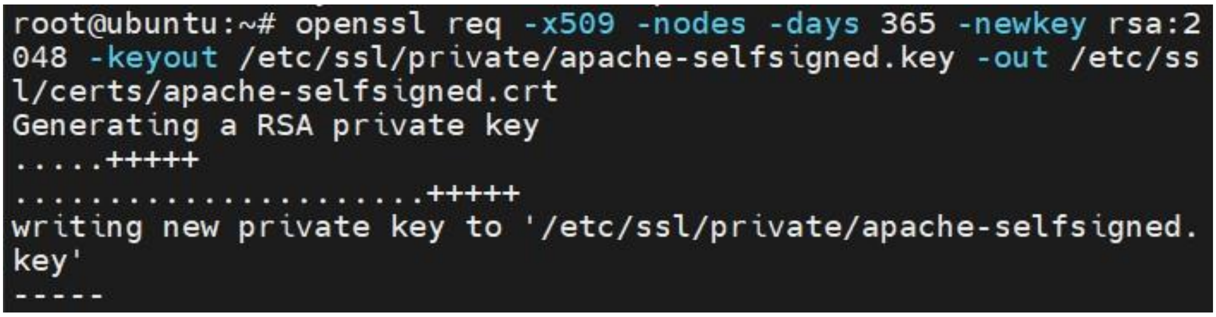
```
root@ubuntu:~# apt update
Hit:1 http://in.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://in.archive.ubuntu.com/ubuntu focal-updates InRelease [128 kB]
Get:3 https://download.docker.com/linux/ubuntu focal InRelease [57.7 kB]
Get:4 https://download.docker.com/linux/ubuntu focal/stable amd64 Packages [48.0 kB]
Hit:5 http://in.archive.ubuntu.com/ubuntu focal-backports InRelease
Get:6 http://in.archive.ubuntu.com/ubuntu focal-security InRelease [128 kB]
Get:7 http://in.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [3,455 kB]
Get:8 http://in.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [538 kB]
```

- **Enabling mode_ssl with command a2enmod**



```
root@ubuntu:~# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Module socache_shmcb already enabled
Module ssl already enabled
```

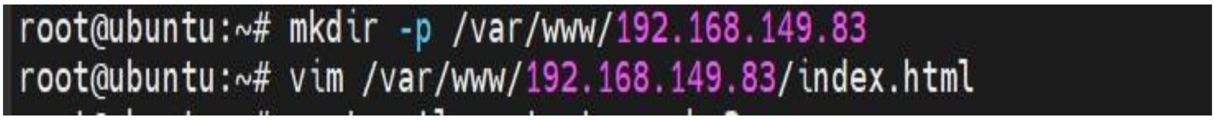
- **Obtain an SSL Certificate**
- Self-Signed SSL Certificate



```
root@ubuntu:~# openssl req -x509 -nodes -days 365 -newkey rsa:2
048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ss
l/certs/apache-selfsigned.crt
Generating a RSA private key
.....+++++
......................+++++
writing new private key to '/etc/ssl/private/apache-selfsigned.
key'
-----
```

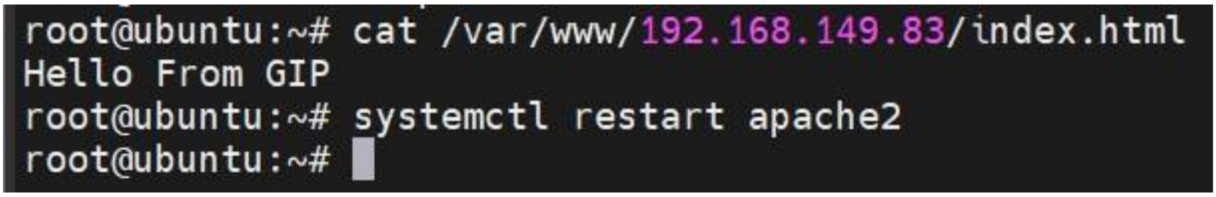
During this process, you'll be prompted to enter information like your country, state, and domain name.

- **Configure Apache to Use the SSL**



```
root@ubuntu:~# mkdir -p /var/www/192.168.149.83
root@ubuntu:~# vim /var/www/192.168.149.83/index.html
```

Make domain IP directory and inside create index.html file write message **"Hello From GIP"** and save the file.

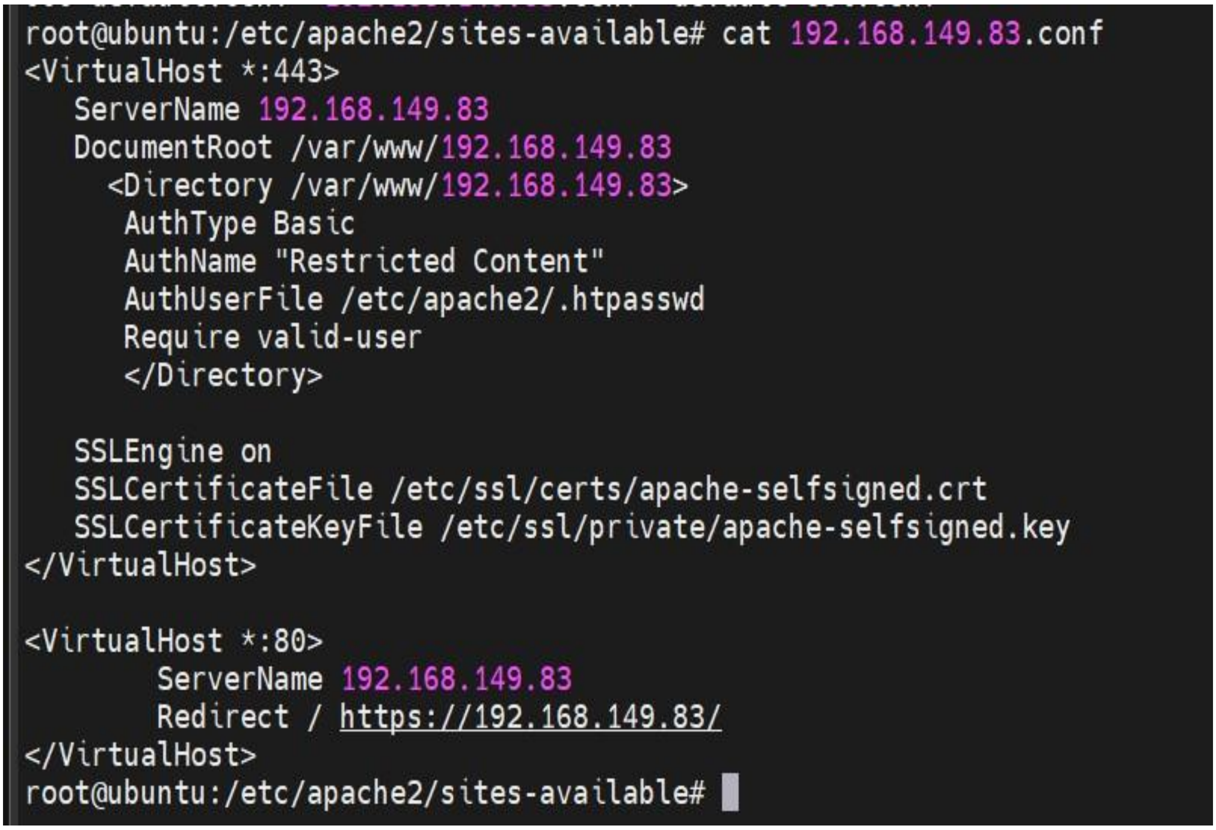


```
root@ubuntu:~# cat /var/www/192.168.149.83/index.html
Hello From GIP
root@ubuntu:~# systemctl restart apache2
root@ubuntu:~#
```

Restart the apache using command **systemctl restart apache2**

- **Configure Apache to Use SSL**

Create a new configuration file for the SSL-enabled site or modify the default SSL configuration and add the following lines to point to your SSL certificate and key files.



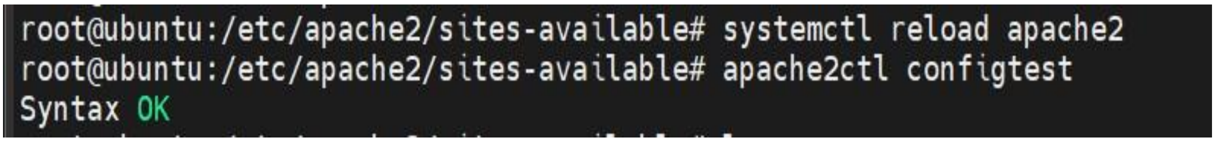
```
root@ubuntu:/etc/apache2/sites-available# cat 192.168.149.83.conf
<VirtualHost *:443>
   ServerName 192.168.149.83
   DocumentRoot /var/www/192.168.149.83
     <Directory /var/www/192.168.149.83>
      AuthType Basic
      AuthName "Restricted Content"
      AuthUserFile /etc/apache2/.htpasswd
      Require valid-user
      </Directory>

   SSLEngine on
   SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
   SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>

<VirtualHost *:80>
        ServerName 192.168.149.83
        Redirect / https://192.168.149.83/
</VirtualHost>
root@ubuntu:/etc/apache2/sites-available#
```

Now save and exit the file.

- **Enable the SSL Site and reload Apache**

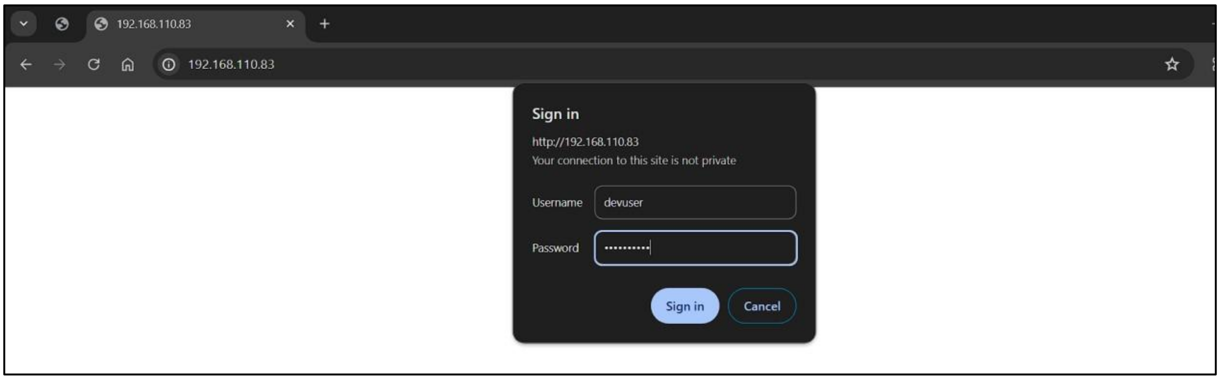


```
root@ubuntu:/etc/apache2/sites-available# systemctl reload apache2
root@ubuntu:/etc/apache2/sites-available# apache2ctl configtest
Syntax OK
```

If output is syntax ok then, your configuration file has no syntax errors. We can safely reload Apache to implement our changes

- **Test HTTPS Access**

Open webserver and navigate to the IP address of your Ubuntu server

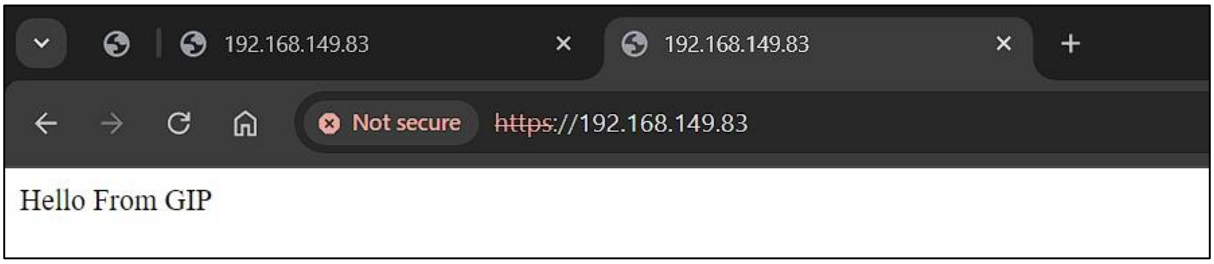


Sign in

http://192.168.110.83
Your connection to this site is not private

Username: devuser

Password: ••••••••••



~~https~~://192.168.149.83

Hello From GIP

After entering the credentials, you should see the message **"Hello From GIP".**

Conclusion:

This document serves as a comprehensive guide to securing an Apache web server with SSL, ensuring your site is accessible via HTTPS. It outlines the necessary steps to obtain an SSL certificate, configure Apache accordingly, and verify the successful implementation of SSL. By following these steps, you can enhance the security of your web server and protect sensitive data transmitted between your server and clients.

Thank You

dasaremahir333@gmail.com