

Configure Apache server

This tutorial aims at teaching how to configure an Apache server. By the end of this tutorial, one should be able to:

1. Install and configure a simple Apache server;
2. Enable automatic serving on users' home folders;
3. Configure HTTPS using certbot.

In this tutorial, it is considered that the subjacent Linux distro is CentOS 7.

Install Apache

First, install Apache using the package manager.

```
sudo yum -y install httpd
```

Enable and start Apache server.

```
sudo systemctl enable httpd.service
sudo systemctl start httpd.service
```

By default CentOS 7 firewall disables all the ports that are not used by the essential services. To allow Apache to communicate with the outside world, HTTP and HTTPS ports have to be enabled.

```
sudo firewall-cmd --add-service={http,https} --permanent
sudo firewall-cmd --reload
```

Additionally to the regular permissions of folders and files, CentOS 7 has another layer of security called Security-Enhanced Linux (SELinux). By default, SELinux is enabled on CentOS 7 and **it is a ridiculous practice to disable it just because it is easy!**. To give Apache privileges to use the home folders, use NFS mounted folders (this might be the case if centralized storage is being used), and communicate with other devices in the network, some SE booleans must be set.

```
sudo setsebool -P httpd_use_nfs 1
sudo setsebool -P httpd_can_network_connect 1
sudo setsebool -P httpd_enable_homedirs true
```

Configure Apache to use home folders

To allow Apache to serve files in users' home folders without changing the ownership of those files, the user launching the web server process must be the user who owns the files, and not Apache's user. First, install ITK.

```
sudo yum install -y httpd-itk
```

Edit `/etc/httpd/conf.modules.d/00-mpm-itk.conf`, uncommenting the `#LoadModule` line by removing the leading `#`.

Replace the content of the configuration file `/etc/httpd/conf.d/userdir.conf` with the following:

[userdir.conf](#)

```
<VirtualHost *:80>
#
# UserDir: The name of the directory that is appended onto a user's
home
# directory if a ~user request is received.
#
# The path to the end user account 'public_html' directory must be
# accessible to the webserver userid. This usually means that
~userid
# must have permissions of 711, ~userid/public_html must have
permissions
# of 755, and documents contained therein must be world-readable.
# Otherwise, the client will only receive a "403 Forbidden"
message.
#
<IfModule mod_userdir.c>
#
# UserDir is disabled by default since it can confirm the
presence
# of a username on the system (depending on home directory
# permissions).
#
UserDir enabled

#
# To enable requests to /~user/ to serve the user's public_html
# directory, remove the "UserDir disabled" line above, and
uncomment
# the following line instead:
#
UserDir web
</IfModule>

#
# Control access to UserDir directories. The following is an
example
# for a site where these directories are writable and each user has
# its own httpd process ran by itself.
#

RewriteEngine on
RewriteRule /~([a-z]+) - [E=ITKUID:$1]
```

```
<Directory "/home/*/web">
    AllowOverride FileInfo AuthConfig Limit Indexes
    Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
    Require method GET POST OPTIONS
    AssignUserIDExpr %{reqenv:ITKUID}
</Directory>
</VirtualHost>
```

Configure HTTPS

HTTPS is preferred over HTTP because it enables encryption thus enhancing security. To enable HTTPS, OpenSSL is required.

```
sudo yum install -y mod_ssl python-certbot-apache
```

For simplicity, this tutorial uses Certbot to acquire and manage certificates.

```
sudo certbot --apache -d example.com -d www.example.com
```

Replace `example.com` and `www.example.com` by your registered domain(s).

Finally, just restart Apache server.

```
sudo systemctl restart httpd
```

Setup default web page by redirecting it

When trying to access the default address `example.com`, one might wish it to be redirected to someone's web directory in his home folder. For that, create the file `/var/www/html/index.html` with the following content:

[index.html](#)

```
<!DOCTYPE html>
<html>
<head>
    <!-- HTML meta refresh URL redirection -->
    <meta http-equiv="refresh"
    content="0; url=https://example.com/~mainuser/">
</head>
<body>
    <p>Redirecting to https://example.com/~mainuser/</p>
</body>
</html>
```

Replace `example.com` by your registered domain, and `mainuser` by the username of the user you want to redirect the primary address to.

From:

<https://joaomiguelvieira.ece.utah.edu/~joaovieira/wiki/> - **João Vieira's Wiki**

Permanent link:

https://joaomiguelvieira.ece.utah.edu/~joaovieira/wiki/doku.php?id=configure_apache_server

Last update: **2019/05/21 19:18**

