TASK 1: How to setup Apache Server

Step 1: Update your local package index

Command:

```
$ sudo apt update
```

Install the apache2 package:

Command:

```
$ sudo apt install apache2
```

Step 2: Configure Your Firewall

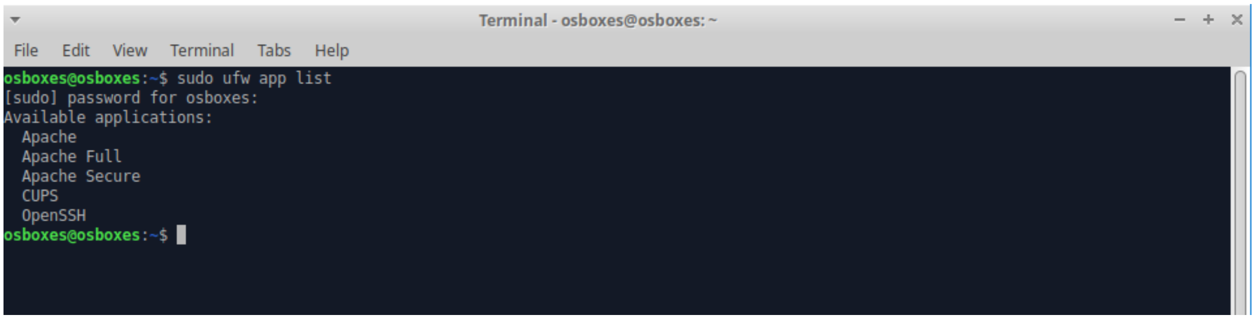
Although the Apache installation process is complete, there is one more additional step. Configure the default UFW firewall to allow traffic on port 80.

Start by displaying available app profiles on UFW:

Command:

```
$ sudo ufw app list
```

The terminal should respond by listing all available application profiles, as seen in the example below.



Terminal - osboxes@osboxes: ~
File Edit View Terminal Tabs Help

```
osboxes@osboxes:~$ sudo ufw app list
[sudo] password for osboxes:
Available applications:
  Apache
  Apache Full
  Apache Secure
  CUPS
  OpenSSH
osboxes@osboxes:~$ 
```

Let’s enable the most restrictive profile that will still allow the traffic you’ve configured, permitting traffic on port 80 (normal, unencrypted web traffic):

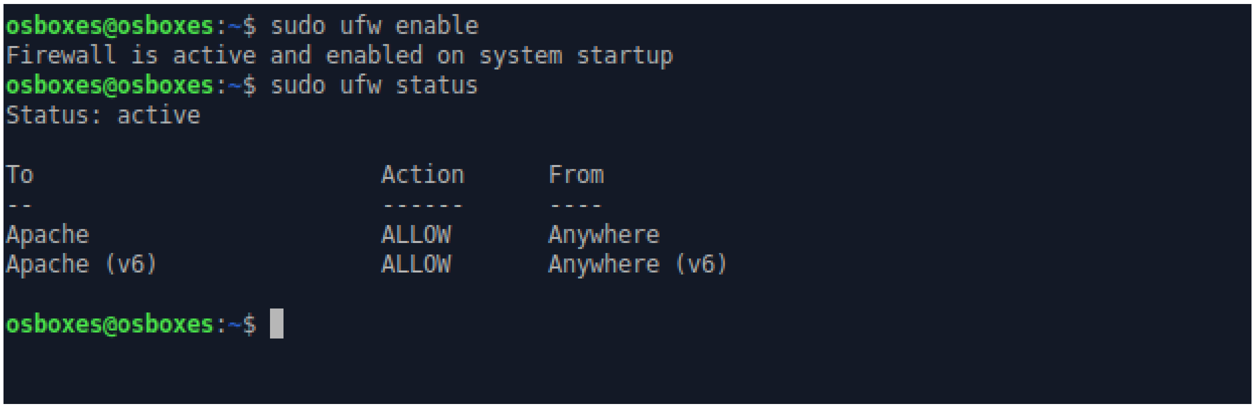
Command:

```
$ sudo ufw allow 'Apache'
```

Verify the change:

```
$ sudo ufw status
```

In case your firewall's status is inactive, you can use the command "sudo ufw enable" to active.



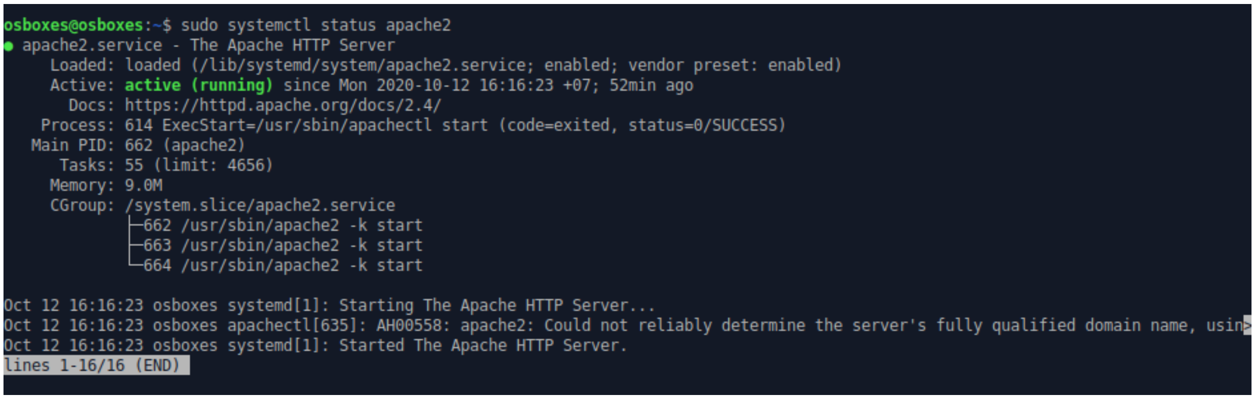
```
osboxes@osboxes:~$ sudo ufw enable
Firewall is active and enabled on system startup
osboxes@osboxes:~$ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
Apache                     ALLOW       Anywhere
Apache (v6)                ALLOW       Anywhere (v6)

osboxes@osboxes:~$ 
```

Step 3: Checking your Web Server

Check with the systemd init system to make sure the service is running by typing:

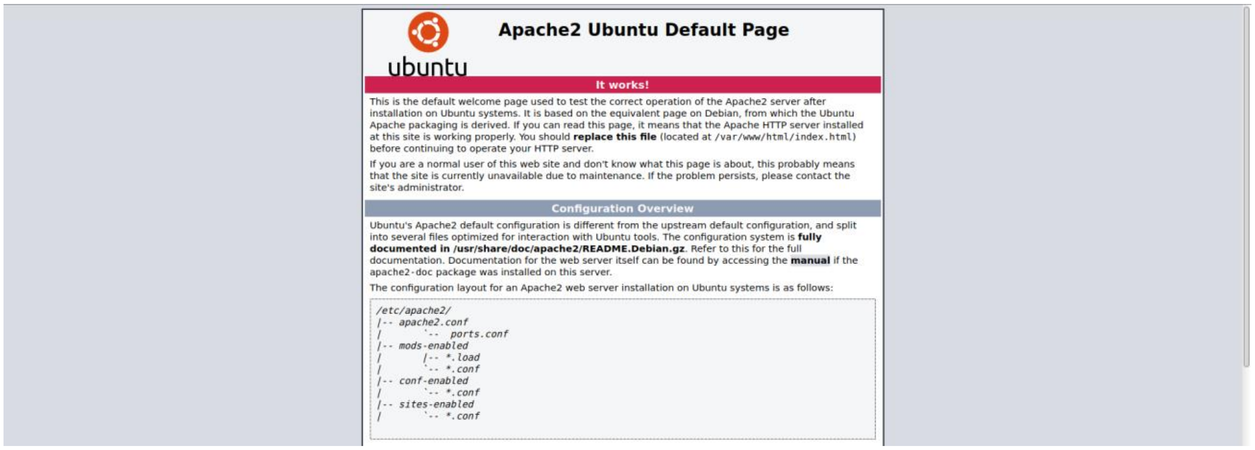
Command:

```
$ sudo systemctl status apache2
```



```
osboxes@osboxes:~$ sudo systemctl status apache2
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2020-10-12 16:16:23 +07; 52min ago
       Docs: https://httpd.apache.org/docs/2.4/
    Process: 614 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
   Main PID: 662 (apache2)
      Tasks: 55 (limit: 4656)
     Memory: 9.0M
     CGroup: /system.slice/apache2.service
             ├─662 /usr/sbin/apache2 -k start
             ├─663 /usr/sbin/apache2 -k start
             └─664 /usr/sbin/apache2 -k start

Oct 12 16:16:23 osboxes systemd[1]: Starting The Apache HTTP Server...
Oct 12 16:16:23 osboxes apachectl[635]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, usin>
Oct 12 16:16:23 osboxes systemd[1]: Started The Apache HTTP Server.
lines 1-16/16 (END)
```

Access the default Apache landing page to confirm that the software is running properly through your IP address: http://your_server_ip



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the apache2-doc package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|       `--  ports.conf
|-- mods-enabled
|       |-- *.load
|       `-- *.conf
|-- conf-enabled
|       `-- *.conf
|-- sites-enabled
|       `-- *.conf
```

Step 4: Set up password Authentication

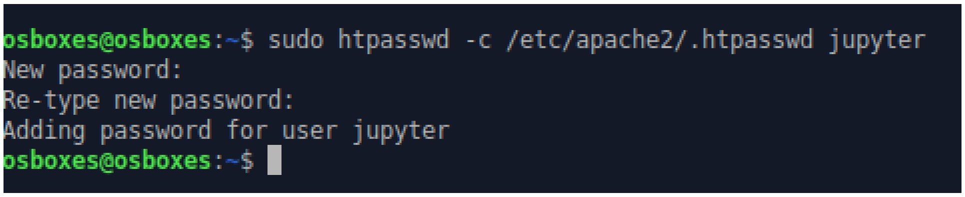
* Installing the Apache Utilities Package

Command:

```
$ sudo apt-get update
$ sudo apt-get install apache2-utils
```

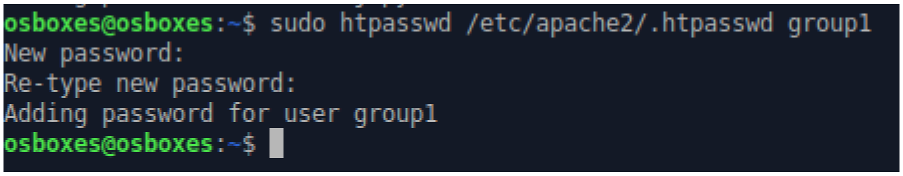
— Creating the Password File

We now have access to the htpasswd command. We can use this to create a password file that Apache can use to authenticate users. We will create a hidden file for this purpose called .htpasswd within our /etc/apache2 configuration directory. The first time we use this utility, we need to add the -c option to create the specified file. We specify a username (jupyter in this example) at the end of the command to create a new entry within the file:



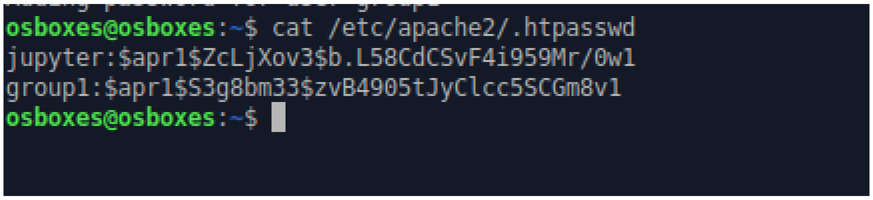
```
osboxes@osboxes:~$ sudo htpasswd -c /etc/apache2/.htpasswd jupyter
New password:
Re-type new password:
Adding password for user jupyter
osboxes@osboxes:~$
```

Leave out the -c argument for any additional users you wish to add:



```
osboxes@osboxes:~$ sudo htpasswd /etc/apache2/.htpasswd group1
New password:
Re-type new password:
Adding password for user group1
osboxes@osboxes:~$
```

If we view the contents of the file, we can see the username and the encrypted password for each record:



```
osboxes@osboxes:~$ cat /etc/apache2/.htpasswd
jupyter:$apr1$ZcLjXov3$b.L58CdCSvF4i959Mr/0w1
group1:$apr1$S3g8bm33$zvB4905tJyClcc5SCGm8v1
osboxes@osboxes:~$
```

STEP 5: — Configuring Apache Password Authentication

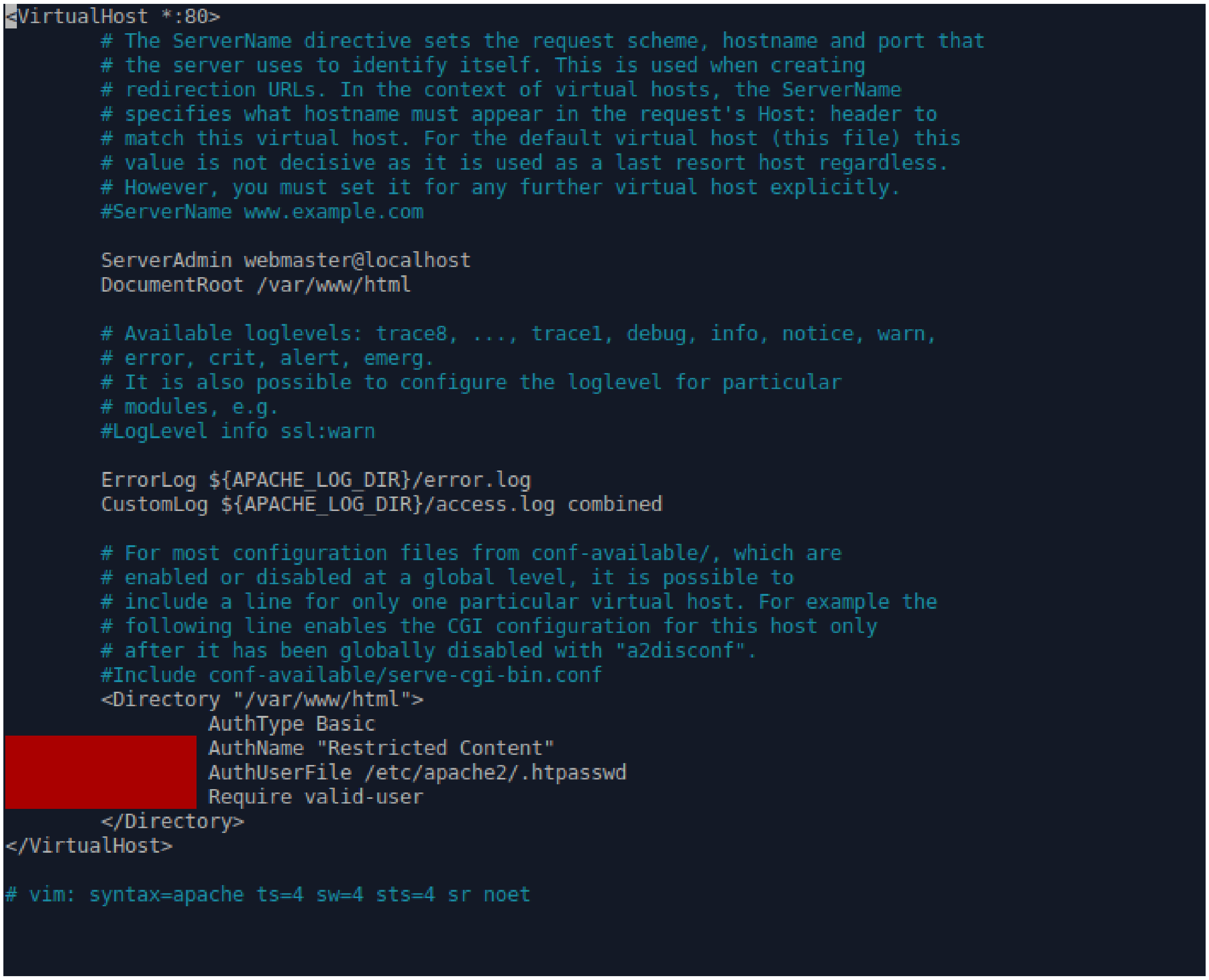
Configuring Access Control within the Virtual Host Definition

Begin by opening up the virtual host file that you wish to add a restriction to. For our example, we’ll be using the 000-default.conf file that holds the default virtual host installed through Ubuntu’s apache package:

```
$ sudo nano /etc/apache2/sites-enabled/000-default.conf
```

Authentication is done on a per-directory basis. To set up authentication, you will need to target the directory you wish to restrict with a <Directory ___> block. In our example, we’ll restrict the entire document root, but you can modify this listing to only target a specific directory within the web space.

Within this directory block, specify that we wish to set up Basic authentication. For the AuthName, choose a realm name that will be displayed to the user when prompting for credentials. Use the AuthUserFile directive to point Apache to the password file we created. Finally, we will require a valid-user to access this resource, which means anyone who can verify their identity with a password will be allowed in:



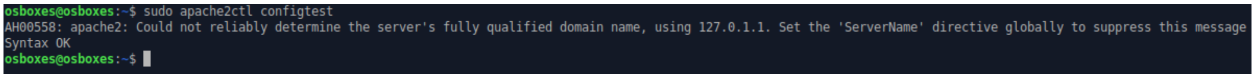
```
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        #ServerName www.example.com

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html

        # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
        # error, crit, alert, emerg.
        # It is also possible to configure the loglevel for particular
        # modules, e.g.
        #LogLevel info ssl:warn

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        # For most configuration files from conf-available/, which are
        # enabled or disabled at a global level, it is possible to
        # include a line for only one particular virtual host. For example the
        # following line enables the CGI configuration for this host only
        # after it has been globally disabled with "a2disconf".
        #Include conf-available/serve-cgi-bin.conf
        <Directory "/var/www/html">
                AuthType Basic
[redacted]      AuthName "Restricted Content"
                AuthUserFile /etc/apache2/.htpasswd
                Require valid-user
        </Directory>
</VirtualHost>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

Save and close the file when you are finished.

Before restarting the web server, you can check the configuration with the following command:

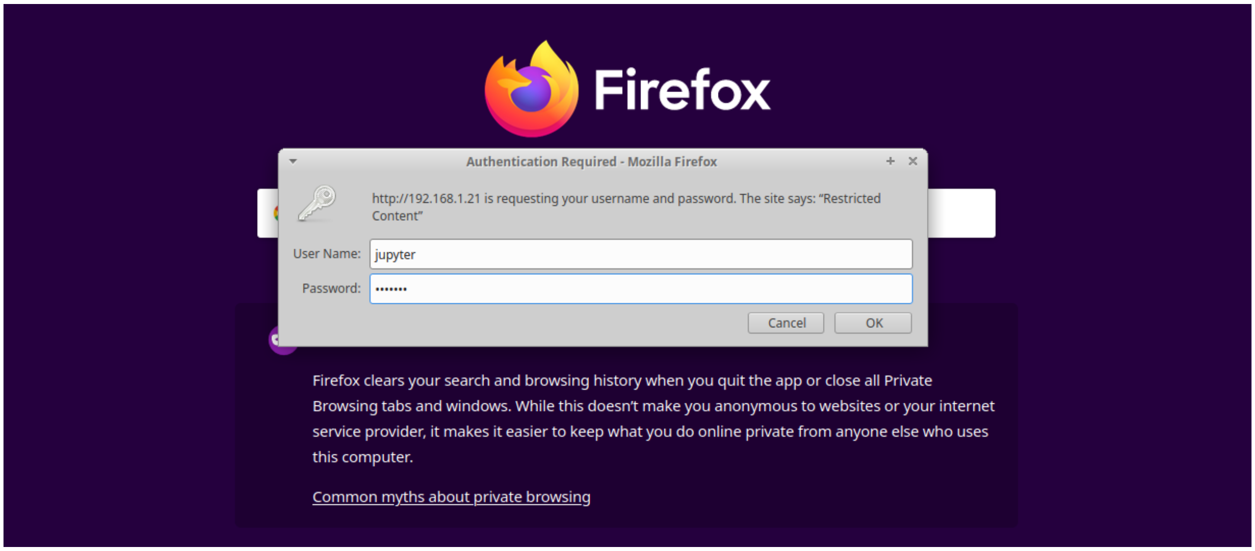
Command:

```
$ sudo apache2ctl configtest
```



```
osboxes@osboxes:~$ sudo apache2ctl configtest
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message
Syntax OK
osboxes@osboxes:~$ 
```

— Confirming Password Authentication


Firefox
Authentication Required - Mozilla Firefox
Cancel
OK
Common myths about private browsing

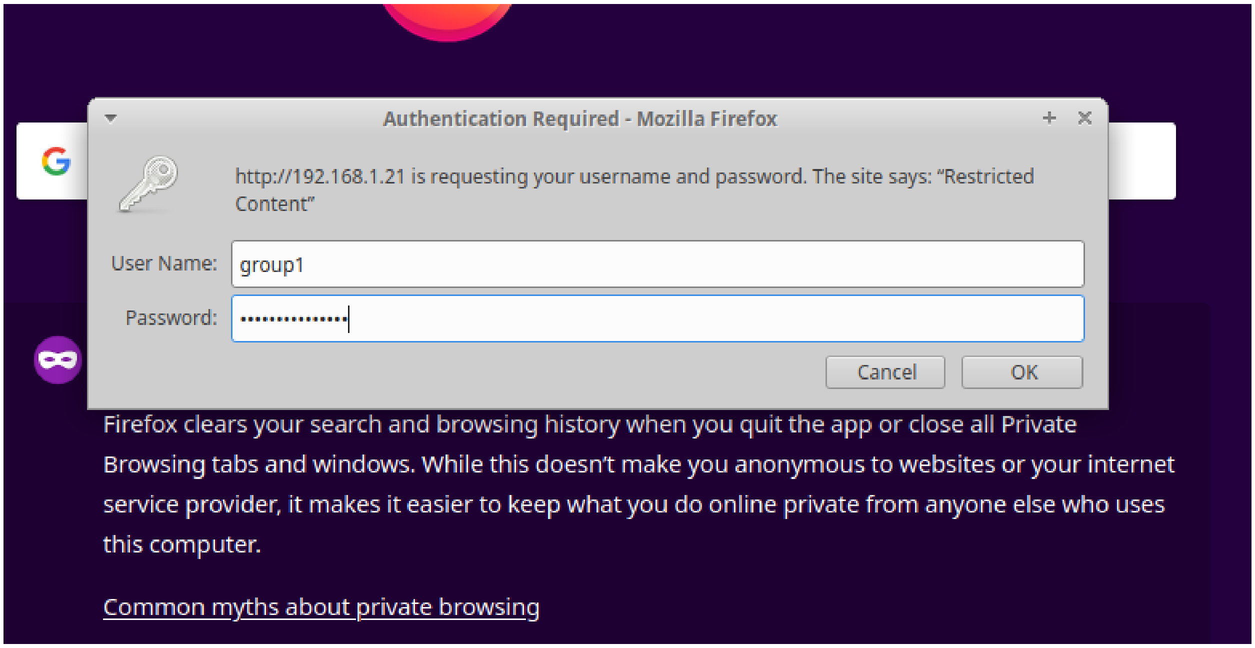
http://192.168.1.21 is requesting your username and password. The site says: “Restricted Content”

User Name: jupyter

Password: •••••••

Firefox clears your search and browsing history when you quit the app or close all Private Browsing tabs and windows. While this doesn’t make you anonymous to websites or your internet service provider, it makes it easier to keep what you do online private from anyone else who uses this computer.

In case I type wrong password, I can not access to the web server:

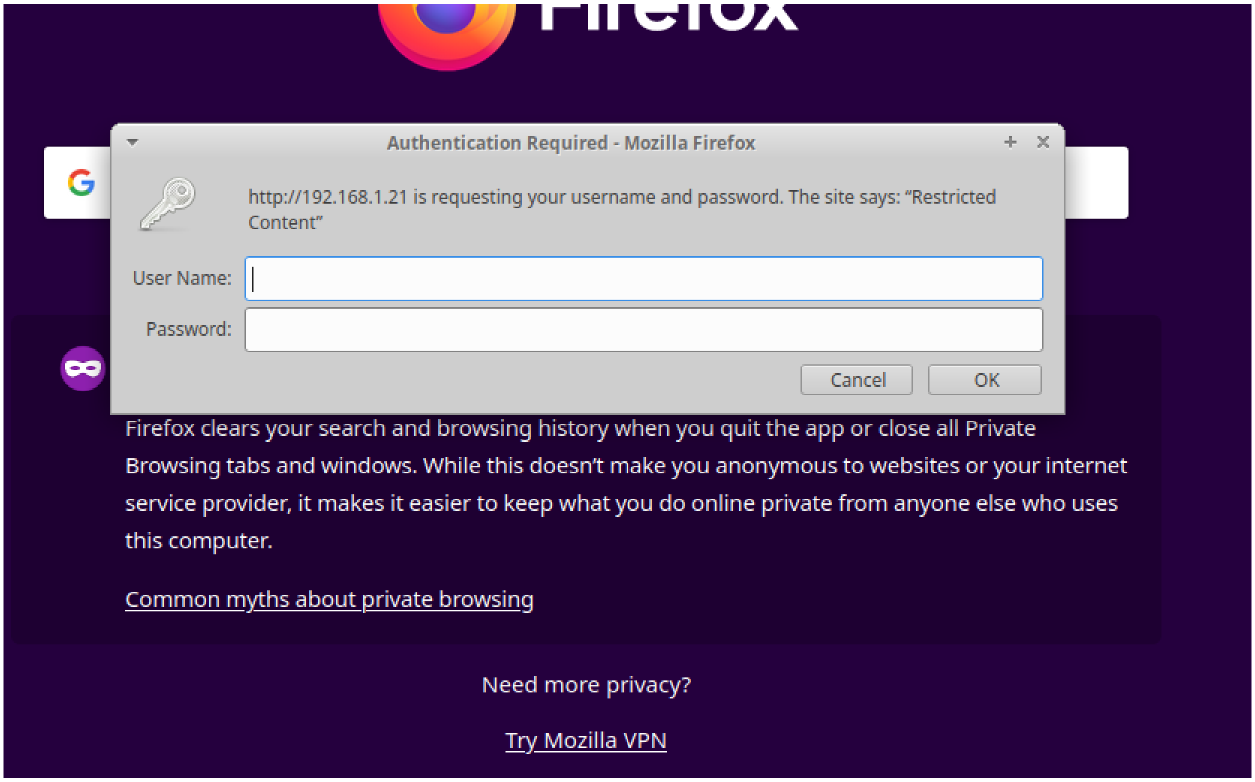

Authentication Required - Mozilla Firefox
Cancel
OK
Common myths about private browsing

http://192.168.1.21 is requesting your username and password. The site says: “Restricted Content”

User Name: group1

Password: •••••••••••••••

Firefox clears your search and browsing history when you quit the app or close all Private Browsing tabs and windows. While this doesn’t make you anonymous to websites or your internet service provider, it makes it easier to keep what you do online private from anyone else who uses this computer.



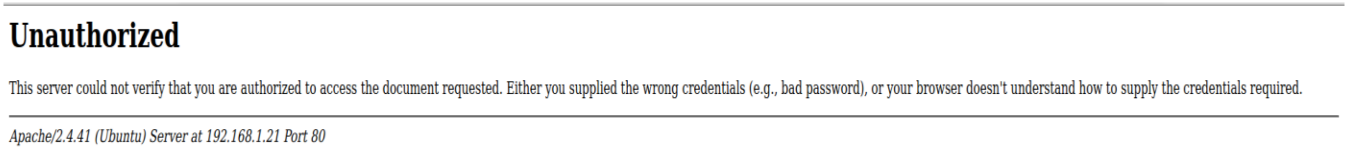
http://192.168.1.21 is requesting your username and password. The site says: “Restricted Content”

Firefox clears your search and browsing history when you quit the app or close all Private Browsing tabs and windows. While this doesn’t make you anonymous to websites or your internet service provider, it makes it easier to keep what you do online private from anyone else who uses this computer.

Common myths about private browsing

Need more privacy?

Try Mozilla VPN

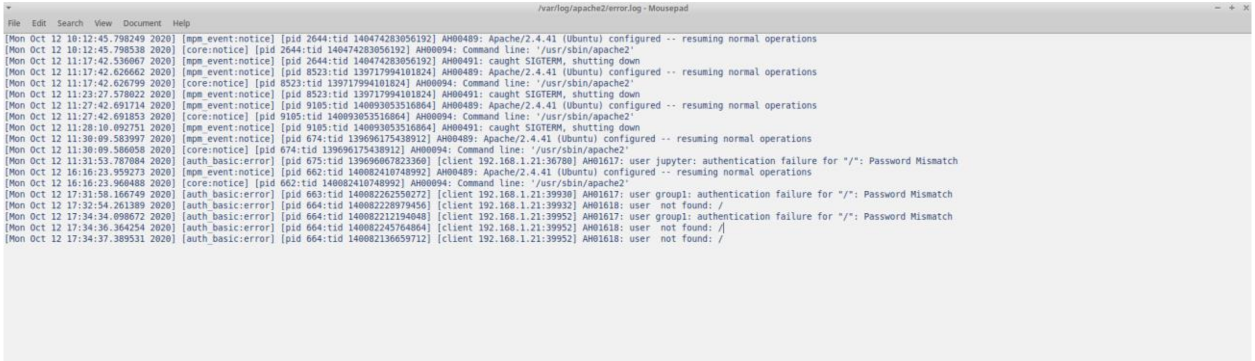


Unauthorized

This server could not verify that you are authorized to access the document requested. Either you supplied the wrong credentials (e.g., bad password), or your browser doesn't understand how to supply the credentials required.

Apache/2.4.41 (Ubuntu) Server at 192.168.1.21 Port 80

In case you want to see all the log files, this is the path: /var/log/apache2/. It contains all the process and action from user with the web server.


/var/log/apache2/error.log - Mousepad
File Edit Search View Document Help

```
[Mon Oct 12 10:12:45.798249 2020] [mpm_event:notice] [pid 2644:tid 140474283056192] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operations
[Mon Oct 12 10:12:45.798538 2020] [core:notice] [pid 2644:tid 140474283056192] AH00094: Command line: '/usr/sbin/apache2'
[Mon Oct 12 11:17:42.536067 2020] [mpm_event:notice] [pid 2644:tid 140474283056192] AH00491: caught SIGTERM, shutting down
[Mon Oct 12 11:17:42.626662 2020] [mpm_event:notice] [pid 8523:tid 139717994101824] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operations
[Mon Oct 12 11:17:42.626799 2020] [core:notice] [pid 8523:tid 139717994101824] AH00094: Command line: '/usr/sbin/apache2'
[Mon Oct 12 11:23:27.578022 2020] [mpm_event:notice] [pid 8523:tid 139717994101824] AH00491: caught SIGTERM, shutting down
[Mon Oct 12 11:27:42.691714 2020] [mpm_event:notice] [pid 9105:tid 140093053516864] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operations
[Mon Oct 12 11:27:42.691853 2020] [core:notice] [pid 9105:tid 140093053516864] AH00094: Command line: '/usr/sbin/apache2'
[Mon Oct 12 11:28:10.092751 2020] [mpm_event:notice] [pid 9105:tid 140093053516864] AH00491: caught SIGTERM, shutting down
[Mon Oct 12 11:30:09.583997 2020] [mpm_event:notice] [pid 674:tid 139696175438912] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operations
[Mon Oct 12 11:30:09.586058 2020] [core:notice] [pid 674:tid 139696175438912] AH00094: Command line: '/usr/sbin/apache2'
[Mon Oct 12 11:31:53.787084 2020] [auth_basic:error] [pid 675:tid 139696067823360] [client 192.168.1.21:36780] AH01617: user jupyter: authentication failure for "/": Password Mismatch
[Mon Oct 12 16:16:23.959273 2020] [mpm_event:notice] [pid 662:tid 140082410748992] AH00489: Apache/2.4.41 (Ubuntu) configured -- resuming normal operations
[Mon Oct 12 16:16:23.960488 2020] [core:notice] [pid 662:tid 140082410748992] AH00094: Command line: '/usr/sbin/apache2'
[Mon Oct 12 17:31:58.166749 2020] [auth_basic:error] [pid 663:tid 140082262550272] [client 192.168.1.21:39930] AH01617: user group1: authentication failure for "/": Password Mismatch
[Mon Oct 12 17:32:54.261389 2020] [auth_basic:error] [pid 664:tid 140082228979456] [client 192.168.1.21:39932] AH01618: user  not found: /
[Mon Oct 12 17:34:34.098672 2020] [auth_basic:error] [pid 664:tid 140082212194048] [client 192.168.1.21:39952] AH01617: user group1: authentication failure for "/": Password Mismatch
[Mon Oct 12 17:34:36.364254 2020] [auth_basic:error] [pid 664:tid 140082245764864] [client 192.168.1.21:39952] AH01618: user  not found: /
[Mon Oct 12 17:34:37.389531 2020] [auth_basic:error] [pid 664:tid 140082136659712] [client 192.168.1.21:39952] AH01618: user  not found: /
```

REFERENCES:

1. https://phoenixnap.com/kb/how-to-install-apache-web-server-on-ubuntu-18-04
2. https://www.digitalocean.com/community/tutorials/how-to-install-the-apache-web-server-on-ubuntu-18-04-quickstart
3. https://www.digitalocean.com/community/tutorials/how-to-set-up-password-authentication-with-apache-on-ubuntu-16-04