Course Content Hide Player

☐ 1. Introduction to HTML, CSS, JavaScript & How websites work? | Web Development Tutorials #1
Free YouTube Video

☐ 2. HTML Tutorial: Installing VS Code & Live Server | Web Development Tutorials #2
Free YouTube Video

☐ 3. HTML Tutorial: Basic Structure of a Website | Web Development Tutorials #3
Free YouTube Video

☐ 4. HTML Tutorial: Title, Script, Link & Meta Tags | Web Development

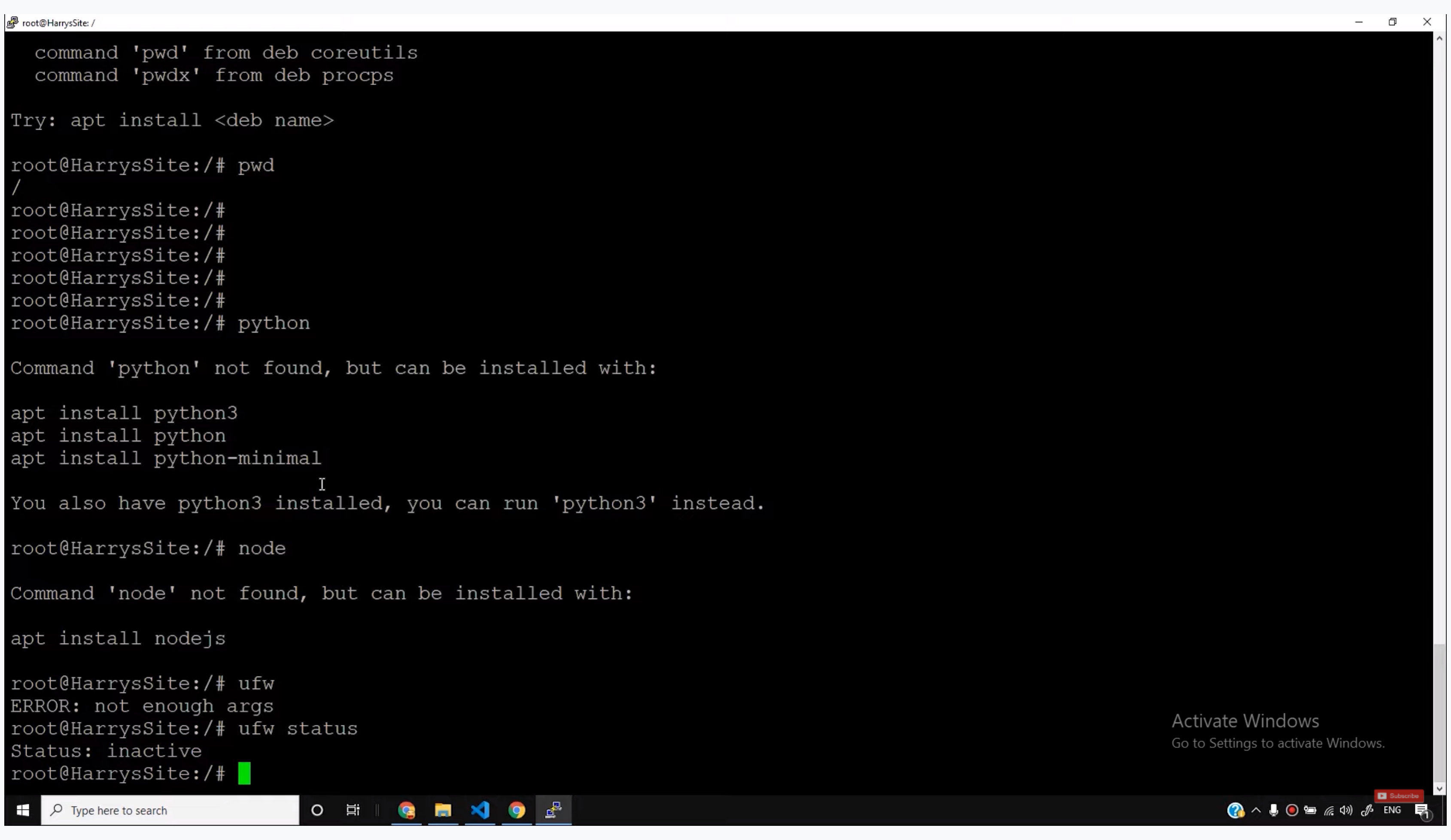
Overview Q&A Files Announcements

Hosting Tutorial: Installing Apache2 Web Server on a VPS | Web Development Tutorials #92

Hosting Tutorial- Installing Apache 2 Web Server on a VPS

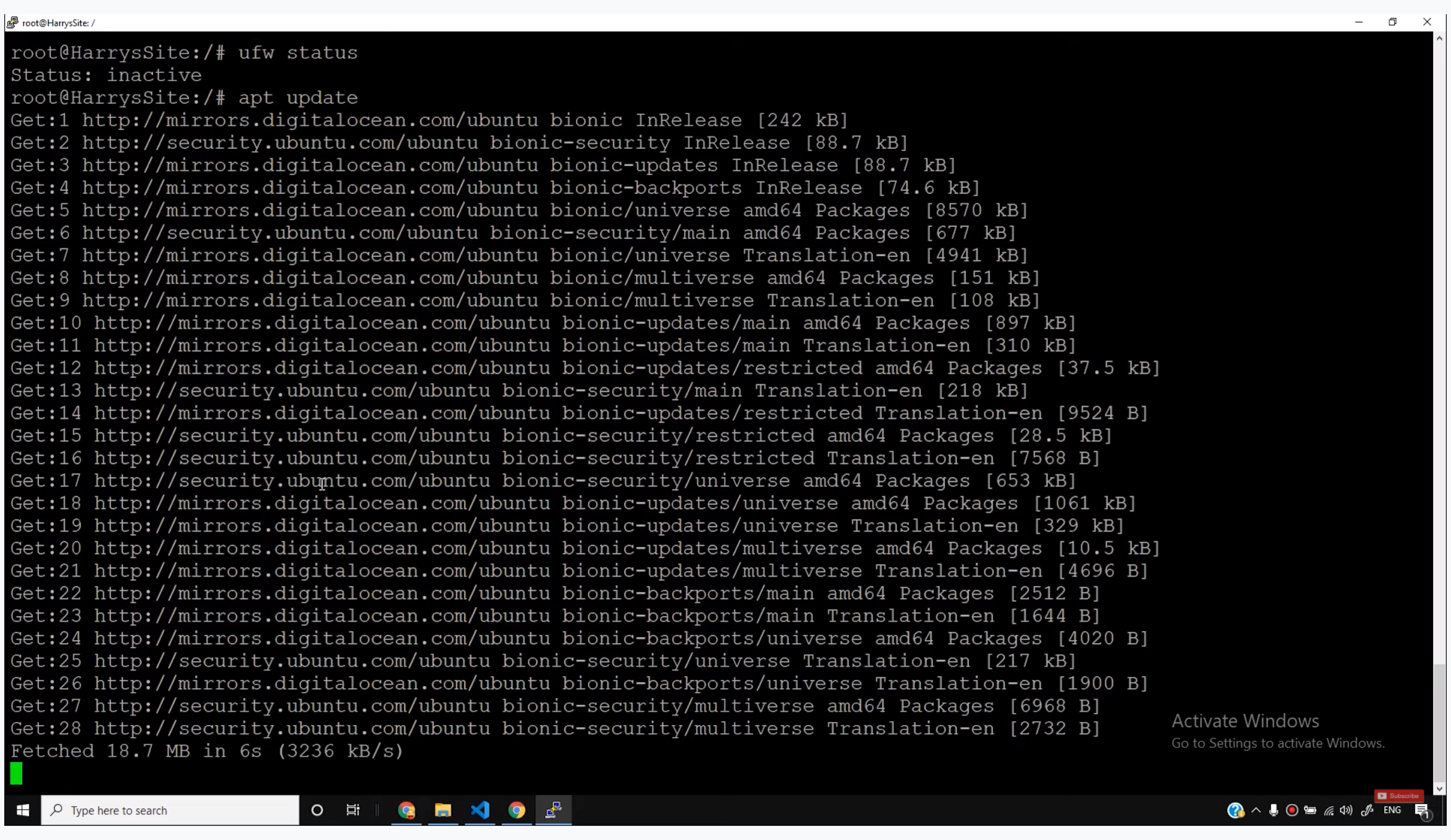
In the last tutorial, we have clearly understood how a person can easily set up his server through Ubuntu. We have also seen how to manage the server in DigitalOcean. Although we can install several other operating systems in the server like python, node etc. but our main objective is to host our static website on it.

To start serving our files, we need to allow ***outbound traffic***. Now if you write ***ufw status***, then you can expect the output as follows-



```
  command 'pwd' from deb coreutils
  command 'pwdx' from deb procps

Try: apt install <deb name>

root@HarrysSite:/# pwd
/
root@HarrysSite:/#
root@HarrysSite:/#
root@HarrysSite:/#
root@HarrysSite:/#
root@HarrysSite:/#
root@HarrysSite:/# python

Command 'python' not found, but can be installed with:

apt install python3
apt install python
apt install python-minimal

You also have python3 installed, you can run 'python3' instead.

root@HarrysSite:/# node

Command 'node' not found, but can be installed with:

apt install nodejs

root@HarrysSite:/# ufw
ERROR: not enough args
root@HarrysSite:/# ufw status
Status: inactive
root@HarrysSite:/# 
```

UFW is a firewall whose status is inactive now. It means that anyone can configure with our system's requirements and hence, we need to activate it. Along with it, we need to install a tried and trusted software. We need to run the command called ***apt update*** and we will get the output as follows-



```
root@HarrysSite:/# ufw status
Status: inactive
root@HarrysSite:/# apt update
Get:1 http://mirrors.digitalocean.com/ubuntu bionic InRelease [242 kB]
Get:2 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Get:3 http://mirrors.digitalocean.com/ubuntu bionic-updates InRelease [88.7 kB]
Get:4 http://mirrors.digitalocean.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://mirrors.digitalocean.com/ubuntu bionic/universe amd64 Packages [8570 kB]
Get:6 http://security.ubuntu.com/ubuntu bionic-security/main amd64 Packages [677 kB]
Get:7 http://mirrors.digitalocean.com/ubuntu bionic/universe Translation-en [4941 kB]
Get:8 http://mirrors.digitalocean.com/ubuntu bionic/multiverse amd64 Packages [151 kB]
Get:9 http://mirrors.digitalocean.com/ubuntu bionic/multiverse Translation-en [108 kB]
Get:10 http://mirrors.digitalocean.com/ubuntu bionic-updates/main amd64 Packages [897 kB]
Get:11 http://mirrors.digitalocean.com/ubuntu bionic-updates/main Translation-en [310 kB]
Get:12 http://mirrors.digitalocean.com/ubuntu bionic-updates/restricted amd64 Packages [37.5 kB]
Get:13 http://security.ubuntu.com/ubuntu bionic-security/main Translation-en [218 kB]
Get:14 http://mirrors.digitalocean.com/ubuntu bionic-updates/restricted Translation-en [9524 B]
Get:15 http://security.ubuntu.com/ubuntu bionic-security/restricted amd64 Packages [28.5 kB]
Get:16 http://security.ubuntu.com/ubuntu bionic-security/restricted Translation-en [7568 B]
Get:17 http://security.ubuntu.com/ubuntu bionic-security/universe amd64 Packages [653 kB]
Get:18 http://mirrors.digitalocean.com/ubuntu bionic-updates/universe amd64 Packages [1061 kB]
Get:19 http://mirrors.digitalocean.com/ubuntu bionic-updates/universe Translation-en [329 kB]
Get:20 http://mirrors.digitalocean.com/ubuntu bionic-updates/multiverse amd64 Packages [10.5 kB]
Get:21 http://mirrors.digitalocean.com/ubuntu bionic-updates/multiverse Translation-en [4696 B]
Get:22 http://mirrors.digitalocean.com/ubuntu bionic-backports/main amd64 Packages [2512 B]
Get:23 http://mirrors.digitalocean.com/ubuntu bionic-backports/main Translation-en [1644 B]
Get:24 http://mirrors.digitalocean.com/ubuntu bionic-backports/universe amd64 Packages [4020 B]
Get:25 http://security.ubuntu.com/ubuntu bionic-security/universe Translation-en [217 kB]
Get:26 http://mirrors.digitalocean.com/ubuntu bionic-backports/universe Translation-en [1900 B]
Get:27 http://security.ubuntu.com/ubuntu bionic-security/multiverse amd64 Packages [6968 B]
Get:28 http://security.ubuntu.com/ubuntu bionic-security/multiverse Translation-en [2732 B]
Fetched 18.7 MB in 6s (3236 kB/s)
```

This function will update all the packages installed in the server. APT is also called the package manager of Ubuntu. It allows us to install the latest set of packages.

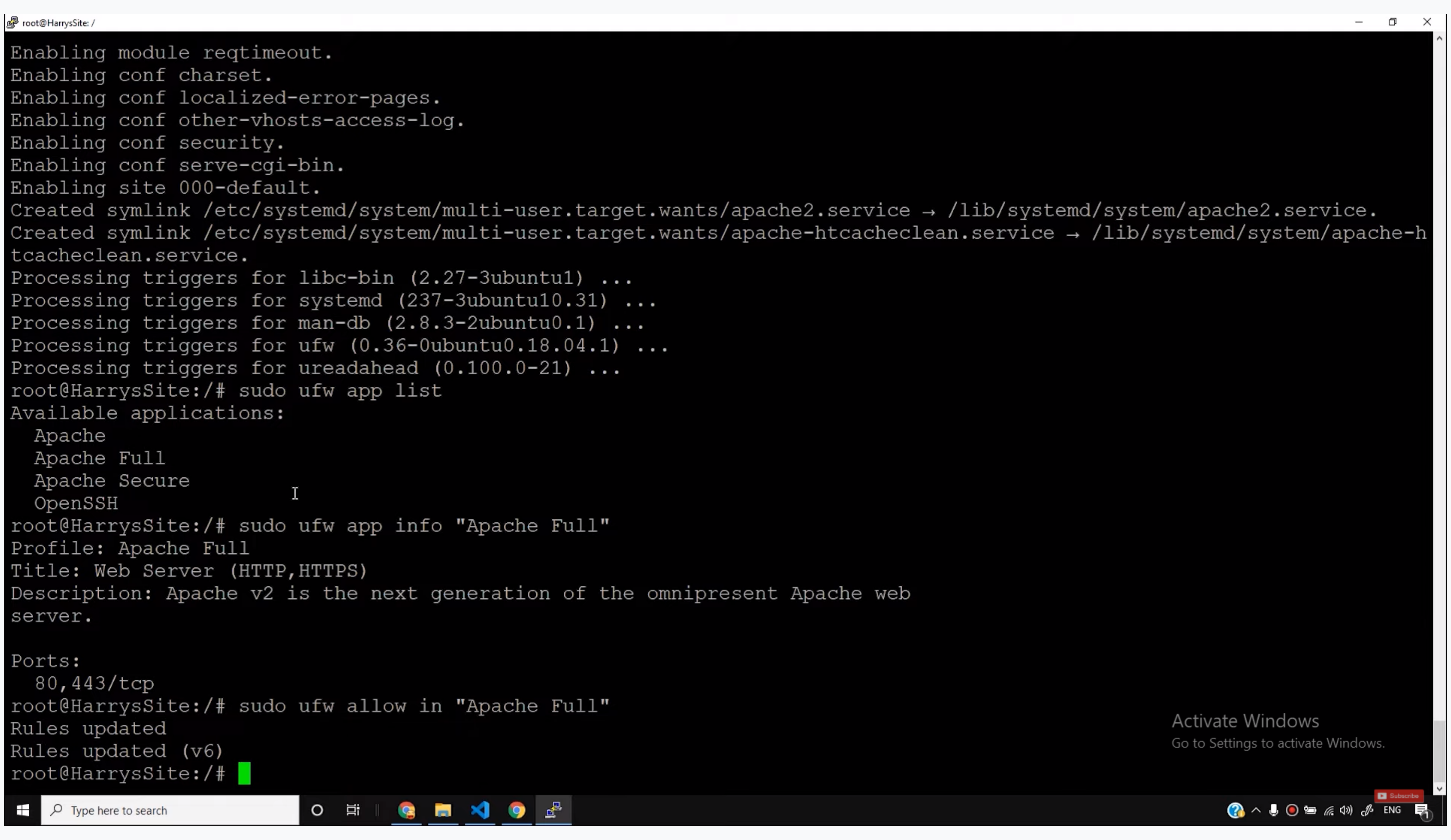
We will now install the software called *lamp stack*. A **LAMP Stack** is a set of open-source software that can be used to create websites and web applications. **LAMP** is an acronym, and these **stacks** typically consist of the Linux operating system, the Apache HTTP Server, the MySQL relational database management system, and the PHP programming language.

We now need to write ***apt install apache2.*** We also need to apply a pseudo if we are the second user. In the beginning, we don;t need to use our root id, rather we should go for the pseudo. Pseudo user simply means, we will create a new user and provide it all the power in terms of benefits.

But if you are already a root user, there is no need of making yourself a pseudo user.

Now we need to install Apache server. The Apache HTTP Server, colloquially called Apache, is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

We need to allow the incoming traffic and we will get the output as follows-



```
Enabling module reqtimeout.
Enabling conf charset.
Enabling conf localized-error-pages.
Enabling conf other-vhosts-access-log.
Enabling conf security.
Enabling conf serve-cgi-bin.
Enabling site 000-default.
Created symlink /etc/systemd/system/multi-user.target.wants/apache2.service → /lib/systemd/system/apache2.service.
Created symlink /etc/systemd/system/multi-user.target.wants/apache-htcacheclean.service → /lib/systemd/system/apache-h
tcacheclean.service.
Processing triggers for libc-bin (2.27-3ubuntu1) ...
Processing triggers for systemd (237-3ubuntu10.31) ...
Processing triggers for man-db (2.8.3-2ubuntu0.1) ...
Processing triggers for ufw (0.36-0ubuntu0.18.04.1) ...
Processing triggers for ureadahead (0.100.0-21) ...
root@HarrysSite:/# sudo ufw app list
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
root@HarrysSite:/# sudo ufw app info "Apache Full"
Profile: Apache Full
Title: Web Server (HTTP,HTTPS)
Description: Apache v2 is the next generation of the omnipresent Apache web
server.

Ports:
  80,443/tcp
root@HarrysSite:/# sudo ufw allow in "Apache Full"
Rules updated
Rules updated (v6)
root@HarrysSite:/# 
```

And if we now move to our website that we have hosted, it will look as follows-

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
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf` . See their respective man pages for detailed information.
- The binary is called apache2. Due to the use of environment variables, in the default

So I hope you must have understood how to host your static website with the help of Apache server. In the upcoming tutorials, we will also learn how to host multiple websites. Till now, stay with the tutorials.

← Previous Next →

Copyright © 2020-2021 CodeWithHarry.com