**Name of Title:** Learning Nginx

**Video Name:** Adding files to the root directory

**Estimated Length:**

**Author Name:** Michael Jenkins

**Chapter\_Section\_Video:**

**Video Objective:**

At the end of this video the learner will be able to unzip the exercise file with the demo site and place the files into the site’s root directory.

**Speaking Points:**

1. Mention exercise files; vagrant VM
2. Check status (vagrant status)
3. Boot up (vagrant up)
4. Connect to the VM (vagrant ssh)
5. Install unzip on the VM
6. Unzip contents into the root directory
7. Set file permissions
8. Open the demo site in a browser

**Script:**

After getting our server configuration in place, we’re ready to add some actual content to our website.

The exercise files for this lesson include a zip file with HTML, images, and other files we’ll need for the site.

It’s important that we run vagrant up inside the directory with these exercise files so we can take advantage of vagrant and Virtualbox’s synced directories. With synced directories, all the files in this directory will be available once we’re logged into the VM. This will keep us from having to copy files up into the VM.

If you’re not following along with the exercise files, you’ll need to use secure copy or SCP to get your zip file up to the server.

Let’s go ahead and connect to the development server with vagrant ssh and switch to the to the root user.

vagrant ssh

sudo su -

Now I can CD to the /vagrant directory. This is a directory that I was telling you about that gets synchronized between the local workstation and the virtual machine.

And if I run LS, we can see the zip file for the demo site.

ls

Before we can do anything with this zip file, we need to make sure the unzip command is installed on the VM. We can check this by running ‘which unzip’:

which unzip

If this returns a path to unzip for you, then you’re all set. In this case, unzip isn’t installed so we need to install it.

We can do that by running “apt install unzip”

apt install unzip

Now we can confirm that unzip is available:

which unzip

Now let’s make sure our root directory is there.

ls /var/www/wisdompetmed.local

In this case we see the index.html file we created during the server configuration. If you get an error like “No such file or directory”, go ahead and create the directory with the mkdir command.

Now let’s unzip the file. We’ll enter: unzip dash O, the file name, and then dash D followed by the root directory. The dash O tells unzip to overwrite our index.html file and the dash D tells unzip where we want the files to be extracted.

unzip -o 'Wisdom\_Pet\_Medicine\_responsive\_website\_LYNDA\_12773.zip' -d /var/www/wisdompetmed.local

Now let’s CD into the root directory and take a look at the files.

cd /var/www/wisdomdpetmed.local

ls -ltr

OK the content is all there and it’s in the right place. Now we need to update the permissions.

Right now the files and directories are owned by the root user but the permissions are set in such a way that the files could be modified by any user. In other words, the directory and files are world writable making the site vulnerable.

To keep our site safe, we need to fix the permissions by locking them down.

We can do this by changing the files and directories to be writable by the root user and read only to everyone else. We also want the directories to have the execute property since that’s needed for them to function properly as directories.

For the files, we can use the find function to do this for us with the command: Find /var/www/wisdomdpetmed.local, dash type f, dash exec, chmod 644, open parenthesis, close parenthesis, backslash, semicolon.

Find /var/www/wisdomdpetmed.local -type f -exec chmod 644 {} \;

And for the directories: Find dot, dash type f, dash exec, chmod 755, open parenthesis, close parenthesis, backslash, semicolon.

find /var/www/wisdomdpetmed.local -type d -exec chmod 755 {} \;

Now if we look at the files and directories again:

ls -ltr

Now we can see that the permissions are set properly.

Now that the files are in place, let’s take a look at the site:

Open browser

<http://192.168.0.3/>

And there’s our site! It took some maneuvering but finally the site is being served.

This is a great accomplishment, but its not the last step in getting our site up and running on nginx. There’s still much more functionality that we can add.

**Conclusion:**

Type your conclusion statement here.

**Script and Media:**

Break the script up into parts and align it with any media (slides, web, CLI, etc.)

| **Part** | **Script** | **Media** |
| --- | --- | --- |
|  |  |  |

**Exercise Files:**

**Basement:**

unzip /vagrant/Wisdom\_Pet\_Medicine\_responsive\_website\_LYNDA\_12773.zip -d /var/www/wisdomdpetmed.local

cd /var/www/wisdomdpetmed.local

find /var/www/wisdomdpetmed.local -type f -exec chmod 644 {} \; -print

find /var/www/wisdomdpetmed.local -type d -exec chmod 755 {} \; -print

rm -rvf /var/www/wisdomdpetmed.local/\_\_MACOSX/

I’ll start by first making sure the VM is running by using the vagrant status command:

vagrant status

My VM is down so I need to run vagrant up:

vagrant up

and wait for th VM to boot up.

>>> MINIMIZER HERE **Before we get started, let me describe what we have in place for this lesson.**

**The VM that I’ll be working with already has the server configuration and root directory in place for our demo site.**

**If you need to set up the server configuration, you can find a complete conf file in the exercise files for this lesson. The exercise files also contain a zip file with the demo site assets.**

**If you’re not using the exercise files, you can use the default nginx configuration and substitute another site for the demo.**

**OK let’s get started.**

**First, let’s make sure we’re in the directory where the exercise files are extracted. If we run LS, we can see the Vagrantfile, the conf file, and the ZIP file for our demo site.**

**CTRL+L**

**ls**

>> START HERE