**Name of Title:** Learning Nginx

**Video Name:** The nginx server configuration file

**Estimated Length:**

**Author Name:** Michael Jenkins

**Chapter\_Section\_Video:**

**Video Objective:**

At the end of this video the learner will know the various parts of the nginx global configuration file.

**Introductory Statement:**

Type your introductory statement here.

**Speaking Points:**

1. Point\_1
2. Point\_2
3. Point\_3
4. Point\_4
5. Point\_5

**Script:**

Let’s take a closer look at the main configuration file: nginx.conf

view /etc/nginx/nginx.conf.

I should also add that the need to edit nginx.conf is very rare. So as I open the file, I’ll even use the view command to make sure I don’t make any unwanted changes:

In this file we’ll find the settings for nginx listed as simple directives on one line or in blocks on multiple lines. These blocks start with a directive and are followed by an open bracket, more directives, and then a closing bracket.

One example of a simple directive is the user setting on the very first line. This defines the user that the nginx process will use after it gets started by the root user.

Another example of a block directive is the http block. This is a directive that contains other directives. For example, a few lines down we find the Logging Settings for the access and error logs; and a few more lines beneath that we find two include statements:

include /etc/nginx/conf.d/\*.conf;

include /etc/nginx/sites-enabled/\*;

These lines tell the main configuration file to process configuration files in other directories, particularly the etc nginx conf.d and sites-enabled directories.

Now, I have to pause for a minute and explain that there are varying schools of thought on how additional configurations get added to nginx when it comes to these two directories.

You’ll notice that the conf.d directory has a wildcard for files ending in .conf and the sites-enabled directory has a wildcard for all files.

So one method for adding configurations is to place a file ending in .conf in the conf.d directory.

Another method is to place configurations in the /etc/nginx/sites-available directory, and then link to them using a symlink from the sites-enabled directory.

Exit nginx.conf

Cd /etc/nginx/sites-enabled

Ls -ltr

In fact, on Ubuntu servers, this is the way that the default server configuration gets set up when nginx is installed.

For this course, we’ll be using the conf.d directory to store our configurations.

Cd /etc/nginx/conf.d

Using symlinks from sites-enabled is a perfectly viable method; but in my experience, I find using the conf.d directory to be more straightforward and easier to maintain once configurations are in place.

There are other configurations that I didn’t go over but again, the defaults in a standard nginx install are perfectly acceptable when you’re just getting up to speed.

If you want more information on the directives in the nginx.conf file or other directives you can use, you can find the complete listing in the documentation on the Nginx.org website.

**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:**

nginx.conf

**Basement:**

As we take a first look at this and other nginx configuration files, one thing you might notice is they look a lot like a program or script. If you have experience with Apache configuration files, this might be

At this point we have nginx installed and we know the key files and directories that we’ll use along with the web server.

If you’re following along and have a server set up, open the nginx.conf file on your server; or you can find a sample nginx.conf in the exercise files.