**Name of Title:** Learning Nginx

**Video Name:**

**Estimated Length:**

**Author Name:** Michael Jenkins

**Chapter\_Section\_Video:**

**Video Objective:**

At the end of this video the learner will know how to use the nginx command line interfaces.

**Introductory Statement:**

Type your introductory statement here.

**Speaking Points:**

1. Status; enabling
2. Starting
3. Stopping
4. Checking config; printing config
5. Point\_5

**Script:**

Like many other linux services, nginx is controlled by its command line interface, or CLI. Let’s take a look at some of the commands we’ll be using as we get more familiar with nginx.

Keep in mind that you need to either be logged in as the root user or preface the command with sudo to enable root privileges as the command is run.

Since I’ll be running several commands that require root privileges, I’ll become the root user by typing:

sudo su -

Now we can check the status of the nginx service with:

systemctl status nginx

This gives a brief report letting us know the service is running, how long its been running, and some details on how the process was started.

This command uses a pager by default so you have to type Q to end the status report and get back the a prompt.

type q and exit

One way to get around this is to use the --no-pager switch. Like this:

CTRL+L

systemctl status nginx --no-pager

This gives us the complete output and then goes right back to the command prompt.

We can also use systemctl to stop and start nginx. We just need to follow systemctl with the desired command and then nginx. For example, starting looks like:

CTRL+L

systemctl start nginx

And stopping nginx looks like:

systemctl stop nginx

One thing to note about the start and stop commands is that they don’t produce any output. To make sure these command ran successfully, It's good practice to follow them with the “is-active” command.

Since I just ran the stop command, I can run:

systemctl is-active nginx

The “inactive” message let's us know that indeed nginx is not running.

Now let’s check the start command:

systemctl start nginx

systemctl is-active nginx

This produces an “active” message so we know that nginx has started and is actively running.

The last systemctl command we’ll need is the reload command:

CTRL+L

systemctl reload nginx

This command tells nginx to reload all configuration files from disk. Because nginx loads its configuration into memory when its first starts, the reload command is needed if we make any changes to the configuration and want to have them applied.

An alternative to this is stopping and starting nginx, but this would cause the service to be offline from the time we ran the stop command until the time we successfully ran the start command. Reloading, on the other hand, keeps the service running.

Like the start and stop commands, the reload command does not produce any output. If we need to confirm the reload, we can run the systemctl status command again...

CTRL+L

systemctl status nginx --no-pager

And at the end of the status report we can see two lines, one saying the service is “Reloading” and another letting us know the service reloaded successfully.

(pause)

Along with the systemctl commands, we can also use the nginx command directly. To get a listing of the switches, we can run:

nginx -h

This lists the version being run and some of the options we can use.

highlight the -t and -T lines

For us, the most useful switches will be the lowercase and uppercase T switches. These let us test the nginx configuration.

Using the lower case t….

CTRL+L

nginx -t

Checks the configuration without stopping or reloading the service. Using this option, we can make changes to the configuration and then test them before reloading it. An “OK” message means the configuration is good.

Using the capital T…

CTRL+L

nginx -T

Tests the configuration, and then prints to the screen. Using a pager like “less” or “more” with this command lets you browse through the output one page at a time.

So if I run this command again and pipe it to less…

nginx -T | less

I can scroll by page with the spacebar and line by line with the error keys.

Since nginx’s configuration can be stored over many different files, using the -T option can be helpful for seeing the entire configuration in one place.

**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:**

List all of the systemctl commands and the nginx commands

**Basement:**