**Name of Title:** Learning Nginx

**Video Name:** Configuring a virtual host Part 2

**Estimated Length:**

**Author Name:** Michael Jenkins

**Chapter\_Section\_Video:**

**Video Objective:**

At the end of this video the learner will finish the vhost configuration and upload upload files the to server using scp or winscp; move the files into the /var/www/wisdompetmed.local directory.

**Introductory Statement:**

Type your introductory statement here.

**Speaking Points:**

1. Default server
2. Server name
3. Index file

**Script:**

Let’s take a look at the simple server configuration that we already have in place.

cat /etc/nginx/confi.d/wisdompetmed.local.conf

server {

listen 80;

root /var/www/wisdompetmed.local  
}

So far, we’ve configured the port the server will be listening on and the root directory where the files for our site will be stored.

Now let’s edit the file and add a few more directives.

vim /etc/nginx/confi.d/wisdompetmed.local.conf

First, let’s add the ‘default\_server” directive to the listen statement.

listen 80 **default\_server**;

Because nginx can serve multiple sites from the same IP address, we can use the default\_server statement to tell nginx to use this server configuration if no other configurations match the name of the site being requested.

Which leads to the next item we need to add: the name of the server, which is also the name of the site:

server\_name wisdompetmed.local www.wisdompetmed.local;

If you’re familiar with Apache vhost configurations, this is similar to setting up a name-based vhost. By including the server name, we let nginx know which sites this server configuration applies to. Otherwise, if we were serving multiple sites from the same IP address, nginx might serve the wrong content.

For more information on how nginx matches server names when a request is received, take a look at the nginx documentation page “How nginx processes a request”

Now let’s add the next directive: index.

The index directive tells nginx what file to use when it responds to a request.

index index.html

In this case, we’re telling nginx to look for a file named ‘index.html’ which is already the default. We can add other names to this directive and nginx will look for them in the order they are listed.

So if the index file will be index.htm or index.php, we can add those and nginx will use the first one that matches in the root directory:

index index.html index.htm index.php;

save the file and exit

Now let’s test the updated configuration to make sure we didn’t introduce any errors. For that we’ll use `nginx -t again`:

nginx -t

And with no errors, we can go ahead and reload the configuration...

systemctl reload nginx

And test the site...

curl localhost

Great everything’s working as planned. Now we can move on to adding the website files.

**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:**

http://nginx.org/en/docs/http/ngx\_http\_index\_module.html

**Basement:**

server {

listen 80 default\_server;

root /var/www/wisdompetmed.local;

server\_name wisdompetmed.local www.wisdompetmed.local;

index index.html index.htm index.php;

}

* Default server
* Server name
* Index files

sudo su -