

Forum:[Hack Forums](#)

Thread: [Nginx: 403 not found](#)

Not Your Average Orange - October 10, 2017, 6:12 pm

Hello, I'm new to the nginx scene. I had it working before, but then I did this.

I was trying to move my nginx site specific files to directories like this:

var/www/site.com/html

Instead of

var/www/html

So I could host more than one site. I was following [this](#) tutorial. I was running as root so I didn't do any permissions stuff.

Let's pretend my site is example.com

Here's my /etc/nginx/nginx.conf file:

<https://pastebin.com/a7Y5uC4E>

/etc/nginx/sites-available/example.com:

<https://Pastebin.com/R49yHGKV>

Help would be appreciated :)

Also, yes, my path for the root is correct

datguy.dev - October 10, 2017, 8:07 pm

With Nginx, often more than not, the directory index is not properly defined in the config. Especially, when you go changing directories. Though if you use sudo to mv the files the permissions are broken as well.

Not Your Average Orange - October 10, 2017, 8:54 pm

(Oct 10, 2017 08:07 PM)datguy.dev Wrote: [\[CITE\]](#) With Nginx, often more than not, the directory index is not properly defined in the config. Especially, when you go changing directories. Though if you use sudo to mv the files the permissions are broken as well.

What do you mean? They lose superuser permissions? How would I reapply those permissions?

datguy.dev - October 10, 2017, 9:29 pm

(Oct 10, 2017 08:54 PM)Not Your Average Orange Wrote: [\[CITE\]](#) What do you mean? They lose superuser permissions? How would I reapply those permissions?

No. That applies to anything you do under super user. You'll lose permissions of the files if you do so. /etc ...

Whereas you should have a user for web based tasks that are not root. If Nignx is running under the www-data user, then your not going to make any progress until that is addressed.

Permissions; sudo chown www-data:www-data && sudo chmod -R 755 \${files}.

Secondly, for multiple sites you'll need to address the config. Because to locate multiple sites it requires a regex. It's a bit late here. I'd be glad to look into it in the morning.

Not Your Average Orange - October 11, 2017, 5:18 am

(Oct 10, 2017 09:29 PM)datguy.dev Wrote: [\[CITE\]](#) No. That applies to anything you do under super

user. You'll lose permissions of the files if you do so. /etc ...

Whereas you should have a user for web based tasks that are not root. If Nginx is running under the www-data user, then your not going to make any progress until that is addressed.

Permissions; sudo chown www-data:www-data && sudo chmod -R 755 \${files}.

Secondly, for multiple sites you'll need to address the config. Because to locate multiple sites it requires a regex. It's a bit late here. I'd be glad to look into it in the morning.

Ok, thanks. I will create a user for that doesn't have root perms by default and adjust perms.

Maybe I should just reinstall nginx as a regular user?

datguy.dev - October 11, 2017, 9:32 am

(Oct 11, 2017 05:18 AM)Not Your Average Orange Wrote: [\[CITE\]](#) Ok, thanks. I will create a user for that doesn't have root perms by default and adjust perms.

Maybe I should just reinstall nginx as a regular user?

It shouldn't matter that you installed nginx as root. In many cases that is normal. There is a difference in the permissions required to install and run nginx and those needed to serve or modify server files. You can run everything nginx as root. But it is not advisable for security and it is potentially destructive for the system as you have experienced with permissions.

In your nginx configuration, you have "user www-data;" specified. Therefore, if any of the www related files only have root permissions set, than the user www-data or group will not be able to serve or merely read those files. Does that make sense?

You just need to check them with "ls -l" and make sure the www-data user and it's group has access. Afterwards, you can add your normal user account to the www group. Logout and back in for the changes to take affect.

The guide you followed is good. Try following it again, but first check the default and working directories of Nginx for your OS.

Not Your Average Orange - October 11, 2017, 10:13 am

(Oct 11, 2017 09:32 AM)datguy.dev Wrote: [\[CITE\]](#) It shouldn't matter that you installed nginx as root. In many cases that is normal. There is a difference in the permissions required to install and run nginx and those needed to serve or modify server files. You can run everything nginx as root. But it is not advisable for security and it is potentially destructive for the system as you have experienced with permissions.

In your nginx configuration, you have "user www-data;" specified. Therefore, if any of the www related files only have root permissions set, than the user www-data or group will not be able to serve or merely read those files. Does that make sense?

You just need to check them with "ls -l" and make sure the www-data user and it's group has access. Afterwards, you can add your normal user account to the www group. Logout and back in for the changes to take affect.

The guide you followed is good. Try following it again, but first check the default and working directories of Nginx for your OS.

Thanks, I think I understand (more) now.

I can do this later when I get to my laptop. I have to edit perms for config files as well, right?

(Oct 11, 2017 10:13 AM)Not Your Average Orange Wrote: [\[CITE\]](#) Thanks, I think I understand (more) now.

I can do this later when I get to my laptop. I have to edit perms for config files as well, right?
Nginx should fail to start if it cannot access the config files.

By default, the main nginx process runs as root. The children as www-data. Therefore, it would be best to leave them for root. They should at least be readable and writable. (664) Whereas, www-data needs your site's files.

It's possible you don't have www users or groups. In that situation, you just need to create them. Here are the defaults:

PHP Code

```
<?php
```

```
sudo systemctl status nginx
```

? nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2017-10-11 13:46:15 EDT; 5min ago

Main PID: 15838 (nginx)

CGroup: /system.slice/nginx.service

+15838 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;

+15839 nginx: worker process

+15840 nginx: worker process

+15841 nginx: worker process

+15842 nginx: worker process

+15843 nginx: worker process

+15844 nginx: worker process

```
ps -p 15838 o pid,user,group,command
```

PID	USER	GROUP	COMMAND
-----	------	-------	---------

15838	root	root	nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
-------	------	------	--

```
ps aux|grep nginx|grep -v grep
```

root	15838	0.0	0.0	124972	1436	?	Ss	13:46	0:00	nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data	15839	0.0	0.0	125336	3200	?	S	13:46	0:00	nginx: worker process
www-data	15840	0.0	0.0	125336	3200	?	S	13:46	0:00	nginx: worker process
www-data	15841	0.0	0.0	125336	3200	?	S	13:46	0:00	nginx: worker process
www-data	15842	0.0	0.0	125336	3200	?	S	13:46	0:00	nginx: worker process
www-data	15843	0.0	0.0	125336	3200	?	S	13:46	0:00	nginx: worker process
www-data	15844	0.0	0.0	125336	3200	?	S	13:46	0:00	nginx: worker process

```
ls -lR /etc/nginx/
```

/etc/nginx/:

total 56

drwxr-xr-x 2 root root 4096 Jul 12 06:34 conf.d

-rw-r--r-- 1 root root 1077 Feb 11 2017 fastcgi.conf

-rw-r--r-- 1 root root 1007 Feb 11 2017 fastcgi_params

-rw-r--r-- 1 root root 2837 Feb 11 2017 koi-utf

-rw-r--r-- 1 root root 2223 Feb 11 2017 koi-win

-rw-r--r-- 1 root root 3957 Feb 11 2017 mime.types

-rw-r--r-- 1 root root 1462 Feb 11 2017 nginx.conf

```
-rw-r--r-- 1 root root 180 Feb 11 2017 proxy_params
-rw-r--r-- 1 root root 636 Feb 11 2017 scgi_params
drwxr-xr-x 2 root root 4096 Oct 11 13:46 sites-available
drwxr-xr-x 2 root root 4096 Oct 11 13:46 sites-enabled
drwxr-xr-x 2 root root 4096 Oct 11 13:46 snippets
-rw-r--r-- 1 root root 664 Feb 11 2017 uwsgi_params
-rw-r--r-- 1 root root 3071 Feb 11 2017 win-utf
```

/etc/nginx/conf.d:

total 0

/etc/nginx/sites-available:

total 4

```
-rw-r--r-- 1 root root 2074 Feb 11 2017 default
```

/etc/nginx/sites-enabled:

total 0

```
lrwxrwxrwx 1 root root 34 Oct 11 13:46 default -> /etc/nginx/sites-available/default
```

/etc/nginx/snippets:

total 8

```
-rw-r--r-- 1 root root 422 Feb 11 2017 fastcgi-php.conf
-rw-r--r-- 1 root root 217 Feb 11 2017 snakeoil.conf
```

Not Your Average Orange - October 11, 2017, 2:41 pm

(Oct 11, 2017 11:07 AM)datguy.dev Wrote: [\[CITE\]](#) Nginx should fail to start if it cannot access the config files.

By default, the main nginx process runs as root. The children as www-data. Therefore, it would be best to leave them for root. They should at least be readable and writable. (664) Whereas, www-data needs your site's files.

It's possible you don't have www users or groups. In that situation, you just need to create them. Here are the defaults:

PHP Code

```
<?php
```

```
sudo systemctl status nginx
```

? nginx.service - A high performance web server and a reverse proxy server

Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)

Active: active (running) since Wed 2017-10-11 13:46:15 EDT; 5min ago

Main PID: 15838 (nginx)

CGroup: /system.slice/nginx.service

+15838 nginx: master process /usr/sbin/nginx -g daemon on; master_process on

+15839 nginx: worker process

+15840 nginx: worker process

+15841 nginx: worker process

+15842 nginx: worker process

+15843 nginx: worker process

+15844 nginx: worker process

```
ps -p 15838 o pid,user,group,command
```

```
PID USER  GROUP  COMMAND
```

```
15838 root   root    nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
```

```
ps aux|grep nginx|grep -v grep
```

```
root   15838  0.0  0.0 124972 1436 ?        Ss   13:46   0:00 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
www-data 15839  0.0  0.0 125336 3200 ?        S    13:46   0:00 nginx: worker process
www-data 15840  0.0  0.0 125336 3200 ?        S    13:46   0:00 nginx: worker process
www-data 15841  0.0  0.0 125336 3200 ?        S    13:46   0:00 nginx: worker process
www-data 15842  0.0  0.0 125336 3200 ?        S    13:46   0:00 nginx: worker process
www-data 15843  0.0  0.0 125336 3200 ?        S    13:46   0:00 nginx: worker process
www-data 15844  0.0  0.0 125336 3200 ?        S    13:46   0:00 nginx: worker process
```

```
ls -lR /etc/nginx/
```

```
/etc/nginx/:
```

```
total 56
```

```
drwxr-xr-x 2 root root 4096 Jul 12 06:34 conf.d
-rw-r--r-- 1 root root 1077 Feb 11 2017 fastcgi.conf
-rw-r--r-- 1 root root 1007 Feb 11 2017 fastcgi_params
-rw-r--r-- 1 root root 2837 Feb 11 2017 koi-utf
-rw-r--r-- 1 root root 2223 Feb 11 2017 koi-win
-rw-r--r-- 1 root root 3957 Feb 11 2017 mime.types
-rw-r--r-- 1 root root 1462 Feb 11 2017 nginx.conf
-rw-r--r-- 1 root root 180 Feb 11 2017 proxy_params
-rw-r--r-- 1 root root 636 Feb 11 2017 scgi_params
drwxr-xr-x 2 root root 4096 Oct 11 13:46 sites-available
drwxr-xr-x 2 root root 4096 Oct 11 13:46 sites-enabled
drwxr-xr-x 2 root root 4096 Oct 11 13:46 snippets
-rw-r--r-- 1 root root 664 Feb 11 2017 uwsgi_params
-rw-r--r-- 1 root root 3071 Feb 11 2017 win-utf
```

```
/etc/nginx/conf.d:
```

```
total 0
```

```
/etc/nginx/sites-available:
```

```
total 4
```

```
-rw-r--r-- 1 root root 2074 Feb 11 2017 default
```

```
/etc/nginx/sites-enabled:
```

```
total 0
```

```
lrwxrwxrwx 1 root root 34 Oct 11 13:46 default -> /etc/nginx/sites-available/default
```

```
/etc/nginx/snippets:
```

```
total 8
```

```
-rw-r--r-- 1 root root 422 Feb 11 2017 fastcgi-php.conf
-rw-r--r-- 1 root root 217 Feb 11 2017 snakeoil.conf
```

Bad news, I don't believe `/etc/nginx/sites-available/example.com` exists anymore

Here's the output of `systemctl`:

Quote: `? nginx.service - A high performance web server and a reverse proxy server`

Loaded: loaded (`/lib/systemd/system/nginx.service`; enabled; vendor preset: enabled)

Active: active (running) since Tue 2017-10-10 19:31:51 EST; 21h ago

Docs: `man:nginx(8)`

Process: 9959 ExecReload=/usr/sbin/nginx -g daemon on; master_process on; -s reload
(code=exited, status=0/SUCCESS)

Main PID: 1218 (nginx)

Tasks: 2 (limit: 4915)

Memory: 8.5M

CPU: 384ms

CGroup: /system.slice/nginx.service

+1218 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;

+9961 nginx: worker process

Warning: Journal has been rotated since unit was started. Log output is incomplete or unavailable.
I have modified the permissions to the www files as well.

datguy.dev - October 11, 2017, 11:07 pm

Any luck? Not sure where you stand.

In my opinion, nginx is great for a lightweight web server. I've seen it used for some of the best websites on the internet. Although, I have found it to be not as easily to configure as compared to other options when it comes to complex setups such as hosting multiple sites. Apache is overall good. Look into it and see if you have different results. Alternatively, you could also do something as simple a python flask web server and run two instances.

Not Your Average Orange - October 12, 2017, 5:23 am

(Oct 11, 2017 11:07 PM)datguy.dev Wrote: [\[CITE\]](#) Any luck? Not sure where you stand.

In my opinion, nginx is great for a lightweight web server. I've seen it used for some of the best websites on the internet. Although, I have found it to be not as easily to configure as compared to other options when it comes to complex setups such as hosting multiple sites. Apache is overall good. Look into it and see if you have different results. Alternatively, you could also do something as simple a python flask web server and run two instances.

Well, my files are all in the www-data group, but I'm still getting a 404. I think I'll just go Flask & Apache. [This](#) tutorial look good?

SuckySucky - October 12, 2017, 9:18 am

just a hunch I have... comment out all PHP include parts in sites-available, restart nginx and see what happens

Not Your Average Orange - October 12, 2017, 12:39 pm

(Oct 12, 2017 09:18 AM)SuckySucky Wrote: [\[CITE\]](#) just a hunch I have... comment out all PHP include parts in sites-available, restart nginx and see what happens

Thanks for the suggestion, but I go the same thing. I think I'm just going to uninstall NGINX and go with Apache & Flask

datguy.dev - October 12, 2017, 1:04 pm

(Oct 12, 2017 05:23 AM)Not Your Average Orange Wrote: [\[CITE\]](#) Well, my files are all in the www-data group, but I'm still getting a 404. I think I'll just go Flask & Apache. [This](#) tutorial look good? You went from 403 to 404. 403 Forbidden: meaning something like the requests for resources are not permitted. (Your config's root site directory does not permit access to / or /index.html) Whereas, 404 Not Found, means the requested resource does not exists. (/does_not_exist.html) Their different errors with different causes.

I've done apache and flask. But not apache+flask. I'm not really sure of the benefit of doing that, but I'd like to try it personally. I'd just use Apache... The flask app I was talking about is a few lines of

code for the app, then you allow firewall traffic through to the port you've specified. Not all that configuring and enabling modules.

Not to suggest too much too soon, but you really should consider a plan for how you would migrate your site. It really sucks once you have a complicated and working site, then having to move it to a new machine or host, where your system packages and configs will most likely be broken. Instead, you could develop everything in Docker right now which will produce a "container." (They have basic pre-made containers btw.) Then to migrate, you just ship the container over and that is really it. You can kind of do this with virtualenv, but it is a bit flaky when it comes to directories.

Not Your Average Orange - October 12, 2017, 2:06 pm

(Oct 12, 2017 01:04 PM)datguy.dev Wrote: [\[CITE\]](#) You went from 403 to 404. 403 Forbidden: meaning something like the requests for resources are not permitted. (Your config's root site directory does not permit access to / or /index.html) Whereas, 404 Not Found, means the requested resource does not exists. (/does_not_exist.html) Their different errors with different causes.

I've done apache and flask. But not apache+flask. I'm not really sure of the benefit of doing that, but I'd like to try it personally. I'd just use Apache... The flask app I was talking about is a few lines of code for the app, then you allow firewall traffic through to the port you've specified. Not all that configuring and enabling modules.

Not to suggest too much too soon, but you really should consider a plan for how you would migrate your site. It really sucks once you have a complicated and working site, then having to move it to a new machine or host, where your system packages and configs will most likely be broken. Instead, you could develop everything in Docker right now which will produce a "container." (They have basic pre-made containers btw.) Then to migrate, you just ship the container over and that is really it. You can kind of do this with virtualenv, but it is a bit flaky when it comes to directories.

Sorry, I originally got a 404.

I kinda wanted to try out flask though. I'll check out docker as well.

What do you mean by migrate? As in nameservers and stuff or local files on the vps?

Also, I've heard of docker before. What would be the benefit of building in docker if I just have a basic static html site?

datguy.dev - October 12, 2017, 2:45 pm

(Oct 12, 2017 02:06 PM)Not Your Average Orange Wrote: [\[CITE\]](#) Sorry, I originally got a 404.

I kinda wanted to try out flask though. I'll check out docker as well.

What do you mean by migrate? As in nameservers and stuff or local files on the vps?

Also, I've heard of docker before. What would be the benefit of building in docker if I just have a basic static html site?

Sorry, I forgot that the term "migration" applies to local files and hosts. I was talking about local files. Can't do much about things like nameservers...

The reasons you'd use Docker is much like why you'd clone a virtual machine image. It saves time and you can perfectly replicate the environment for your application. It also provides means for easy backup in a situation where you can rollback the site to an earlier configuration.

In an simple analogy, would you rather reinstall Windows from square one or use a Windows image you created that includes all your programs and preference. System configurations and updates. What if you had to reinstall Windows 1000 times? Things like moving from one VPS to another is common. I think one method is overall smarter than the other.

Not Your Average Orange - October 12, 2017, 2:52 pm

(Oct 12, 2017 02:45 PM)datguy.dev Wrote: [\[CITE\]](#) Sorry, I forgot that the term "migration" applies to local files and hosts. I was talking about local files. Can't do much about things like nameservers...

The reasons you'd use Docker is much like why you'd clone a virtual machine image. It saves time and you can perfectly replicate the environment for your application. It also provides means for easy backup in a situation where you can rollback the site to an earlier configuration.

In an simple analogy, would you rather reinstall Windows from square one or use a Windows image you created that includes all your programs and preference. System configurations and updates. What if you had to reinstall Windows 1000 times? Things like moving from one VPS to another is common. I think one method is overall smarter than the other.

Gotcha.

So the docker image would come with all system files & the application configured to my needs, assuming their is a repo like that?

Were your referencing to me 'docking' with NGINX, Apache or Flask? I'd like to explore Flask, but I'm fine with NGINX/Apache as well. Even better, both ?

I've also got some scripts, although I guess I can just transport those.

datguy.dev - October 12, 2017, 3:53 pm

(Oct 12, 2017 02:52 PM)Not Your Average Orange Wrote: [\[CITE\]](#) Gotcha.

So the docker image would come with all system files & the application configured to my needs, assuming their is a repo like that?

Were your referencing to me 'docking' with NGINX, Apache or Flask? I'd like to explore Flask, but I'm fine with NGINX/Apache as well. Even better, both ?

I've also got some scripts, although I guess I can just transport those.

You install docker and can pull images from the application or download them manually.

The point of the pre-made images is to simplify and streamline things. For a large docker setup, you'd want to separate services and or reserve resources as much as possible.

Whereas you most likely want to go with the Ubuntu base image. You can customize it to your hearts content then commit changes. From that image, you'll have nginx running right along with Python, or whatever you choose to install.

Remember that problem you had with permissions? Imagine if you had created a container right before you made those server configurations. You got to a point where you needed to reinstall. You could have just cloned the base container and tried again knowing the container is proper. Whereas if you uninstall, your not quite sure if all the configs are gone and if the ports are not bound to an address etc. Docker is one of those things that are hands-on now, so it doesn't have to be later.

Mr. Miyagi - October 12, 2017, 4:02 pm

Can you post your configuration for /etc/nginx/conf.d/default.conf, I always found it config multiple sites there.

If you need any help, shoot me a PM and I can help you get it setup.

Not Your Average Orange - October 12, 2017, 4:22 pm

(Oct 12, 2017 03:53 PM)datguy.dev Wrote: [\[CITE\]](#) You install docker and can pull images from the application or download them manually.

The point of the pre-made images is to simplify and streamline things. For a large docker setup, you'd want to separate services and or reserve resources as much as possible.

Whereas you most likely want to go with the Ubuntu base image. You can customize it to your hearts content then commit changes. From that image, you'll have nginx running right along with Python, or whatever you choose to install.

Remember that problem you had with permissions? Imagine if you had created a container right before you made those server configurations. You got to a point where you needed to reinstall. You could have just cloned the base container and tried again knowing the container is proper. Whereas if you uninstall, your not quite sure if all the configs are gone and if the ports are not bound to an address etc. Docker is one of those things that are hands-on now, so it doesn't have to be later. Thanks. So docker packages can be installed within the server, or do I have to download an ISO and reinstall?

(Oct 12, 2017 04:02 PM)Mr. Miyagi Wrote: [\[CITE\]](#) Can you post your configuration for /etc/nginx/conf.d/default.conf, I always found it config multiple sites there. If you need any help, shoot me a PM and I can help you get it setup. Ok. On mobile right now though.

datguy.dev - October 12, 2017, 4:39 pm

(Oct 12, 2017 04:22 PM)Not Your Average Orange Wrote: [\[CITE\]](#) Thanks. So docker packages can be installed within the server, or do I have to download an ISO and reinstall? I don't follow that terminology. ? You install Docker, run the image, and install packages.

1. docker search ubuntu
2. docker pull ubuntu
3. docker run -it ubuntu
4. at the prompt, apt-get update && apt-get install nginx python etc
5. get nginx working
6. type exit
7. [commit your changes](#)

\$ docker ps

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
PORTS	NAMES			
c3f279d17e0a	ubuntu:12.04	/bin/bash	7 days ago	Up 25 hours
desperate_dubinsky				
197387f1b436	ubuntu:12.04	/bin/bash	7 days ago	Up 25 hours
focused_hamilton				

\$ docker commit c3f279d17e0a svendowideit/testimage:version3

f5283438590d

\$ docker images

REPOSITORY	TAG	ID	CREATED	SIZE
svendowideit/testimage	version3	f5283438590d	16 seconds ago	335.7 MB

lwristis - October 12, 2017, 4:41 pm

datguy.dev sure is a HIGH quality member.

LEGION - October 13, 2017, 3:59 pm

If you're still having the issue make sure your symbolic links are set correctly.

You need to use full paths. That was my mistake and it took me ages to find.
I've successfully got 3 domains running on the same IP/Server now though.

(Oct 12, 2017 04:39 PM)datguy.dev Wrote: [\[CITE\]](#) 5. *get nginx working*

datguy.dev - October 13, 2017, 6:46 pm

(Oct 13, 2017 03:59 PM)Stirling Moss Wrote: [\[CITE\]](#) [img]h ttps://i.imgflip.com/1xkzdp.jpg[/img]
I previously stated nginx wasn't simple to configure. The OP asked about how to use nginx with Docker. There is no point in adding basic nginx steps as the process is the same. The OP had already tried them. To solve this would probably require remote assistance.

Targus - October 14, 2017, 7:44 pm

Is the issue fixed, OP? Or are you looking into an alternative?

P.S. From what I've noticed in your original post, if you want to segregate your website folders like var/www/site.com/html or var/www/site2.com/html, you might want to look into [nginx's server blocks feature](#). This is similar to how Apache's VirtualHost function works. Would be nice if you can get the hang of how this works because the server block feature allows you to host multiple websites on the same server.

The base/template is already there in your default config under sites-available:

Code # Virtual Host configuration for example.com

```
#  
# You can move that to a different file under sites-available/ and symlink that  
# to sites-enabled/ to enable it.  
#  
#server {  
#    listen 80;  
#    listen [::]:80;  
#  
#    server_name example.com;  
#  
#    root /var/www/example.com;  
#    index index.html;  
#  
#    location / {  
#        try_files $uri $uri/ =404;  
#    }  
#}
```

Not Your Average Orange - October 14, 2017, 9:57 pm

Hey everyone. Sorry for the delay. I went with datguy's suggestion and used apache. Everything is working now.