

[Subscribe](#)
[Share](#)
[Contents](#)


# How To Move an Nginx Web Root to a New Location on Ubuntu 18.04

  
2

Posted July 12, 2018

© 6.2k

NGINX

BLOCK STORAGE

STORAGE

UBUNTU

UBUNTU 18.04

By: Melissa Anderson

By: Kathleen Juell

Not using **Ubuntu 18.04**? Choose a different version:

Ubuntu 16.04



Automated: Docker

[request](#)

Automated: Bash

[request](#)

## Introduction

SCROLL TO TOP

On Ubuntu, the [Nginx web server](#) stores its documents in `/var/www/html`, which is typically located on the root filesystem with rest of the operating system. Sometimes, though, it's helpful to move the document root to another location, such as a separate mounted filesystem. For example, if you serve multiple websites from the same Nginx instance, putting each site's document root on its own volume allows you to scale in response to the needs of a specific site or client.

In this guide, you will move an Nginx document root to a new location.

## Prerequisites

To complete this guide, you will need:

- An Ubuntu 18.04 server and a non-root user with sudo privileges. You can learn more about how to set up a user with these privileges in our [Initial Server Setup with Ubuntu 18.04](#) guide.
- Nginx installed, following [How To Install Nginx on Ubuntu 18.04](#).
- A TLS/SSL certificate configured for your server. You have three options:
  - You can get a free certificate from [Let's Encrypt](#) by following [How to Secure Nginx with Let's Encrypt on Ubuntu 18.04](#).
  - You can also generate and configure a self-signed certificate by following [How to Create a Self-signed SSL Certificate for Nginx in Ubuntu 18.04](#).
  - You can [buy one from another provider](#) and configure Nginx to use it by following Steps 2 through 6 of [How to Create a Self-signed SSL Certificate for Nginx in Ubuntu 18.04](#).

We will use the domain name **example.com** in this tutorial, but you should substitute this with your own domain name.

- A new location for your document root. In this tutorial, we will use the `/mnt/volume-nyc3-01` directory for our new location. If you are using Block Storage on DigitalOcean, [this guide](#) will show you how to create and attach your volume. Your new document root location is configurable based on your needs, however. If you are moving your document root to a different storage device, you will want to select a location under the device's mount point.

## Step 1 — Copying Files to the New Location

On a fresh installation of Nginx, the document root is located at `/var/www/html`. By following the prerequisite guides, however, you created a new document root, `/var/www/example.com/html`. You may have additional document roots as well. In this step, we will establish the location of our document roots and copy the relevant files to their new location.

You can search for the location of your document roots using `grep`. Let's search in the `/etc/nginx/sites-enabled` directory to limit our focus to active sites. The `-R` flag ensures that `grep` will print both the line with the `root` directive and the full filename in its output:

```
$ grep -R "root" /etc/nginx/sites-enabled
```

If you followed the prerequisite tutorials on a fresh server, the result will look like this:

#### Output

```
/etc/nginx/sites-enabled/example.com:      root /var/www/example.com/html;
/etc/nginx/sites-enabled/default:         root /var/www/html;
/etc/nginx/sites-enabled/default:         # deny access to .htaccess files, if Apache's document
/etc/nginx/sites-enabled/default:#       root /var/www/example.com;
```

If you have pre-existing setups, your results may differ from what's shown here. In either case, you can use the feedback from `grep` to make sure you're moving the desired files and updating the appropriate configuration files.

Now that you've confirmed the location of your document root, you can copy the files to their new location with `rsync`. Using the `-a` flag preserves the permissions and other directory properties, while `-v` provides verbose output so you can follow the progress of the sync:

**Note:** Be sure there is no trailing slash on the directory, which may be added if you use tab completion. When there's a trailing slash, `rsync` will dump the contents of the directory into the mount point instead of transferring it into a containing `html` directory.

```
$ sudo rsync -av /var/www/example.com/html /mnt/volume-nyc3-01
```

You will see output like the following:

#### Output

```
sending incremental file list
created directory /mnt/volume-nyc3-01
html/
html/index.html

sent 318 bytes  received 39 bytes  714.00 bytes/sec
total size is 176  speedup is 0.49
```

With our files in place, let's move on to modifying our Nginx configuration to reflect these changes.

## Step 2 — Updating the Configuration Files

Nginx makes use of both global and site-specific configuration files. For background about the hierarchy of configuration files, take a look at "How To Configure The Nginx Web Server On a Virtual Private Server". You will modify the `server` block file for our `example.com` project: `/etc/nginx/sites-enabled`

SCROLL TO TOP

**Note:** Remember that in your case `example.com` will be `your_domain_name`, and that you will be modifying the server block files that were outputted when you ran the `grep` command in Step 1.

Start by opening `/etc/nginx/sites-enabled/example.com` in an editor:

```
$ sudo nano /etc/nginx/sites-enabled/example.com
```

Find the line that begins with `root` and update it with the new root location. In our case this will be `/mnt/volume-nyc3-01/html`:

```
                                /etc/nginx/sites-enabled/example.com

server {

    root /mnt/volume-nyc3-01/html;
    index index.html index.htm index.nginx-debian.html;
    . . .
}
. . .
```

Keep an eye out for any other places that you see the original document root path outputted by `grep` in Step 1, including in aliases or rewrites. You will also want to update these to reflect the new document root location.

When you've made all of the necessary changes, save and close the file.

## Step 3 — Restarting Nginx

Once you've finished making the configuration changes, you can restart Nginx and test the results.

First, make sure the syntax is correct:

```
$ sudo nginx -t
```

If everything is in order, it should return:

### Output

```
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

If the test fails, track down and fix the problems.

Once the test passes, restart Nginx:

SCROLL TO TOP

```
$ sudo systemctl restart nginx
```

When the server has restarted, visit your affected sites and ensure they're working as expected. Once you're comfortable that everything is in order, don't forget to remove the original copy of the data:

```
$ sudo rm -Rf /var/www/example.com/html
```

You have now successfully moved your Nginx document root to a new location.

## Conclusion

In this tutorial, we covered how to change the Nginx document root to a new location. This can help you with basic web server administration, like effectively managing multiple sites on a single server. It also allows you to take advantage of alternative storage devices such as network block storage, which can be helpful in scaling a web site as its needs change.

If you're managing a busy or growing web site, you might be interested in learning [how to set up Nginx with HTTP/2](#) to take advantage of its high transfer speed for content. You can also learn more about improving the production experience in this [comparison of five ways to improve your production web application server setup](#).

By: Melissa Anderson

By: Kathleen Juell

♡ Upvote (2)

📄 Subscribe

🔗 Share



We just made it easier for you to deploy faster.

[TRY FREE](#)

### Related Tutorials

[How To Upgrade Nginx In-Place Without Dropping Client Connections](#)

SCROLL TO TOP

[How To Target Your Users with Nginx Analytics and A/B Testing](#)

[How To Deploy a PHP Application with Kubernetes on Ubuntu 16.04](#)

[How To Ensure Code Quality with SonarQube on Ubuntu 18.04](#)

[How To Set Up a Private Docker Registry on Ubuntu 18.04](#)

## 0 Comments

Leave a comment...

[Log In to Comment](#)



This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.



Copyright © 2019 DigitalOcean™ Inc.

[Community](#) [Tutorials](#) [Questions](#) [Projects](#) [Tags](#) [Newsletter](#) [RSS](#) 

[SCROLL TO TOP](#)

[Distros & One-Click Apps](#) [Terms, Privacy, & Copyright](#) [Security](#) [Report a Bug](#) [Write for DOnations](#) [Shop](#)

