

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



Posted July 11, 2018 8.2k [APACHE](#) [STORAGE](#) [BLOCK STORAGE](#) [UBUNTU](#) [UBUNTU 18.04](#)

By: [Melissa Anderson](#) By: [Kathleen Juell](#)

Introduction

On Ubuntu, the Apache 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 Apache 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 Apache 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.
- Apache installed, following How To Install the Apache Web Server on Ubuntu 18.04.
- SSL configured for your domain following How To Secure Apache with Let's Encrypt on 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 Apache, 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 also have additional document roots in corresponding `VirtualHost` directives. 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/apache2/sites-enabled` directory to limit our focus to active sites. The `-R` flag ensures that `grep` will print both the `DocumentRoot` and the full filename in its output:

```
$ grep -R "DocumentRoot" /etc/apache2/sites-enabled
```

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

Output

```
/etc/apache2/sites-enabled/example.com-le-ssl.conf: DocumentRoot /var/www/example.com/html
/etc/apache2/sites-enabled/example.com.conf:      DocumentRoot /var/www/example.com/html
```

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
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 Apache configuration to reflect the.

[SCROLL TO TOP](#)

Step 2 — Updating the Configuration Files

Apache 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 Apache Web Server on an Ubuntu or Debian VPS](#). We will modify the virtual host files for our `example.com` project: `/etc/apache2/sites-enabled/example.com.conf` and `/etc/apache2/sites-enabled/example.com-le-ssl.conf`, which was created when we configured SSL certificates for `example.com`.

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

Start by opening `/etc/apache2/sites-enabled/example.com.conf`:

```
$ sudo nano /etc/apache2/sites-enabled/example.com.conf
```

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

```
/etc/apache2/sites-enabled/example.com.conf

<VirtualHost *:80>
    ServerAdmin sammy@example.comn
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /mnt/volume-nyc3-01/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    RewriteEngine on
    RewriteCond %{SERVER_NAME} =www.example.com [OR]
    RewriteCond %{SERVER_NAME} =example.com
    RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI} [END,NE,R=permanent]
</VirtualHost>
```

Let's also add directives to ensure that the server will follow the symbolic links in the directory:

```
/etc/apache2/sites-enabled/example.com.conf

. . .
<Directory /mnt/volume-nyc3-01/html>
    Options FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>
```

Keep an eye out for the `DocumentRoot` that `grep` outputted in Step 1, including in aliases. You will also want to update these to reflect the new document root location.

After saving these changes, let's turn our attention to the SSL configuration. Open `/etc/apache2/sites-enabled/example.com-le-ssl.conf`:

```
$ sudo nano /etc/apache2/sites-enabled/example.com-le-ssl.conf
```

Modify the `DocumentRoot` to reflect the new location, `/mnt/volume-nyc3-01/html`:

```
                                /etc/apache2/sites-enabled/example.com-le-ssl.conf

<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerAdmin sammy@example.com
    ServerName example.com
    ServerAlias www.example.com
    DocumentRoot /mnt/volume-nyc3-01/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
    . . .
</VirtualHost>
</IfModule>
```

You have now made the necessary configuration changes to reflect the new location of your document root.

Step 3 — Restarting Apache

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

First, make sure the syntax is right with `configtest`:

```
$ sudo apachectl configtest
```

On a fresh installation you will get feedback that looks like this:

Output

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.0.1:80
Syntax OK
```

If you want to suppress the top line, just add a `ServerName` directive to your main (global) Apache configuration file at `/etc/apache2/apache2.conf`. The `ServerName` can be your server's domain or IP address. This is just a message, however, and doesn't affect the functionality of your site. As long as the output contains `Syntax OK`, you are ready to continue.

Use the following command to restart Apache:

SCROLL TO TOP

```
$ sudo systemctl reload apache2
```

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

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

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

Conclusion

In this tutorial, we covered how to change the Apache document root to a new location. This can help you with basic web server administration, like effectively hosting 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 load test your web server to identify performance bottlenecks before you encounter them in production](#). 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 (1)

✚ Subscribe

🔗 Share



We just made it easier for you to deploy faster.

[TRY FREE](#)

Related Tutorials

[How To Migrate your Apache Configuration from 2.2 to 2.4 Syntax.](#)

[SCROLL TO TOP](#)

2 Comments

Leave a comment...

Log In to Comment

^ [haalmaguer88](#) *July 30, 2018*

0 also edit /etc/apache2/apache2.conf to add or modify <Directory>.

^ [shuttlesx35](#) *October 30, 2018*

0 After doing this... nothing working anymore.

Forbidden

You don't have permission to access /index.php on this server.
Apache/2.4.29 (Ubuntu) Server at localhost Port 80





Copyright © 2019 DigitalOcean™ Inc.

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

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