How To Move an Apache Web Root to a New Location on Ubuntu 16.04

PostedJuly 27, 2016 78.5k APACHE STORAGE BLOCK STORAGE UBUNTU UBUNTU 16.04

Introduction

On Ubuntu, by default, the Apache2 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 one'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, we'll show you how to move an Apache document root to a new location.

Prerequisites

To complete this guide, you will need:

- An Ubuntu 16.04 server with a non-root user with sudo privileges. You can learn more about how to set up a user with these privileges in our <u>Initial Server Setup with Ubuntu 16.04</u> guide.
- An Apache2 web server: If you haven't already set one up, the Apache section of the in-depth article, <u>How To Install Linux, Apache, MySQL, PHP</u> (LAMP) stack on Ubuntu 16.04, can guide you.
- A new location for your document root: The new document root location is completely configurable based on your needs. 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.

In this example, we will use the /mnt/volume-nyc1-01 directory. If you are using Block Storage on DigitalOcean, this guide will show you how to mount your drive before continuing with this tutorial.

Step 1 - Copying files to the new location

On a fresh installation of Apache, the document root is located at /var/www/html. If you're working with an existing server, however, you may have a significantly different setup including multiple document roots in corresponding VirtualHost directives. You can search for the location of additional document roots using grep. We'll 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 filename in its output:

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

The result will look something like the output below, although the names and number of results are likely to be different on an existing installation:

Output

sites-enabled/000-default.conf DocumentRoot /var/www/html

Use the feedback from grep to make sure you're moving the files that you want to move and updating their appropriate configuration files.

Now that we've confirmed the location of our document root, we'll 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.

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/html /mnt/volume-nyc1-01

Now we're ready to update the configuration.

Step 2 — Updating the configuration files

Apache2 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.

If you're working with an existing installation, you should modify the virtual host files you found earlier with the grep command. For our example, we're going to look at the two Virtual Host files that ship with Apache by default, 000-default.conf and default-ssl.conf.

We'll start by editing the 000-default.conffile:

sudo nano /etc/apache2/sites-enabled/000-default.conf

Next we'll find the line that begins with DocumentRoot and update it with the new location.

Note: You should look for other places the original path showed up, and change those to the new location as well. With a default installation, there's the DocumentRoot and a **Directory** block you'll need to change. On an existing installation, you may find things like aliases and rewrites that need updating, too. Wherever you see the original document root's path in the output of **grep**, you'll need to investigate.

/etc/apache2/sites-enabled/000-default.conf

<VirtualHost *:80>

ServerAdmin webmaster@localhost

DocumentRoot /mnt/volume-nyc1-01/html

<Directory />

Options FollowSymLinks AllowOverride None

</Directory>

<Directory /mnt/volume-nyc1-01/html/>

Options Indexes FollowSymLinks MultiViews AllowOverride None Require all granted

</Directory>

After we save these changes, we'll turn our attention to the SSL configuration. On a fresh installation, SSL won't be configured yet, but you'll probably want to update the ssl-default.conf to avoid some troubleshooting later if you don't remember that you need to make the change.

sudo nano /etc/apache2/sites-available/ssl-default.conf /etc/apache2/sites-available/ssl-default.conf

<IfModule mod_ssl.c>

<VirtualHost _default_:443> ServerAdmin webmaster@localhost DocumentRoot /mnt/volume-nyc1-01

. . .

Note: If SSL is not enabled, then the ssl-default.conf file is located only in /etc/apache2/sites-available. If you enable SSL with a2ensite, a symlink is created from the file in sites-availableto /etc/apache2/sites-enabled. In that case, the file can be edited from either directory.

Step 3 — Restarting Apache

Once you've finished the configuration changes, you can make sure the syntax is right with configtest:

sudo apachectl configtest

You will get feedback from apachectl configtest with a fresh install:

Output

AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message Syntax OK

As long as you get Syntax OK, restart the web server. Otherwise, track down and fix the problems it reported.

Use the following command to restart Apache:

sudo systemctl reload apache2

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

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, an important step 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 <u>five ways to</u> improve your production web application server setup. To set the alias name of document path

example : Alias /content/ /path/to/content/