Linux Server

- User and Group permissions, with chmod, and Apache

User and Group permissions, with chmod, and Apache

I've been scouring the internet for good information on setting up user and group permissions for Apache. I'll link some resources on the bottom here, but here's what I found:

Basics

There are three sets of permissions to worry about with any directory/file:

- User - What the owner of the file can do
- Group - What users of the same group can do
- Other - What anyone else can do

Correspondingly, users have a username (unique to each user). Users can also be part of a group - In fact, multiple users can be part of the same group.

> **Note:** The `chmod` command can accept numeric integers, such as 0664, which relate to user permissions. See this to help create these, if you wish

I will cover using chmod. Chmod is used to modify the permissions of a directory or file.

Usage:

```
chmod -flags permissions /path/to/dir/or/file
```

Flags

-R

`chmod -R …` will recursively go through the directory provided and change all file/directory permissions as specified.

Changing Permissions

You can define for whom the permissions you are setting apply with these:

- **u** = user
- **g** = group
- **o** = other

You can add or remove permissions using these:

- **+** will add permissions
- **-** will remove permissions

You can set these permissions:

- **r** = read
- **w** = write
- **x** = execute

Use this knowledge to setup Apache

Assumptions:

- Apache is run as user www-data and group www-data.
- Server web root is /var/www

First

We need to set the owner/group of the web root (and any directories/files therein):

1

1

```
$ sudo chown -R www-data:www-data /var/www
```

Second

We need to setup the proper permissions for users and groups. We do some blanket commands restricting access, and then open access up as much as we need to.

To start, make it so no-one but the current user (www-data) can access the web-root content. We use 'go', meaning apply to 'group' and 'other'. We use '-', which means remove permissions. We use 'rwx' to remove read, write and execute permissions.

1

1

```
$ chmod go-rwx /var/www
```

Next, allow users of the same group (and 'other') to enter the `/var/www` directory. This is *not* done recursively. Once again, we use 'group' and 'other' but we use '+' to allow the execute ('x') permission.

1

1

```
$ chmod go+x /var/www
```

Next, change all directories and files in the web root to the same group (www-data) - just in case there are files in there currently:

1

1

```
$ chgrp -R www-data /var/www
```

Next, let's do another "reset" of sorts - Make it so only the user can access web content:

1

1

```
$ chmod -R go-rwx /var/www
```

And finally, make it so anyone in the same group can ready/write and execute directories/files in the web root.

1

1

```
$ chmod -R g+rx /var/www
```

> I actually give group write permissions as well, for users which need to modify content, such as users used to deploy code. That looks like this:

1

1

```
$ chmod -R g+rwx /var/www
```

> Often going through all of these steps isn't necessary, but this is a useful exercise to see how these commands work!