Dan Benson

CTEC145 Web Server Technology

Alias and Redirects

5/15/16

In this document, I want to explain what Apache Redirects and Aliases are and how one might use them in a live configuration. In Apache we use Redirects to control incoming traffic to bring them to another domain under our control. When we use an Alias directive, it is commonly to connect and share resources over multiple domains on the same server or when you’ve changed original content directories. The two are similar in that they allow more control through vHosts so clients are finding the correct content, but the job of the Redirect is to bring the client to a new URL, where Aliasing can deliver content on a domain in which the content lives in a different file structure or remotely. I will show you some examples of how to setup your vHost directives to properly use a redirect and alias respectively.

Redirects in Apache are a very useful tool for use when we want to direct incoming requests to other URL’s on our server or elsewhere. When we use a redirect, the server is essentially evaluating the incoming request URL and returning with information that forces the client to the new URL set in our vHost directive. Looking at this further, many times a client will omit “www” header information when typing in your URL, when this happens the server recognizes a difference in “www.requestedURL.com” and “requestedURL.com.” When this happens you want to make sure no 404 errors or default content is shown to the client. Also when a company has multiple domains and/or expanded domains for their main site, they want to direct requests to those secondary domains to their primary site. These are real world examples of why someone would want to use HTTP Redirects and shows the importance of them for content delivery.

Often times, content we store on our servers (or remotely) is needed for proper site display. Such as when you host videos or images offsite in the cloud or content management system. The Alias directive helps deliver content to our clients by allowing the requested files to be stored in a directory that does not share the root directory of the domains default file system. Another situation that commonly happens that requires an Alias, is when someone builds a website and hosts the content locally, then moves the content later on and needs a way to link the new directory so Apache knows where to find it. We can also use Aliases to link content stored in one location that is used over multiple domains. As you can see, this is another powerful tool in Apache that allows further control of content delivery.

There are some similarities and differences between the functions of a Redirect and Alias. They can be used together or separately depending on your needs, but when used together they can help define strict paths for content delivery and correct domain name usage. It would be wise for any server administrator to setup adequate and proper Redirect / Alias directives in their httpd config to eliminate user input error, duplicate content and increase security. In many cases redirects, aliases and rewrites are used concurrently to complete the tasks they were designed for, rather than using an Alias in lieu of a hard redirect. This allows total control and helps the user viewing the content and domain name to know where they are and have the proper accessible content delivered to them.

Here are some examples of vHost directives used to setup a redirect and alias configuration.

Simple HTTP Redirect:

* **Redirect / http://www.redirectedURL.com/**
  + This redirects the client to the specified URL.

Other types of Redirects:

* **Redirect permanent /files http://www.redirectedURL.com/files/**
  + This redirects and indicates that the URL to the folder “files” has moved to this the specified URL permanently (301).
* **Redirect temp /files http://www.redirectedURL.com/files/**
  + This redirects and indicates that the URL to the folder “files” has moved to this the specified URL temporarily (302).
* **Redirect seeother /files http://www.redirectedURL.com/files/**
  + This redirects and indicates that the URL to the folder “files” has moved to this the specified URL as replaced.
* **Redirect gone /files http://www.redirectedURL.com/files/**
  + This redirects and indicates that the URL to the folder “files” has been removed.

Simple Alias Directive:

* **Alias / /newcontentfolder/newimagefolder/**
  + This sets an alias to deliver content requested in the root directory to the new directory in the same file system named “newimagefolder”.

More Alias Directive examples:

* **Alias /oldimagefolder /newimagefolder**
  + This sets an alias to deliver content requested in the “oldimagefolder” to the new directory in the same file system named “newimagefolder”.
* **Alias /videos /ftp/user/videos**
  + This sets an alias to deliver content requested in the relative videos directory to the ftp upload folder for the given user’s video folder.

Here is a complete example of how to use a Redirect and Alias together.

<VirtualHost \*:80>

ServerName www.mydomain.com

Alias /inventorySpreadsheets /usr/local/shared/currentInventory

Redirect 301 /inventorySpreadsheets http://www.myotherdomain.com/currentInventory

</VirtualHost>

This vHost rule (omitting rewrite data for class scope purposes) tells a client requesting www.mydomain.com on port 80 to permanently redirect to www.myotherdomain.com and to find its current inventory spreadsheets in the currentInventory folder, while the actual spreadsheets are located elsewhere as indicated.

As you can see, Redirects and Aliases help administrators properly direct requests to the correct content. They are similar in that they tell clients where to find content on a particular domain name and its file structure. They are different in that Redirects are used primarily to forcibly change the agent URI when they are used to catch secondary domain requests or when the original domain has changed, whereas an Alias points to the content location when it differs from its original location or when it is simply not stored in the default folder. This allows broad control over how to setup their file system and how to manage owned domains when they all share the same single resolved IP address.