

The Foolproof Guide to: Apache Virtual Host Configuration

Disclaimer: No warranties are made or implied.

This guide was developed and tested with recent Debian and Ubuntu (Hardy and Lenny) releases.

*License: This Document may be used and copied under the same terms as Apache:
(<http://httpd.apache.org/docs/2.2/license.html>).*

Apache defaults to serving webpages from `/var/www/` and often that is good enough. But sometimes we need to host multiple sites on one server, and that can get a little tricky.

Preparation:

The list below may seem obvious, use it as a checklist.

1. Check Your Hostname

`/etc/hostname` should contain a single line with the Fully Qualified Domain Name of your Server, that is the box, not necessarily any website it may host. The same hostname should be returned by the `hostname` command. If you need to change `/etc/hostname`, you should reboot for the change to take effect.

2. Check your Hosts File.

The first two entries in your hosts file should be something like this:

```
# cat /etc/hosts
127.0.0.1    localhost
192.168.10.62  userv1.brain.demo  userv1
```

If you have a dynamic address and the second line isn't present, don't add it manually.

3. Confirm that you have a static IP Address. You can do this one of two ways:

- On a Debian Type Distribution look for `/etc/network/interfaces`, on Red Hat try `/etc/sysconfig/network-scripts/ifcfg-eth?` where ? is the interface you wish to configure. Make sure that they are set with a Static Address.
- Reserve an IP Address in your DHCP server(s), leaving your web server configured to obtain an address through DHCP. If you do this confirm that your server always receives the assigned address.

4. Create Entries in DNS for each Virtual Host you wish to have, you should use an alias (CNAME) for each virtual host that points to the primary hostname of your server. The advantage of the Alias is that should you ever change the ip address of your host, the virtual hosts will not need to be updated. If you have not set a static address (or reserved), you must use an alias because your host could change its' IP address unexpectedly.

5. Test DNS. Make sure that on your host all of the hostnames resolve correctly, ping and nslookup are good tools for this. Test it on at least one other machine, ideally from someplace else on the network and/or Internet, depending on the nature of the website(s).

Apache Virtual Host Configuration

Install Apache.

Install and test apache as appropriate for your distribution. The default configuration should serve an “it works” page from `/var/www`.

Create a directory for each of your virtual hosts, and put a placeholder `index.html` page in each so you can confirm apache is serving the correct page, something like

```
<html><head><title>schooner.brain.demo</title></head>
<body><h1>The Temporary Page for: schooner.brain.demo</h1></body></html>
```

In the directory `/etc/apache2/` copy `httpd.conf`, `apache2.conf`, and `ports.conf` (for vendor sample files like this I use the extension `.sam`, other people might choose `.backup` or `.original`). **`apache2.conf`** is the main configuration file, I have a working one at the end of the document, it is likely that your default one does not need much changing. On Debian/Ubuntu you can safely comment out the include line for `httpd.conf` (it is available if you are re-using older configurations) but you will also need `ports.conf` file (or to place its' commands in `apache2.conf`).

ports.conf

```
NameVirtualHost *:80
Listen 80
<IfModule mod_ssl.c> # virtual hosts do not currently support ssl
    Listen 443      # you could comment this section out if you're not using ssl.
</IfModule>
```

NameVirtualHost tells apache that you have virtual hosts, ***:80** says all ip addresses port 80. In a more complex configuration you may specify the ip address and alternate ports. If specifying multiple IP address and port combinations you can use multiple NameVirtualHost Statements.

Listen 80 is necessary because by default apache doesn't listen on any ports.

Word of Warning.

There is one very important thing about NameVirtualHost that the Apache manpage forgets to mention. Each NameVirtualHost statement should only appear once in your configuration and it should be in `ports.conf` (or `apache2.conf` if you don't use `ports.conf`).

Many items on the web make it look like it belongs in your Virtual Host file.

In `/etc/apache2` run `grep -R NameVirtualHost *` to make sure you haven't made this mistake.

Some Administrators avoid modifying the vendor provided default files to avoid complications from future upgrades, they create an extra configuration file just to hold their NameVirtualHosts directives, which they include by either adding an explicit include for it or by placing it in `sites-enabled`. If you are one of them, this is equally valid, so long as you obey the Foolproof Advice of placing them all in one specific place.

Virtual Hosts Definitions.

Underneath `/etc/apache2/` there are two directories: `sites-available` and `sites-enabled`. Virtual Hosts definitions files are placed in `sites-available`. It is likely that you already have a default site defined in a file titled `default`. The document below is my default which hasn't been changed much since the install. For now using the default configuration should be fine.

sites-available/default

```
<VirtualHost *:80>
    ServerName userv2.brainbuz.org
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/
    <Directory /var/www/>
        DirectoryIndex index.html /var/www/index.html
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Order allow,deny
        allow from all
    </Directory>
    ScriptAlias /cgi-bin/ /var/www/cgi-bin/
    <Directory "/var/www/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>
    ErrorLog /var/log/apache2/error.log
    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn
    CustomLog /var/log/apache2/access.log combined
    ServerSignature On
</VirtualHost>
```

If you intend to actually host something there, using the default may not be fine. Consider the `AllowOverride None` directive, it means that subordinate configurations will be ignored, without it, the default is to permit subordinate configurations. My advice is to build your own Virtual Host file up, check the values in examples you're using against the documentation to see if what they suggest is appropriate for what you want. To add your own virtual host, open a new file in a text editor, and save it in `sites-available`, though not mandatory, I recommend having the filename match the website name.

A virtual host configuration file (`sites-available/schooner.brain.demo`).

```
# schooner.brain.demo
<VirtualHost *:80 >
    ServerAdmin postmaster@brain.demo
    ServerName schooner.brain.demo
    DocumentRoot /var/www/schooner/
    <Directory /var/www/schooner/ >
        DirectoryIndex index.html /var/www/schooner/index.html
    </Directory>
    ScriptAlias /cgi-bin /var/www/schooner/cgi-bin
    <Directory /var/www/schooner/cgi-bin >
        Options +ExecCGI
        Order allow,deny
```

Apache Virtual Host Configuration

```
    AllowOverride None
    Allow from all
  </Directory>
</VirtualHost>
```

The enclosing tag is `VirtualHost`, in the opening tag `*:80` matches an existing `NameVirtualHosts` directive. Most of the remaining values should be self-explanatory. The Internal Directory Tags allow setting of options for a directory. My example shows a documents directory, the `Directory` Index sets the default document. Placing the `NameVirtualHosts` directive in the same file will sometimes work but can cause grief later on. Multiple copies of the same `NameVirtualHosts` directive and incorrect matching by the `VirtualHost` tag are easy errors to make that will make you not happy.

Enabling Your Virtual Hosts

By default only the default site is going to be enabled for you. To enable a site you simply create a link in `sites-enabled`. To make this even easier there are two scripts `a2ensite` and `a2dissite` which will prompt you with a list of sites that can be enabled or disabled. Then use `'/etc/init.d/apache2 reload'` to activate.

```
Uersv1:/etc/apache2# a2ensite
Your choices are: default default-ssl dhingy.brain.demo schooner.brain.demo
sloop.brain.demo yacht.brain.demo
Which site(s) do you want to enable (wildcards ok)?
schooner.brain.demo
Enabling site schooner.brain.demo.
Run '/etc/init.d/apache2 reload' to activate new configuration!
```

Serving up a script instead of a page

It is common practice today to have an entire site run by a script. Drupal, Joomla, Django and Catalyst all operate in this manner. The following configuration changes my default site to serve a simple cgi script which tells me which page I've arrived at.

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  RewriteEngine On
  RewriteRule ^(/cgi-bin/index\.cgi/\.*)$ - [L]
  RewriteRule ^(.*)$ /cgi-bin/index\.cgi/$1 [PT,L]
  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI
    Order allow,deny
    Allow from all
  </Directory>
</VirtualHost>
```

The key here is the rewrite engine, which is turned off by default. To enable the rewrite engine use the command `a2enmod`. Just like `a2ensite`, `a2enmod` will prompt you with a list of available modules, both also let you use a wildcard or

Apache Virtual Host Configuration

enter an item directly on the command line. To activate the module you will need to restart Apache with `'/etc/init.d/apache2 restart'`.

```
#!/usr/bin/perl
# index.cgi
use CGI ;
use strict ;
my $query = CGI->new ;
my $url = $query->url() ;
print $query->header ;
print $query->start_html( -title=>'The Page You Have Arrived At', -BGCOLOR=>'cyan') ;
print $query->h1("You are at $url") ;
print $query->end_html ;
```

Appendix: sample apache2.conf

```
ServerRoot "/etc/apache2"
LockFile /var/lock/apache2/accept.lock

PidFile ${APACHE_PID_FILE}
Timeout 300
KeepAlive On
MaxKeepAliveRequests 100
KeepAliveTimeout 15

<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers   5
    MaxSpareServers   10
    MaxClients        150
    MaxRequestsPerChild  0
</IfModule>

<IfModule mpm_worker_module>
    StartServers      2
    MaxClients        150
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadsPerChild   25
    MaxRequestsPerChild  0
</IfModule>

# These need to be set in /etc/apache2/envvars
User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

AccessFileName .htaccess

<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
```

Apache Virtual Host Configuration

```
DefaultType text/plain
HostnameLookups Off
ErrorLog /var/log/apache2/error.log
LogLevel warn

# Include module configuration:
Include /etc/apache2/mods-enabled/*.load
Include /etc/apache2/mods-enabled/*.conf

# Include all the user configurations:
# Include /etc/apache2/httpd.conf

# Include ports listing
Include /etc/apache2/ports.conf

#
# The following directives define some format nicknames for use with
# a CustomLog directive (see below).
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Define an access log for VirtualHosts that don't define their own logfile
CustomLog /var/log/apache2/other_vhosts_access.log vhost_combined

# Include generic snippets of statements
Include /etc/apache2/conf.d

# Include the virtual host configurations:
Include /etc/apache2/sites-enabled/
```

Tips and Credits

When looking for answers to Apache questions the two best places to look are the Apache Documentation itself and the Apache Wiki. The Apache Wiki may not have what you're looking for, but unlike a lot of other resources answering Apache questions out on the web, it will generally be correct.

<http://httpd.apache.org/docs/2.2/>

<http://wiki.apache.org/httpd/FrontPage>

This document was written by John Karr: <http://brainbuz.org/techinfo>.

Technical reviewers: Chris J. Fearnley and J.P. Vossen.