Setup a Virtual Host/Website

Contents

Goals

After reading this guide, you should understand how to setup and manage a website on your Linux server system. (Please note that here we are discussing a typical Linux system, with no control panel. However, even if you do have a control panel, most of the configurations are the same, with the exception of file locations.)

Setup a Website in CentOS

Create the Document Root

Before starting, you have to create the document root, which is the directory where your site contents should be stored, and give necessary permissions and ownerships. In this case, we are going to set our document root under "/var/www/html/example_docs".

```
[root@CentOS ~]# mkdir /var/www/html/example_docs/
```

Sample Index File

Let's create a sample index file. An index file is the page that loads by default when you access your website. (You can use the file for testing purposes and delete or modify the file after testing.)

1) Change your current directory to our document root.

```
[root@CentOS ~]# cd /var/www/html/example_docs/
```

2) Now create the index file using vi, vim, nano, or your favourite editing tool. (Note: vim is not installed by default. You can use vi instead.)

```
[root@CentOS example_docs]# vim index.html
```

3) Add the following lines in the file:

```
<html>
<title> Index page </title>
<body bgcolor=blue > </body>
<A> SUCCESS!!!<br>Your Website is working fine. </A>
</html>
```

4) If you are using vi or vim editor, please press "Esc + shift + : + x" to save the file.

5) Now you have successfully created our test index file.

Configuration

Now we are going to create a website named "www.example.com". (Simply replace example.com with your domain name.)

(Note: You do not have to install httpd because httpd service is typically installed with CentOS by default. If httpd is not found, please install the service using yum. The command will be as follows:

yum install httpd)

1) Open the Apache configuration file using any of your file editors, for example using vi or vim. You can find your Apache configuration file (httpd.conf) under

```
[root@CentOS logs]# vim /etc/httpd/conf/httpd.conf
```

/etc/httpd/conf/httpd.conf.

2) Next, you have to specify your document root with the directive "Directory" with necessary permissions. (Please do not worry about the line number; you can place the root anywhere in the configuration file.) Please follow the example below.

```
<Directory /var/www/html/example_docs>
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
<Directory>
```

3) Then, find the directive "DirectoryIndex". You can specify your index file here. (This will

```
DirectoryIndex index.html index.html.var
```

be the default page that loads when you access your domain.)

4) There is two ways that you can configure your domain: IP-based Apache and name-based Apache. Using the IP-based method, you have to use separate IP addresses for each domain. In contrast, using the name-based method, you can share an IP address with a number of domains. Here we are going to configure your domain using the IP-based Apache method.

Please go to the end of the "httpd.conf" file and add the following lines:

```
NameVirtualHost *:80
<VirtualHost *:80>
    DocumentRoot /var/www/html/example_docs
    ServerName www.example.com
    ErrorLog logs/www.example.com-error_log
</VirtualHost>
```

NameVirtualHost – This directive is used to specify the IP that you will use for your domains and also the httpd service port. Here we provide the wildcard "*" instead of a specific IP address, in order to assign any available IP to our domain. (You can assign the same IP to all your domains or you can assign separate IPs for each domain.)

DocumentRoot – Your domains document root directory path.

ServerName – Your server FQDN (Fully Qualified Domain Name).

ErrorLog – The error log path.

5) Now you can restart the httpd service to see if there are any errors.

```
[root@CentOS ~]# /etc/init.d/httpd restart
Stopping httpd:                                          [  OK  ]
Starting httpd:                                          [  OK  ]
```

6) You have now successfully configured your website! (Note that you have to register your domain name first, then point your domain to an IP address and forward the DNS. Reverse entries are must if you want the domain to resolve from the outside.)

How to Check If Your Website is Working

Now that you have successfully configured your website, how do you know if the website is loading successfully within your system? Before registering your domain and setting up DNS entries, please follow the steps below:

1) Open the file "/etc/hosts" and point your IP to a created domain name.

```
# Do not remove the following line, or various programs
# that require network functionality will fail.
127.0.0.1       localhost.localdomain localhost
::1             localhost6.localdomain6 localhost6
10.0.0.239 www.example.com
```

2) Now open and comment on the entries in "/etc/resolv.conf". Put a "#" at the start of each line so that the system will only check the "/etc/hosts" file and will not look for other resolvers in /etc/resolv.conf.

```
#nameserver 8.8.8.8
#nameserver 8.8.4.4
```

3) Now you can type your website address (the name you gave in "/etc/hosts") in the browser to see if the index page is loading. If you do not have a graphical interface, you

can use the command "elinks" or "htmlview". If these components are not installed, you can install them using the yum installation utility. The command will be as follows:

yum install elinks (for installing "elinks")

yum install htmlview" (for installing "htmlview")

4) Now you will receive your default page (index page) and will have successfully setup your website! (Note that "SUCCESS!!! Your Website is working fine." is the contents from the index page and if you are using any other index page it will display the contents from that index page.)
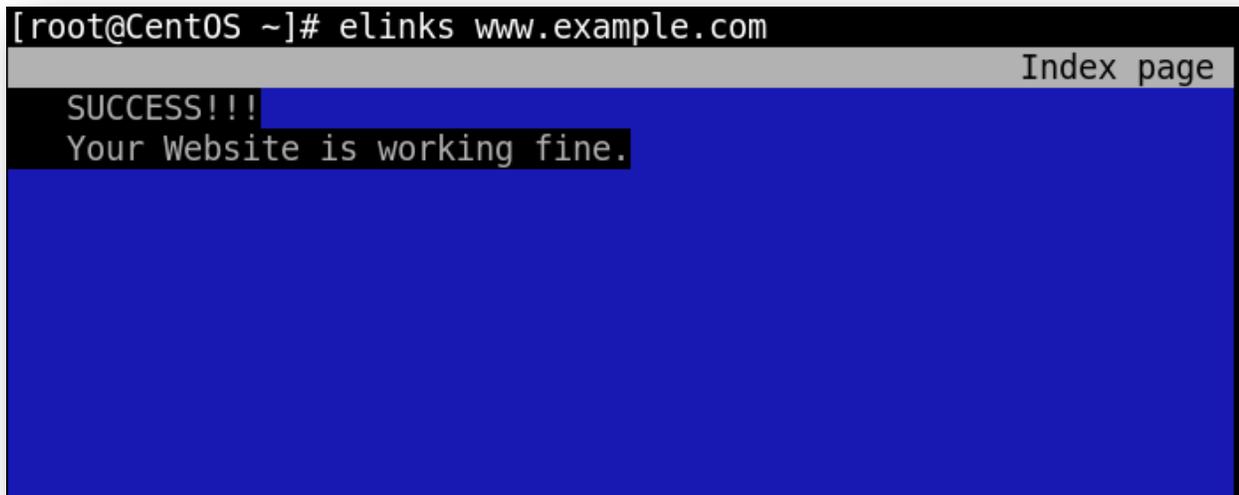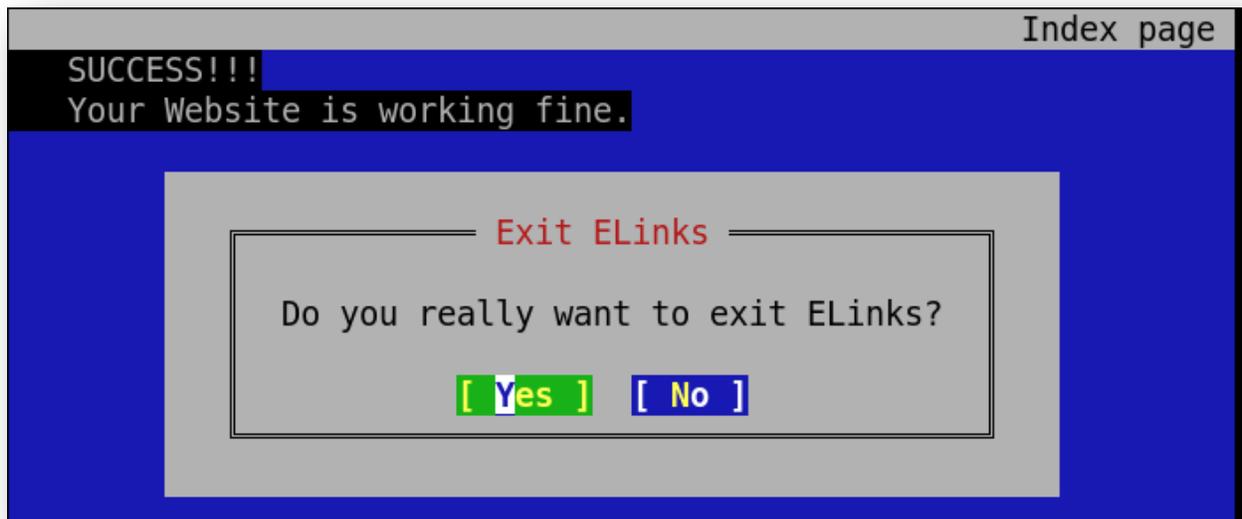
```
[root@CentOS ~]# elinks www.example.com
                                                          Index page
   SUCCESS!!!
   Your Website is working fine.
```

5) Press "q" to exit the window.

6) You have successfully setup your website! (Don't forget to remove the "#" from /etc/resolv.conf.)

Setup a Website in Ubuntu

(Note: In Ubuntu, after installation, a sample index file will be created automatically under the "/var/www" directory. For simplicity, we are going to use "/var/www" as our document root and the index.html under that directory as our test index page.)

```
SUCCESS!!!
Your Website is working fine.
```

```
========= Exit ELinks =========

     Do you really want to exit ELinks?

           [ Yes ]    [ No ]
```

Configuration

Now we are going to create a website named "www.example.com". (Simply replace example.com with your domain name.)

1) Open the configuration file httpd.conf and edit using any of your file editors. For example, you can use vi , vim or nano. You can find this file under "/etc/apache2/httpd.conf".

(Note: This is actually an empty file. The actual Apache configuration file is "apache2.conf", but it is a large configuration file and it may be difficult for you to find its contents and modify them. The easy way to manage this file is to edit and add the configurations in httpd.conf, which are included by default in "apache2.conf".)

```
root@Debian:~# vim /etc/apache2/httpd.conf
```

2) First, tell Apache what the index file name is. You can use the directive "DirectoryIndex" for that. Add the following line in the file:

```
DirectoryIndex index.html
```

3) Then, specify the document root and provide the necessary permissions.

7

```
<Directory "/var/www/">
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
</Directory>
```

4) Now we are going to add the configuration for your virtual host. For that, add the following lines into the file (we give the wildcard "*" with the directive "VirtualHost" so that we can use any IP address available with this address):

```
<VirtualHost *:80>
    DocumentRoot /var/www/
    ServerName www.example.com
    ErrorLog ${APACHE_LOG_DIR}/www.example.com_error.log
</VirtualHost>
```

5) The full file is given below.

6) You have completed the configuration. Now you can restart the apache service for the configurations to take effect and to know if there are any errors.

```
DirectoryIndex index.html

<Directory "/var/www/">
Options Indexes FollowSymLinks
AllowOverride None
Order allow,deny
Allow from all
</Directory>

<VirtualHost *:80>
    DocumentRoot /var/www/
    ServerName www.example.com
    ErrorLog ${APACHE_LOG_DIR}/www.example.com_error.log
</VirtualHost>
```

```
root@Debian:~# /etc/init.d/apache2 restart
Restarting web server: apache2 ... waiting .
```

7) Now you have successfully setup your website on your Ubuntu system.

How to Check If Your Website is Working

```
#nameserver 8.8.8.8
#nameserver 8.8.4.4
```

Now that you have successfully configured your website, how do you know if the website is loading successfully within your system? Before registering your domain name and setting up DNS entries, please follow the steps below:

1) Open the file "/etc/hosts" file and point your IP to a created domain.

```
127.0.0.1        localhost
10.0.0.249 www.example.com

# The following lines are desirable for IPv6 capable hosts
::1      ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

2) Now open and comment on the entries in "/etc/resolv.conf". Put a "#" at the start of each line so that the system will only check the "/etc/hosts" file and will not look for other resolvers in /etc/resolv.conf.

3) Now you can type your website address (the name you gave in "/etc/hosts") in the browser to see if the index page is loading. If you do not have a graphical interface, you can use the commands "elinks" or "htmlview". If these components are not installed, you can install them using the "apt-get" installation utility. The command will be as follows.

apt-get install elinks (for installing "elinks")

apt-get install htmlview" (for installing "htmlview")

4) Now you will get your default page (index page), and you have successfully setup your website!

```
root@Debian:~# elinks www.example.com


                                         http://www.example.com/
                    It works!

 This is the default web page for this server.

 The web server software is running but no content has
 been added, yet.
```

5) Press "q" to exit the window.

6) You have successfully setup your website! (Don't forget to remove the "#" from /etc/resolv.conf.)

```
                                         http://www.example.com/
                    It works!

 This is the default web page for this server.

 The web server software is running but no content has
 been added, yet.
          ┌──────── Exit ELinks ────────┐
          │                             │
          │  Do you really want to exit ELinks? │
          │                             │
          │      [ Yes ]    [ No ]      │
          │                             │
          └─────────────────────────────┘
```

Conclusion

Now you have successfully setup your website. This is a setup with minimal configuration; you can customize or secure your website further according to your needs. If you encountered any issues with this guide, please also note that there is additional information available in our Wiki database at http://myhosting.com/kb. Finally, we encourage you to contact our technical support team by email at vps@myhosting.com, or calling us at 1-866-289-5091 with any questions or concerns.