Web Servers

By,
Subrahmanya Bhat,
Dept. MCA, CCIS
Srinivas University
Mangaluru.

- A web server is a network service that serves content to a client over the web.
- This typically means web pages, and any other documents

- Web servers are also known as HTTP servers
- They use the hypertext transport protocol (HTTP).

- The web server available in Red Hat Enterprise Linux 7 is Apache HTTP Server, version 2.4, httpd

- It is a open source web server developed by the Apache Software Foundation

- To be able to use the **httpd** service, make sure you have the httpd installed.
- You can do so by using the following command
  - ***yum install httpd***

Starting the Service

- To run the **httpd** service, type the following at a shell prompt as root
  - ***systemctl start httpd.service***

- If you want the service to start automatically at boot time, use the following command
  - ***systemctl enable httpd.service***

Stopping the Service

- To stop the running **httpd** service, type the following at a shell prompt as root
  - ***systemctl stop httpd.service***

- To prevent the service from starting automatically at boot time, type
  - ***systemctl disable httpd.service***

Restarting the Service

- To restart the service completely, enter the following command as root
  - ***systemctl restart httpd.service***

- To only reload the configuration, as root, type
  - ***systemctl reload httpd.service***

- To reload the configuration without affecting active requests, enter the following command as root
  - ***apachectl graceful***

- This causes the running **httpd** service to reload its configuration file.
- Any requests currently being processed will continue to use the old configuration.

Verifying the Service Status

- To verify that the **httpd** service is running, type the following at a shell prompt
  - ***systemctl is-active httpd.service***

Editing Configuration Files

- When the **httpd** service is started, by default, it reads the configuration from
  - **/etc/httpd/conf/httpd.conf**
  - &
  - **/etc/httpd/conf.d/**

- To check the configuration for possible errors, type the following at a shell prompt
  - ***apachectl configtest***

Ex

]# **apachectl config test**

Syntax OK

- Being a modular application, the httpd service is distributed along with a number of **Dynamic Shared Objects** (**DSO**s), which can be dynamically loaded or unloaded at runtime as necessary.

Virtual Hosts (Vhost)

- Apache Virtual Hosts (host) are used to run more than one web site(domain) using a single IP address.

- In other words you can have multiple web sites(domains) but a single server.

Virtual Hosts (Vhost)

- Different sites will be shown depending on the user’s requested URL.
- Best part is you can have any number of virtual hosts in a single server.
- It simply means you can have any number of web sites(domains) in a single server.

With Virtual Hosting

- Requests from each domain will be mapped into respective *document root*.
- *document root* is where all the files of the website are located(could be public_html)

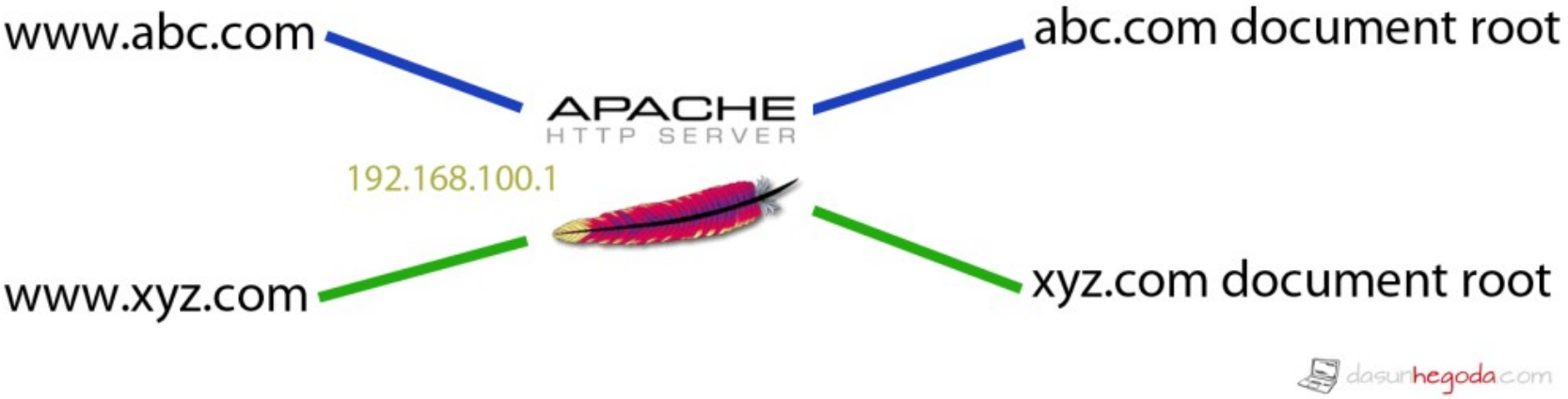
Two methods of Virtual Hosting

- Name-Based Virtual Hosting
- IP-Based Virtual Hosting

Name-Based Virtual Hosting

- Most of the time you will be using name-based virtual host configuration
- When a request is made to the Apache web server, it looks for the hostname in the HTTP header in the given request.
- Depending on the hostname, request will be served.

Name-Based Virtual Hosting



Name-Based Virtual Hosting

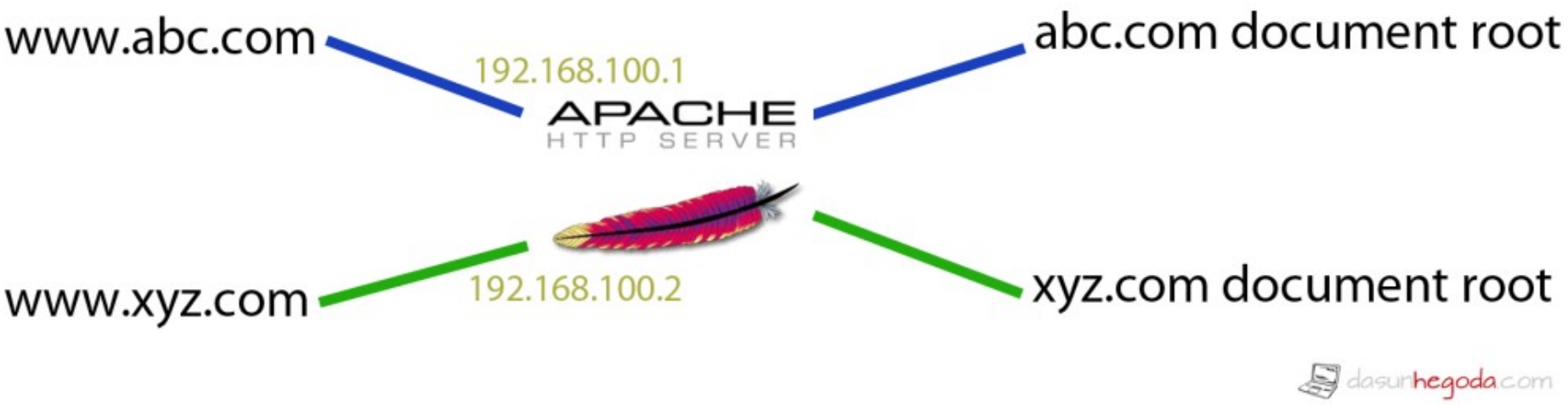
- The server has got only one ip-address but multiple web sites(domains) will be pointing to the same server.

- In this scenario we need to have two virtual hosts, one for xyz.com and one for abc.com.

IP-Based Virtual Hosting

- In this scenario the physical server should have two ip addresses
- Server should have two ethernet cards, each one of them are configured to the particular ip-address of the corresponding website
- There is only one physical server running Apache but two IPs.

IP-Based Virtual Hosting



Setting Up Virtual Hosts

- Apache HTTP Server's built in virtual hosting allows the server to provide different information based on which IP address, host name, or port is being requested

Setting Up Virtual Hosts

- To create a name-based virtual host, create configuration file named **httpd-vhosts.conf** in **/etc/httpd/conf.d/directory**

Ex. httpd-vhosts.conf

<VirtualHost *:80>
ServerAdmin webmaster@ penguin.example.com
DocumentRoot
"/www/docs/penguin.example.com"
ServerName penguin.example.com
ServerAlias www.penguin.example.com
ErrorLog "/var/log/httpd/dummy-
host.example.com-error_log"
CustomLog "/var/log/httpd/dummy-
host.example.com-access_log" common
</VirtualHost>

- *VirtualHost *:80* – Virtual hosts will be listening on the default port 80(443 for https)
- *ServerAdmin* – Server Admin’s email
- *DocumentRoot* – Path where web site files are located
- *ServerName* – Server name

- *ServerAlias* – Alternate names
- *ErrorLog* – File contains any errors that it encounters in processing requests
- *CustomLog* – All requests processed by the server. Access log file

- Note that ServerName must be a valid DNS name assigned to the machine

- You can find the example configuration file at **/usr/share/doc/httpd-VERSION/httpd-vhosts.conf**

Setting Up an SSL Server

- Secure Sockets Layer (SSL) is a cryptographic protocol that allows a server and a client to communicate securely.
- Its extended and improved version called Transport Layer Security (TLS) gives both privacy and data integrity.

- The Apache HTTP Server in combination with a module **mod_ssl**, uses the OpenSSL toolkit to provide the SSL/TLS support, is commonly referred to as the SSL server

- Red Hat Enterprise Linux also supports the use of Mozilla NSS as the TLS implementation.
- Support for Mozilla NSS is provided by the **mod_nss** module.

- Unlike an HTTP connection that can be read and possibly modified by anybody who is able to intercept it, the use of SSL/TLS over HTTP, referred to as HTTPS, prevents any inspection or modification of the transmitted content

Enabling **mod_ssl** Module

- If you intend to set up an SSL or HTTPS server using **mod_ssl**, you can not have the another application or module, like mod_nss configured to use the same port.
- Port 443 is the default port for HTTPS.

Enabling **mod_ssl** Module

- To set up an SSL server using the mod_ssl module and the OpenSSL toolkit, install the mod_ssl and openssl packages.
- Enter the following command as root:
-]# yum install mod_ssl openssl

Enabling **mod_ssl** Module

- This will create the mod_ssl configuration file at /etc/httpd/conf.d/ssl.conf
- This file is included in the main Apache HTTP Server configuration file by default.
- For the module to be loaded, restart the httpd service
-]# systemctl restart httpd

Enabling specifc **SSL & TLS**

- Two options to enable specific versions of the SSL and TLS protocol
- 1) Do it globally by adding the SSLProtocol directive in the “ ## SSL Global Context” section in **ssl.conf** file and removing it everywhere
- 2) Edit the default entry under “ # SSL Protocol support” in all “ VirtualHost” sections.

- After modifying the ssl.conf file, Restart the Apache daemon as follows:
-]# systemctl restart httpd

Configure the Firewall

- Red Hat Enterprise Linux does not allow HTTP and HTTPS traffic by default.

- To enable HTTP using the command line, issue the following command as **root**

- ***firewall-cmd --add-service http***

- To enable HTTPS using the command line, issue the following command as **root**

- ***firewall-cmd --add-service https***

- Note that these changes will not persist after the next system start.

- To make permanent changes to the firewall, repeat the commands adding the **--permanent** option

- ***firewall-cmd --add-service http --permanent***

- To check what services the firewall is configured to allow

- ***firewall-cmd --list-all***

Ex

#***firewall-cmd --list-all***

public (default, active)
interfaces: em1
sources:
services: dhcpv6-client ssh
output truncated

- example taken from a default installation, the firewall is enabled but HTTP and HTTPS have not been allowed to pass

- Once the HTTP and HTTPS firewall services are enabled, the services line will appear like

- **services: dhcpv6-client http https ssh**

Ex

#***firewall-cmd --list-all***

public (default, active)
interfaces: em1
sources:
services: dhcpv6-client http https ssh
output truncated

Assignments 1 Marks

1)Which is the web server available with RedHat enterprise server?
2)Give the command to install a Web Server in RedHat Linux
3)How will you set the Web Server to run by default in RedHat Linux server?
4)Which Protocol is used by the Web Servers to communicate with Clients?
5)Name the configuration file related to Virtual Hosting
6)What is SSL?

Assignments 7 Marks

1)What is a Web server? Give the significance of Web Server in RedHat Linux.
2)Give the procedure to set a RedHat Enterprise Server as a Web Server.
3)What is a Virtual Host? Give the types and significance of Virtual Hosting.
4)What is Virtual Host? Give the procedure to set up your RedHat Linux for virtual hosting.
5)What is SSL? Give the functionality of SSL
6)How to make http server as a secured server implementation? Explain