Linux Web Server (Apache) Workshop:

Part 1: Installation

Step 1: Install apache web server:
Check if it is already installed:
rpm -q httpd
httpd-2.4.6-45.el7.centos.i686
Otherwise install it:
yum install httpd

Step 2: Run & permanent the httpd service:
systemctl start httpd
netstat -antp | grep 80
systemctl enable httpd

Step 3: Permit https and https on firewall
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --reload

Step 4: Test with graphical browsers (optional)
Check if firefox is installed:
rpm -q firefox
firefox-45.4.0-1.el7.centos.i686
Otherwise install firefox:
yum install firefox
Open the firefox and go to our server home:
firefox http://127.0.0.1/ &

Step 5: apache package overview:
rpm -qc httpd | grep
/etc/httpd/conf.d/autoindex.conf
/etc/httpd/conf.d/userdir.conf
/etc/httpd/conf.d/welcome.conf
/etc/httpd/conf.modules.d/00-base.conf
/etc/httpd/conf.modules.d/00-dav.conf
/etc/httpd/conf.modules.d/00-lua.conf
/etc/httpd/conf.modules.d/00-mpm.conf
/etc/httpd/conf.modules.d/00-proxy.conf
/etc/httpd/conf.modules.d/00-systemd.conf
/etc/httpd/conf.modules.d/01-cgi.conf
/etc/httpd/conf/httpd.conf <<< Apache Master Config file
/etc/httpd/conf/magic
/etc/logrotate.d/httpd
/etc/sysconfig/htcacheclean
/etc/sysconfig/httpd

Part 2: Static content

Tip: Web servers usually map web URLs to files.
For example on Redhat/Centos apache will map:
http://127.0.0.1/ -- to --> /var/www/html
Step 1: Create a static page on web root:
cd /var/www/html/

echo "<html><body>Fanavaran Anisa</body></html>" > index.html
Now test it with firefox/elinks:
firefox http://127.0.0.1/index.html

Step 2: Address an image:
cp /usr/share/httpd/icons/apache_pb.gif /var/www/html/apache.gif
Now test it with firefox:
firefox http://127.0.0.1/apache.gif

Step 3: Address a complete page:
cd /var/www/html/
echo "<html><body><h1>Hello</h1></body></html>" > test2.html
Now test it with firefox:
firefox http://127.0.0.1/test2.html

Part 3: Dynamic content (CGI) with bash
Tip: CGI is an executeable program which apache will execute and send back the results to web browsers.
CGI programs have some limitations and standards on their input(s) and output(s):

Step 1: Create a Bash shell script as CGI:
cd /var/www/cgi-bin/
Create test.sh as :
#!/bin/bash
echo "Content-Type: text/plain"
echo
echo this is a simple bash CGI
date
df -h

Step 2: Make the program executable:
chmod +x test.sh
Step 3: Test CGI (bash script) program
Now test it with firefox:
firefox http://127.0.0.1/cgi-bin/test.sh

Part 4: Changing Apache modes
Tip: There are 3 modes for apache on RHEL/CentOS
- prefork mode: Multi-Processing Module (MPM)
- worker mode: Using threads.
(not fully compatible with all modules)
- event mode: simillar to worker.

Change the mode:
modify /etc/httpd/conf.modules.d/00-mpm.conf

Restart the service:
service httpd restart
httpd -V

Part 5: Enable user dirctory
Step 1: Modify the main config for UserDir Change the following lines in main file:
(file is /etc/httpd/conf.d/userdir.conf)

comment line 17
#UserDir disable
uncomment line 24
UserDir public_html
Restart the server:
service httpd restart

Step 2: Add & create a user dircetory:
useradd lpi
chmod go+rx /home/lpi
mkdir /home/lpi/public_html
chmod go+rx /home/lpi/public_html
Step 3: Put some content in directory & test:
echo test > /home/lpi/public_html/index.html
firefox http://localhost/~lpi/

Part 6: Overwriting configuration with .htaccess
Step 1: Modify the config for UserDir
Change the following lines in main file:
(file is /etc/httpd/conf.d/auth_basic.conf)

```
<Directory /var/www/html/test>
   AuthType Basic
   AuthName "Basic Authentication"
   AuthUserFile /etc/httpd/conf/.htpasswd
   require valid-user
</Directory>
```

Step 2: add a user : create a new file with "-c" (add the "-c" option only for the initial registration)
htpasswd -c /etc/httpd/conf/.htpasswd anisa
New password: # set password
Re-type new password: # confirm

Step 3: Restart the server:
service httpd restart

Step 4: Provide the content
echo Anisa > /var/www/html/test/index.html

Step 5: Test password protected area:
firefox http://localhost/test/ &

Part 7: Virtual Hosting (Port based)
Step 1: Create home directory for 2 sites:
mkdir /var/www/lpir{_com,_org}
cd /var/www/lpir_com
echo "<html><body><h1>lpir.com website</h1></body></html>" > index.html
cd /var/www/lpir_org
echo "<html><body><h1>lpir.org website</h1></body></html>" > index.html

Step 2: Configure 2 port based virtual hosting:
Edit file /etc/httpd/conf/httpd.conf and add the following:

Listen 8080

<VirtualHost *:8080>
DocumentRoot /var/www/lpir_com
</VirtualHost>
Listen 8090
<VirtualHost *:8090>
DocumentRoot /var/www/lpir_org
</VirtualHost>

Step 3: restart server & test:
service httpd restart
firefox http://127.0.0.1:8080/ &
firefox http://127.0.0.1:8090/ &

Part 8: Virtual Hosting (IP based)

Step 1: Check you network configurations,
you should have 2 IP addresses:
ip addr show | grep "inet " # check your ip
Now set you secondary IP address as:
ifconfig eth0:0 192.168.100.2nd

Step 2: Edit file /etc/httpd/conf/httpd.conf and
add the following:
<VirtualHost 192.168.100.1st:80>
DocumentRoot /var/www/lpir_com
</VirtualHost>
<VirtualHost 192.168.100.2nd:80>
DocumentRoot /var/www/lpir_org
</VirtualHost>

Step 3: restart server & test:
service httpd restart
firefox http://192.168.100.1st/ &
firefox http://192.168.100.2nd/ &

Part 9: Virtual Hosting (Name based)

Step 1: Add 2 domain names to your host and
modify file /etc/hosts as:
127.0.0.1 www.lpir.com lpir.com
127.0.0.1 www.lpir.org lpir.org

Step 2: Edit file /etc/httpd/conf/httpd.conf and
add the following:
NameVirtualHost *:80
<VirtualHost *:80>
ServerName lpir.com
ServerAlias www.lpir.com
DocumentRoot /var/www/lpir_com
</VirtualHost>
<VirtualHost *:80>
ServerName lpir.org
ServerAlias www.lpir.org
DocumentRoot /var/www/lpir_org

</VirtualHost>

Step 3: restart server & test:
service httpd restart
firefox http://www.lpir.com/ &
firefox http://www.lpir.org/ &

Part 10: Working with models (PHP)
Tip: LAMP is Linux Apache MySQL & PHP
Step 1: Install MySQL (MariaDB)
yum install mariadb-server mariadb

Step 2: Start the DB
systemctl start mariadb
systemctl enable mariadb.service

Step 3: Remove some dangerous defaults
mysql_secure_installation

Step 4: Install PHP
yum install php php-mysql
systemctl restart httpd.service

Step 5:
echo "<?php phpinfo(); ?>" > /var/www/html/info.php
firefox http://127.0.0.1/info.php

Part 11: Create Self-Signed SSL Certificates with OpenSSL

Step 1: Check and Install the required packages
rpm -ql | grep “httpd\|mod_ssl\|openssl”

Step 2: Generate Your Private Key
Create a 1024-bit RSA private key for your Apache server
openssl genrsa -des3 -out server.key 1024

View the content:
openssl rsa -noout -text -in server.key

Step 3: Create a Certificate Signing Request (CSR)
Create a Certificate Signing Request (CSR) that incorporates the server key you just generated:
openssl req -new -key server.key -out server.csr

View the details:
openssl req -noout -text -in server.csr

Step 4: Sign Your Certificate Signing Request
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt

Step 5: Configuring Apache to use SSL
<VirtualHost 192.168.10.3:443>

serverName www.lpir.org
DocumentRoot /var/www/html
SSLCertificateFile /etc/httpd/conf/server.crt
SSLCertificateKeyFile /etc/httpd/conf/server.key
SSLengine on
</VirtualHost>

Step 6: Test the result
firefox https://www.lpir.org &

Good Luck
Fanavaran Anisa - 2017