**13. Setup a virtual host with an alternative document root. Extend your web to include a virtual for the site http://wwwX.example.com. Set the document root as /virtual. Download http://content.example/pub/rhce/www.html and Place this to document root of the virtual host. Rename the file as index.html. Note: The other websites configures for your server must still accessible. wwwX.example.com is already provided by the name server on example.com**

#mkdir /virtual

#cd /virtual

#wget http://content.example/pub/rhce/www.html

#mv www.html index.html

#chcon -t httpd\_sys\_content\_t /virtual -R

Or

#chcon --reference /var/www/html /virtual -R

#ls -ldZ /virtual [for check SE context]

#cd /etc/httpd/conf.d/vhosts.conf

#vim vhosts.conf

<VirtualHost serverX.example.com:80>

ServerName serverX.example.com

DocumentRoot /var/www/html

</VirtualHost>

<Directory "/var/www/html/topsecret">

Require host serverX.example.com

</Directory>

**<VirtualHost wwwX.example.com:80>**

**ServerName wwwX.example.com**

**DocumentRoot /virtual**

**</VirtualHost>**

**<Directory /virtual>**

**Require all granted**

**</Directory>**

:wq!

#systemctl restart httpd

#curl wwwX.example.com

**14. Configure website http://webappX.example.com:9189/myapp on serverX with the document root /dynamic. Site should executes appweb.wsgi. Page is already provided on http://content.example.com/pub/appweb.wsgi. Content of the script should not be modified.**

#yum install mod\_wsgi -y

#mkdir /dynamic

#cd /daynamic

#wget http://content.example.com/pub/appweb.wsgi

#chcon -t httpd\_sys\_content\_t /dynamic -R

Or

#chcon --reference /var/www/html /dynamic -R

#ls -ldZ /virtual [for check SE context]

#cd /etc/httpd/conf.d/vhosts.conf

#vim vhosts.conf

<VirtualHost serverX.example.com:80>

ServerName serverX.example.com

DocumentRoot /var/www/html

</VirtualHost>

<Directory "/var/www/html/topsecret">

Require host serverX.example.com

</Directory>

<VirtualHost wwwX.example.com:80>

ServerName wwwX.example.com

DocumentRoot /virtual

</VirtualHost>

<Directory /virtual>

Require all granted

</Directory>

**<VirtualHost webappX.example.com:9189>**

**ServerName webappX.example.com**

**WSGIScriptAlias /myapp /dynamic/appweb.wsgi**

**</VirtualHost>**

**<Directory /dynamic>**

**Require all granted**

**</Directory>**

**Listen 9189**

:wq!

#semanage port -l | grep http

#semanage port -a -t http\_port\_t -p tcp 9189

#semanage port -l | grep http

#firewall-cmd --permanent --add-port=9189

#firewall-cmd --reload

#firewall-cmd --list-all

#systemctl restart httpd

[check from foundation machine firefox **webappX.example.com:9189/myapp**]

**15. Restore a database on serverX from the backup file http://content.example.com/new\_inventory.dump. The database name should be Contacts. It should be access only within the localhost. Set a password for root user as "Postroll". Other than the root user, the user andrew able to read the query from the above mentioned database. The user should be authenticated with the password as "Postroll".**

#yum groupinstall mariadb mariadb-client -y

#firewall-cmd –permanent –add-service=mysql

#firewall-cmd –reload

#firewall-cmd –list-all

#systemctl start maraiadb

#systemctl enable mariadb

#systemctl restart mariadb

#mysql\_secure\_installation

[set root password as "Postroll", Disallow root login from remote and Reload privileges for mariadb]

#mysql -u root -p

Password: Postroll

Or

#mysql -u root -pPostroll

<none>create database Contacts;

<none>show databases;

Exit

#wget http://content.example.com/new\_inventory.dump

#mysql -u root -pPostroll Contacts < new\_inventory.dump

#mysql -u root -pPostroll

<none>show databases;

<none>use mysql;

<mysql>create user andrew@localhost identified by 'Postroll';

<mysql>grant select on Contacts.\* to andrew@localhost identified by 'Postroll';

<mysql>show grants for andrew@localhost;

Exit

**16. Contacts database queries using a database on the system serverX and use the appropriate SQL query to answer the following questions: 1. Find the first\_name of the person who's password is lrsji . Create a file in /root called fname.ext and write the name there. 2. Find the number of persons who live in Dhaka. Create a file in /root called number.txt and write the number.**

#mysql -u root -pPostroll

<none>show databases;

<none>use Contacts;

<Contacts> show tables;

<Contacts> select \* from user\_info where password='lrsji';

<Contacts> select count(\*) from user\_info where address='Dhaka';

Exit

#echo result\_of\_1st\_query > /root/fname.txt

#echo result\_of\_2nd\_query > /root/number.txt

**17. Create a script on serverX called /root/random with following details.**

**When run as /root/random postconf, should bring the output as "postroll"**

**When run as /root/random postroll, should bring the output as "postconf"**

**When run with any other argument or without argument, should bring the stdrr as "/root/random postconf | postroll"**

**18. Create a script on serverX called /root/createusers. When this script is with the testfile argument, it should add all the users from the file ownload the file from http://content.example.com/pub/testfile**

**All users should have the login shell as /bin/false, password not required. When this script is called with anyother argument, it should print the message as "Input File Not Found". When this scriptis run without argument, it should display "Usage: /root/createusers" NOTE: If the users are added no need to delete.**

**19. Configure serverXwith the following requirements:**

**Share the /nfsshare directory within the example.com domain clients only, share must be writeable.**

**Share the /nfssecure, enable krb5p security to secure access to the NFS share from URL** [**http://classroom.example.com/pub/keytabs**](http://classroom.example.com/pub/keytabs)**/serverX.keytab. Create a directory named as protected under /nfssecure The exported directory should have read/write access from all sub-domains of the example.com domain. Ensure the directory /nfssecure/protected should be owned by the user harry with read/write permission.**

**21. Mount /nfsshare directory on desktopXunder /public directory persistently at system boot time.**

**Mount /nfssecure/protected with krb5 secured share on desktop beneath /secure/protected with keytab** [**http://classroom.example.com/pub/keytabs/desktopX.keytab**](http://classroom.example.com/pub/keytabs/desktopX.keytab)

**The user harry able to write files on /secure directory.**

**22. Share the /sambadir directory via SMB on server**

**-Your SMB server must be a member of the TESTGROUP workgroup.**

**-The share’s name must be data.**

**-The data share must be available to example.com domain clients only.**

**-The data share must be browse able.**

**-Susan must have read access to the share, authentication with the same password password if necessary.**

**23. Configure the serverX to share /opstack with SMB share name must be cluster.**

**The user frankenstin readable, writeable and accessible to the /opstack SMB share.**

**The user martin has read access to the /opstack SMB share.**

**Both users should have SMB password ‘SaniTago’.**