Please read below points very carefully.

All the ip addresses and subnets which belong to the mention hostnames are available in the instruction page.

Regarding ip address setting, its clearly mentioned if you need you can maintain static network using given static ip's, But no need to change ip address settings as the assigned address are persistent across reboots.

1) Configure selinux.
- Configure your systems that should be running in Enforcing.

Solution

# vim /etc/selinux/config
SELINUX=enforcing

After reboot and verify with this command
getenforce

Or else use setenforce 1 command, and then no need to reboot the systems.

2) Configure repository.

- Create a Repository for your virtual machines. The URI is http://station.network0.example.com/content/rhel7.0/x86_64/dvd

Solution

vim /etc/yum.repos.d/local.repo
[localrepo]
name = Local Repo for RHCE Exam
baseurl = http://station.network0.example.com/content/rhel7.0/x86_64/dvd
gpgcheck = 0
enabled = 1

yum clean all
yum repolist

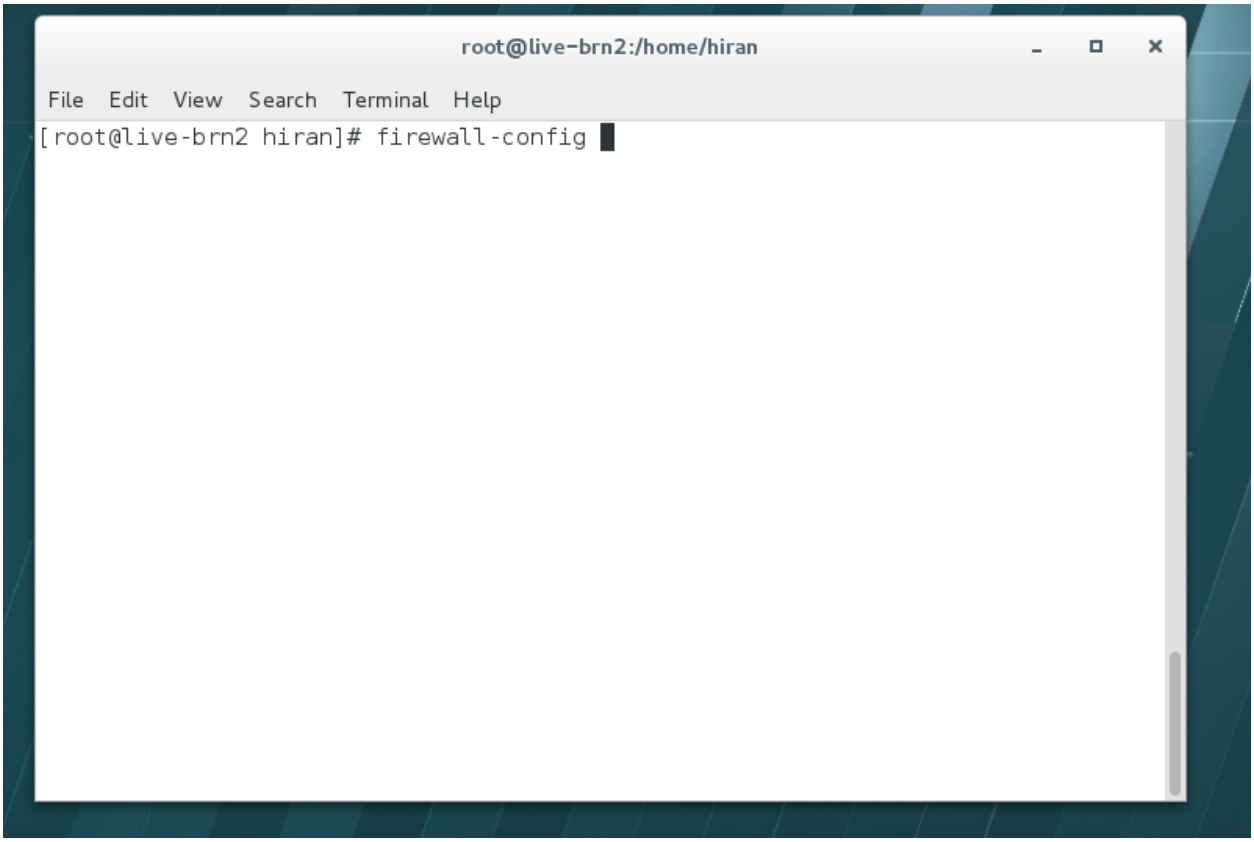
3) SSH configuration.

- Configure SSH access on your virtual hosts as follows.

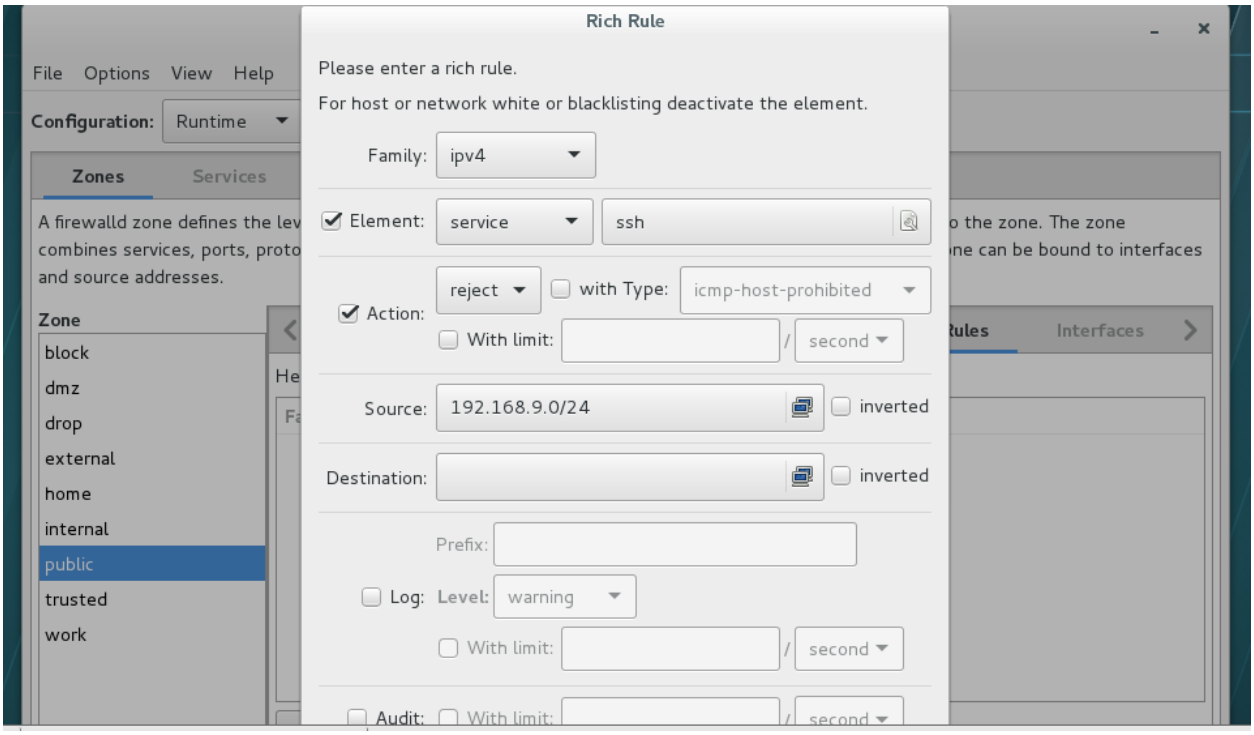
- Clients within my22ilt.org should NOT have access to ssh on your systems

Solution

You can do this by adding a rich rule.



First select permanent from the drop down, and then go to the richrules tab, and then add a rich rule as follows.

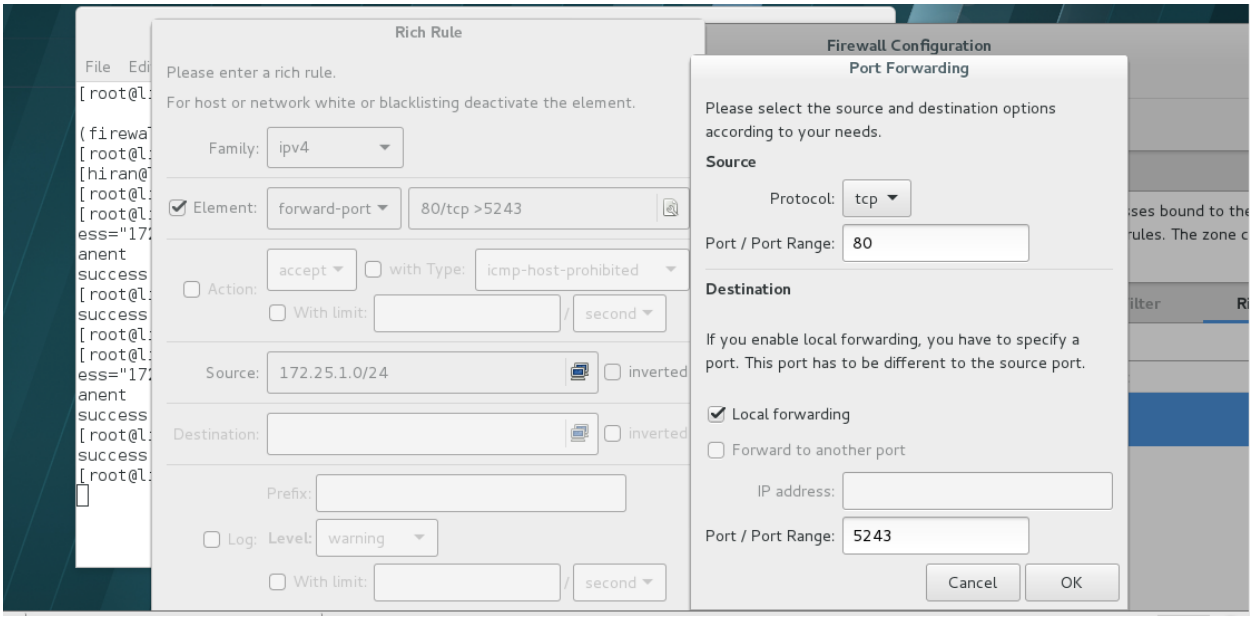


4) Configure port forwarding.
- Configure serverX to forward traffic incoming on port 80/tcp
from source network 172.25.X.0/255.255.255.0 to port on 5243/tcp.

Solution

firewall-cmd --add-rich-rule='rule family="ipv4" source address="172.25.1.0/24" forward-port port="80" protocol="tcp" to-port="5423"' –permanent

Or else use firewall-config gui and add a rich rule as follows.



5) Customize User Environment.
- Create a command called qstat on both serverX and desktopX.
- It should able to execute the following command
(ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm)
- The command shoud be executable by all users.

Solution

vim /etc/bashrc
alias qstat='ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm'

or

vim /usr/local/bin/qstat

'ps -eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm'

Chmod a+x /usr/local/bin/qstat

6) Configure ipv6 network.
- Configure eth0 with a static ipv6 addresses as follows.
- Configure a Static IPv6 address in serverX as fddb:fe2a:ab1e::c0a8:64/64.
- Configure a Static IPv6 address in desktopX as fddb:fe2a:ab1e::c0a8:02/64.
- Both machines are able to communicate within the network fddb:fe2a:able/64

- The changes should be permanent even after the reboot.

Solution

On ServerX:
nmcli conn show ----> to find the connection name that attaches to the eth0 interface
nmcli conn modify "System eth0" ipv6.addresses fddb:fe2a:ab1e::c0a8:64/64
nmcli conn modify "System eth0" connection.autoconnect true
nmcli conn modify "System eth0" ipv6.method manual
nmcli conn down "System eth0"
nmcli conn up "System eth0"

7) Link aggregation
- Configure your serverX and desktopX, which watches for link changes and selects an active port for data transfers.
- serverX should have the address as 192.168.X.10/255.255.255.0.
- desktopX should have the address as 192.168.X.11/255.255.255.0.
(Note: where X is your station number)

Solution

You have to run lab teambrdige setup script in the servers to create the two interface eno1 & eno2
At the exam they will provide 2 additional interfaces, So make sure not to touch your interface which uses for ssh connectivity.
On Server Machine:

Go to documentation available under /usr/share/doc and filter results for activebackup

#grep –irn “activebackup” * from here you can get the json format for the runner

#nmcli device show
#nmcli con add type team con-name Team1 ifname Team1 config '{"runner": {"name": "activebackup"}}'
#nmcli con modify Team1 ipv4.addresses 192.168.1.10/24
#nmcli con modify Team1 ipv4.method manual
#nmcli con add type team-slave con-name Team1-slave1 ifname eno1 master Team1
#nmcli con add type team-slave con-name Team1-slave2 ifname eno2 master Team1
#nmcli con up Team1
#nmcli con up Team1-slave1

#nmcli con up Team1-slave2
Verification & Testing:
#teamdctl Team1 state
#nmcli dev dis eth1 ---> Disconnect device for verification
#nmcli con up Team1-slave1
#teamnl Team1 ports
#teamnl Team1 getoption activeport
#teamnl Team1 setoption activeport PORT_NUMBER
#ping –I Team1 192.168.1.11

8) SMTP Configuration.
- Configure the SMTP mail service on serverX and desktopX which relay the mail only from local system through station.network0.example.com, all outgoing mail have their sender domain as example.com. Ensure that mail should not store locally.
- Verify the mail server is working by sending mail to a natasha user.
- Check the mail on both serverX and desktopX with the below URL
http://station.network0.example.com/system1
http://station.network0.example.com/system2

Solution

Please read the documentation under /usr/share/doc
# yum install postfix
# systemctl enable postfix
# systemctl start postfix
# vim /etc/postfix/main.cf
inet_interfaces = all
mydestination =
myorigin = example.com
mynetworks = 127.0.0.0/8, [::1]/128
relayhost = [station.network0.example.com]
local_transport = error: local delivery disabled
# systemctl restart postfix

9) NFS server.
- Configure serverX with the following requirements.
- Share the /nfsshare directory within the example.com domain clients only, share must be writable.

- Share the /nfssecure, enable krb5p security to secure access to the NFS share from URI http://station.network0.example.com/pub/keytabs/serverX.keytab
- Create a directory named as protected under /nfssecure
- The exported directory should have read/write access from all subdomains of the example.com domain.
- Ensure the directory /nfssecure/protected should be owned by the user harry with read/write permission.

Solution

Kindly notice that user harry mentioned here is a Kerberos user, in our classroom environment we have ldapuserx who is a Kerberos user, So instead of harry use ldapuserx for testing.

Before starting make sure to run lab nfs-krb5 setup on both server and desktop

#yum install –y nfs-utils
#mkdir –p /nfsshare
#chmod 0777 /nfsshare or chown nfsnobody /nfsshare
#vim /etc/exports
/nfsshare *.example.com(rw)
#systemctl restart nfs-server
#systemctl enable nfs-server
#firewall-cmd --permanent --add-service=nfs
#firewall-cmd --reload

#mkdir –p /nfssecure
#wget –O /etc/krb5.keytab http://station.network0.example.com/pub/keytabs/serverX.keytab
#systemctl enable nfs-secure-server
#mkdir /nfssecure/protected
#vim /etc/exports
/nfssecure *.example.com(rw,sec=krb5p,sync)

#chown ldapuserx /nfssecre/protected
Better to do like this:
#setfacl –m u:ldapuserx:rwX /nfssecure/protected
#exportfs -r
#semanage fcontext -a -t public_content_rw_t "/nfsshare(/.*)?"
#semanage fcontext -a -t public_content_rw_t "/nfssecure(/.*)?"

#restorecon -Rv /nfssecure/
#firewall-cmd --permanent --add-service=rpc-bind
#firewall-cmd --permanent --add-service=mountd
#firewall-cmd –reload
#systemctl restart nfs-server
#systemctl restart nfs-secure-server
#systemctl enable nfs-secure-server

10) Configure nfs mount.
- Mount /nfsshare directory on desktopX under /public directory persistently at system boot time.
- Mount /nfssecure/protected with krb5p secured share on desktopX beneath /secure/protected provided with keytab http://station.network0.example.com/pub/keytabs/desktopX.keytab
- The user harry able to write files on /secure directory

Solution

#yum install –y nfs-utils
wget –O /etc/krb5.keytab http://station.network0.example.com/pub/keytabs/desktopX.keytab
#systemctl start nfs-secure
#systemctl enable nfs-secure
#mkdir –p /public
#vim /etc/fstab
server1.example.com:/nfsshare /public nfs defaults,sync 0 0
#mkdir –p /secure/protected
#vim /etc/fstab
server1.example.com:/nfssecure/protected /secure/protected nfs defaults,sec=krb5p,sync 0 0
Verification:
#ssh ldapuserx@localhost
#cd /secure/protected
#echo “Is it writeable?” >> test.txt

11) Configure smb access.
- Share the /sambadir directory via SMB on serverX
- Your SMB server must be a member of the TESTGROUP workgroup
- The share name must be data

- The data share must be available to example.com domain clients only
- The data share must be browseable
- susan must have read access to the share, authenticating with the same password "password", if necessary
- Configure the serverX to share /opstack with SMB share name must be cluster.
- The user frankenstein has readable,writeable,accesseable to the /opstack SMB share.
- The user martin has read access to the /opstack SMB share.
- Both users should have the SMB passwd "SaniTago".

Solution

#yum install samba samba-client
#systemctl start smb nmb
#systemctl enable smb nmb
#firewall-cmd --permanent --add-service=samba
#firewall-cmd --reload
#mkdir -p /sambadir
#semanage fcontext -a -t samba_share_t "/sambadir(/.*)?"
#restorecon -Rv /sambadir
#setfacl –m u:susan:r-X /sambadir
#vim /etc/samba/smb.conf
#workgroup = TESTGROUP
[data]
comment = data share
path = /sambadir
browseable = yes
valid users = susan
read only =yes
hosts allow = 172.25.1. #(ifconfig and get your ip and only use the 3 octels)
#grep –i "susan" /etc/passwd
(It it return nothing then create a user first)
#useradd -s /sbin/nologin susan
#smbpasswd -a susan

#mkdir –p /opstack
#semanage fcontext -a -t samba_share_t "/ opstack (/.*)?"
#restorecon -Rv / opstack
#vim /etc/samba/smb.conf
[cluster]
comment = opstack share
path = /opstack
write list = frankenstein

writable = no
useradd -s /sbin/nologin frankenstein
useradd -s /sbin/nologin martin
smbpasswd –a Frankenstein
smbpasswd –a martin

Allow Frankenstein write access & Martin read access to the directory

1) setfacl -m u:frankenstein:rwX /opstack/
2) setfacl -m u:martin:r-X /opstack/

12) smb multiuser mount.
- Mount the samba share /opstack permanently beneath /mnt/smbspace on desktopX as a multiuser mount.
- The samba share should be mounted with the credentials of frankenstein.

Solution

#yum –y install cifs-utils samba-client
#mkdir –p /mnt/smbspace
#vim /root/smb-multiuser.txt
username=frankenstein
password= SaniTago
chmod 0600 /root/multiuser.txt
#vim /etc/fstab
//server1/cluster /mnt/smbspace cifs defaults,sec=ntlmssp,credentials=/root/smb-multiuser.txt,multiuser 0 0

13) Webserver.
- Implement a webserver for the site http://serverX.example.com
- Download the webpage from http://station.network0.example.com/pub/rhce/rhce.html
- rename the downloaded file in to index.html.
- copy the file into the document root.
- Do not make any modification with the content of the index.html.
- Clients within my22ilt.org should NOT access the webserver on your systems

Solution

#yum install httpd httpd-manual
#systemctl start httpd
#systemctl enable httpd
#firewall-cmd --permanent --add-service=http

#firewall-cmd –reload
#wget http://station.network0.example.com/pub/rhce/rhce.html
#cp rhce.html /var/www/html/index.html
#cd /etc/httpd/conf.d/
#vim server1.conf
<VirtualHost *:80>
ServerAdmin webmaster@server1.example.com
ServerName server1.example.com
DocumentRoot /var/www/html
CustomLog "logs/server1_access_log" combined
ErrorLog "logs/server1_error_log"
</VirtualHost>
<Directory "/var/www/html">
<RequireAll>
Require all granted
Require not host my22ilt.org
</RequireAll>
</Directory>
#systemctl restart httpd

IGNORE THIS, IF YOU HAVE SET IT INSIDE THE APACHE CONFIG

#ORDER of rule is important, First deny/reject than Allow

#firewall-cmd --permanent --add-rich-rule='family="ipv4" source address=" my22ilt.org" service name="http" log prefix="http denied" level="info" limit value="3/m" reject'
#firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source address="0.0.0.0/0" service name="http" accept'
#firewall-cmd --reload

14) secured webserver
- configure the website https://serverX.example.com with TLS
- SSLCertificate file http://classroom.example.com/pub/rhce/tls/certs/system1.networkX.crt
- SSLCertificatekeyfile
http://classroom.example.com/pub/rhce/tls/private/system1.networkX.key
- SSL CA certificate file http://classroom.example.com/pub/example-ca.crt

Solution

#yum install –u mod_ssl

#wget http://classroom.example.com/pub/rhce/tls/certs/system1.network1.crt
#wget http://classroom.example.com/pub/rhce/tls/private/system1.network1.key
#wget http://classroom.example.com/pub/example-ca.crt
#cp system1.network1.crt /etc/pki/tls/certs/
#cp system1.network1.key /etc/pki/tls/private/
#cp example-ca.crt /etc/pki/tls/certs/

Very Important, Fix the Permission on Key File

#chmod 0600 /etc/pki/tls/private/system1.network1.key
#vim /etc/httpd/conf.d/server1.conf
(Add the following)
<VirtualHost *:443>
ServerName server1.example.com
DocumentRoot /var/www/html
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/localhost.crt
SSLCertificateKeyFile /etc/pki/tls/private/localhost.key
#SSLCertificateChainFile /etc/pki/tls/certs/server-chain.crt
</VirtualHost>

#firewall-cmd --permanent --add-service=https
#firewall-cmd –reload

15) Webpage content modification.
- Implement website for http://serverX.example.com/owndir
- Create a directory named as "owndir" under the document root of webserver Download http://station.network0.example.com/pub/rhce/restrict.html
- rename the file into index.html
- The content of the owndir should be visible to everyone browsing from your local system but should not be accessible from other location

Solution

#mkdir /var/www/html/owndir
#restorecon –Rv /var/www/html
#cd /var/www/html/owndir
#wget http://station.network0.example.com/pub/rhce/restrict.html
#mv restrict.html intex.html
#vi /etc/httpd/conf.d/server1.conf
(Add this)

<Directory "/var/www/html/owndir">
AllowOverride None
Require all Denied
Require local
</Directory>
systemctl restart httpd

16) Virtual hosting.
- Setup a virtual host with an alternate document root.
- Extend your web to include a virtual for the site http://vhostsX.example.com
- Set the document root as /usr/local/vhosts
- Download http://station.network0.example.com/pub/rhce/vhost.html
- rename it as index.html
- place this document root of the virtual host
- Note: The other websites configures for your server must still accessible.
vhosts.networkX.example.com is already provide by the name server on example.com

Solution

Check that the mentioned document root exists by:
#cd /usr/local/vhosts
If it doesn't exist then create it:
#mkdir /usr/local/vhosts
#cd /usr/local/vhosts
#wget http://station.network0.example.com/pub/rhce/vhost.html
#mv vhost.html index.html
#semanage fcontext -a -t httpd_sys_content_t "/usr/local/vhosts(/.*)?"
#restorecon -Rv /usr/local/vhosts/
Create the configuration of new virtual host:
#vim /etc/httpd/conf.d/vhosts.conf
<VirtualHost *:80>
ServerAdmin webmaster@vhosts1.example.com
ServerName vhosts1.example.com
DocumentRoot /usr/local/vhosts
CustomLog "logs/vhosts_access_log" combined
ErrorLog "logs/vhosts_error_log"
</VirtualHost>
<Directory "/usr/local/vhosts">
AllowOverride None
Allow open access:
Require all granted
</Directory>

#systemctl restart httpd

17) Dynamic Webpage Configuration.
- configure website http://wsgiX.example.com:8961 on system1 with the documentroot /var/www/scripts
- Site should executes webapp.wsgi.
- Page is already provided on http://classroom.example.com/pub/webapp.wsgi
- Content of the script should not be modified.

Solution

yum install -y mod_wsgi
mkdir –p /var/www/scripts
cd /var/www/scripts
wget http://classroom.example.com/pub/webapp.wsgi
restorecon –Rv /var/www/scripts
vim /etc/httpd/conf/httpd.conf
Listen 8961
vim /etc/httpd/conf.d/wsgi1.conf
<VirtualHost *:8961>
ServerAdmin webmaster@wsgi1.example.com
ServerName wsgi1.example.com
DocumentRoot /var/www/scripts # We don't need it,only testing
WSGIScriptAlias / /var/www/scripts/webapp.wsgi
CustomLog "logs/wsgi_access_log" combined
ErrorLog "logs/wsgi_error_log"
</VirtualHost>
<Directory "/var/www/scripts">
AllowOverride None
Allow open access:
Require all granted
</Directory>
firewall-cmd --permanent --add-port=8961/tcp
firewall-cmd --reload
semanage port -a -t http_port_t -p tcp 8961
systemctl status httpd
Verification:
From Server2,do this
links --dump http://wsgi1.example.com:8961
Should present with the desired page

18) Script1
- 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 stderr as "/root/random postconf|postroll"

Solution

vim /root/random
#!/bin/bash

If [$# -eq 0] ; then
Echo "/root/random postconf|postroll"
Elif [$1 == 'postroll'] ; then
Echo "postconf"
Elif [$1 == 'postconf'] ; then
Echo "postroll"

chmod +x /root/random

20) Configure SCSI storage.
- Create a new 1GB target on your serverX.example.com.
- The block device name should be data_block
- The server should export an iscsi disk called iqn.2014-10.com.example:serverX.
- This target should only be allowed to desktopX

Solution

yum install –y targetcli
systemctl start target
systemctl enable target
firewall-cmd --permanent --add-port=3260/tcp
firewall-cmd –reload
#targetcli
backstores/block/create data_block /dev/sdb1
iscsi/ create iqn.2014-10.com.example:server1
cd iscsi/iqn.2014-10.com.example:server1/tpg1/
acls create iqn.2014-10.com.example:desktop1
luns/ create backstores/block/data_block
portals Server_IP(172.25.x.11) 3260
exit

21) ISCSI Initiator
- The serverX.example.com provides an iscsi port(3260).
connect the disk with desktopX.example.com and configure filesystem with the following requirements.
- Create 800MB partition on ISCSI block device and assign the filesystem as xfs.
- Mount the volume under /mnt/initiator at the system boot time.
- The filesystem should contains the copy of
http://station.network0.example.com/pub/iscsi.txt.
- The file sould be owned by root with 0644 permission.
- NOTE: content of the file should not be modified.

Solution

yum install –y iscsi-initiator-utils
vim /etc/iscsi/initiatorname.iscsi
InitiatorName=iqn.2014-11.com.example:desktop1
systemctl start iscsi
systemctl start iscsid
systemctl enable iscsi
systemctl enable iscsid
iscsiadm --mode discoverydb --type sendtargets --portal server1.example.com --discover
iscsiadm --mode node --targetname iqn.2014-11.com.example:server1 --portal server1.example.com:3260 --login
Verification:
iscsiadm –m session –P 3 (it should show the State: running)
lsblk
fdisk /dev/sdb
Create the partition of 800M
mkfs.xfs /dev/sdb1
mkdir –p /mnt/initiator
mount /dev/sdb1 /mnt/initiator
blkid /dev/sdb1
vim /etc/fstab
UUID=c9213938-6753-4001-b939-4b5720c8ec5e /mnt/initiator xfs _netdev 0 0
cd /mnt/initiator
wget http://station.network0.example.com/pub/iscsi.txt
chown root iscsi.txt
chmod 0644 iscsi.txt

22) Mariadb
- Restore a database on serverX from the backup file
http://classroom.example.com/pub/rhce/backup.mdb.
- 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".

Solution

yum groupinstall -y mariadb mariadb-client
systemctl start mariadb
systemctl enable mariadb
(We don’t need to open firewall port because it says that only access from localhost)
mysql_secure_installation

```
wget http://classroom.example.com/pub/rhce/backup.mdb
mysql –u root –p
CREATE DATABASE Contacts;
CREATE USER andrew@localhost IDENTIFIED BY 'Postroll';
GRANT SELECT ON Contacts.* TO andrew@localhost;
mysql -u root -p Contacts < backup.mdb
```