**Domain Name:**

System1:  system1.district10.example.com

System2:  system2.district10.example.com

**IP Address:**

System1:172.24.10.110/24

System1:172.24.10.120/24

**Name Server: 172.24.10.250**

**Gateway:172.24.10.254**

**Root password : zaldebro**

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

2.Configure repository.  Create a Repository for your virtual machines. The URI is<http://station.district0.example.com/content/rhel7.0/x86_64/dvd>

3. SSH configuration.

Clients within my133ilt.org should NOT have access to ssh on your systems

Clients with domain district10.example.com should be able to access the systems

4. Port forwarding

Configure system1 to forward traffic incoming on port 80/tcp from source network 172.24.X.0/255.255.255.0 to port on 5243/tcp

5. User Environment.

Create a command called qstat on both system1 and system2.  It should able to execute the following command(ps eo pid,tid,class,rtprio,ni,pri,psr,pcpu,stat,wchan:14,comm)

The command should be executable by all users.

6. Ipv6 network.

Configure eth0 with a static ipv6 addresses as follows.

Configure a Static IPv6 address in system1 as fddb:fe2a:ab1e::c0a8:64/64.

Configure a Static IPv6 address in system2 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

7. Link aggregation  Configure your system1 and system2, which watches for link changes and selects an active port for data transfers.  System1 should have the address as 172.24.10.10/255.255.255.0.  System2 should have the address as 172.24.10.20/255.255.255.0

8. SMTP Configuration.  Configure the SMTP mail service on system1 and system2 which relay the mail only from local system through station.network0.example.com, all outgoing mail have their sender domain as example.com.  Verify the mail server is working by sending mail to a local  user clarke.

Check the mail on both system1 and system2 with the below URL

[http://rhcert.district0.example.com](http://rhcert.district0.example.com/)

9. NFS server.

Configure system1 with the following requirements.

Share the /nfsshare directory within the example.com domain clients only, share must not be writable.  Share the /nfssecure, enable krb5p security to secure access to the NFS share from URI​​http://station.network0.example.com/pub/keytabs/system1.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.

10. Nfs mount

Mount /nfsshare directory on system2 under /public directory persistently at system boot time.  Mount /nfssecure/protected with krb5p secured share on system2 beneath /secure/protected provided with keytab http://station.network0.example.com/pub/keytabs/system2.keytab

The user harry able to write files on /secure directory

11. Smb access

Share the /sambadir directory via SMB on system1  Your SMB server must be a member of the STAFF workgroup  The share name must be data .The data share must be available to district10.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

12.SAMBA Mount

Share /opstack with SMB share name must be cluster.

The user frankenstein has readable,writeable,accessible to the /opstack SMB share.  The user martin has read access to the /opstack SMB share.  Both users should have the SMB passwd "SaniTago".

The share must be browseable

Mount the samba share /opstack permanently beneath /mnt/smbspace on system2 as a multiuser mount.  The samba share should be mounted with the credentials of martin.

13.  Webserver.

Implement a webserver for the site​​ http://system1.district10.example.com

Download the webpage from​​http://station.district0.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.

Webserver must be available to clients with domain district10.example.com

Clients within my22ilt.org should NOT access the webserver on your systems

14) Secured webserver

configure the website​​https://system1.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/exampleca.crt

15) Webpage content modification.

Implement website for ​​http://system1.district10.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

User harry can edit the contents of the directory

16) Virtual hosting

Setup a virtual host with an alternate document root.  Extend your web to include a virtual for the site​​http://www.district10.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

17. Dynamic Webpage Configuration.

Configure website​​http://dynamic.district10.example.com:8899 on system1 with the documentroot /var/www/scripts  Site should executes webapp.wsgi.

Page is already provided on​ ​http://station.district0.example.com/pub/webapp.wsgi  Content of the script should not be modified.

18) Script1

Create a script on system1 called /root/random with following details.  When run as /root/random user, should bring the output as "user"  When run as /root/random kernel, should bring the output as "user"  When run with any other argument or without argument, should bring the stderr as "/root/random user|kernel"

19) Script2

Create a script on system1 called /root/createusers  When this script is called with the argument, it should add all the users from the file  Download the file from​​http://station.district0.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 script is run without any argument, it should display "Usage: /root/createusers"

20.  ISCSI Storage

Create a new 3GB target on your system1.district10.example.com.  The logical block name should be lvm  The server should export an iscsi disk called iqn.2015-12.com.example.district10:system1.  This target should only be allowed to system2

21. ISCSI Initiator

The system1 provides an iscsi port(3260).

connect the disk with system2.district10.example.com and configure filesystem with the following requirements.

Create 2040 MB partition on ISCSI block device and assign the filesystem as ext3.

Mount the volume under /mnt/initiator at the system boot time.

22.MaridDB Configuration

Configure  a MariaDB on System1 with a database name Contacts.

The Database must be accessible locally only.

The root password must be zaldebro.

Apart from root, only the user Zyuichi must be able to query the Contacts Database.

Zyuichi must be identified by zaldebro.

Restore a database on system1 from the backup file <http://station.district0.example.com/pub/rhce/backup.mdb>

23.MariaDB Query

Find the first name of user with password “ecosystem