Lab 1: Working with virtualization – 20 Minutes

KVM is an open source hardware virtualization software through which we can create and run multiple Linux based and windows based virtual machines simultaneously. KVM is known as Kernel based Virtual Machine because when we install KVM package then KVM module is loaded into the current kernel and turns our Linux machine into a hypervisor.

Step 1: Before proceeding KVM installation, let's check whether your system's CPU supports Hardware Virtualization.

Run the beneath command from the console.

```
# grep -E '(vmx|svm)' /proc/cpuinfo
```

Step 2: Install KVM and its associated packages.

```
# yum install qemu-kvm qemu-img virt-manager libvirt libvirt-python libvirt-client virt-install virt-viewer bridge-utils
```

Step 3: Start and enable libvirtd service.

```
# systemctl start libvirtd
```

```
# systemctl enable libvirtd
```

Step 4: Run the beneath command to check whether KVM module is loaded or not

```
# lsmod | grep kvm
```

Step 5: Start virt-manager and check its option to create a VM

```
# virt-manager
```

Explore options to create New VM and ask for any doubt you have.

Lab 2: Configure Squid server – 20 Minutes

A proxy server has many use cases. it could range from personal internet access to restrict organization systems/servers to access the external world or to limit external internet access for a set of servers on the cloud.

The best way to configure a proxy server is by using the Squid proxy. It is a widely used proxy server.

Step 1: Install squid

yum -y install squid

Step 2: Start and enable squid server.

systemctl start squid

systemctl enable squid

Step 3: Check the status of squid server.

systemctl status squid

Step 4: Open "/etc/squid/squid.conf" configuration file for squid and starts doing the modifications as per requirement.

Acl comnet src "ipaddressofyourmachine/subnetmask"

save and quit the file

Step 5: Restart the squid service

systemctl restart squid

Step 6: Verify the connectivity from the curl command on squid server

curl -x http://ipofyourserver:3128 -I http://www.facebook.com

Step 7: Create "/etc/squid/blocksites" file and add the following lines

.facebook.com

save and quit the file.

Step 8: Add the configuration in squid.conf to block the site.

Acl block1 dstdomain “/etc/squid/blocksites”

http_access deny block1

Step 9: Restart the squid service

systemctl restart squid

Step 10: Verify the connectivity from the curl command on squid server. You should see the forbidden error now.

curl -x http://ipofyourserver:3128 -I http://www.facebook.com

Lab 3: Configure Ldap Server – 30 Minutes

Step 1: Install the following LDAP RPM packages on LDAP server

yum -y install openldap compat-openldap openldap-clients openldap-servers openldap-servers-sql openldap-devel

Step 2: Start the LDAP service and enable it for the auto start of service on system boot.

systemctl enable slapd

systemctl enable slapd

Step 3: Run the command to create ldap root password. We will use this password throughout the lab

slappasswd -h {SSHA} -s redhat

Keep the output of previous command safe, it will be used to in later configuration

Step 4: OpenLDAP servers configuration files are found in /etc/openldap/slapd.d/. To start with the configuration of LDAP, we would need to update the variables “olcSuffix” and “olcRootDN“.

vim db.ldif

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=example,dc=com # change the domain name
```

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=ldapadm,dc=example,dc=com # change the domain name
```

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootPW
olcRootPW: {SSHA}d/thexcQUuSfe3rx3gRaEhHpNJ52N8D3   # replace the password hash
```

save and quit the file

Step 5: Send the configuration to the ldap server

#ldapmodify -Y EXTERNAL -H ldapi:/// -f db.ldif

Step 6: Make a changes to /etc/openldap/slapd.d/cn=config/olcDatabase={1}monitor.ldif (Do not edit manually) file to restrict the monitor access only to ldap root (ldapadm) user not to others.

vi monitor.ldif

dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcAccess
olcAccess: {0}to * by dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external, cn=auth" read by dn.base="cn=ldapadm,dc=example,dc=com" read by * none

save and quit the file

ldapmodify -Y EXTERNAL -H ldapi:/// -f monitor.ldif

Step 7: Copy the sample database configuration file to /var/lib/ldap and update the file permissions.

cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
chown ldap:ldap /var/lib/ldap/*

Step 8: Add the cosine and nis LDAP schemas.

ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif

Step 9: Generate base.ldif file for your domain.

vim base.ldif

dn: dc=example,dc=com
dc: example
objectClass: top
objectClass: domain

dn: cn=ldapadm ,dc=example,dc=com
objectClass: organizationalRole
cn: ldapadm
description: LDAP Manager

dn: ou=People,dc=example,dc=com
objectClass: organizationalUnit
ou: People

dn: ou=Group,dc=example,dc=com
objectClass: organizationalUnit
ou: Group

save and quit the file

Step 10: Build the director structure

ldapadd -x -W -D "cn=ldapadm,dc=itzgeek,dc=local" -f base.ldif

Step 11: Create user config and add in the system

vi user.ldif

dn: uid=testuser,ou=People,dc=example,dc=com
ObjectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: testuser
uid: testuser
uidNumber: 9999
gidNumber: 100
homeDirectory: /home/testuser
loginShell: /bin/bash
gecos: test user account
userPassword: {crypt}x
shadowLastChange: 17058
shadowMin: 0
shadowMax: 99999
shadowWarning: 7

ldapadd -x -W -D "cn=ldapadm,dc=example,dc=com" -f user.ldif

Step 12: Verify the ldap entry

ldapsearch -x cn=testuser-b dc=example,dc=com