Part 1

System Configuration and Management

PRINCE BAJAJ

System Configuration and Management

1. Configure server and client machines with IP addresses 192.168.122.10 and 192.168.122.20 respectively and set the DNS Address on both machines to 192.168.122.254.

Commands: (On server.example.com)

ip addr **(To check the IP Address configuration on interface eth0)**

nmcli connection add con-name server ifname eth0 type ethernet ip4 192.168.122.10/24 **(To assign IP Address on eth0 interface)**

nmcli connection modify server ipv4.dns 192.168.122.254 ipv4.method manual **(To set DNS IP and make the connection static)**

nmcli connection up server **(To Activate the connection)**

nmcli device status **(To check the device status)**

nmcli connection show server **(To Display connection)**

Commands: (On client.example.com)

ip addr **(To check the IP Address configuration on interface eth0)**

nmcli connection add con-name client ifname eth0 type ethernet ip4 192.168.122.20/24 **(To assign IP Address on eth0 interface)**

nmcli connection modify client ipv4.dns 192.168.122.254 ipv4.method manual **(To set DNS IP and make the connection static)**

nmcli connection up client **(To Activate the connection)**

nmcli device status **(To check the device status)**

nmcli connection show client **(To Display connection)**

Note – You can configure Default Gateway for interface by using gw4 option with nmcli conn add command.

You can also add Default Gateway after adding connection with Ipv4 address with nmcli conn modify with ipv4.addresses set to "IP Address+space+Gateway IP"

2. Configure Server and Client machines to have selinux running in enforcing mode.

Commands: (On both server.example.com and client.example.com Machines)

getenforce or sestatus (To check status of selinux)

vim /etc/selinux/config (To Set selinux in enforcing mode)

SELINUX=enforcing

SELINUXTYPE=targeted (This is set by default)

:wq

systemctl reboot (Reboot the system to make changes effective)

:wq

3. Configure both server and client machines to use the repository located at **/root/server-repo** and **/root/client-repo** respectively.

<u>Commands: (On server.example.com)</u>

vim /etc/yum.repos.d/server.repo **(To configure system to use already created repository)**

[server]

name = server-repo

baseurl = <u>file:///root/server-repo</u>

gpgcheck = 0

enabled = 1   :wq

yum clean all **(To clear cache)**

yum repolist **(To check Packages in yum repository)**

yum groups list hidden **(To display all Group Packages in yum repository)**

<u>Commands: (On client.example.com)</u>

vim /etc/yum.repos.d/client.repo **(To configure system to use already created repository)**

[client]

name = client-repo

baseurl = <u>file:///root/client-repo</u>

gpgcheck = 0

enabled = 1   :wq

yum clean all **(To clear cache)**

yum repolist **(To check Packages in yum repository)**

yum groups list hidden **(To display all Group Packages)**

4. Configure server and client machine with ipv6 addresses **2020::1/64** and **2020::2/64** respectively. Ipv4 configurations should not be affected.

Commands: (On server.example.com)

    ip addr **(To check the IP Address configuration on interface eth0)**

    nmcli connection modify server ipv6.addresses 2020::1/64 ipv6.method manual **(To assign IPv6 and make the connection static)**

    nmcli connection up server **(To Activate the connection)**

    systemctl restart network.service **(To Restart Network service)**

    nmcli connection show server **(To Display connection)**

Commands: (On client.example.com)

    ip addr **(To check the IP Address configuration on interface eth0)**

    nmcli connection modify client ipv6.addresses 2020::2/64 ipv6.method manual **(To assign IPv6 and make the connection static)**

    nmcli connection up client **(To Activate the connection)**

    systemctl restart network.service **(To Restart Network service)**

    nmcli connection show server **(To Display connection)**

5. Configure teamed interface between server and client machine :

    a) Use eth1 and eth2 interfaces on both systems.

    b) Assign IP address 30.0.0.1/24 on server and 30.0.0.2/24 on client machine to teamed interface.

    c) Configure teamed interface with active-backup scheme.

    d) Check the reachability using ping command from both machines

Commands: (On server.example.com)

    ip addr **(To Verify status od eth1 and eth2 interfaces)**

    nmcli device disconnect eth1 **(To disconnect device eth1)**

    nmcli device disconnect eth2 **(To disconnect device eth2)**

    nmcli connection add type team con-name TEAMS ifname TEAMED_S **(To create teamed Interface)**

    nmcli connection modify TEAMS team.config '{"runner": {"name": "activebackup" }}' **(To define runner for teamed interface)**

    nmcli connection modify TEAMS ipv4.addresses 30.0.0.1/24 ipv4.method manual **(To assign IP address to teamed interface and make connection static)**

    nmcli connection add type slave con-name SLAVE1 type team-slave ifname eth1 master TEAMED_S **(To add eth1 interface slave to teamed interface)**

    nmcli connection add type slave con-name SLAVE2 type team-slave ifname eth2 master TEAMED_S **(To add eth2 interface slave to teamed interface)**

    nmcli connection up SLAVE1 **(To activate SLAVE1 connection)**

    nmcli connection up SLAVE2 **(To activate SLAVE2 connection)**

    nmcli connection up TEAMS **(To activate teamed interface connection)**

Note - I will suggest to use modified JSON format config file to define the runner for Teamed interface to avoid issue due to wrong syntax

<u>Commands: (On client.example.com)</u>

ip addr **(To Verify status od eth1 and eth2 interfaces)**

nmcli device disconnect eth1 **(To disconnect device eth1)**

nmcli device disconnect eth2 **(To disconnect device eth2)**

nmcli connection add type team con-name TEAMC ifname TEAMED_C **(To create teamed Interface)**

nmcli connection modify TEAMC team.config '{"runner": {"name": "activebackup" }}' **(To define runner for teamed interface)**

nmcli connection modify TEAMC ipv4.addresses 30.0.0.1/24 ipv4.method manual **(To assign IP address to teamed interface and make connection static)**

nmcli connection add type slave con-name SLAVE1 type team-slave ifname eth1 master TEAMED_C **(To add eth1 interface slave to teamed interface)**

nmcli connection add type slave con-name SLAVE2 type team-slave ifname eth2 master TEAMED_C **(To add eth2 interface slave to teamed interface)**

nmcli connection up SLAVE1 **(To activate SLAVE1 connection)**

nmcli connection up SLAVE2 **(To activate SLAVE2 connection)**

nmcli connection up TEAMC **(To activate teamed interface connection)**

Note - I will suggest to use modified JSON format config file to define the runner for Teamed interface to avoid issue due to wrong syntax

6. Configure a static route on **server.example.com** for destination 10.1.1.0/24 via 192.168.122.40.

a> Route configuration should be persistent after reboot

b> eth0 should be used as exit interface

c > Also configure interface to use Default Gateway 192.168.122.1

Commands: (On server.example.com)

ip route add 10.1.1.0/24 via 192.168.122.40 **(To add route in run time)**

ip route show or route -n **(To verify the route)**

To add Persistent Static route With command line :

nmcli connection modify server ipv4.routes "10.1.1.0/24 192.168.122.40"

To add Persistent Static route With config file:

vim /etc/sysconfig/network-scripts/route-server **(for persistent static route definition)**

10.1.1.0/24 via 192.168.122.40 dev eth0

:wq

To add Default Gateway with Command line :

nmcli connection modify server ipv4.addresses "192.168.122.10/24 192.168.122.1"

systemctl restart network **(To Restart network service)**

route -n **(Verify the static route)**

7. Configure system and client machines to authenticate using Kerberos. Kerberos server is configured on FreeIPA server **ipaserver.example.com**

        a> Verify using klist –k if host principle is present on both machines

Commands: (On both server.example.com & client.example.com)

    yum install ipa-client (To install IPA Client package)

    systemctl status sssd (To check status of System Security Service)

    cat /etc/sssd/sssd.conf (To check if host is already joined)

    klist -k (To check if host principal is present in krb5.keytab file)

    ipa-client-install (To install IPA client to join IPA domain on ipaserver.example.com)

    klist -k (To verify if host Principal is present now)

Also check status of sssd service and verify from sssd.conf file. sssd service must be active and enabled now