

69. Configure and demonstrate the use of NIS and NFS. (on CentOS/redhat container)

Show the imp steps and file name of configurations. (on answer sheet)

Create 5 users and make two groups, demonstrate the NIS and NFS concepts by example on LAN connected Linux OS.

NIS configuration Centos

Commands:

Open terminal and run server container:

```
docker run --name nisserver --privileged -it centos:centos6 /bin/bash
```

In another terminal, run client container:

```
docker run --name nisclient --privileged -it centos:centos6 /bin/bash
```

NOTE: Inside both client and server containers, first run below 2 commands, else you will not be able to install packages:

```
sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
```

```
sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
```

Server container configuration

(inside container)

```
yum -y update
```

```
yum -y install rpcbind ypserv nano passwd
```

```
root@g14:/home/aaditya# docker run --name nisserver --privileged -it centos:centos6 /bin/bash
[root@c9c543f971fa /]# sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*
[root@c9c543f971fa /]# sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*
[root@c9c543f971fa /]# yum update -y && yum -y install ypserv rpcbind
```

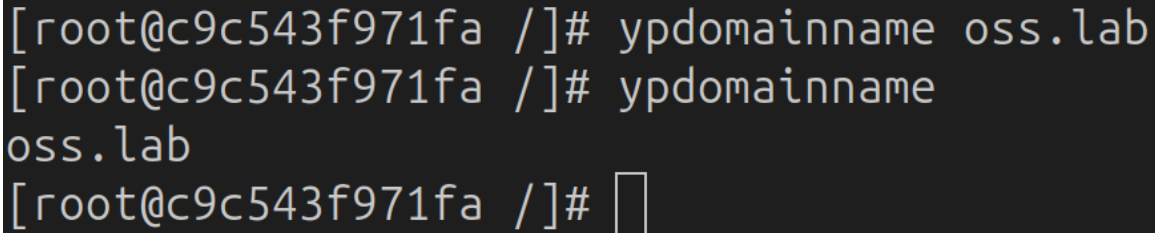
Set hostname:

```
hostname srvr
```

```
[root@c9c543f971fa /]# hostname srvr
[root@c9c543f971fa /]# hostname
srvr
[root@c9c543f971fa /]#
```

Now run:

[ypdomainname oss.lab](#) (can be anything but make sure it is unique. if domain exists already, it may create problems)

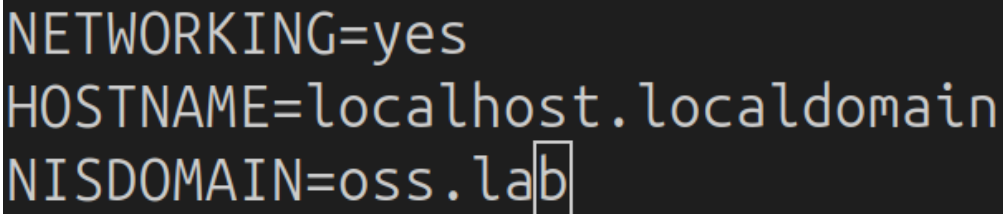


```
[root@c9c543f971fa /]# ypdomainname oss.lab
[root@c9c543f971fa /]# ypdomainname
oss.lab
[root@c9c543f971fa /]#
```

Open `/etc/sysconfig/network` file - [nano /etc/sysconfig/network](#)

Add following line at the end:

[NISDOMAIN=oss.lab](#)



```
NETWORKING=yes
HOSTNAME=localhost.localdomain
NISDOMAIN=oss.lab
```

For docker containers, the IP address will look something like: 172.17.0.x

Check it using `ifconfig` command.

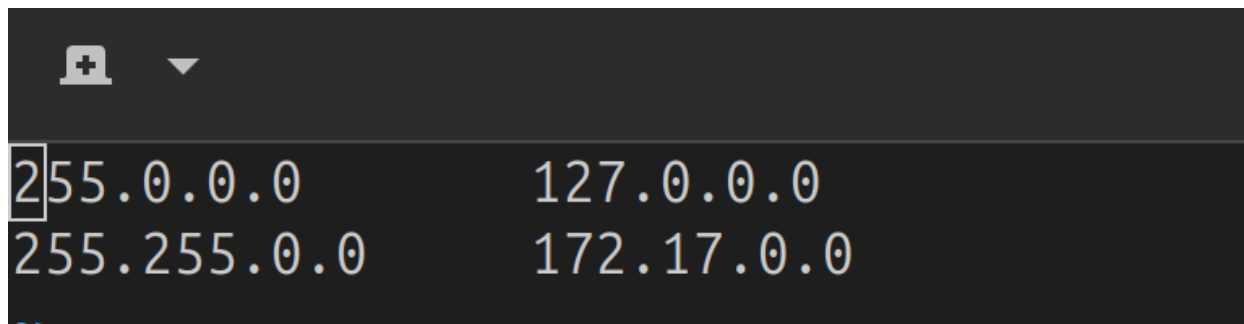
To use `ifconfig`, run command: [yum install net-tools -y](#)

```
[root@c9c543f971fa /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 02:42:AC:11:00:02
          inet addr:172.17.0.2  Bcast:172.17.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:40696 errors:0 dropped:0 overruns:0 frame:0
          TX packets:23096 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:78267666 (74.6 MiB)  TX bytes:1776908 (1.6 MiB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
```

[nano /var/yp/securenets](#)

Add IP network range you want to allow to access NIS server. Since subnet mask is 255.255.0.0, just add first two octets ie. 172.17.0.0



```
255.0.0.0      127.0.0.0
255.255.0.0    172.17.0.0
```

In the **client container**, set the hostname as “clnt” or something of your choice and note its IP address using [ifconfig](#) command.

```
[root@81de9d04472c /]# hostname clnt
[root@81de9d04472c /]# hostname
clnt
```

Also in **client container**, set the domain name.
[ypdomainname oss.lab](#) (same domain name as before)

Going back to server container..

`nano /etc/hosts`

Add following lines in the given format:

Format ->

<server Ip address> <server-hostname.domainname> <server-hostname>

<client Ip address> <client-hostname.domainname> <client-hostname>

```
127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.17.0.2     srvr.oss.lab srvr
172.17.0.3     clnt.oss.lab clnt
~
~
~
~
~
~
```

Now run commands:

`service rpcbind start`

`service ypserv start`

`service ypxfrd start`

`service yppasswdd start`

`/usr/lib64/yp/ypinit -m`

```
[root@c9c543f971fa /]# /usr/lib64/yp/ypinit -m

At this point, we have to construct a list of the hosts which will run NIS
servers.  srvr.oss.lab is in the list of NIS server hosts.  Please continue to add
the names for the other hosts, one per line.  When you are done with the
list, type a <control D>.
    next host to add:  srvr.oss.lab
    next host to add:  
```

Verify that the name is correct and press Ctrl+D, then press y.

```
@c9c543f971fa:/
[root@c9c543f971fa /]# /usr/lib64/yp/ypinit -m

At this point, we have to construct a list of the hosts which will run NIS
servers.  srvr.oss.lab is in the list of NIS server hosts.  Please continue to add
the names for the other hosts, one per line.  When you are done with the
list, type a <control D>.
    next host to add:  srvr.oss.lab
    next host to add:
The current list of NIS servers looks like this:

srvr.oss.lab

Is this correct? [y/n: y] y
We need a few minutes to build the databases...
Building /var/yp/oss.lab/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory `/var/yp/oss.lab'
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating hosts.byname...
Updating hosts.byaddr...
Updating rpc.byname...
Updating rpc.bynumber...
Updating services.byname...
Updating services.byservicename...
Updating netid.byname...
Updating protocols.bynumber...
Updating protocols.byname...
Updating mail.aliases...
gmake[1]: Leaving directory `/var/yp/oss.lab'

srvr.oss.lab has been set up as a NIS master server.

Now you can run ypinit -s srvr.oss.lab on all slave server.
```

According to problem statement, you need to create 5 users and 2 groups. To add a newuser:

`useradd newuser`

`passwd newuser`

Then add any password of your choice.

To create a group:

`groupadd groupname`

To add user to a group:

`usermod -aG <groupname> <username>`

```
Sat Nov 9 11:30:11 A
@c9c543f971fa:/

[root@c9c543f971fa /]# useradd newuser
[root@c9c543f971fa /]# passwd newuser
Changing password for user newuser.
New password:
BAD PASSWORD: it is based on a dictionary word
BAD PASSWORD: is too simple
Retype new password:
passwd: all authentication tokens updated successfully.
[root@c9c543f971fa /]#
```

To apply this to NIS database:

`cd /var/yp/`
`make`

```
[root@c9c543f971fa /]# cd /var/yp
[root@c9c543f971fa yp]# make
gmake[1]: Entering directory `/var/yp/oss.lab'
Updating passwd.byname...
Updating passwd.byuid...
Updating group.byname...
Updating group.bygid...
Updating netid.byname...
gmake[1]: Leaving directory `/var/yp/oss.lab'
[root@c9c543f971fa yp]#
```

Client container configuration:

(inside container)

`yum -y update`
`yum -y install ypbind rpcbind authconfig nano`

`nano /etc/hosts`

Add IP addresses of server and client as done before

```

127.0.0.1      localhost
::1           localhost ip6-localhost ip6-loopback
fe00::0       ip6-localnet
ff00::0       ip6-mcastprefix
ff02::1       ip6-allnodes
ff02::2       ip6-allrouters
172.17.0.3     clnt.oss.lab clnt
172.17.0.2     srvr.oss.lab srvr

```

`authconfig --enablenis --nisdomain=<domain> --nisserver=<server> --enablemkhomedir --update`
Refer screenshot

```

[root@81de9d04472c /]# authconfig --enablenis --nisdomain=oss.lab --nisserver=srvr.oss.lab --enablemkhomedir --update
[root@81de9d04472c /]#

```

Try switching to 'newuser' that we created on NIS server.

`su - newuser`

```

[root@81de9d04472c /]# service ypbind restart
Shutting down NIS service: [ OK ]
Starting NIS service: [ OK ]
Binding NIS service: . [ OK ]
[root@81de9d04472c /]# su - newuser
Creating directory '/home/newuser'.
[newuser@clnt ~]$

```

If directory is created as shown, you have configured everything correctly.

NFS configuration Centos

Run the container:

In centos docker containers, NFS does not work because the filesystem is not supported. Instead, you need to mount a directory from the host machine.

First create a directory on the host machine (not container)

`mkdir /mnt/nfsshare`

Now we will mount this onto container when we run the container.

`docker run --privileged -it -v /mnt/nfsshare:/mntnfsshare centos:centos6 /bin/bash`

```
@377eb1c88d12:/
root@g14:/home/aaditya# docker run --name nfsserver --privileged -it -v /mnt/nfsshare:/mnt/nfsshare centos:centos6 /bin/bash
[root@377eb1c88d12 /]# ls /mnt/nfsshare/
[root@377eb1c88d12 /]#
```

(inside container)

`sed -i 's/mirrorlist/#mirrorlist/g' /etc/yum.repos.d/CentOS-*`

`sed -i 's|#baseurl=http://mirror.centos.org|baseurl=http://vault.centos.org|g' /etc/yum.repos.d/CentOS-*`

`yum update -y`

`yum install -y nfs-utils nano net-tools`

Go to /mnt/nfsshare folder and add some files.

```
[root@377eb1c88d12 /]# cd /mnt/nfsshare/
[root@377eb1c88d12 nfsshare]# touch hello
[root@377eb1c88d12 nfsshare]# vi hello
[root@377eb1c88d12 nfsshare]#
```

`nano /etc/exports`

Add folder to export and IP address of clients. You need not create a client container, just use your host machine's IP address.

Format ->

`<folder to export> <ip address of client/host machine>(rw, sync)`

To allow access to all, you can set * instead of IP address


```
/mnt/nfsshare *(rw,sync)
```

```
~
```

```
~
```

```
~
```

```
~
```

Then run: `exportfs -arv`

```
[root@377eb1c88d12 /]# vi /etc/exports
```

```
[root@377eb1c88d12 /]# exportfs -arv
```

```
exporting */mnt/nfsshare
```

Start NFS service by running commands:

`rpcbind`

`rpc.nfsd`

`rpc.mountd`

`rpc.statd`

Client configuration:

Create a directory for mounting the nfs directory onto client:

`mkdir clientshare`

```
root@g14:/home/aaditya# mkdir clientshare
```

`showmount -e <ip address of server container>`

You should see something like this:

```
aaditya@g14:~$ showmount -e 172.17.0.2
```

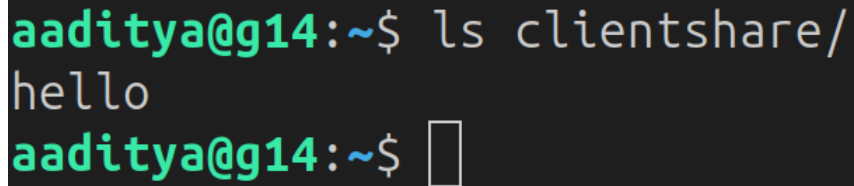
```
Export list for 172.17.0.2:
```

```
/mnt/nfsshare *
```

```
aaditya@g14:~$
```

Then mount this directory onto client:

```
sudo mount -t nfs 172.17.0.2:/mnt/nfsshare/ clientshare/
```

A terminal window with a dark background. The prompt is 'aaditya@g14:~\$'. The user has entered 'ls clientshare/' and the output is 'hello'. The prompt is now 'aaditya@g14:~\$' followed by a cursor.

```
aaditya@g14:~$ ls clientshare/  
hello  
aaditya@g14:~$
```

Now, you should be able to access the files from /mnt/nfsshare.

For more information, refer these articles:

<https://www.tecmint.com/install-nfs-server-on-centos-8/>

https://www.server-world.info/en/note?os=CentOS_7&p=nis&f=1