



**MALAD KANDIVALI EDUCATION SOCIETY'S
NAGINDAS KHANDWALA COLLEGE OF COMMERCE,
ARTS & MANAGEMENT STUDIES & SHANTABEN NAGINDAS
KHANDWALA COLLEGE OF SCIENCE
MALAD [W], MUMBAI – 64
(AUTONOMOUS)**

**(Reaccredited 'A' Grade by NAAC)
(AFFILIATED TO UNIVERSITY OF MUMBAI)
(ISO 9001:2015)**

CERTIFICATE

Name: Mr./Ms. Yash Panchal

Roll No: 533

Programme: BSc CS

Semester: V

This is certified to be a bonafide record of practical works done by the above student in the college laboratory for the course **Linux Server Administration** (Course Code: **1852UCSPR**) for the partial fulfillment of Fifth Semester of BSc CS during the academic year 2020-2021.

The journal work is the original study work that has been duly approved in the year 2020-2021 by the undersigned.

External Examiner

Mr. Gangashankar Singh
(Subject-In-Charge)

Date of Examination:

(College Stamp)

Name: **Yash Panchal**Roll No: **533**

Subject: Linux Server Administration

Class: T.Y.B.Sc.CS

INDEX

Sr. No	Date	Topic	Sign
1	D- 05/08/20	Install DHCP server	
2	D- 10/08/20	Initial settings: Add a user, Network settings, Configure services and List of services	
3	D- 17/08/20	Configure NFS server to share directories or files on your network	
4	D- 07/09/20	SSH Server - Password Authentication Configure SSH server	
5	D- 12/09/20	Install Samba to share folders or files between Windows and Linux	
6	D- 19/09/20	Configure NTP, Install and Configure NTP Server, Configure NTP Client	
7	D- 26/09/20	Install MySQL to configure database server	
8	D- 10/10/20	Configure NIS in order to share user accounts in your local network.	

PRACTICAL NO: 1**AIM:** Install DHCP server.**THEORY:**

1. DHCP stands for dynamic host configuration protocol and is a network protocol used on IP networks where a DHCP server automatically assigns an IP address and other information to each host on the network so they can communicate efficiently with other endpoints.
2. DHCP is based on a client-server model and based on discovery, offer, request, and ACK.
3. Every device on a TCP/IP-based network must have a unique unicast IP address to access the network and its resources. Without DHCP, IP addresses for new computers or computers that are moved from one subnet to another must be configured manually; IP addresses for computers that are removed from the network must be manually reclaimed.
4. DHCP is a client-server protocol in which servers manage a pool of unique IP addresses, as well as information about client configuration parameters, and assign addresses out of those address pools.
5. DHCP-enabled clients send a request to the DHCP server whenever they connect to a network.

Advantages of using DHCP:-

- Centralized management of IP addresses
- Ease of adding new clients to a network
- Reuse of IP addresses reducing the total number of IP addresses that are required

Disadvantage of using DHCP:-

- IP conflict can occur

COMMANDS & OUTPUT:

1. Command to install dhcp-server:

```
$ sudo apt-get install -y isc-dhcp-server
```

```
yash@yash:~$ sudo apt-get install -y isc-dhcp-server
[sudo] password for yash:      TRANS.TBL    VBoxWindowsAdditions...
Reading package lists... Done
Building dependency tree
Reading state information... Done
isc-dhcp-server is already the newest version (4.4.1-2.1+b2).
```

2. Then we need to configure our dhcpcd.conf by directly editing through a path:

```
$ sudo nano /etc/dhcp/dhcpcd.conf
```

yash@yash:~\$ sudo nano /etc/dhcp/dhcpcd.conf

```
GNU nano 5.2
# This declaration allows BOOTP clients to get dynamic addresses,
# which we don't really recommend.

#subnet 10.254.239.32 netmask 255.255.255.224 {
#  range dynamic-bootp 10.254.239.40 10.254.239.60;
#  option broadcast-address 10.254.239.31;
#  option routers rtr-239-32-1.example.org;
#}

# A slightly different configuration for an internal subnet.
subnet 192.168.106.0 netmask 255.255.255.0 {
    range 192.168.106.200 192.168.106.225;
    option domain-name-servers 192.168.106.128,8.8.8.8;
    option domain-name "kali";
    option routers 192.168.106.255;
    option broadcast-address 192.168.106.255;
    default-lease-time 600;
    max-lease-time 7200;
}

# Hosts which require special configuration options can be listed in
# host statements. If no address is specified, the address will be
# allocated dynamically (if possible), but the host-specific information
# will still come from the host declaration.
```

3. To know about ip address of our machine the following command can be given as:

```
$ ip addr
```

yash@yash:~\$ ip addr

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:aa:3a:41 brd ff:ff:ff:ff:ff:ff
    inet 192.168.106.128/24 brd 192.168.106.255 scope global noprefixroute eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feaa:3a41/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

4. Then we configure the subnet and the netmask, after configuring it then save the file and exit. (press[esc] and :wq!)

5. Now we must edit /etc/default/isc-dhcp-server:

```
$ sudo vi /etc/default/isc-dhcp-server
```

```
$ cat /etc/default/isc-dhcp-server
```

```
yash@yash:~$ sudo vi /etc/default/isc-dhcp-server
yash@yash:~$ cat /etc/default/isc-dhcp-server
# Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dhcp-server)

# Path to dhcpcd's config file (default: /etc/dhcp/dhcpcd.conf).
DHCPDv4_CONF=/etc/dhcp/dhcpcd.conf
#DHCPDv6_CONF=/etc/dhcp/dhcpcd6.conf

# Path to dhcpcd's PID file (default: /var/run/dhcpcd.pid).
DHCPDv4_PID=/var/run/dhcpcd.pid
#DHCPDv6_PID=/var/run/dhcpcd6.pid

# Additional options to start dhcpcd with.
#       Don't use options -cf or -pf here; use DHCPD_CONF/ DHCPD_PID instead
#OPTIONS=""

# On what interfaces should the DHCP server (dhcpcd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="eth0"
INTERFACESv6=""
```

6. Now start & check the status of the service:

```
$ sudo systemctl start isc-dhcp-server.service
```

```
$ sudo systemctl status isc-dhcp-server.service
```

```
yash@yash:~$ sudo systemctl start isc-dhcp-server
yash@yash:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Fri 2020-10-16 00:17:05 IST; 8s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 2191 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 4 (limit: 2515)
   Memory: 5.2M
      CGroup: /system.slice/isc-dhcp-server.service
              └─2204 /usr/sbin/dhcpcd -4 -q -cf /etc/dhcp/dhcpcd.conf eth0

Oct 16 00:17:03 yash systemd[1]: Starting LSB: DHCP server ...
Oct 16 00:17:03 yash isc-dhcp-server[2191]: Launching IPv4 server only.
Oct 16 00:17:03 yash dhcpcd[2204]: Wrote 0 leases to leases file.
Oct 16 00:17:03 yash dhcpcd[2204]: Server starting service.
Oct 16 00:17:05 yash isc-dhcp-server[2191]: Starting ISC DHCPv4 server: dhcpcd.
Oct 16 00:17:05 yash systemd[1]: Started LSB: DHCP server.
```

7. If the service gets failed, then we can use ifconfig eth0 <ip address> netmask <0.0.0.0>

```
$ sudo ifconfig eth0 192.168.106.128 netmask 255.255.255.0
```

```
$ ip addr
```

```
yash@yash:~$ sudo ifconfig eth0 192.168.106.128 netmask 255.255.255.0
yash@yash:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:aa:3a:41 brd ff:ff:ff:ff:ff:ff
        inet 192.168.106.128/24 brd 192.168.106.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fea:3a41/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

And then restart and check the status of **isc-dhcp-server**

8. Now to test the DHCP:

```
$ sudo dhcpcd -T eth0
```

```
yash@yash:~$ sudo dhcpcd -T eth0
Internet Systems Consortium DHCP Server 4.4.1
Copyright 2004-2018 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
Config file: /etc/dhcp/dhcpcd.conf
Database file: /var/lib/dhcp/dhcpcd.leases
PID file: /var/run/dhcpcd.pid
Wrote 0 leases to leases file.
Lease file test successful, removing temp lease file: /var/lib/dhcp/dhcpcd.leases.1602305743
yash@yash:~$ █
```

PRACTICAL NO: 2

AIM: Initial settings: Add a user, Network settings, Configure services and List of services.

THEORY:

1. Adding a user:

A user is a person who utilizes a computer or network service.

A user is the one who works on the machine or have the permissions to read, write and execute particular or all files.

A user can be added individually or they can be added in a group too.

2. Network settings:

The ip address, ethernet connections and other network connections can be seen through using the command:

\$ sudo ip addr

3. Configuring services:

Services are also known as daemons which efficiently run in the background, services usually have a 'd' in last of their name to identify it as a daemon for example - smbd , nmbd, etc services can be looked up in the /etc/ folder as mostly the services lie there for example: \$sudo nano /etc/dhcp/dhcpd.conf

COMMANDS & OUTPUT:

1. Adding a user:

\$ sudo adduser <username>

```
yash@yash:~$ sudo adduser yashp0
[sudo] password for yash:      VBoxWind-      VBoxWind-
Adding user `yashp0' ...      wsAdditions..      wsAdditions..
Adding new group `yashp0' (1011) ...
Adding new user `yashp0' (1010) with group `yashp0' ...
Creating home directory `/home/yashp0' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:      VBoxDarwin-      VBoxWind-
passwd: password updated successfully      pkg      VBoxWind-
Changing the user information for yashp0      wsAdditions..      wsAdditions..
Enter the new value, or press ENTER for the default
      Full Name []: YashP      VBoxWind-      VBoxWind-
      Room Number []:
      Home Work Phone []: 0N.I-      VBoxDarwin-      VBoxWind-
      Home Phone []:
      Other []:
Is the information correct? [Y/n] Y
```

2. Network settings:

```
$ sudo ip addr
```

```
yash@yash:~$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:aa:3a:41 brd ff:ff:ff:ff:ff:ff
        inet 192.168.106.128/24 brd 192.168.106.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:fea:3a41/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

3. Configuring services:

Here, I've taken SSH service for demonstration

```
yash@yash:~$ sudo systemctl start ssh
yash@yash:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-16 00:04:49 IST; 1h 10min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Main PID: 750 (sshd)
        Tasks: 1 (limit: 2515)
      Memory: 2.9M
        CGroup: /system.slice/ssh.service
                └─750 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 16 00:04:48 yash systemd[1]: Starting OpenBSD Secure Shell server ...
Oct 16 00:04:49 yash sshd[750]: Server listening on 0.0.0.0 port 22.
Oct 16 00:04:49 yash sshd[750]: Server listening on :: port 22.
Oct 16 00:04:49 yash systemd[1]: Started OpenBSD Secure Shell server.
yash@yash:~$ sudo systemctl stop ssh
yash@yash:~$ sudo systemctl restart ssh
yash@yash:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-16 01:15:30 IST; 2s ago
     Docs: man:sshd(8)
           man:sshd_config(5)
     Process: 3265 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
   Main PID: 3266 (sshd)
      Tasks: 1 (limit: 2515)
     Memory: 1.3M
        CGroup: /system.slice/ssh.service
                └─3266 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 16 01:15:30 yash systemd[1]: Starting OpenBSD Secure Shell server ...
Oct 16 01:15:30 yash sshd[3266]: Server listening on 0.0.0.0 port 22.
Oct 16 01:15:30 yash sshd[3266]: Server listening on :: port 22.
Oct 16 01:15:30 yash systemd[1]: Started OpenBSD Secure Shell server.
yash@yash:~$
```

4. To check all the running services the command is :

```
$ sudo service --status-all
```

```
yash@yash:~$ sudo service --status-all
[sudo] password for yash:      TRANSTBL
[ - ] apache-htcacheclean
[ - ] apache2
[ - ] apparmor
[ - ] atftpd
[ - ] avahi-daemon
[ + ] binfmt-support
[ - ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ - ] cryptdisks
[ - ] cryptdisks-early
[ + ] dbus
[ - ] dns2tcp AUTORUN-NF VBoxDarwin-Additions.pkg
[ - ] gdomap      NF      AdditionsUn
[ + ] haveged
[ - ] hwclock.sh
[ - ] inetsim
[ + ] inetutils-inetd
[ - ] iodined
[ - ] ipsec    autorun.sh VBoxLinuxA-
[ - ] isc-dhcp-server
[ - ] keyboard-setup.sh
[ + ] kmod
[ + ] lightdm
[ - ] miredo
[ + ] mysql    runasroot.sh VBoxSolaris-Additions.pkg
[ - ] named
[ + ] networking
[ - ] nfs-common
[ + ] nfs-kernel-server
[ - ] nginx
[ - ] nis
[ - ] nmbd
[ - ] ntp
[ - ] openvpn
[ - ] plymouth
[ + ] plymouth-log
[ - ] postfix
[ - ] postgresql
[ - ] pppd-dns
[ + ] procps
[ - ] ptunnel
[ - ] pulseaudio-enable-autospawn
[ - ] redsocks
[ + ] rpcbind runasroot.sh VBox
[ - ] rsync
[ + ] rsyslog
[ - ] rwhod
[ - ] samba-ad-dc
[ - ] saned
[ - ] screen-cleanup
[ - ] selinux-autorelabel
[ + ] smartmontools
[ - ] smbd
[ - ] snmpd
[ + ] ssh
[ - ] sslh
[ + ] stunnel4
[ - ] sudo
[ - ] sysstat
[ + ] udev
[ + ] ufw
[ - ] vsftpd
[ - ] x11-common
[ - ] xl2tpd
yash@yash:~$
```

PRACTICAL NO: 3

AIM: Configure NFS server to share directories or files on your network.

THEORY:

1. The Network File System (NFS) is a client/server application that lets a computer user view and optionally store and update files on a remote computer as though they were on the user's own computer.
2. The NFS protocol is one of several distributed file system standards for network-attached storage (NAS). NFS is kind of Google Drive.

There are 3 types of NFS:

- a. NFSv2: -
 - In these requests are based on per host basis and not on a per user.
 - Convenient for whole system.
 - TCP or UDP support.
 - File size limitation is less than 2GB.
- b. NFSv3: -
 - Bugs fixes of NFSv2, more features and more performance gain.
 - TCP or UDP support.
 - Support 2GB on server and more than 2GB on client.
 - Support is based on server file limit.
- c. NFSv4: -
 - Uses stateful protocol TCP or SCTP (Stream Control Transmission Protocol).
 - More security features and Kerberos.
 - Here, client authentication is per user basis and not per host basis.
 - RPC (Remote Procedure Calls).
 - It listens on port 2049.

COMMANDS & OUTPUT:

1. Install NFS server:

```
$ sudo apt-get -y install kernel-server  
$ sudo apt-get -y install nfs-kernel-server
```

```
yash@yash:~$ sudo apt-get install -y nfs-kernel-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
nfs-kernel-server is already the newest version (1:1.3.4-4).  
The following packages were automatically installed and are no longer  
needed:  
  galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl  
  libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic-neologd  
  mysql-community-server-core openjdk-8-jre python3.7 python3.7-minimal  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 195 not upgraded.
```

2. Command to enable a service:

```
$ sudo systemctl enable nfs-kernel-server
```

```
yash@yash:~$ sudo systemctl enable nfs-kernel-server
Synchronizing state of nfs-kernel-server.service with SysV service scri
Executing: /lib/systemd/systemd-sysv-install enable nfs-kernel-server
```

3. Now user needs to create a directory to show the function of NFS that is the mounting process:

```
$ mkdir demo_dir1
$ vi b.txt
$ ls b.txt
$ mv b.txt demo_dir1
$ ls /home/yash/demo_dir1
```

```
yash@yash:~$ mkdir demo_dir1
yash@yash:~$ vi b.txt
yash@yash:~$ ls b.txt
b.txt
yash@yash:~$ cat b.txt
Hii 6073YashPanchal :)
yash@yash:~$ ls /home/yash/demo_dir1
yash@yash:~$ ls /home/yash/demo_dir1
a.txt
yash@yash:~$
```

4. Now we have to mention our /directory_path ip_address (permissions), we need to give rw (read,write) permission,

```
$ sudo vi /etc/exports
$ cat /etc/exports
```

```
yash@yash:~$ sudo vi /etc/exports
yash@yash:~$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#           to NFS clients. See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes      hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_subtree_check)
#
# Example for NFSv4:
# /srv/nfs4      gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes  gss/krb5i(rw,sync,no_subtree_check)
#
/home/yash/demo_dir1 127.0.0.1(rw)
```

After mentioning the /directory_path ip_address (permissions), it should look like this.

/home/yash/demo_dir 127.0.0.1(rw)

5. Now to check the exports:

```
$ sudo exportfs -ra
```

```
yash@yash:~$ sudo exportfs -ra
exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtree_check' specified for export "127.0.0.1:/home/yash/demo_dir".
Assuming default behaviour ('no_subtree_check').
NOTE: this default has changed since nfs-utils version 1.0.x
```

6. To allow file sharing between systems on your network, install:

```
$ sudo apt-get -y install nfs-common
```

```
yash@yash:~$ sudo apt-get install -y nfs-common
Reading package lists... Done
Building dependency tree...
Reading state information... Done
nfs-common is already the newest version (1:1.3.4-4).
The following packages were automatically installed and are no longer required:
  galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl lib
  libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic mecab-
  mysql-community-server-core openjdk-8-jre python3.7 python3.7-minimal rsync
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 195 not upgraded.
yash@yash:~$
```

Note: Before mounting make sure that the user have made a directory to mount the file, if not the write the command to do so:

```
$ sudo mkdir /home/yash_directory
```

```
yash@yash:~$ sudo mkdir /home/yash_directory
```

7. Now to mount the file and then the user can check for the mounted file in the directory that user has made:

```
$ sudo mount -t nfs 127.0.0.1:/home/yash/demo_dir /home/yash_directory
$ ls
```

```
yash@yash:~$ sudo mount -t nfs 127.0.0.1:/home/yash/demo_dir /home/yash_directory
yash@yash:~$ cd /home/yash_directory
yash@yash:/home/yash_directory$ ls
b.txt
```

PRACTICAL NO: 4**AIM:** SSH Server - Password Authentication Configure SSH server.**THEORY:**

1. SSH Stands for - Secured Shell.
2. It is a method for secure remote login from one computer to another.
3. It provides several alternative options for strong authentication.
4. The connection can also be used for terminal access, file transfers & for tunnelling other applications.
5. It protects the communications security and integrity with strong encryption.
6. It is a secure alternative to the non-protected login protocols (such as telnet) and insecure file transfer methods (such as FTP).

COMMANDS & OUTPUT:

1. Install SSH server:

```
$ sudo apt-get -y install openssh-server
```

```
yash@yash:~$ sudo apt-get -y install openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  openssh-client openssh-sftp-server
Suggested packages:
  keychain libpam-ssh monkeysphere ssh-askpass molly-guard
The following packages will be upgraded:
  openssh-client openssh-server openssh-sftp-server
```

2. Now enable & check the status of SSH:

```
$ sudo systemctl enable ssh  
$ sudo systemctl status ssh
```

```
yash@yash:~$ sudo systemctl enable ssh  
Synchronizing state of ssh.service with SysV service script with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable ssh  
yash@yash:~$ sudo systemctl status ssh  
● ssh.service - OpenBSD Secure Shell server  
  Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabled)  
  Active: active (running) since Fri 2020-10-16 01:15:30 IST; 20min ago  
    Docs: man:sshd(8)  
          man:sshd_config(5)  
   Main PID: 3266 (sshd)  
     Tasks: 1 (limit: 2515)  
    Memory: 1.3M  
   CGroup: /system.slice/ssh.service  
           └─3266 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups  
  
Oct 16 01:15:30 yash systemd[1]: Starting OpenBSD Secure Shell server ...  
Oct 16 01:15:30 yash sshd[3266]: Server listening on 0.0.0.0 port 22.  
Oct 16 01:15:30 yash sshd[3266]: Server listening on :: port 22.  
Oct 16 01:15:30 yash systemd[1]: Started OpenBSD Secure Shell server.
```

Now if there is any configuration to be changed, then it can be changed from the sshd file.

Note: To be on the safe side, user can install ufw (Uncomplicated Firewall), ufw provides a user-friendly way to create an IPv4 or IPv6 host-based firewall.

To install ufw, the command is:

```
$ sudo apt-get -y install ufw
```

```
yash@yash:~$ sudo apt-get install ufw  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
ufw is already the newest version (0.36-7).  
The following packages were automatically installed and are no longer  
galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-p  
libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic  
mysql-community-server-core openjdk-8-jre python3.7 python3.7-minim  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 195 not upgraded.  
yash@yash:~$ █
```

3. Now To configure user's server to allow incoming SSH connections:

```
$ sudo ufw allow ssh
```

```
yash@yash:~$ sudo ufw allow ssh
Skipping adding existing rule
Skipping adding existing rule (v6)
```

```
$ sudo ufw allow ssh
```

```
yash@yash:~$ sudo ufw reload
Firewall reloaded
```

4. After this user can connect to the ssh localhost by the command:

```
$ ssh localhost      (OR)      $ ssh 127.0.0.1
```

```
yash@yash:~$ ssh localhost
yash@localhost's password:
Linux yash 5.5.0-kali2-amd64 #1 SMP Debian 5.5.17-1kali1 (2020-04-21) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
Last login: Mon Oct  5 17:15:24 2020 from 127.0.0.1
yash@yash:~$ ssh 127.0.0.1
yash@127.0.0.1's password:
Linux yash 5.5.0-kali2-amd64 #1 SMP Debian 5.5.17-1kali1 (2020-04-21) x86_64

The programs included with the Kali GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have mail.
Last login: Fri Oct 16 01:39:14 2020 from ::1
yash@yash:~$
```

After this user can enter the root password to connect to the localhost.

PRACTICAL NO: 5

AIM: Install Samba to share folders or files between Windows and Linux.

THEORY:

1. Samba shares Linux files and printers with Windows systems, and also gives Linux users access to files on Windows systems.
2. It allows Linux to work with the Windows operating system, as both a server and a client.

Several Samba packages are included with most of the Linux distributions:-

- Samba: Provides an SMB/Common Internet File System (CIFS) server that can be used to provide network services to SMB/CIFS clients.
- Samba-client: Provides some SMB/CIFS clients to complement the built-in SMB/CIFS file system in Linux. These clients allow access to SMB/CIFS shares and printing to SMB/CIFS printers.
- Samba-common: Provides files necessary for both the server and client Samba packages.

Types of Samba servers used while configuring Samba:-

- Stand-Alone Server.
- Domain Member Server.
- Domain Controller.

Samba Daemons and Services:-

- smbd: The server daemon that provides file-sharing and printing services to Windows clients. It is also responsible for user authentication, resource locking, and data sharing through the SMB protocol.
- nmbd: The NetBIOS nameserver daemon replies to name-service requests produced by SMB/CIFS in Windows-based systems. It also provides browsing support in the Windows Network Neighbourhood view.

COMMANDS & OUTPUT (Kali Linux Part):

1. First, we need to install Samba Server:

```
$ sudo apt-get -y install samba
```

```
yash@yash:~$ sudo apt-get -y install samba
[sudo] password for yash:
Reading package lists... Done
Building dependency tree
Reading state information... Done
samba is already the newest version (2:4.12.5+dfsg-3).
The following packages were automatically installed and are no longer required:
  galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl
  libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic_mecab
  mysql-community-server-core openjdk-8-jre python3.7 python3.7-minimal
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 195 not upgraded.
```

2. Then we need to start & check status of smbd:

```
$ sudo systemctl start smbd
$ sudo systemctl status smbd
```

```
yash@yash:~$ sudo systemctl start smbd
yash@yash:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-09-02 09:18:09 IST; 17s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 1764 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 1768 (smbd)
   Status: "smbd: ready to serve connections ..."
    Tasks: 4 (limit: 2520)
   Memory: 25.9M
      CPU: 0.000 CPU(s) since started
      CGroup: /system.slice/smbd.service
              └─1768 /usr/sbin/smbd --foreground --no-process-group

Sep 02 09:18:07 yash systemd[1]: Starting Samba SMB Daemon ...
Sep 02 09:18:09 yash systemd[1]: Started Samba SMB Daemon.
```

3. Then we need to start & check status of nmbd:

```
$ sudo systemctl start nmbd
$ sudo systemctl status nmbd
```

```
yash@yash:~$ sudo systemctl start nmbd
yash@yash:~$ sudo systemctl status nmbd
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/lib/systemd/system/nmbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-16 01:44:40 IST; 3s ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 4449 (nmbd)
   Status: "nmbd: ready to serve connections ..."
    Tasks: 1 (limit: 2515)
   Memory: 11.3M
      CPU: 0.000 CPU(s) since started
      CGroup: /system.slice/nmbd.service
              └─4449 /usr/sbin/nmbd --foreground --no-process-group

Oct 16 01:44:39 yash systemd[1]: Starting Samba NMB Daemon ...
Oct 16 01:44:40 yash nmbd[4449]: [2020/10/16 01:44:40.329794,  0] ../../lib/util/become_daemon.c:135(daemon_ready)
Oct 16 01:44:40 yash nmbd[4449]:   daemon_ready: daemon 'nmbd' finished starting up and ready to serve connections
Oct 16 01:44:40 yash systemd[1]: Started Samba NMB Daemon.
```

4. First, we make a samba_shared directory in /home/yash

Then we make few txt files in /home/yash/samba_shared

```
$ sudo mkdir samba_shared
$ touch a.txt b.txt c.txt
```

```
yash@yash:/etc/samba$ ls /home/yash/samba_shared
a.txt b.txt c.txt
```

5. Now,

```
$ cd /etc/samba
$ ls /etc/samba
```

```
yash@yash:~$ cd /etc/sa,ba
bash: cd: /etc/sa,ba: No such file or directory
yash@yash:~$ cd /etc/samba
yash@yash:/etc/samba$ ls
gdbcommands  smb.conf  tls
```

6. Now take the backup of smb.conf

```
$ sudo cp smb.conf smb.conf_bkp
$ cat smb.conf_bkp
```

```
yash@yash:/etc/samba$ sudo cp smb.conf smb.conf_bkp
yash@yash:/etc/samba$ cat smb.conf_bkp
#
# Sample configuration file for the Samba suite for Debian GNU/Linux.
#
#
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options most of which
# are not shown in this example
#
# Some options that are often worth tuning have been included as
# commented-out examples in this file.
# - When such options are commented with ";", the proposed setting
#   differs from the default Samba behaviour
# - When commented with "#", the proposed setting is the default
#   behaviour of Samba but the option is considered important
#   enough to be mentioned here
#
# NOTE: Whenever you modify this file you should run the command
# "testparm" to check that you have not made any basic syntactic
# errors.

===== Global Settings =====

[global]
```

7. Now we need to do few configuration in /etc/smb.conf

Then delete all the things inside smb.conf and then write:

```
[samba_shared]
comment = Welcome to samba
path = /home/yash/samba_shared
browseable = yes
read only = yes
guest ok = no
```

```
$ sudo nano smb.conf  
$ sudo cat smb.conf
```

```
yash@yash:/etc/samba$ sudo nano smb.conf  
yash@yash:/etc/samba$ sudo cat smb.conf  
[samba_share]  
comment = welcome to samba  
path = /home/yash/samba_shared  
browseable = yes  
read only = yes  
yash@yash:/etc/samba$ █
```

8. Now download smbclient:

```
$ sudo apt-get -y install smbclient
```

```
yash@yash:/etc/samba$ sudo apt-get -y install smbclient  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
smbclient is already the newest version (2:4.12.5+dfsg-3).  
The following packages were automatically installed and are no longer required:  
galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libglib2.0-bin li  
libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic mecab-ipadic-utf8 me  
mysql-community-server-core openjdk-8-jre python3.7 python3.7-minimal rsync ruby-did-you  
Use 'sudo apt autoremove' to remove them.  
0 upgraded, 0 newly installed, 0 to remove and 195 not upgraded.  
yash@yash:/etc/samba$ █
```

9. Now restart services of both smbd and nmbd & check their status too

```
$ sudo systemctl restart smbd  
$ sudo systemctl restart nmbd
```

```
yash@yash:/etc/samba$ sudo systemctl restart smbd
yash@yash:/etc/samba$ sudo systemctl restart nmbd
yash@yash:/etc/samba$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-16 01:53:26 IST; 18s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
   Process: 4585 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 4589 (smbd)
   Status: "smbd: ready to serve connections ..."
    Tasks: 4 (limit: 2515)
   Memory: 17.3M
      CGroup: /system.slice/smbd.service
              ├─4589 /usr/sbin/smbd --foreground --no-process-group
              ├─4592 /usr/sbin/smbd --foreground --no-process-group
              ├─4593 /usr/sbin/smbd --foreground --no-process-group
              └─4594 /usr/sbin/smbd --foreground --no-process-group

Oct 16 01:53:25 yash systemd[1]: Starting Samba SMB Daemon ...
Oct 16 01:53:26 yash smbd[4589]: [2020/10/16 01:53:26.990889,  0] ..../lib/util/become_daemon.c:135(daemon_ready)
Oct 16 01:53:26 yash smbd[4589]:  daemon_ready: daemon 'smbd' finished starting up and ready to serve connections
Oct 16 01:53:26 yash systemd[1]: Started Samba SMB Daemon.
yash@yash:/etc/samba$ sudo systemctl status nmbd
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/lib/systemd/system/nmbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-10-16 01:53:34 IST; 19s ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 4601 (nmbd)
   Status: "nmbd: ready to serve connections ..."
    Tasks: 1 (limit: 2515)
   Memory: 2.3M
      CGroup: /system.slice/nmbd.service
              └─4601 /usr/sbin/nmbd --foreground --no-process-group

Oct 16 01:53:34 yash systemd[1]: Starting Samba NMB Daemon ...
Oct 16 01:53:34 yash nmbd[4601]: [2020/10/16 01:53:34.159855,  0] ..../lib/util/become_daemon.c:135(daemon_ready)
Oct 16 01:53:34 yash nmbd[4601]:  daemon_ready: daemon 'nmbd' finished starting up and ready to serve connections
Oct 16 01:53:34 yash systemd[1]: Started Samba NMB Daemon.
yash@yash:/etc/samba$
```

10. First do:

```
$ whoami
```

```
yash@yash:/etc/samba$ whoami
yash
yash@yash:/etc/samba$
```

11. And then:

```
$ sudo pdbedit -a -u $(whoami)
```

```
yash@yash:/etc/samba$ sudo pdbedit -a -u $(whoami)
new password:
retype new password:
Unix username:      yash
NT username:
Account Flags: [U-A-D-B-]
User SID:          S-1-5-21-1561718998-541519598-3414635832-1000
Primary Group SID: S-1-5-21-1561718998-541519598-3414635832-513
Full Name:         yash panchal
Home Directory:   \\yash\yash
HomeDir Drive:
Logon Script:
Profile Path:     \\yash\yash\profile
Domain:           YASH
Account desc:
Workstations:
Munged dial:
Logon time:       0
Logoff time:      Wed, 06 Feb 2036 20:36:39 IST
Kickoff time:     Wed, 06 Feb 2036 20:36:39 IST
Password last set: Fri, 16 Oct 2020 01:55:58 IST
Password can change: Fri, 16 Oct 2020 01:55:58 IST
Password must change: never
Last bad password : 0
Bad password count : 0
Logon hours       : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
yash@yash:/etc/samba$
```

12. Type and retype your password:

\$ sudo pdbedit -L (to check all users)

```
yash@yash:/etc/samba$ sudo pdbedit -L
yash:1000:yash panchal
YASHP:1003:
```

13. To check for your ip:

\$ ip addr

```
yash@yash:/etc/samba$ sudo ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:aa:3a:41 brd ff:ff:ff:ff:ff:ff
        inet 192.168.106.128/24 brd 192.168.106.255 scope global noprefixroute eth0
            valid_lft forever preferred_lft forever
        inet6 fe80::a00:27ff:feaa:3a41/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

14. This will make any connection possible to connect:

```
$ sudo ufw allow from (your ip addr) any app Samba
```

In my case it was:

```
$ sudo ufw allow from 192.168.56.0/24 to any app Samba
```

```
yash@yash:/etc/samba$ sudo ufw allow from 192.168.56.0/24 to any app Samba
Skipping adding existing rule
```

15. Now to reload your firewall use:

```
$ sudo ufw reload
```

```
yash@yash:/etc/samba$ sudo ufw reload
Firewall reloaded
```

16. To check whether it's working or not, use:

```
$ sudo smbclient -U <username> -L //your ip
```

In my case it was:

```
$ sudo smbclient -U yash -L //192.168.106.128
```

```
yash@yash:/etc/samba$ sudo smbclient -U yash -L //192.168.106.128
Enter WORKGROUP\yash's password:
```

Sharename	Type	Comment
samba_share	Disk	welcome to samba
IPC\$	IPC	IPC Service (Samba 4.12.5-Debian)
documnets	Disk	sharefolder
SMB1 disabled -- no workgroup available		

17. Now restart both smbd and nmbd & check their status too

```
$ sudo systemctl restart smbd
```

```
$ sudo systemctl restart nmbd
```

```
$ sudo systemctl status smbd
```

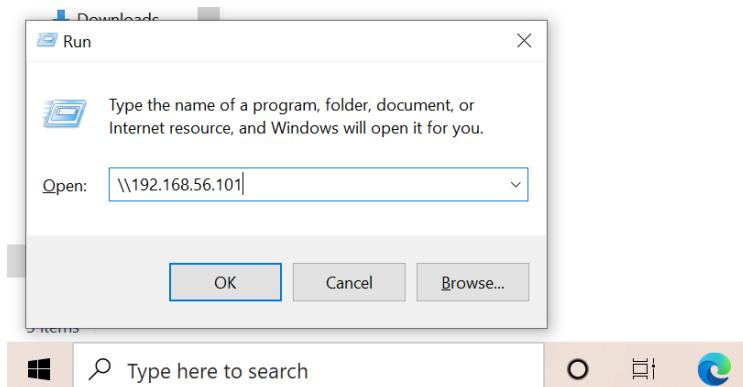
```
$ sudo systemctl status nmbd
```

```
yash@yash:/etc/samba$ sudo systemctl restart smbd
yash@yash:/etc/samba$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; disabled; vendor pr
   Active: active (running) since Fri 2020-10-16 02:07:27 IST; 14s ago
     Docs: man:smbd(8) man:/usr/share/man/man8/smbd.8.gz
           man:samba(7) man:/usr/share/doc/samba-4.1.15/Debian/man/samba.7.gz
           man:smb.conf(5)
   Process: 5785 ExecStartPre=/usr/share/samba/update-apparmor-samba-profil
 Main PID: 5789 (smbd)
   Status: "smbd: ready to serve connections ... "
    Tasks: 4 (limit: 2515)
   Memory: 7.5M
      CGroup: /system.slice/smbd.service
```

```
yash@yash:/etc/samba$ sudo systemctl restart nmbd
yash@yash:/etc/samba$ sudo systemctl status nmbd
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/lib/systemd/system/nmbd.service; disabled; vendor present)
   Active: active (running) since Fri 2020-10-16 02:08:20 IST; 3s ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
 Main PID: 5815 (nmbd)
   Status: "nmbd: ready to serve connections ... "
    Tasks: 1 (limit: 2515)
   Memory: 2.3M
      CGroup: /system.slice/nmbd.service
              └─5815 /usr/sbin/nmbd --foreground --no-process-group
```

Windows OS Part:

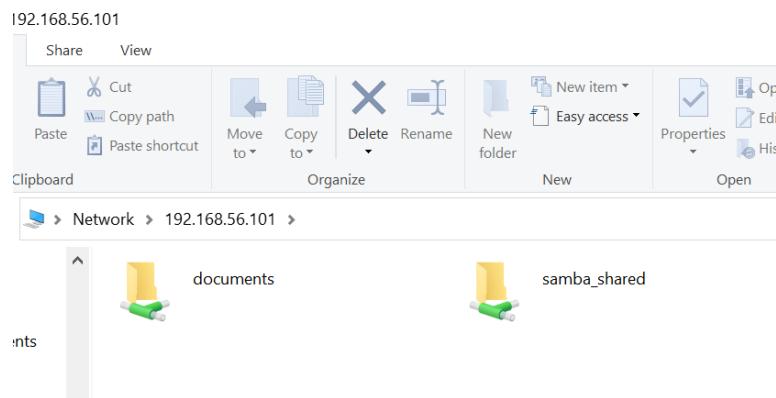
- Once smbd and nmbd services are active now switch back to your windows OS and press windows + R key on keyboard and now a pop up box will open and their write smbclient ip address, here in my case it was //192.168.56.101 and press OK.



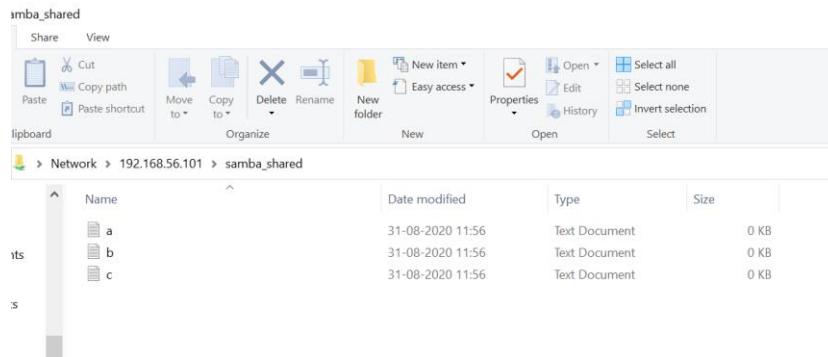
- Now again one more pop up box will open, write the credentials of your Windows or Kali or whatever it would be.



- Now in top of the address bar you can see Network > 192.168.56.101 (your ip address) in this you can see samba_shared and documents folder.



4. In samba_shared server you can see all the txt files that we've made earlier, using touch command.



- Once all your samba_shared files are visible in windows, then the sharing of files/folders have been successfully done between Windows and Kali Linux.

PRACTICAL NO: 6

AIM: Configure NTP, Install and Configure NTP Server, Configure NTP Client.

THEORY:

1. NTP- (Port number - 123)
 - NTP stands for Network Time Protocol, and it is an Internet protocol used to synchronize the clocks of computers to sometime reference.
 - NTP is an Internet standard protocol.
 - NTP time servers work within the TCP/IP suite and rely on User Datagram Protocol (UDP) port 123.
2. SNTP-
 - SNTP (Simple Network Time Protocol) is basically also NTP,
 - As a full implementation of the NTP protocol seemed too complicated for many systems, a simplified version of the protocol, namely SNTP had been defined.
3. FEATURES:
 - a. NTP needs some reference clock that defines the true time to operate. All clocks are set towards that true time.
 - b. NTP uses UTC as reference time.
 - c. NTP is a fault-tolerant protocol that will automatically select the best of several available time sources to synchronize to.
 - d. NTP is highly scalable: - A synchronization network may consist of several reference clocks.
Each node of such a network can exchange time information either bidirectional or unidirectional.
 - e. Having available several time sources, NTP can select the best candidates to build its estimate of the current time.
The protocol is highly accurate, using a resolution of less than a nanosecond (about 2^{32} seconds).

COMMANDS & OUTPUT:

1. We need update:

```
$ sudo apt update -y
$ uname -a
```

```
yash@yash:~$ sudo apt update -y
[sudo] password for yash:
Hit:1 http://repo.mysql.com/apt/debian wheezy InRelease
Get:2 http://ftp.harukasan.org/kali kali-rolling InRelease [30.5 kB]
Get:3 http://ftp.harukasan.org/kali kali-rolling/main Sources [13.2 MB]
Get:4 http://ftp.harukasan.org/kali kali-rolling/non-free Sources [125 kB]
Get:5 http://ftp.harukasan.org/kali kali-rolling/contrib Sources [61.2 kB]
Get:6 http://ftp.harukasan.org/kali kali-rolling/main amd64 Packages [16.6 MB]
Get:7 http://ftp.harukasan.org/kali kali-rolling/non-free amd64 Packages [200 kB]
Get:8 http://ftp.harukasan.org/kali kali-rolling/contrib amd64 Packages [99.7 kB]
Fetched 30.3 MB in 1min 39s (307 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
992 packages can be upgraded. Run 'apt list --upgradable' to see them.
yash@yash:~$ uname -a
Linux yash 5.5.0-kali2-amd64 #1 SMP Debian 5.5.17-1kali1 (2020-04-21) x86_64 GNU/Linux
```

2. Command to install NTP:

```
$ sudo apt install ntp
```

```
yash@yash:~$ sudo apt install ntp
Reading package lists... Done
Building dependency tree
Reading state information... Done
ntp is already the newest version (1:4.2.8p14+dfsg-2).
ntp set to manually installed.
The following packages were automatically installed and are no longer required:
  galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libhtml-template-
  mecab-utils mysql-client mysql-community-client mysql-community-client-core mysql-commun
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 992 not upgraded.
```

3. Command to install SNTP:

```
$ sudo apt install sntp
```

```
yash@yash:~$ sudo apt install sntp
Reading package lists... Done
Building dependency tree
Reading state information... Done
sntp is already the newest version (1:4.2.8p15+dfsg-1).
The following packages were automatically installed and are no longer required:
  galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libglib2.0-bin libhtml-temp
  libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils my
  mysql-community-server-core openjdk-8-jre python3.7 python3.7-minimal rsync ruby-did-you-mean
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 195 not upgraded.
yash@yash:~$
```

4. Now:

```
$ sudo ntp --version
$ cd /etc
$ ls -lrth *ntp.conf*
```

```
yash@yash:~$ sudo ntp --version
ntp 4.2.8p14@1.3728-o Tue Mar 10 07:38:28 UTC 2020 (1)
yash@yash:~$ cd /etc
yash@yash:/etc$ ls -lrth *ntp*
-rw-r--r-- 1 root root 2.2K Mar 10 2020 ntp.conf
```

5. Then start & check status of NTP:

```
$ sudo systemctl start ntp
$ sudo systemctl status ntp
```

```
yash@yash:/etc$ sudo systemctl start ntp
yash@yash:/etc$ sudo systemctl status ntp
● ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; disabled; vendor preset: disabled)
  Active: active (running) since Wed 2020-10-07 09:29:31 IST; 7s ago
    Docs: man:ntpd(8)
   Process: 3108 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 3114 (ntpd)
    Tasks: 2 (limit: 2520)
   Memory: 2.3M
      CGroup: /system.slice/ntp.service
              └─3114 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 107:112

Oct 07 09:29:31 yash ntpd[3114]: Listen normally on 5 eth0 [fe80::a00:27ff:feaa:3a41%2]:123
Oct 07 09:29:31 yash ntpd[3114]: Listening on routing socket on fd #22 for interface updates
Oct 07 09:29:31 yash ntpd[3114]: kernel reports TIME_ERROR: 0x41: Clock Unsynchronized
Oct 07 09:29:31 yash ntpd[3114]: kernel reports TIME_ERROR: 0x41: Clock Unsynchronized
Oct 07 09:29:31 yash systemd[1]: Started Network Time Service.
Oct 07 09:29:37 yash ntpd[3114]: Soliciting pool server 103.134.252.11
Oct 07 09:29:37 yash ntpd[3114]: Soliciting pool server 162.159.200.1
Oct 07 09:29:38 yash ntpd[3114]: Soliciting pool server 162.159.200.123
Oct 07 09:29:38 yash ntpd[3114]: Soliciting pool server 172.105.40.191
Oct 07 09:29:38 yash ntpd[3114]: Soliciting pool server 5.189.141.35
```

6. Now we need to configure NTP:

```
$ sudo vim /etc/ntp.conf
```

```
yash@yash:/etc$ sudo vim /etc/ntp.conf
```

- After opening **ntp.conf** file scroll down and see-

You need to talk to an NTP server or two (or three)

- Below this line write:

server localhost

```
# /etc/ntp.conf, configuration for ntpd; see ntp.conf(5) for help

driftfile /var/lib/ntp/ntp.drift

# Leap seconds definition provided by tzdata
leapfile /usr/share/zoneinfo/leap-seconds.list

# Enable this if you want statistics to be logged.
#statsdir /var/log/ntpstats/

statistics loopstats peerstats clockstats
filegen loopstats file loopstats type day enable
filegen peerstats file peerstats type day enable
filegen clockstats file clockstats type day enable

# You do need to talk to an NTP server or two (or three).
server localhost

# pool.ntp.org maps to about 1000 low-stratum NTP servers. Your server will
# pick a different set every time it starts up. Please consider joining the
# pool: <http://www.pool.ntp.org/join.html>
pool 0.debian.pool.ntp.org iburst
pool 1.debian.pool.ntp.org iburst
pool 2.debian.pool.ntp.org iburst
pool 3.debian.pool.ntp.org iburst
```

7. Then restart & check status of NTP:

```
$ sudo systemctl restart ntp
$ sudo systemctl status ntp
```

```
yash@yash:/etc$ sudo systemctl restart ntp
yash@yash:/etc$ sudo systemctl status ntp
● ntp.service - Network Time Service
  Loaded: loaded (/lib/systemd/system/ntp.service; disabled; vendor preset: disabled)
  Active: active (running) since Wed 2020-10-07 09:42:18 IST; 4s ago
    Docs: man:ntpd(8)
  Process: 3404 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 3410 (ntpd)
   Tasks: 2 (limit: 2520)
  Memory: 1.3M
    CGroup: /system.slice/ntp.service
            └─3410 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 107:112

Oct 07 09:42:18 yash ntpd[3410]: Listen and drop on 0 v6wildcard [::]:123
Oct 07 09:42:18 yash ntpd[3410]: Listen and drop on 1 v4wildcard 0.0.0.0:123
Oct 07 09:42:18 yash ntpd[3410]: Listen normally on 2 lo 127.0.0.1:123
Oct 07 09:42:18 yash ntpd[3410]: Listen normally on 3 eth0 10.0.2.15:123
Oct 07 09:42:18 yash ntpd[3410]: Listen normally on 4 lo [::1]:123
Oct 07 09:42:18 yash ntpd[3410]: Listen normally on 5 eth0 [fe80::a00:27ff:fea:3a41%2]:123
Oct 07 09:42:18 yash ntpd[3410]: Listening on routing socket on fd #22 for interface updates
Oct 07 09:42:18 yash ntpd[3410]: kernel reports TIME_ERROR: 0x41: Clock Unsynchronized
Oct 07 09:42:18 yash ntpd[3410]: kernel reports TIME_ERROR: 0x41: Clock Unsynchronized
Oct 07 09:42:18 yash systemd[1]: Started Network Time Service.
```

8. Now to query NTP and check its connection:

```
$ sudo ntpq -p
```

remote	refid	st	t	when	poll	reach	delay	offset	jitter
<hr/>									
0.debian.pool.n	.POOL.	16	p	-	64	0	0.000	+0.000	0.000
1.debian.pool.n	.POOL.	16	p	-	64	0	0.000	+0.000	0.000
2.debian.pool.n	.POOL.	16	p	-	64	0	0.000	+0.000	0.000
3.debian.pool.n	.POOL.	16	p	-	64	0	0.000	+0.000	0.000
localhost	.INIT.	16	u	-	64	0	0.000	+0.000	0.000
-ntp.in.eria.one	14.139.60.102	2	u	13	64	1	7.563	-25.115	5.451
#ns2.newsnet.li	251.4.108.178	3	u	17	64	1	210.194	-73.555	4.537
+time.cloudflare	10.35.14.16	3	u	16	64	1	58.387	-27.563	8.829
-5.189.141.35	(m 17.253.54.123	2	u	15	64	1	188.216	-4.822	3.296
#server.antechne	217.31.202.100	2	u	13	64	1	163.387	-29.353	9.163
-103.134.252.11	104.211.76.226	2	u	15	64	1	40.160	-26.140	6.807
+ec2-13-126-27-1	169.254.169.123	4	u	20	64	1	4.579	-28.070	6.294
-139.59.15.185	179.43.76.147	2	u	21	64	1	27.181	-19.362	3.407
-vpn.oe9hamnet.a	.PPS.	1	u	19	64	1	151.245	-32.959	4.747
-45.86.70.11	173.212.222.171	2	u	21	64	1	268.129	-15.343	1.815
*139.59.55.93	17.253.82.253	2	u	20	64	1	23.703	-27.284	7.500
-43.240.66.74	103.134.252.11	3	u	20	64	1	29.738	-10.711	7.555
#sv1.ggsrv.de	205.46.178.169	2	u	17	64	1	138.739	-35.905	4.125

PRACTICAL NO: 7**AIM:** Install MySQL to configure database server.**THEORY:**

- a. MySQL 8.0 is the latest stable release of MySQL relational database management system.
- b. MySQL is a free to use Database Management System (RDBMS) that uses Structured Query Language (SQL).
- c. MySQL is designed to be stable, reliable, and flexible to use.
- d. Oracle provides Debian packages for installing MySQL on Debian or Debian-like Linux systems. The packages are available through two different channels:
 - i. The MySQL APT Repository, supporting Debian and Ubuntu platforms. "Installing MySQL on Linux Using the MySQL APT Repository".
 - ii. The MySQL Developer Zone's Download Area.

COMMANDS & OUTPUT:

1. We need update:

```
$ sudo apt update
```

```
yash@yash:~$ sudo apt update
Hit:1 http://repo.mysql.com/apt/debian jessie InRelease
Hit:2 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
263 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2. Then install wget:

```
$ sudo apt install -y wget
```

```
yash@yash:~$ sudo apt install -y wget
Reading package lists... Done
Building dependency tree
Reading state information... Done
wget is already the newest version (1.20.3-1+b3).
0 upgraded, 0 newly installed, 0 to remove and 263 not upgraded.
```

3. Then we need to wget the MySQL deb files:

```
$ sudo apt install -y wget https://dev.mysql.com/get/mysql-apt-config\_0.8.15-1\_all.deb
```

```
yash@yash:~$ sudo wget https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb
--2020-10-17 16:30:11-- https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb
Resolving dev.mysql.com (dev.mysql.com) ... 137.254.60.11
Connecting to dev.mysql.com (dev.mysql.com)|137.254.60.11|:443 ... connected.
HTTP request sent, awaiting response ... 302 Found
Location: https://repo.mysql.com//mysql-apt-config_0.8.15-1_all.deb [following]
--2020-10-17 16:30:22-- https://repo.mysql.com//mysql-apt-config_0.8.15-1_all.deb
Resolving repo.mysql.com (repo.mysql.com) ... 184.51.94.120
Connecting to repo.mysql.com (repo.mysql.com)|184.51.94.120|:443 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 35532 (35K) [application/x-debian-package]
Saving to: 'mysql-apt-config_0.8.15-1_all.deb.6'

mysql-apt-config_0.8.15-1_all.deb.6          100%[=====] 2020-10-17 16:30:28 (1.63 MB/s) - 'mysql-apt-config_0.8.15-1_all.deb.6' saved [35532/35532]
```

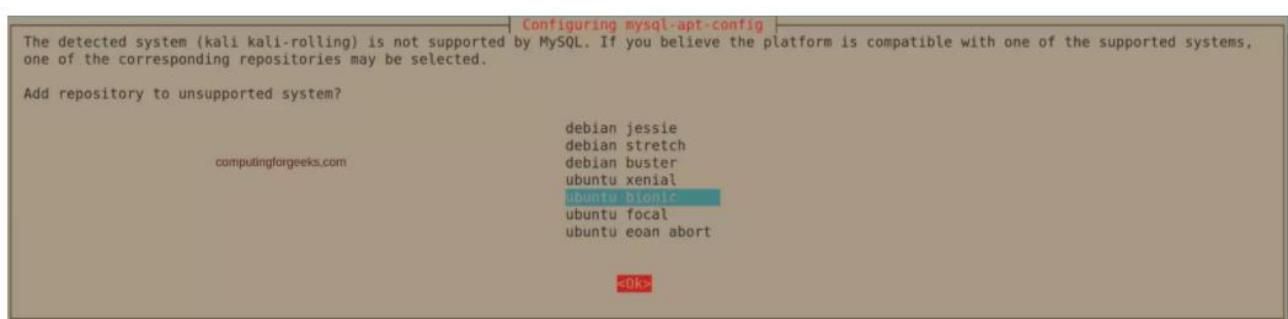
4. Install the deb files:

```
$ sudo dpkg -i mysql-apt-config_0.8.15-1_all.deb
```

```
yash@yash:~$ sudo dpkg -i mysql-apt-config_0.8.15-1_all.deb
(Reading database ... 253650 files and directories currently installed.)
Preparing to unpack mysql-apt-config_0.8.15-1_all.deb ...
Unpacking mysql-apt-config (0.8.15-1) over (0.8.15-1) ...
Setting up mysql-apt-config (0.8.15-1) ...
Warning: apt-key should not be used in scripts (called from postinst maintainer script of the package mysql-apt-config).
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
```

As Kali Linux is not officially supported version, choose the Ubuntu Bionic release.

Select <OK> and press <Enter> key to confirm version installation.



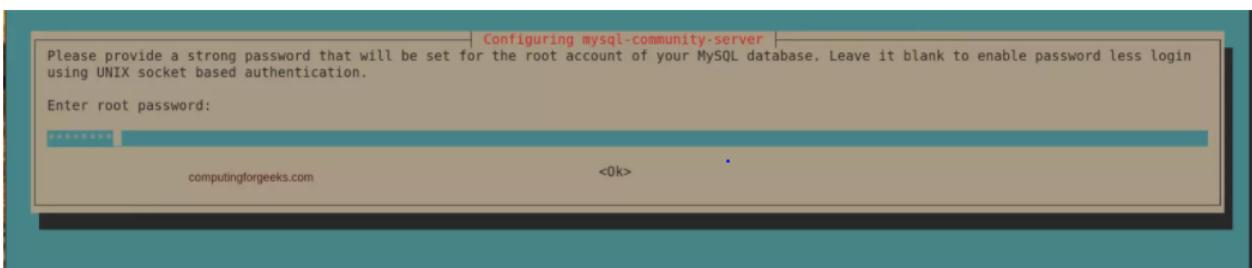
5. We need to & update install mysql-community-server:

```
$ sudo apt update  
$ sudo apt install mysql-community-server
```

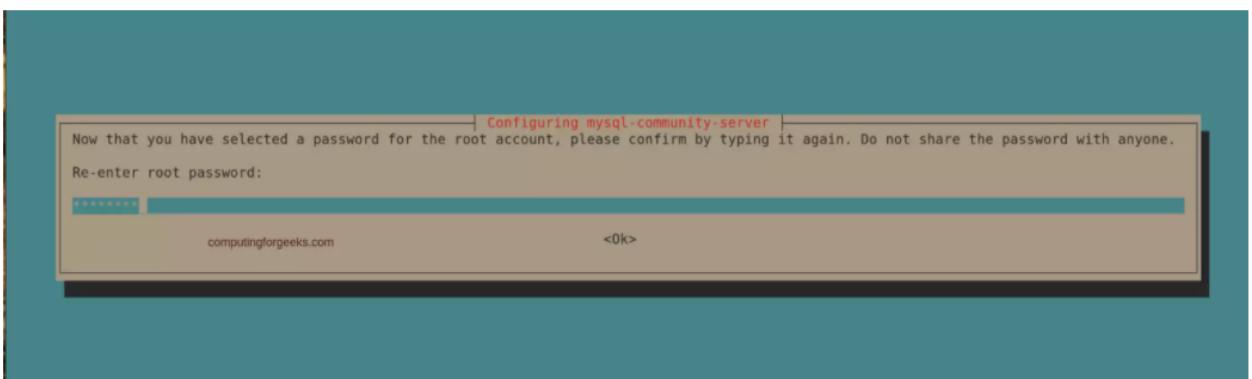
```
yash@yash:~$ sudo apt update  
Hit:1 http://repo.mysql.com/apt/debian jessie InRelease  
Get:2 http://repo.mysql.com/apt/debian jessie/mysql-5.7 Sources [949 B]  
Get:3 http://repo.mysql.com/apt/debian jessie/mysql-5.7 amd64 Packages [3,384 B]  
Hit:4 http://kali.download/kali kali-rolling InRelease  
Fetched 4,333 B in 16s (264 B/s)  
Reading package lists ... Done  
Building dependency tree  
Reading state information ... Done  
263 packages can be upgraded. Run 'apt list --upgradable' to see them.  
yash@yash:~$ sudo apt install mysql-community-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libaio1 libmecab2 mysql-client mysql-common mysql-community-client  
The following NEW packages will be installed:  
  libaio1 libmecab2 mysql-client mysql-common mysql-community-client mysql-community-server  
0 upgraded, 6 newly installed, 0 to remove and 263 not upgraded.  
Need to get 43.5 MB/43.5 MB of archives.  
After this operation, 289 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://kali.download/kali kali-rolling/main amd64 libmecab2 amd64 0.996-14+b1 [221 kB]  
Get:2 http://repo.mysql.com/apt/debian jessie/mysql-5.7 amd64 mysql-community-client amd64 5.7.30-1debian8 [11.8 MB]  
Get:3 http://repo.mysql.com/apt/debian jessie/mysql-5.7 amd64 mysql-client amd64 5.7.30-1debian8 [72.4 kB]  
Get:4 http://repo.mysql.com/apt/debian jessie/mysql-5.7 amd64 mysql-community-server amd64 5.7.30-1debian8 [31.4 MB]  
Fetched 43.5 MB in 1min 21s (540 kB/s)  
Preconfiguring packages ...
```

Accept license agreement in the next screens to begin installation.

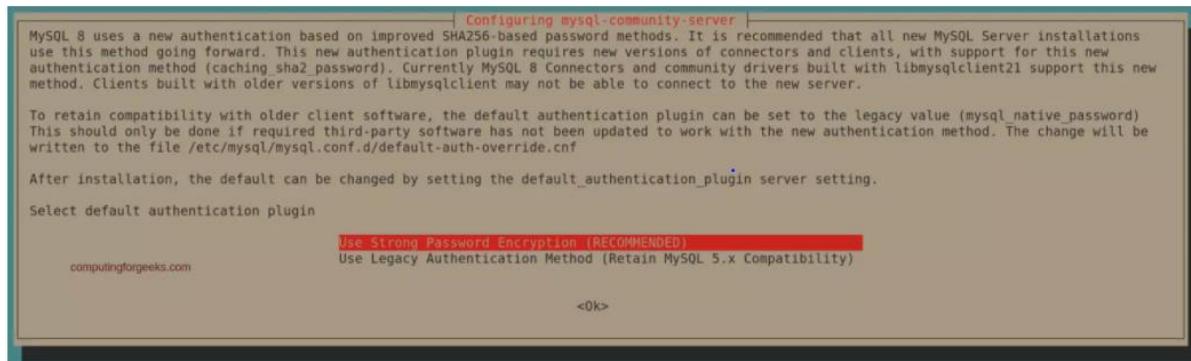
Set root password for your MySQL database server.



Confirm your root password.



Select the default authentication plugin.



6. Now enable & check MySQL status:

```
$ sudo systemctl enable --now mysql  
$ sudo systemctl start --now mysql  
$ sudo systemctl status mysql
```

```
yash@yash:~$ sudo systemctl enable --now mysql  
Synchronizing state of mysql.service with /lib/systemd/systemd-sysv-install.  
Executing: /lib/systemd/systemd-sysv-install enable mysql  
Created symlink /etc/systemd/system/multi-user.target.wants/mysql.service → /lib/systemd/system/mysql.service.  
yash@yash:~$ sudo systemctl start mysql  
yash@yash:~$ sudo systemctl status mysql  
● mysql.service - MySQL Community Server  
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: disabled)  
   Active: active (running) since Sat 2020-10-17 16:38:00 IST; 28s ago  
     Process: 5130 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)  
     Process: 5165 ExecStart=/usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid (code=exited, status=0/SUCCESS)  
   Main PID: 5167 (mysqld)  
     Tasks: 27 (limit: 2515)  
    Memory: 179.9M  
      CGroup: /system.slice/mysql.service  
              └─5167 /usr/sbin/mysqld --daemonize --pid-file=/var/run/mysqld/mysqld.pid  
  
Oct 17 16:38:00 yash systemd[1]: Starting MySQL Community Server ...  
Oct 17 16:38:00 yash systemd[1]: Started MySQL Community Server.
```

7. To test if the database server is working fine by creating a test database:

```
$ sudo mysql -u root -p
```

```
yash@yash:~$ sudo mysql -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.30 MySQL Community Server (GPL)

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.
```

```
Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.
```

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> create database yashp;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| sys |
| yashp |
+-----+
5 rows in set (0.00 sec)
```

```
mysql> exit
Bye
```

PRACTICAL NO: 8

AIM: Configure NIS in order to share user accounts in your local network.

THEORY:

- NIS stands for Network Information Systems.
- NIS, or Network Information Systems, is a network service that allows authentication and login information to be stored on a centrally located server.
- This includes the username and password database for login authentication, database of user groups, and the locations of home directories.
- NIS-sharing flat files (sharing resources).
- Files stored on certain machines, kind of database, can be queried, in tabular form, at least one column should be there.
- It is kind of kind of database in which each table has unique key, one or more column.
- And with query we need to identify it.
- There are 2 types:
 1. Master – Main Server
 2. Slave – Secondary server, load balancing

If master server goes down then the Slave can take over Master server.

- Client can join without server acceptance
- NIS daemons - NIS use client server architecture with one master server (and possibly one or several secondary slave servers, necessary for reliability) and many clients. NIS uses several daemons on the server and on all clients to enable the NIS system. The daemons are:

- I. The **ypserv** daemon:
 - a. Runs on master and slave servers
 - b. Answers ypbind requests from clients
 - c. Responds to client information requests
- II. The **ypbind** daemon runs on all NIS client systems:
 - a. Makes initial client-to-server binding requests
 - b. Stores binding information in the /var/yp/binding/domain directory
 - c. Rebinds to another server if the connection is lost with the initial server
 - d. Requests NIS map information at the library-call level
- III. The **ypasswd** daemon:
 - a. Allows users to change their passwords
 - b. Updates the passwd and shadow files on the master server
 - c. Updates the NIS password map

d. Provides or “pushes” the NIS password map to all slave servers

IV. The **ypxfrd** daemon:

- a. Runs on the NIS master server only
- b. Responds to requests, generated in the slave servers by using the ypxfr command to pull the maps from the master
- c. Transfers NIS maps at a high speed

V. The **rpc.ypupdated** daemon:

- a. runs on the NIS master server only
- b. Updates NIS maps using the configuration stored in the /var/yp/updaters file

COMMANDS & OUTPUT:

1. First update kali:

```
$ sudo apt-get update
```

```
yash@yash:~$ sudo apt-get update
[sudo] password for yash:
Hit:2 http://repo.mysql.com/apt/debian jessie InRelease
Get:1 http://kali.download/kali kali-rolling InRelease [30.5 kB]
Get:3 http://kali.download/kali kali-rolling/main Sources [13.2 MB]
Get:4 http://kali.download/kali kali-rolling/non-free Sources [125 kB]
Get:5 http://kali.download/kali kali-rolling/contrib Sources [61.8 kB]
Get:6 http://kali.download/kali kali-rolling/main amd64 Packages [16.6 MB]
Get:7 http://kali.download/kali kali-rolling/non-free amd64 Packages [200 kB]
Get:8 http://kali.download/kali kali-rolling/contrib amd64 Packages [100 kB]
Fetched 30.4 MB in 1min 16s (397 kB/s)
```

```
$ sudo apt update
```

```
yash@yash:~$ sudo apt update
Hit:1 http://kali.download/kali kali-rolling InRelease
Hit:2 http://repo.mysql.com/apt/debian jessie InRelease
Reading package lists... Done
Building dependency tree
Reading state information... Done
195 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2. Now install NIS:

```
$ sudo apt install nis
```

```
yash@yash:~$ sudo apt install nis
Reading package lists... Done
Building dependency tree
Reading state information... Done
nis is already the newest version (3.17.1-5).
The following packages were automatically installed and are no longer
needed:
  galera-3 libaio1 libconfig-inifiles-perl libdbd-mysql-perl lib
  libpython3.7-stdlib libreadline5 libterm-readkey-perl mecab-ipadic-utf8
  mysql-community-server-core openjdk-8-jre python3.7 python3.7-
  python3.7-minimal
Use 'sudo apt autoremove' to remove them.
```

3. Now user must configure NIS:

```
$ sudo nano /etc/default/nis
```

```
yash@yash:~$ sudo nano /etc/default/nis
```

- After opening **/etc/default/nis** file scroll down and see-
Are we a NIS server and if so what kind (values: false, slave, master)?
- And write:
NISERVER=master
- And then scroll below
Are we a NIS client?
- And write:
NISCLIENT=false

```
GNU nano 5.2
#
# /etc/default/nis      Configuration settings for the NIS daemons.
#
# Are we a NIS server and if so what kind (values: false, slave, master)?
NISSERVER=master
# AdditionsIn
#
# Are we a NIS client?
NISCLIENT=false

# Location of the master NIS password file (for yppasswdd).
# If you change this make sure it matches with /var/yp/Makefile.
YPPWDDIR=/etc

# Do we allow the user to use ypchsh and/or ypchfn ? The YPCHANGEOK
# fields are passed with -e to yppasswdd, see it's manpage.
# Possible values: "chsh", "chfn", "chsh,chfn"
YPCHANGEOK=chsh

# NIS master server. If this is configured on a slave server then ypinit
# will be run each time NIS is started.
NISMASTER=
# AdditionsIn

# Additional options to be given to ypserv when it is started.
YPSERVARGS=
# AdditionsIn
#
# Additional options to be given to ypbind when it is started.
YPBINDARGS=

# Additional options to be given to yppasswdd when it is started. Note
# that if -p is set then the YPPWDDIR above should be empty.
YPPASSWDDARGS=
# AdditionsIn

# Additional options to be given to ypxfrd when it is started.
```

4. Take copy of NIS:

```
$ sudo cp /etc/default/nis /etc/default/nis_client
```

5. In /etc/ypserv.securenets, we need to write our IP address and subnet mask:

```
$ sudo nano /etc/ypserv.securenets
```

```
yash@yash:~$ sudo cp /etc/default/nis /etc/default/nis_client
yash@yash:~$ sudo nano /etc/ypserv.securenets
```

```
GNU nano 5.2                                         /etc/ypserv.securenets

# securenets  This file defines the access rights to your NIS server
#              for NIS clients (and slave servers - ypxfrd uses this
#              file too). This file contains netmask/network pairs.
#              A clients IP address needs to match with at least one
#              of those.
#
#              One can use the word "host" instead of a netmask of
#              255.255.255.255. Only IP addresses are allowed in this
#              file, not hostnames.
#
# Always allow access for localhost
255.0.0.0      127.0.0.0

# This line gives access to everybody. PLEASE ADJUST!
0.0.0.0        0.0.0.0
255.255.255.0  10.0.2.0
```

6. Now for gmake (**GNU make** - called simply make on Linux systems) is a tool to help you build a program from its source.

```
$ which gmake
$ sudo dpkg -S `which gmake`
```

```
yash@yash:~$ which gmake
/usr/bin/gmake
yash@yash:~$ sudo dpkg -S `which gmake`
make: /usr/bin/gmake
```

7. Now start & check the status of ypserv:

- ypserv: yellow pages (directory), it's a server where a client can query to see what service is there.
- ypbind: Client binds tools
- ypxfrd: nis database is with master, this command sends the database to the slave (secondary server) - load balancing

```
$ sudo systemctl start ypserv
$ sudo systemctl status ypserv
```

```
yash@yash:~$ sudo systemctl start ypserv
yash@yash:~$ sudo systemctl status ypserv
● nis.service - LSB: Start NIS client and server daemons.
  Loaded: loaded (/etc/init.d/nis; generated)
  Active: active (running) since Wed 2020-10-14 09:50:33 IST; 12s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 3141 ExecStart=/etc/init.d/nis start (code=exited, status=0/SUCCESS)
  Tasks: 6 (limit: 2515)
  Memory: 2.2M
  CGroup: /system.slice/nis.service
          └─3150 /usr/sbin/ypserv
              ├─3153 /usr/sbin/rpc.yppasswdd -D /etc -e chsh
              ├─3156 /usr/sbin/rpc.ypxfrd
              └─3164 /usr/sbin/ypbind -broadcast

Oct 14 09:47:37 yash systemd[1]: Starting LSB: Start NIS client and server daemons....
Oct 14 09:47:37 yash nis[3141]: Setting NIS domainname to: yash.
Oct 14 09:48:31 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 09:49:25 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 09:50:19 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 09:50:33 yash nis[3141]: Starting NIS services: ypserv yppasswdd ypxfrd ypbindbin
Oct 14 09:50:33 yash nis[3141]: .
Oct 14 09:50:33 yash systemd[1]: Started LSB: Start NIS client and server daemons..
```

8. Now run the yp:

```
$ sudo /usr/lib/yp/ypinit -m
```

```
yash@yash:~$ sudo /usr/lib/yp/ypinit -m
```

At this point, we have to construct a list of the hosts which will run NIS servers. yash 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: yash  
next host to add:
```

The current list of NIS servers looks like this:

```
yash  
Is this correct? [y/n: y]  
We need a few minutes to build the databases ...  
Building /var/yp/yash/ypservers ...  
Running /var/yp/Makefile ...  
gmake[1]: Entering directory '/var/yp/yash'  
Updating passwd.byname ...  
Updating passwd.byuid ...  
Updating group.byname ...  
Updating group.bygid ...  
Updating hostsbyname ...
```

yash has been set up as a NIS master server.

Now you can run ypinit -s yash on all slave server.

9. Now start & check status of NIS:

```
$ sudo systemctl start nis  
$ sudo systemctl status nis
```

```
yash@yash:~$ sudo systemctl start nis
yash@yash:~$ sudo systemctl status nis
● nis.service - LSB: Start NIS client and server daemons.
   Loaded: loaded (/etc/init.d/nis; generated)
   Active: active (running) since Wed 2020-10-14 09:50:33 IST; 20min ago
     Docs: man:systemd-sysv-generator(8)
     Tasks: 6 (limit: 2515)
    Memory: 2.2M
   CGroup: /system.slice/nis.service
           └─3150 /usr/sbin/ypserv
             ├─3153 /usr/sbin/rpc.yppasswdd -D /etc -e chsh
             ├─3156 /usr/sbin/rpc.ypxfrd
             └─3164 /usr/sbin/ypbind -broadcast

Oct 14 10:02:57 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:03:51 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:04:45 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:05:39 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:06:33 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:07:27 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:08:21 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:09:15 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:10:09 yash ypbind[3164]: broadcast: RPC: Timed out.
Oct 14 10:11:03 yash ypbind[3164]: broadcast: RPC: Timed out.
```

10. Restart and check status of rpcbind NIS:

```
$ sudo systemctl restart rpcbind nis
$ sudo systemctl status rpcbind nis
```

```
yash@yash:~$ sudo systemctl restart rpcbind nis
yash@yash:~$ sudo systemctl status rpcbind nis
● rpcbind.service - RPC bind portmap service
   Loaded: loaded (/lib/systemd/system/rpcbind.service; disabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-10-14 10:16:52 IST; 3min 7s ago
TriggeredBy: ● rpcbind.socket
   Docs: man:rpcbind(8)
   Main PID: 3804 (rpcbind)
     Tasks: 1 (limit: 2515)
    Memory: 656.0K
   CGroup: /system.slice/rpcbind.service
           └─3804 /sbin/rpcbind -f -w

Oct 14 10:16:52 yash systemd[1]: Starting RPC bind portmap service...
Oct 14 10:16:52 yash systemd[1]: Started RPC bind portmap service.

● nis.service - LSB: Start NIS client and server daemons.
   Loaded: loaded (/etc/init.d/nis; generated)
   Active: active (running) since Wed 2020-10-14 10:19:49 IST; 13s ago
```

11. Now we need to Add a user:

```
$ sudo adduser yash9
```

```
yash@yash:~$ sudo adduser yash9
Adding user `yash9' ...
Adding new group `yash9' (1012) ...
Adding new user `yash9' (1011) with group `yash9' ...
Creating home directory `/home/yash9' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for yash9
Enter the new value, or press ENTER for the default
      Full Name []: YashP
      Room Number []:
      Work Phone []:
      Home Phone []:
      Other []:
Is the information correct? [Y/n] Y
```

12. Now to make your user:

```
$ cd /var/yp
$ sudo make
```

```
yash@yash:~$ cd /var/yp
yash@yash:/var/yp$ sudo make
[sudo] password for yash:
gmake[1]: Entering directory '/var/yp/(none)'
Updating passwd.byname ...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating passwd.byuid ...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating group.bynam...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating group.bygid ...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating hosts.bynam...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating hosts.byaddr ...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating netid.bynam...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating shadow.bynam...
```

13. Now add a hostname in /etc/hosts: (I've added **kali.yash**)

```
$ sudo nano /etc/hosts
```

```
yash@yash:/var/yp$ sudo nano /etc/hosts
```

```
GNU nano 5.2
127.0.0.1      localhost      TRANS.TBL    VBoxWindo-
127.0.1.1      yash          wsAdditions..
10.0.2.15     kali.yash
# The following lines are desirable for IPv6 capable hosts
::1      localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

14. In /etc/yp.conf/ write the domain and server

In my case I've written domain as Yash and server as kali.yash

```
$ sudo nano /etc/yp.conf
```

yash@yash:/var/yp\$ sudo nano /etc/yp.conf

```
GNU nano 5.2                                         /etc/yp.conf
#
# yp.conf      Configuration file for the ypbnd process. You can define
#               NIS servers manually here if they can't be found by
#               broadcasting on the local net (which is the default).
#
#               See the manual page of ypbnd for the syntax of this file.
#
# IMPORTANT:   For the "ypserver", use IP addresses, or make sure that
#               the host is in /etc/hosts. This file is only interpreted
#               once, and if DNS isn't reachable yet the ypserver cannot
#               be resolved and ypbnd won't ever bind to the server.
#
# ypserver ypserver.network.com
# domain yash server kali.yash
```

15. Now, do \$ ls -lthr

```
$ cat ypservers
```

yash@yash:/var/yp\$ ls -lthr

```
total 40K
-rw-r--r-- 1 root root 185 Aug 16 21:14 nicknames
-rw-r--r-- 1 root root 17K Aug 16 21:14 Makefile
drwxr-xr-x 2 root root 4.0K Oct 14 09:36 '(none)'
drwxr-xr-x 2 root root 4.0K Oct 14 09:47 binding
-rw-r--r-- 1 root root 5 Oct 14 09:51 ypservers
drwxr-xr-x 2 root root 4.0K Oct 14 10:24 yash
```

yash@yash:/var/yp\$ cat ypserver

```
cat: ypserver: No such file or directory
```

yash@yash:/var/yp\$ cat ypservers

```
yash
```

16. Now do hostsbyname name to see all your host name & domain name

```
$ sudo cat hostsbyname  
$ sudo domainname
```

17. Now ping, just to check everything's working:

```
$ ping kali
```

```
yash@yash:/var/yp/(none)$ ping kali
PING kali (10.0.2.15) 56(84) bytes of data.
64 bytes from kali (10.0.2.15): icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from kali (10.0.2.15): icmp_seq=2 ttl=64 time=0.078 ms
64 bytes from kali (10.0.2.15): icmp_seq=3 ttl=64 time=0.044 ms
64 bytes from kali (10.0.2.15): icmp_seq=4 ttl=64 time=0.120 ms
64 bytes from kali (10.0.2.15): icmp_seq=5 ttl=64 time=0.391 ms
64 bytes from kali (10.0.2.15): icmp_seq=6 ttl=64 time=0.203 ms
^C
--- kali ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5115ms
rtt min/avg/max/mdev = 0.044/0.147/0.391/0.121 ms
```

```
$ ping yash
```

```
yash@yash:/var/yp/(none)$ ping yash
PING yash (127.0.1.1) 56(84) bytes of data.
64 bytes from yash (127.0.1.1): icmp_seq=1 ttl=64 time=0.031 ms
64 bytes from yash (127.0.1.1): icmp_seq=2 ttl=64 time=0.065 ms
64 bytes from yash (127.0.1.1): icmp_seq=3 ttl=64 time=0.106 ms
64 bytes from yash (127.0.1.1): icmp_seq=4 ttl=64 time=0.101 ms
64 bytes from yash (127.0.1.1): icmp_seq=5 ttl=64 time=0.040 ms
64 bytes from yash (127.0.1.1): icmp_seq=6 ttl=64 time=0.044 ms
64 bytes from yash (127.0.1.1): icmp_seq=7 ttl=64 time=0.040 ms
64 bytes from yash (127.0.1.1): icmp_seq=8 ttl=64 time=0.049 ms
64 bytes from yash (127.0.1.1): icmp_seq=9 ttl=64 time=0.050 ms
^C
--- yash ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8233ms
rtt min/avg/max/mdev = 0.031/0.058/0.106/0.025 ms
```

```
$ ping kali.yash
```

```
yash@yash:/var/yp/(none)$ ping kali.yash
PING kali.yash (10.0.2.15) 56(84) bytes of data.
64 bytes from kali.yash (10.0.2.15): icmp_seq=1 ttl=64 time=0.026 ms
64 bytes from kali.yash (10.0.2.15): icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from kali.yash (10.0.2.15): icmp_seq=3 ttl=64 time=0.087 ms
64 bytes from kali.yash (10.0.2.15): icmp_seq=4 ttl=64 time=0.052 ms
^C
--- kali.yash ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3071ms
rtt min/avg/max/mdev = 0.026/0.052/0.087/0.022 ms
```

18. Now again run the yp:

```
$ sudo /usr/lib/yp/ypinit -m
```

```
yash@yash:/var/yp/(none)$ sudo /usr/lib/yp/ypinit -m
```

At this point, we have to construct a list of the hosts which will run NIS servers. yash 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: yash
next host to add: kali.yash
next host to add:
```

The current list of NIS servers looks like this:

```
yash
kali.yash
```

Is this correct? [y/n: y] y
We need a few minutes to build the databases ...

yash has been set up as a NIS master server.

Now you can run ypinit -s yash on all slave server.

19. Now configure **nsswitch.conf** for NIS:

- The Name Service Switch (NSS) configuration file, /etc/nsswitch.conf, is used by the GNU C Library and certain other applications to determine the sources from which to obtain name-service information in a range of categories, and in what order.
- Each category of information is identified by a database name.

```
$ sudo nano /etc/nsswitch.conf
```

```
yash@yash:/var/yp/(none)$ sudo nano /etc/nsswitch.conf
```

```
GNU nano 5.2
# /etc/nsswitch.conf
#
# Example configuration of GNU Name Service Switch functionality.
# If you have the `glibc-doc-reference` and `info` packages installed, try:
# `info libc "Name Service Switch"` for information about this file.
#
#      HOME   GROUP  SERVICE
passwd:    files  systemd  nis
group:     files  systemd  nis
shadow:    files   nis
gshadow:   files
hosts:     files  mdns4_minimal [NOTFOUND=return] dns  nis
networks:  files
protocols: db    files
services:  db    files
ethers:    db    files
rpc:       db    files
netgroup:  nis
cert      VBoxDarwin-
          AdditionsUn...
```

20. Now reboot Kali and Login with the user that we have added above:

```
$ sudo reboot
```

```
yash@yash:/var/yp/(none)$ sudo reboot now
```

21. If Login is successful then NIS server is working.

