



**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. Shibu Mohapatra

Roll No: 530 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: **Shibu Mohapatra**

Roll No: **530**

Subject: Linux Server Administration

Class: T.Y.B.Sc.CS

INDEX

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

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

1. Dynamic Host Configuration Protocol Daemon (DHCPD), the DHCP server, is responsible for serving IP addresses and other relevant information upon client request.
2. Since DHCP is broadcast-based, a server will have to be present on each subnet for which DHCP service is to be provided.
3. Dynamic Host Configuration Protocol (DHCP) is a network management protocol used to automate the process of configuring devices on IP networks,
4. Thus allowing them to use network services such as DNS, NTP, and any communication protocol based on UDP or TCP.
5. A DHCP server dynamically assigns an IP address and other network configuration parameters to each device on a network so they can communicate with other IP networks.
6. DHCP is an enhancement of an older protocol called BOOTP. DHCP is an important part of the DDI solution.

Features of DHCP can be explained as:

- DHCPOFFER - The server will host (342 bytes).
- DHCPREQUEST - Client responds by broadcasting
- DHCPDISCOVER - First message generated between server & client (342 or 576 bytes).
- DHCPDECLINE - If IP address already in use, DHCP declines server.
- DHCPRELEASE - Client sends DHCP release packet to server to release IP address & cancel any remaining lease time.
- DHCPINFORM - Domain name.
- DHCPACK/DHCPNAK - Server will make entry [static ip to server].

COMMANDS & OUTPUT:

1. Command to install dhcp-server:

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

```
shibu@kali:~$ sudo apt-get install -y isc-dhcp-server
[sudo] password for shibu:
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 dhcpd.conf by directly editing through a path:

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

```
shibu@kali:~$ sudo nano /etc/dhcp/dhcpd.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 a (OR) $ ip addr
```

```
shibu@kali:~$ 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 08:00:27:9e:27:4a 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:fe9e:274a/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
```

```
File Actions Edit View Help

shibu@kali:~$ sudo vi /etc/default/isc-dhcp-server
shibu@kali:~$ 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
```

```
shibu@kali:~$ sudo systemctl start isc-dhcp-server
shibu@kali:~$ 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 Mon 2020-10-05 12:55:05 IST; 8s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 3192 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 8 (limit: 1677)
   Memory: 7.2M
    CGroup: /system.slice/isc-dhcp-server.service
            └─3076 /usr/sbin/dhcpcd -4 -q -cf /etc/dhcp/dhcpcd.conf eth0
                ├─3205 /usr/sbin/dhcpcd -4 -q -cf /etc/dhcp/dhcpcd.conf eth0

Oct 05 12:55:03 kali systemd[1]: Starting LSB: DHCP server...
Oct 05 12:55:03 kali isc-dhcp-server[3192]: Launching IPv4 server only.
Oct 05 12:55:03 kali dhcpcd[3205]: Wrote 0 leases to leases file.
Oct 05 12:55:03 kali dhcpcd[3205]: Server starting service.
Oct 05 12:55:05 kali isc-dhcp-server[3192]: Starting ISC DHCPV4 server: dhcpcd.
Oct 05 12:55:05 kali 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
shibu@kali:~$ sudo ifconfig eth0 192.168.106.128 netmask 255.255.255.0
shibu@kali:~$ 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 08:00:27:9e:27:4a 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:fe9e:274a/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
```

```
shibu@kali:~$ 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.1601882728
```

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 one who works on the machine simultaneously or it has the permissions to read, write and execute one file or all files. A user can be added individually or in a group.

- ## 2. Network settings:

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

\$ sudo ip a (OR) \$ sudo ip addr

- ### 3. Configuring services:

The services that run in the background are also known as daemons, services usually have a 'd' in last of their name, this denotes as a daemon. There are 5 services,

- a. Enable
 - b. Start
 - c. Status
 - d. Stop
 - e. Restart

COMMANDS & OUTPUT:

- ## 1. Adding a user:

```
$ sudo adduser <username>
```

```
FIRE Actions Edit View Help  
shibu@kali:~$ sudo adduser shibu7  
[sudo] password for shibu:  
Adding user `shibu7' ...  
Adding new group `shibu7' (1004) ...  
Adding new user `shibu7' (1003) with group `shibu7' ...  
Creating home directory `/home/shibu7' ...  
Copying files from `/etc/skel' ...  
New password:  
Retype new password:  
passwd: password updated successfully  
Changing the user information for shibu7  
Enter the new value, or press ENTER for the default  
Full Name []: ShibuM  
Room Number []:  
Work Phone []:  
Home Phone []:  
Other []:  
Is the information correct? [Y/n] Y
```

2. Network settings:

\$ sudo ip a (OR) \$ sudo ip addr

```
shibu@kali:~$ 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 08:00:27:9e:27:4a brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
            valid_lft 82425sec preferred_lft 82425sec
        inet6 fe80::a00:27ff:fe9e:274a/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

3. Configuring services:

Here, I've taken SSH service for demonstration

```
shibu@kali:~$ sudo systemctl start ssh
shibu@kali:~$ 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 Sun 2020-10-04 13:47:24 IST; 1h 38min ago
    Docs: man:sshd(8)
           man:sshd_config(5)
   Main PID: 664 (sshd)
      Tasks: 1 (limit: 1677)
     Memory: 2.9M
        CGroup: /system.slice/ssh.service
                 └─664 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 04 13:47:23 kali systemd[1]: Starting OpenBSD Secure Shell server...
Oct 04 13:47:24 kali sshd[664]: Server listening on 0.0.0.0 port 22.
Oct 04 13:47:24 kali sshd[664]: Server listening on :: port 22.
Oct 04 13:47:24 kali systemd[1]: Started OpenBSD Secure Shell server.

shibu@kali:~$ sudo systemctl stop ssh
shibu@kali:~$ sudo systemctl restart ssh
shibu@kali:~$ 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 Sun 2020-10-04 15:26:02 IST; 3s ago
    Docs: man:sshd(8)
           man:sshd_config(5)
   Process: 3330 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
  Main PID: 3331 (sshd)
     Tasks: 1 (limit: 1677)
    Memory: 1.3M
        CGroup: /system.slice/ssh.service
                 └─3331 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 04 15:26:02 kali systemd[1]: Starting OpenBSD Secure Shell server...
Oct 04 15:26:02 kali sshd[3331]: Server listening on 0.0.0.0 port 22.
Oct 04 15:26:02 kali sshd[3331]: Server listening on :: port 22.
Oct 04 15:26:02 kali systemd[1]: Started OpenBSD Secure Shell server.
```

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

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

```
vikas@Vikas:~$ sudo service --status-all
[ - ] apache-htcacheclean
[ - ] apache2
[ - ] apparmor
[ - ] atftpd
[ - ] avahi-daemon
[ + ] binfmt-support
[ - ] bluetooth
[ - ] console-setup.sh
[ + ] cron
[ - ] cryptdisks
[ - ] cryptdisks-early
[ + ] dbus
[ - ] dnsztcp
[ - ] gdomap
[ + ] haveged
[ - ] hwclock.sh
[ - ] inetsim
[ + ] inetutils-inetd
[ - ] iodined
[ - ] ipsec
[ - ] isc-dhcp-server
[ - ] keyboard-setup.sh
[ + ] kmod
[ + ] lightdm
[ - ] miredo
[ - ] mysql
[ + ] networking
[ - ] nfs-common
[ - ] nginx
[ - ] nis
[ - ] nmbd
[ - ] ntp
[ - ] openvpn
[ - ] plymouth
[ + ] plymouth-log
[ - ] postgresql
[ - ] pppd-dns
[ + ] procps
[ - ] ptunnel
[ - ] pulseaudio-enable-autospawn
[ - ] redsocks
[ - ] rpcbind
```

[+]	kmod
[+]	lightdm
[-]	miredo
[-]	mysql
[+]	networking
[-]	nfs-common
[-]	nginx
[-]	nis
[-]	nmbd
[-]	ntp
[-]	openvpn
[-]	plymouth
[+]	plymouth-log
[-]	postgresql
[-]	pppd-dns
[+]	procps
[-]	ptunnel
[-]	pulseaudio-enable-autospawn
[-]	redsocks
[-]	rpcbind
[+]	rsync
[+]	rsyslog
[-]	rwhod
[-]	samba-ad-dc
[-]	saned
[-]	screen-cleanups
[-]	selinux-autorelabel
[+]	smartmontools
[-]	smbd
[-]	snmpd
[+]	ssh
[-]	sshd
[+]	stunnel4
[-]	sudo
[-]	sysstat
[+]	udev
[+]	ufw
[+]	virtualbox-guest-utils
[-]	x11-common
[-]	xl2tpd

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
```

```
shibu@kali:~$ sudo apt-get -y install kernel-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
E: Unable to locate package kernel-server
shibu@kali:~$ sudo apt-get -y install 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 required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readke
  mysql-community-client-core mysql-community-server-core python3-deprecation py
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 811 not upgraded.
```

2. Command to enable a service:

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

```
shibu@kali:~$ sudo systemctl enable nfs-kernel-server
Synchronizing state of nfs-kernel-server.service with SysV service sc
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/shibu/demo_dir1
```

```
shibu@kali:~$ mkdir demo_dir1
shibu@kali:~$ b.txt
bash: b.txt: command not found
shibu@kali:~$ vi b.txt
shibu@kali:~$ ls b.txt
b.txt
shibu@kali:~$ cat b.txt
hello everyone
shibu@kali:~$ mv b.txt demo_dir1
shibu@kali:~$ ls /home/shibu/demo_dir1
b.txt
```

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
```

```
shibu@kali:~$ sudo vi /etc/exports
shibu@kali:~$ cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
#           to NFS clients.  See exports(5).
# /srv/nfs4      127.0.0.1(rw)
# 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/shibu/demo_dir1 127.0.0.1(rw)
```

After mentioning the /directory_path ip_address (permissions), it should look like this.
/home/shibu/demo_dir 127.0.0.1(rw)

5. Now to check the exports:

```
$ sudo exportfs -ra
```

```
shibu@kali:~$ sudo exportfs -ra
exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtree_check' specified for export "127.0.0.1:/home/shibu/demo_dir1".
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
```

```
shibu@kali:~$ sudo apt-get -y install 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:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 libcor
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-perl 1
  mysql-community-client-core mysql-community-server-core python3-deprecation python3-f1
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 811 not upgraded.
```

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/shibu_directory
```

```
shibu@kali:~$ sudo mkdir /home/shibu_directory1
```

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/shibu/demo_dir /home/shibu_directory
$ ls
```

```
shibu@kali:~$ sudo mount -t nfs 127.0.0.1:/home/shibu/demo_dir1 /home/shibu_directory1
shibu@kali:~$ cd /home/shibu_directory1
shibu@kali:/home/shibu_directory1$ ls
b.txt
shibu@kali:/home/shibu_directory1$
```

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

1. SSH Stands for - Secured Shell.
2. To tackle the issue of remote login vs password security a solution called Secured Shell (SSH) was developed.
3. It is a suite of network communication tools that are collectively based on protocol/standard i.e. guided by Internet Engineering Task Force (IETF).
4. The SSH command provides a secure encrypted connection between two hosts over an insecure network.
5. The connection can also be used for terminal access, file transfers & for tunnelling other applications.
6. OpenSSH - Open Source Implementations
SSH provides an authentication mechanism based on cryptographic keys, called the "Public Key Authentication".

Port:

- a. SSH client connection - Port number: 22
- b. Telnet client connection - Port number: 23
If we need to change, enter port number between 1024 and 32767.

COMMANDS & OUTPUT:

1. Install SSH server:

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

```
shibu@kali:~$ sudo apt-get install -y openssh-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
openssh-server is already the newest version (1:8.3p1-1).
The following packages were automatically installed and are no longer required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readke
  mysql-community-client-core mysql-community-server-core python3-deprecation py
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 811 not upgraded.
```

2. Now enable & check the status of SSH:

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

```

shibu@kali:~$ 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
shibu@kali:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
    Loaded: loaded (/lib/systemd/system/ssh.service; enabled; vendor preset: disabled)
   NETW Active: active (running) since Sun 2020-10-04 15:26:02 IST; 31min ago
      Docs: man:sshd(8)
             man:sshd_config(5)
   Main PID: 3331 (sshd)
      Tasks: 1 (limit: 1677)
     Memory: 1.3M
      CGroup: /system.slice/ssh.service
              └─3331 sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups

Oct 04 15:26:02 kali systemd[1]: Starting OpenBSD Secure Shell server...
Oct 04 15:26:02 kali sshd[3331]: Server listening on 0.0.0.0 port 22.
Oct 04 15:26:02 kali sshd[3331]: Server listening on :: port 22.
Oct 04 15:26:02 kali svstemd[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
```

```

shibu@kali:~$ sudo apt-get -y install ufw
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libgn
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-perl libunbound8 mecab-ipadic mecab-ipadic-utf8 mecab-utils
  mysql-community-client-core mysql-community-server-core python3-deprecation python3-flask-session python3-pcapfile python3-winrm python3.1
Use 'sudo apt autoremove' to remove them.
The following packages will be upgraded:
  ufw
1 upgraded, 0 newly installed, 0 to remove and 810 not upgraded.
Need to get 167 kB of archives.
After this operation, 1,024 B of additional disk space will be used.
Get:1 http://ftp.harukasan.org/kali kali-rolling/main amd64 ufw all 0.36-7 [167 kB]
Fetched 167 kB in 3s (60.9 kB/s)
Preconfiguring packages ...
(Reading database ... 279201 files and directories currently installed.)
Preparing to unpack .../archives/ufw_0.36-7_all.deb ...
Unpacking ufw (0.36-7) over (0.36-6) ...
Setting up ufw (0.36-7) ...
Processing triggers for man-db (2.9.2-1) ...
Processing triggers for kali-menu (2020.3.0) ...
Processing triggers for rsyslog (8.2004.0-1) ...
Processing triggers for systemd (245.5-3) ...

```

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

```
$ sudo ufw allow ssh
```

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

```
$ sudo ufw allow ssh
```

```
shibu@kali:~$ 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
```

```
shibu@kali:~$ ssh localhost
shibu@localhost's password:
Linux kali 5.6.0-kali2-amd64 #1 SMP Debian 5.6.14-1kali1 (2020-05-25) 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.
Last login: Wed Sep 16 10:06:32 2020 from 127.0.0.1
shibu@kali:~$ ssh 127.0.0.1
shibu@127.0.0.1's password:
Linux kali 5.6.0-kali2-amd64 #1 SMP Debian 5.6.14-1kali1 (2020-05-25) 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.
Last login: Sun Oct 4 16:00:07 2020 from ::1
shibu@kali:~$ █
```

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 is an open source implementation of the Server Message Block/Common Internet File System (SMB/CIFS) protocol suite.
2. Samba transparently provides file and print sharing services to Windows clients as well as other networked clients running other operating systems.
3. It does this using the native Microsoft networking protocols SMB/CIFS.
4. From a system administrator's point of view,
5. this means you can deploy a Linux/UNIX-based server without having to install Network File System (NFS) and some kind of
6. Linux/UNIX compatible authentication support on all the Windows clients in the network.
7. Instead, the clients can use their native tongue to talk to the server—which means fewer hassles for you and seamless integration for your users.

COMMANDS & OUTPUT (Kali Linux Part):

1. First, we need to install Samba Server:
\$ sudo apt-get -y install samba

```
shibu@kali:~$ sudo apt-get -y install samba
[sudo] password for shibu:
Reading package lists... Done
Building dependency tree
Reading state information... Done
samba is already the newest version (2:4.12.5+dfsg-3).0.0-Prac3_NFS_1.ppa - Prac3_NFS_2.ppa
The following packages were automatically installed and are no longer required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 lib
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-per
  mysql-community-client-core mysql-community-server-core python3-deprecation python3
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 810 not upgraded.
```

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

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

```
shibu@kali:~$ sudo systemctl start smbd
shibu@kali:~$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-10-04 16:08:29 IST; 10s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
   Process: 4591 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 4595 (smbd)
   Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 1677)
     Memory: 25.8M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/smbd.service
               └─4595 /usr/sbin/smbd --foreground --no-process-group
                  ├─4598 /usr/sbin/smbd --foreground --no-process-group
                  ├─4599 /usr/sbin/smbd --foreground --no-process-group
                  ├─4600 /usr/sbin/smbd --foreground --no-process-group

Oct 04 16:08:27 kali systemd[1]: Starting Samba SMB Daemon...
Oct 04 16:08:29 kali smbd[4595]: [2020/10/04 16:08:29.433922,  0] ../../lib/util/become_daemon.c:135(daemon_ready)
Oct 04 16:08:29 kali systemd[1]: Started Samba SMB Daemon.
Oct 04 16:08:29 kali smbd[4595]: daemon_ready: daemon 'smbd' finished starting up and ready to serve connections
```

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

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

```
shibu@kali:~$ sudo systemctl start nmbd
shibu@kali:~$ sudo systemctl status nmbd
● nmbd.service - Samba NMB Daemon
   Loaded: loaded (/lib/systemd/system/nmbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Sun 2020-10-04 16:08:52 IST; 16s ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
   Main PID: 4608 (nmbd)
   Status: "nmbd: ready to serve connections..."
      Tasks: 1 (limit: 1677)
     Memory: 2.7M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/nmbd.service
               └─4608 /usr/sbin/nmbd --foreground --no-process-group

Oct 04 16:08:52 kali systemd[1]: Starting Samba NMB Daemon...
Oct 04 16:08:52 kali nmbd[4608]: [2020/10/04 16:08:52.847548,  0] ../../lib/util/become_daemon.c:135(daemon_ready)
Oct 04 16:08:52 kali systemd[1]: Started Samba NMB Daemon.
Oct 04 16:08:52 kali nmbd[4608]: daemon_ready: daemon 'nmbd' finished starting up and ready to serve connections
```

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

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

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

```
shibu@kali:/etc/samba$ ls /home/shibu/samba_shared
a.txt  b.txt  c.txt  po                png
```

5. Now,
\$ cd /etc/samba
\$ ls /etc/samba

```
shibu@kali:~$ cd /etc/samba
shibu@kali:/etc/samba$ cat samba.conf
cat: samba.conf: No such file or directory
shibu@kali:/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

```
shibu@kali:/etc/samba$ sudo cp smb.conf smb.conf_bkp
shibu@kali:/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]

## Browsing/Identification ##

# Change this to the workgroup/NT-domain name your Samba server will part of
```

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/shibu/samba_shared
browseable = yes
read only = yes
guest ok = no
```

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

```
shibu@kali:/etc/samba$ sudo nano smb.conf
shibu@kali:/etc/samba$ sudo cat smb.conf
[samba_shared]
comment = Welcome to samba
path = /home/shibu/samba_shared
browseable = yes
read only = yes
guest ok = no
```

- Now download smbclient:

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

```
shibu@kali:/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:  libappindicator3-0.1 libconfig-inifiles-perl
 libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-perl libunbound8 mecab-ipadic-utf8 mysql-community-client-core mysql-community-server-core python3-deprecation python3-flask-session python3-mysqldb
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 810 not upgraded.
```

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

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

```
shibu@kali:/etc/samba$ sudo systemctl restart smbd
shibu@kali:/etc/samba$ sudo systemctl restart nmbd
shibu@kali:/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 Sun 2020-10-04 16:17:06 IST; 14s ago
     Docs: man:smbd(8)
     CPU: 8ms
       Tasks: 4 (limit: 1677)
      Memory: 8.0M
        CPU: 8ms
      CGroup: /system.slice/smbd.service
              └─4684 /usr/sbin/smbd --foreground --no-process-group
                  ├─4686 /usr/sbin/smbd --foreground --no-process-group
                  ├─4687 /usr/sbin/smbd --foreground --no-process-group
                  └─4688 /usr/sbin/smbd --foreground --no-process-group

Oct 04 16:17:05 kali systemd[1]: Starting Samba SMB Daemon...
Oct 04 16:17:06 kali smbd[4684]: [2020/10/04 16:17:06.239511,  0] ../../lib/util/become_daemon.c:135(daemon_ready)
Oct 04 16:17:06 kali systemd[1]: Started Samba SMB Daemon.
Oct 04 16:17:06 kali smbd[4684]: daemon_ready: daemon 'smbd' finished starting up and ready to serve connections
```

```
shibu@kali:/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 Sun 2020-10-04 16:17:10 IST; 14s ago
    Docs: man:nmbd(8)
          man:samba(7)
          man:smb.conf(5)
   Main PID: 4693 (nmbd)
     Status: "nmbd: ready to serve connections..."
       Tasks: 1 (limit: 1677)
      Memory: 2.3M
        CGroup: /system.slice/nmbd.service
                  └─4693 /usr/sbin/nmbd --foreground --no-process-group

Oct 04 16:17:10 kali systemd[1]: Starting Samba NMB Daemon...
Oct 04 16:17:10 kali systemd[1]: Started Samba NMB Daemon.
Oct 04 16:17:10 kali nmbd[4693]: [2020/10/04 16:17:10.952819,  0] ../../lib/util/become_daemon.c:135(daemon_ready)
Oct 04 16:17:10 kali nmbd[4693]: daemon ready: daemon 'nmbd' finished starting up and ready to serve connections
```

10. First do:

```
$ whoami
```

```
shibu@kali:/etc/samba$ whoami
shibu
```

11. And then:

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

```
shibu@kali:/etc/samba$ sudo pdbedit -a -u $(whoami)
new password:
retype new password:
Unix username: shibu
NT username:
Account Flags: [U ]
User SID: S-1-5-21-819542290-2812997002-1514989171-1000
Primary Group SID: S-1-5-21-819542290-2812997002-1514989171-513
Full Name: shibu mohapatra
Home Directory: \\kali\shibu
HomeDir Drive:
Logon Script:
Profile Path: \\kali\shibu\profile
Domain: KALI
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, 04 Sep 2020 12:07:58 IST
Password can change: Fri, 04 Sep 2020 12:07:58 IST
Password must change: never
Last bad password : 0
Bad password count : 0
Logon hours : FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
```

12. Type and retype your password:

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

```
shibu@kali:/etc/samba$ sudo pdbedit -L
shibu:1000:shibu mohapatra
HP:1002:Shibu
```

13. To check for your ip:

\$ ip a (OR) \$ ip addr

```
shibu@kali:/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 08:00:27:9e:27:4a brd ff:ff:ff:ff:ff:ff
        inet 192.168.56.101/24 brd 192.168.56.255 scope global dynamic noprefixroute eth0
            valid_lft 436sec preferred_lft 436sec
        inet6 fe80::a00:27ff:fe9e:274a/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

```
shibu@kali:/etc/samba$ sudo ufw allow from 192.168.56.0/24 to any app Samba
Rule added
shibu@kali:/etc/samba$ sudo ufw reload
Firewall reloaded
```

15. Now to reload your firewall use:

\$ sudo ufw reload

```
shibu@kali:/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 shibu -L //192.168.56.101

```
shibu@kali:/etc/samba$ sudo smbclient -U shibu -L //192.168.56.101
Enter WORKGROUP\shibu's password:

      Sharename          Type      Comment
      -----            ----      -----
    samba_shared        Disk      Welcome to samba
      IPC$              IPC       IPC Service (Samba 4.12.5-Debian)
    documents           Disk      sharedfolder
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
```

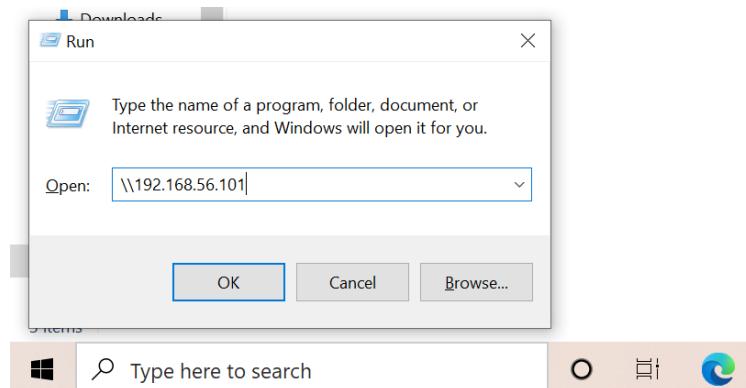
```
shibu@kali:/etc/samba$ sudo systemctl restart smbd
shibu@kali:/etc/samba$ sudo systemctl status smbd
● smbd.service - Samba SMB Daemon
   Loaded: loaded (/lib/systemd/system/smbd.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2020-09-04 12:21:43 IST; 7s ago
     Docs: man:smbd(8)
           man:samba(7)
           man:smb.conf(5)
  Process: 3186 ExecStartPre=/usr/share/samba/update-apparmor-samba-profile (code=exited, status=0/SUCCESS)
 Main PID: 3195 (smbd)
   Status: "smbd: ready to serve connections..."
    Tasks: 4 (limit: 1677)
```

```
shibu@kali:/etc/samba$ sudo systemctl restart nmbd
shibu@kali:/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 2020-09-04 12:22:17 IST; 9s ago
     Docs: man:nmbd(8)
           man:samba(7)
           man:smb.conf(5)
  Main PID: 3208 (nmbd)
    Status: "nmbd: ready to serve connections..."
     Tasks: 1 (limit: 1677)
    Memory: 2.4M
   CGroup: /system.slice/nmbd.service
           └─3208 /usr/sbin/nmbd --foreground --no-process-group

Sep 04 12:22:17 kali systemd[1]: Starting Samba NMB Daemon...
Sep 04 12:22:17 kali systemd[1]: Started Samba NMB Daemon.
Sep 04 12:22:17 kali nmbd[3208]: [2020/09/04 12:22:17.084232,  0] ../../lib/util/become_daemon.c:135(daemon_ready)
Sep 04 12:22:17 kali nmbd[3208]:   daemon ready: daemon 'nmbd' finished starting up and ready to serve connections
```

Windows OS Part:

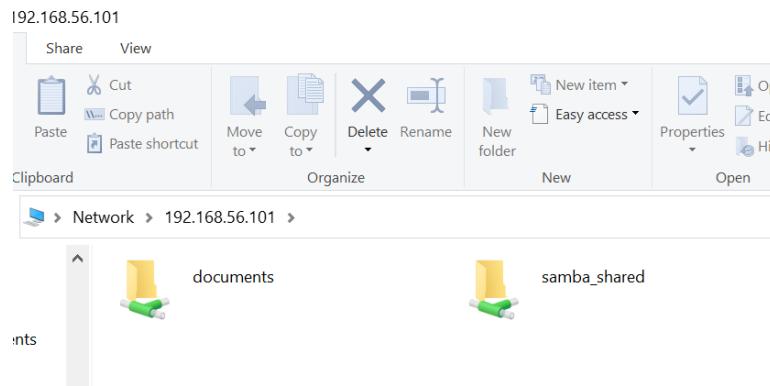
- Once smbd and nmbd services are active now switch back to your windows OS and press windows + R key on keyboard and now a pop up box will open and their write smbclient ip address, here in my case it was //192.168.56.101 and press OK.



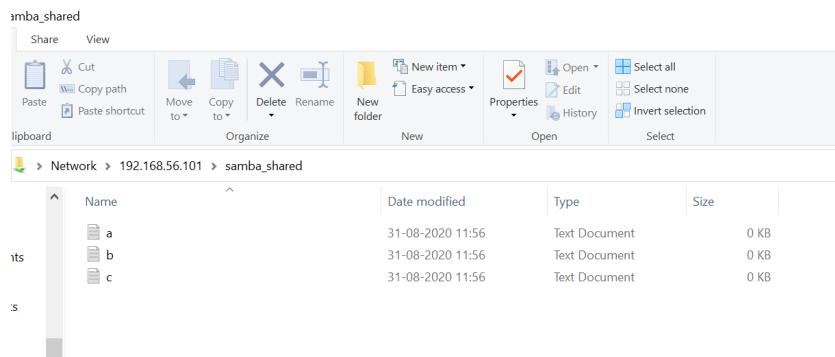
- Now again one more pop up box will open, write the credentials of your Windows or Kali or whatever it would be.



- 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.



5. 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
```

```
shibu@kali:~$ sudo apt update -y
[sudo] password for shibu:
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/contrib Sources [61.2 kB]
Get:4 http://ftp.harukasan.org/kali kali-rolling/main Sources [13.2 MB]
Get:5 http://ftp.harukasan.org/kali kali-rolling/non-free Sources [125 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 20s (379 kB/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
856 packages can be upgraded. Run 'apt list --upgradable' to see them.
shibu@kali:~$ uname -a
Linux kali 5.6.0-kali2-amd64 #1 SMP Debian 5.6.14-1kali1 (2020-05-25) x86_64 GNU/Linux
```

2. Command to install NTP:

```
$ sudo apt install ntp
```

```
shibu@kali:~$ 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:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 libconfig-inifiles-perl libdbd-mysql-perl libdbd-
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-perl libunbound8 mecab-ipadic mecab-ipadic-utf8
  mysql-community-client-core mysql-community-server-core python3-deprecation python3-flask-session python3-pcapfile python3-winc
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 856 not upgraded.
```

3. Command to install SNTP:

```
$ sudo apt install sntp
```

```
shibu@kali:~$ sudo apt install sntp
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 libconfig-inifiles-perl
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-perl libunbound8 mecab-
  mysql-community-client-core mysql-community-server-core python3-deprecation python3-flask-session python3-
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  sntp
0 upgraded, 1 newly installed, 0 to remove and 856 not upgraded.
Need to get 170 kB of archives.
After this operation, 295 kB of additional disk space will be used.
Get:1 http://ftp.harukasan.org/kali kali-rolling/main amd64 sntp amd64 1:4.2.8p14+dfsg-2 [170 kB]
Fetched 170 kB in 3s (50.5 kB/s)
Selecting previously unselected package sntp.
(Reading database ... 279201 files and directories currently installed.)
Preparing to unpack .../sntp_1%3a4.2.8p14+dfsg-2_amd64.deb ...
Unpacking sntp (1:4.2.8p14+dfsg-2) ...
Setting up sntp (1:4.2.8p14+dfsg-2) ...
Processing triggers for kali-menu (2020.3.0) ...
Processing triggers for man-db (2.9.2-1) ...
```

4. Now:

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

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

```
shibu@kali:/etc$ sudo systemctl start ntp
shibu@kali:/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 10:04:01 IST; 6s ago
     Docs: man:ntpd(8)
  Process: 3168 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 3174 (ntpd)
    Tasks: 2 (limit: 1677)
   Memory: 2.2M
      CPU: 0.000 CPU(s) (idle)
     CGroup: /system.slice/ntp.service
             └─3174 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 107:112
certs@vbox-darwin:~
```

Oct 07 10:04:04 kali ntpd[3174]: Soliciting pool server 13.126.27.131
Oct 07 10:04:04 kali ntpd[3174]: Soliciting pool server 162.159.200.123
Oct 07 10:04:04 kali ntpd[3174]: Soliciting pool server 139.59.55.93
Oct 07 10:04:05 kali ntpd[3174]: Soliciting pool server 185.216.231.25
Oct 07 10:04:05 kali ntpd[3174]: Soliciting pool server 5.103.139.163
Oct 07 10:04:05 kali ntpd[3174]: Soliciting pool server 172.105.40.191
Oct 07 10:04:05 kali ntpd[3174]: Soliciting pool server 157.119.108.165
Oct 07 10:04:06 kali ntpd[3174]: Soliciting pool server 43.240.66.74
Oct 07 10:04:06 kali ntpd[3174]: Soliciting pool server 2606:4700:f1::123
Oct 07 10:04:06 kali ntpd[3174]: Soliciting pool server 45.86.70.11

6. Now we need to configure NTP:

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

```
shibu@kali:/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
```

```
shibu@kali:/etc$ sudo systemctl restart ntp
shibu@kali:/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 10:06:37 IST; 2s ago
    Docs: man:ntpd(8)
  Process: 3228 ExecStart=/usr/lib/ntp/ntp-systemd-wrapper (code=exited, status=0/SUCCESS)
 Main PID: 3234 (ntpd)
    Tasks: 2 (limit: 1677)
   Memory: 1.3M
      CGroup: /system.slice/ntp.service
              └─3234 /usr/sbin/ntpd -p /var/run/ntpd.pid -g -u 107:112

Oct 07 10:06:37 kali ntpd[3234]: Listen normally on 3 eth0 10.0.2.15:123
Oct 07 10:06:37 kali ntpd[3234]: Listen normally on 4 lo [::]:123
```

8. Now to query NTP and check its connection:

```
$ sudo ntpq -p
```

```
shibu@kali:/etc$ sudo ntpq -p
      remote           refid      st t when poll reach   delay    offset  jitter
=====
 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         .STEP.       16 u    -  64    0    0.000  +0.000  0.000
time.cloudflare 10.222.8.61    3 u     1  64    1   60.644  -32.380  4.541
 static.226.144. 194.58.200.20  2 u     2  64    1  191.303  -34.253  74.370
+ec2-13-126-27-1 169.254.169.123  4 u     3  64    1   60.274  -24.145  8.731
*time.cloudflare 10.222.8.61    3 u     3  64    1   76.669  -25.906  10.419
+139.59.55.93      17.253.82.253  2 u     1  64    1   50.248  -34.079  6.430
+ntp.in.eria.one 14.139.60.102   2 u     2  64    1   47.032  -29.365  2.310
-5.103.139.163.s .GPS.        1 u     3  64    1  200.256  -28.706  11.338
-185.216.231.25 17.253.26.125   2 u     8  64    1  283.164  -27.644  7.593
+157.119.108.165 162.159.200.123  4 u     7  64    1   58.852  -28.390  19.976
+43.240.66.74      103.134.252.11  3 u     7  64    1   39.250  -29.274  22.204
shibu@kali:/etc$
```

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
```

```
shibu@kali:~$ 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
361 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

2. Then install wget:

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

```
shibu@kali:~$ 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).
The following packages were automatically installed and are no longer required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 libconfig-i
  libpython3.7-minimal libpython3.7-stdlib libqhull7 libreadline5 libterm-readkey-perl libunb
  mysql-community-client-core mysql-community-server-core python3-deprecation python3-flask-s
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 861 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
```

```
shibu@kali:~$ sudo wget https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb
--2020-10-11 15:26:54-- 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]
--2020-10-11 15:26:56-- https://repo.mysql.com//mysql-apt-config_0.8.15-1_all.deb
Resolving repo.mysql.com (repo.mysql.com)... 23.50.245.17
Connecting to repo.mysql.com (repo.mysql.com)|23.50.245.17|: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.2'

sudo: systemctl: command not found
mysql-apt-config_0.8.15-1_all.deb.2: 100%[=====] 35532/35532
shibu@kali:~/etc$ sudo systemctl status ntp
2020-10-11 15:26:56 (3.29 MB/s) - 'mysql-apt-config_0.8.15-1_all.deb.2' saved [35532/35532]
```

4. Install the deb files:

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

```
shibu@kali:~$ sudo dpkg -i mysql-apt-config_0.8.15-1_all.deb
configuration.
(Reading database ... 279293 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)
OK
```

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

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



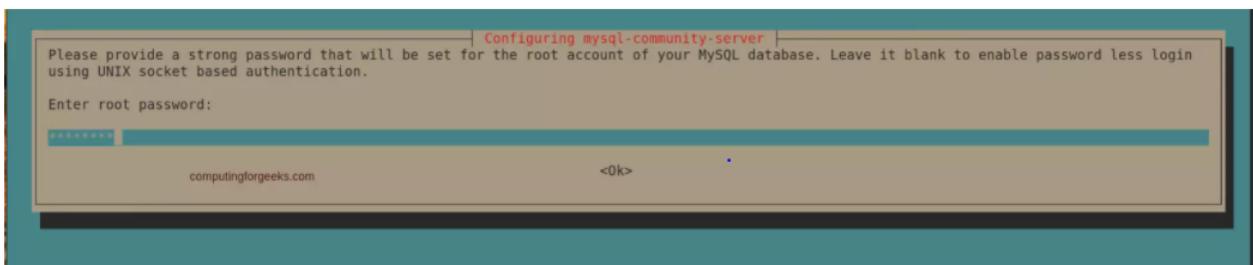
5. We need to & update install mysql-community-server:

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

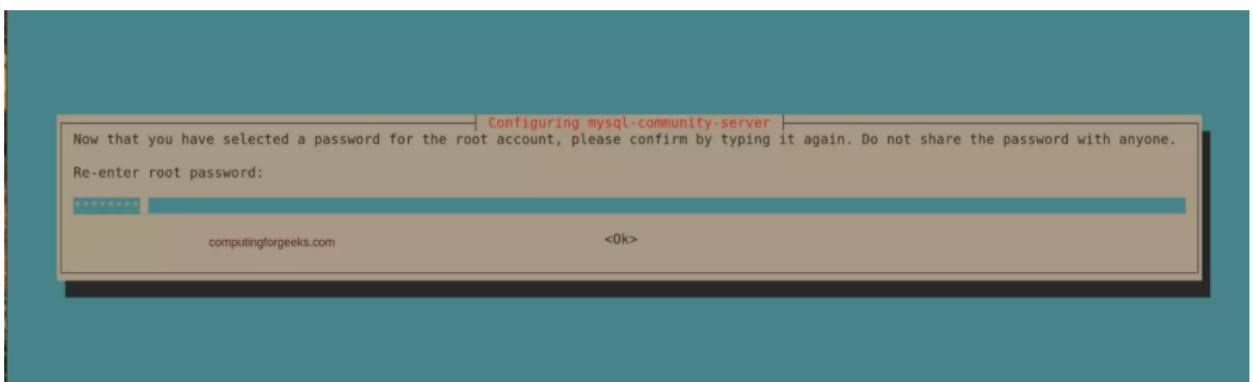
```
shibu@kali:~$ sudo apt update  
Hit:1 http://repo.mysql.com/apt/debian jessie InRelease  
Hit:2 http://kali.download/kali kali-rolling InRelease  
Get:3 http://repo.mysql.com/apt/debian/jessie/mysql-tools-preview amd64 Packages [920 B]  
Fetched 920 B in 1s (647 B/s)  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
You don't have unlimited access, but only if  
861 packages can be upgraded. Run 'apt list --upgradable' to see them.  
shibu@kali:~$ sudo apt install mysql-community-server  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Some packages could not be installed. This may mean that you have  
requested an impossible situation or if you are using the unstable  
distribution that some required packages have not yet been created  
or been moved out of Incoming.  
The following information may help to resolve the situation: the network!  
#disallow auth  
The following packages have unmet dependencies:  
  mysql-community-server: Depends: mysql-client (= 5.7.30-1debian8) but 8.0.21-1ubuntu18.04 is to be installed  
E: Unable to correct problems, you have held broken packages.
```

Accept license agreement in the next screens to begin installation.

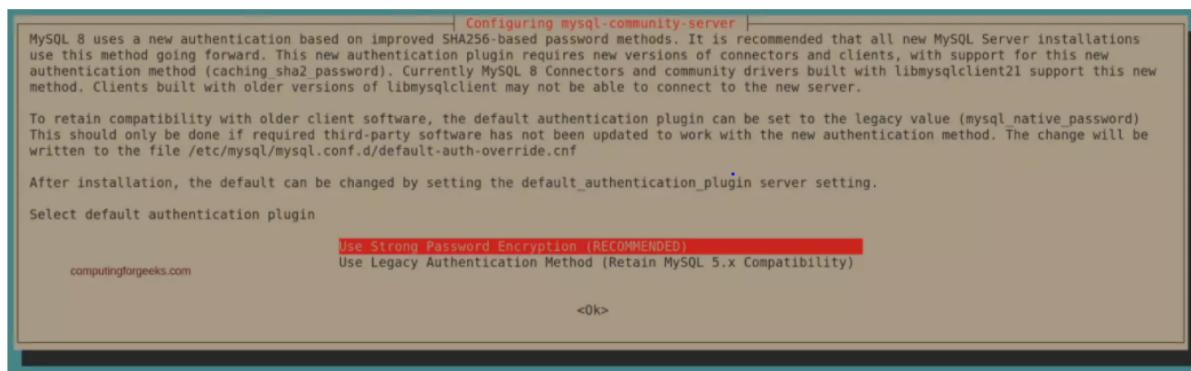
Set root password for your MySQL database server.



Confirm your root password.



Select the default authentication plugin.



6. Now enable & check MySQL status:

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

```
shibu@kali:~$ sudo systemctl enable --now mysql
Synchronizing state of mysql.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable mysql
Failed to enable unit: Unit file /etc/systemd/system/mysql.service is masked.
shibu@kali:~$ sudo systemctl status mysql
● mysql.service   sudo systemctl start ntp
● ntp           Loaded: masked (Reason: Unit mysql.service is masked.)
● ntp           Active: inactive (dead) Service
```

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

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

```
shibu@kali:~$ sudo mysql -u root -p
Enter password: 
ERROR 2002 (HY000): Can't connect to local MySQL server through socket '/var/run/mysqld/mysqld.sock' (2)
shibu@kali:~$
```

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 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 balancingIf 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

```
shibu@kali:~$ sudo apt-get update
Hit:1 http://repo.mysql.com/apt/debian jessie InRelease
Get:2 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 16s (1,879 kB/s)
Reading package lists... Done
shibu@kali:~$
```

```
$ sudo apt update
```

```
shibu@kali:~$ sudo apt update
```

- ## 2. Now install NIS:

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

```
shibab@kali:~$ sudo apt-get install -y 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 required:
  fonts-glyphicons-halflings galera-3 gir1.2-appindicator3-0.1 libappindicator3-1 libconfig-inifiles-
  libhtml-template-perl libicu63 libjsoncpp1 libmecab2 libmpdec2 libpython3.7-minimal libpython3.7-st
  mecab-ipadic-utf8 mecab-utils mysql-client mysql-community-client mysql-community-client-core mysql
  python3-flask-session python3-mimetypes python3-mimerender python3-pcapfile python3-winrm python3.7
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 193 not upgraded.
```

3. Now user must configure NIS:

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

shibu@kali:~\$ 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)?
NISERVER=master
# Are we a NIS client?
NISCLIENT=false

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

# Do we allow the user to use ypchsh and/or ypchnfn ? The YPCHANGEOK
# fields are passed with -e to ypsswdd, 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=

# Additional options to be given to ypserv when it is started.
YPSERVARGS=
    -l 3x      VBoxLinuxA
# Additional options to be given to ypbind when it is started.
YPBINDARGS=

# Additional options to be given to ypsswdd when it is started. Note
# that if -p is set then the YPPWDDIR above should be empty.
YPPASSWDDARGS=
    -l 3x      VBoxLinuxA

# 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
```

```
shibu@kali:~$ sudo cp /etc/default/nis /etc/default/nis_client
shibu@kali:~$ 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 (**G**NU 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`
```

```
shibu@kali:~$ which gmake
/usr/bin/gmake
shibu@kali:~$ 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
```

```

shibu@kali:~$ sudo systemctl start ypserv
shibu@kali:~$ 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; 7s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 4308 ExecStart=/etc/init.d/nis start (code=exited, status=0/SUCCESS)
    Tasks: 6 (limit: 1672)
   Memory: 2.7M
      CGroup: /system.slice/nis.service
              └─4318 /usr/sbin/ypserv
                  ├─4321 /usr/sbin/rpc.yppasswdd -D /etc -e chsh
                  ├─4323 /usr/sbin/rpc.ypxfrd
                  └─4331 /usr/sbin/ypbind -broadcast

Oct 14 09:47:36 kali systemd[1]: Starting LSB: Start NIS client and server daemons....
Oct 14 09:47:36 kali nis[4308]: Setting NIS domainname to: shibu.
Oct 14 09:48:31 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 09:49:25 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 09:50:19 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 09:50:33 kali nis[4308]: Starting NIS services: ypserv yppasswdd ypxfrd ypbindbinding to YP server....
Oct 14 09:50:33 kali nis[4308]: .
Oct 14 09:50:33 kali systemd[1]: Started LSB: Start NIS client and server daemons..

```

8. Now run the yp:

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

```

shibu@kali:~$ sudo /usr/lib/yp/ypinit -m > mtu 1500 qdisc pfifo_fast state UP qdisc link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
At this point, we have to construct a list of the hosts which will run NIS servers. kali 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>. preferred_lft forever
shibu@kali:~$ next host to add: kali
shibu@kali:~$ next host to add:
The current list of NIS servers looks like this:

```

kali

```

Is this correct? [y/n: y] y
We need a few minutes to build the databases...
Building /var/yp/shibu/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory '/var/yp/shibu'
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 netgroup...

```

```

shibu@kali:~$ ip a
kali has been set up as a NIS master server.
  link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
Now you can run ypinit -s kali on all slave server.

```

9. Now start & check status of NIS:

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

```
shibu@kali:~$ sudo systemctl start nis
shibu@kali:~$ 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: 1672)
  Memory: 2.7M
    CGroup: /system.slice/nis.service
            └─4318 /usr/sbin/ypserv
              ├─4321 /usr/sbin/rpc.ypasswdd -D /etc -e chsh
              ├─4323 /usr/sbin/rpc.ypfrd
              └─4331 /usr/sbin/ypbind -broadcast

Oct 14 10:02:56 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:03:50 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:04:44 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:05:38 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:06:32 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:07:26 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:08:20 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:09:14 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:10:08 kali ypbind[4331]: broadcast: RPC: Timed out.
Oct 14 10:11:02 kali ypbind[4331]: broadcast: RPC: Timed out.
```

10. Restart and check status of rpcbind NIS:

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

```
shibu@kali:~$ sudo systemctl restart rpcbind nis
shibu@kali:~$ 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 10:16:44 IST; 44s ago
     Docs: man:rpcbind(8)
   Main PID: 4639 (rpcbind)
   Tasks: 1 (limit: 1672)
  Memory: 664.0K
    CGroup: /system.slice/rpcbind.service
           └─4639 /sbin/rpcbind -f -w

Oct 14 10:16:43 kali systemd[1]: Starting RPC bind portmap service...
Oct 14 10:16:44 kali 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 (exited) since Wed 2020-10-14 10:16:55 IST; 34s ago
     Docs: man:systemd-sysv-generator(8)
   Process: 4649 ExecStart=/etc/init.d/nis start (code=exited, status=0/SUCCESS)

Oct 14 10:16:44 kali systemd[1]: Starting LSB: Start NIS client and server daemons...
Oct 14 10:16:55 kali nis[4649]: Starting NIS services: ypserv yppasswdd ypxfrd ypbindbinding to YP server.....
Oct 14 10:16:55 kali nis[4649]: .
Oct 14 10:16:55 kali systemd[1]: Started LSB: Start NIS client and server daemons..
```

11. Now we need to Add a user:

```
$ sudo adduser shibu9
```

```
shibu@kali:~$ sudo adduser shibu9
Adding user `shibu9' ... Adding new group `shibu9' (1005) ...
Adding new user `shibu9' (1004) with group `shibu9' ...
Creating home directory `/home/shibu9' ...
Copying files from `/etc/skel'.... cft forever
New password: [REDACTED]
Retype new password: [REDACTED]
passwd: password updated successfully
Changing the user information for shibu9
Enter the new value, or press ENTER for the default
  Full Name []: shibu9@redacted_ltt_Forever
  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
```

```
shibu@kali:~$ cd /var/yp
shibu@kali:/var/yp$ sudo make
gmake[1]: Entering directory '/var/yp/shibu'
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.byname...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating group.bygid...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating netidbyname...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating shadowbyname...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating gmake[1]: Leaving directory '/var/yp/shibu'
```

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

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

```
shibu@kali:/var/yp$ sudo nano /etc/hosts

GNU nano 5.2
127.0.0.1      localhost
127.0.1.1      kali
127.0.0.1      shibu.com
10.0.2.15      kali.shibu

# 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 Shibu and server as kali.shibu

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

```
shibu@kali:/var/yp$ sudo nano /etc/yp.conf
```

```
GNU nano 5.2                                     /etc/yp.conf

# yp.conf      Configuration file for the ypbind process. You can define
#               NIS servers manually here if they can't be found by
#               broadcasting on the local net (which is the default).
#
# Home   run See the manual page of ypbind 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 ypbind won't ever bind to the server.

# ypserver ypserver.network.com
domain shibu server kali.shibu
```

15. Now, do \$ ls -l rth

```
$ cat ypservers
```

```
shibu@kali:~/var/yp$ ls -lrth
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 10:25 shibu
shibu@kali:~/var/yp$ cat ypservers
kali
```

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
```

```
shibu@kali:/var/yp/(none)$ ping kali
PING kali (127.0.1.1) 56(84) bytes of data.
64 bytes from kali (127.0.1.1): icmp_seq=1 ttl=64 time=0.068 ms
64 bytes from kali (127.0.1.1): icmp_seq=2 ttl=64 time=0.088 ms
64 bytes from kali (127.0.1.1): icmp_seq=3 ttl=64 time=0.086 ms
64 bytes from kali (127.0.1.1): icmp_seq=4 ttl=64 time=0.088 ms
64 bytes from kali (127.0.1.1): icmp_seq=5 ttl=64 time=0.056 ms
64 bytes from kali (127.0.1.1): icmp_seq=6 ttl=64 time=0.075 ms
64 bytes from kali (127.0.1.1): icmp_seq=7 ttl=64 time=0.102 ms
64 bytes from kali (127.0.1.1): icmp_seq=8 ttl=64 time=0.067 ms
^C
--- kali ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7150ms
rtt min/avg/max/mdev = 0.056/0.078/0.102/0.013 ms
```

```
$ ping shibu.com
```

```
shibu@kali:/var/yp/(none)$ ping shibu.com
PING shibu.com (127.0.0.1) 56(84) bytes of data.
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.050 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.097 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.085 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.055 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.088 ms
^C
--- shibu.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4102ms
rtt min/avg/max/mdev = 0.050/0.075/0.097/0.018 ms
```

```
$ ping kali.shibu
```

```
shibu@kali:~$ ping kali.shibu
PING kali.shibu (10.0.2.15) 56(84) bytes of data.
64 bytes from kali.shibu (10.0.2.15): icmp_seq=1 ttl=64 time=0.046 ms
64 bytes from kali.shibu (10.0.2.15): icmp_seq=2 ttl=64 time=0.104 ms
64 bytes from kali.shibu (10.0.2.15): icmp_seq=3 ttl=64 time=0.056 ms
64 bytes from kali.shibu (10.0.2.15): icmp_seq=4 ttl=64 time=0.092 ms
64 bytes from kali.shibu (10.0.2.15): icmp_seq=5 ttl=64 time=0.094 ms
^C
--- kali.shibu ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4106ms
rtt min/avg/max/mdev = 0.046/0.078/0.104/0.022 ms
```

18. Now again run the yp:

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

```
shibu@kali:/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. kali 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: kali
next host to add: kali.shibu
next host to add:
```

The current list of NIS servers looks like this:

```
kali
kali.shibu
```

```
Is this correct? [y/n: y]
We need a few minutes to build the databases...
Building /var/yp/shibu/ypservers...
Running /var/yp/Makefile...
gmake[1]: Entering directory '/var/yp/shibu'
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.byname...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating group.bygid...
failed to send 'clear' to local ypserv: RPC: Program not registeredUpdating hostsbyname...
```

kali has been set up as a NIS master server.

Now you can run `ypinit -s kali` 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
```

```
shibu@kali:/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.

passwd:      files      systemd nis
group:       files      systemd nis
shadow:      files      nis
gshadow:     files
Trash:        TRANSIENT

hosts:        files      mdns4_minimal [NOTFOUND=return] dns nis
networks:    files

protocols:   db  files
services:    db  files
ethers:      db  files
rpc:         db  files

netgroup:    nis

```

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

```
$ sudo reboot
```

```
shibu@kali:/var/yp/(none)$ sudo reboot
```

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

