Managing a Network Using Linux

Info 2416
Server Operating Systems
(S11)

Prepared By:	Gurkamal Bassi (100360291) Sukhveer Sohi (100371170) Premgeet Singh (100362923) Gurkeerat Singh (100364062)
Submitted to:	Kenward Chin
Submission date:	14 April 2020

Contents

Members Contribution:

Gurkamal Bassi:	Compared different Linux distributions.
	Installed Ubuntu Distribution for the project.
	Installed and configured the DHCP server.
Sukhveer Sohi:	Installed and configured the DNS server
	Installed and Configured user authentication over the network
Premgeet Singh:	Installed and configured Apache Server
	Challenged faced in the project
Gurkeerat Singh:	Designed the webpage for the group project
	Compared WINDOWS SERVER VS LINUX Server

Introduction

Managing a network simply means setting up, administering and troubleshooting a network. Basically, the purpose of a computer network is to share the resources (files and documents) on other devices, hardware devices (for example - printers) and to be able to communicate within and with other networks. Network management can be done using different operating systems like Microsoft Windows Server, Novell Open Enterprise Server, Linux Based OS like Ubuntu server, openSUSE, etc. Mostly all the OS used for network management can achieve the same functionalities.

In this course, we extensively studied managing the network using Microsoft Windows Server, for our project we have chosen to work on a LINUX based OS. LINUX based OS is also very popular among big companies like Oracle, IBM, and Amazon.

Major Goals of our project:

a) *Installing a LINUX operating system on a virtual box.*

b) *Setting up and configuring various services such as*

DHCP

DNS

Web Server

User authentication service over the network.

c) *Creating folder shares using NFS or SAMBA that could be gained by other clients.*

Weakness and Strengths of Linux distributions:

1) **Debian**:

Strengths –

a) Packaging System - Debian GNU/Linux has a packaging system that helps to install new applications, set up old ones and supervises the system without being dependent on libraries and even there is no need to re-write the configuration files.

b) Easy to Install - Debian is one of the easiest to install OS. It can be easily installed from CD, DVD, over the network or even a USB-stick.

c) A lot of Software - Debian includes more than 59000 different types of software and all are available for free. Most of them are already installed by an installer in Debian and are ready to use upon installation of OS.

Weakness –

a) Problem with Free Software - In Debian adding software to the system is as easy as assessing a service from storage. But even this is difficult for some users. Therefore, they depend on using other derivatives like Linux Mint or Ubuntu in which it is easier to get the software (non-free drivers) or some tools like Flash.

b) Usage of Systemd – Since the introduction of the system as an administrative tool many users are not comfortable using it as it is too powerful. And some of the users consider this introduction as a conspiracy by Red Hat.

2) **Fedora**

Strengths –

a) Fast Boot - Fedora OS is famous for its fast boot. Upon turning on the PC running Fedora the boot happens in less than 20 seconds. While it might not be the fastest in the world. But it is very fast for a complete Linux distribution.

b) Graphics - There are a lot of features in FEDORA that lets users control the system. For example, users can control the language settings, users, authentications, network shares, web servers, and firewalls, etc. Moreover, we can configure 3D support for graphic cards and more convenient color management.

Weakness –

There are not many cons of FEDORA but of the cons is that the new version of the OS comes out every six to nine months and we cannot go back to earlier versions so our work can get a bit messed up.

3) **Ubuntu**

Strengths-

a) This OS is one of the easiest to install with a very easy to use interface.

b) The apt-package is the most efficient way of installing programs among the other available ways. Plus, Ubuntu comes with Ubuntu Software Centre.

Weakness –

a) apt is not user friendly for non-Linux users.

b) Ubuntu gives a very substandard support for the printers.

Advantages of Managing the network using LINUX

➔ **Stability-** Linux servers are the most stable platform and the main reason for this is because there is no need to reboot the system even for months. It is unlikely that your system will freeze or there will be a lag in the performance.

➔ **Overall Performance-** On various networks and workstations Linux provides an environment that’s powerful, reliable and stable. It can be connected to multiple devices without any issues.

➔ **Affordability-** Linux being an open-source solution brings the affordability with the package. Also, the setup cost is very low. There are many free applications designed to run with it. Overall, it is much cheaper to run than the Windows Server.

➔ **Security-** Linux servers are highly secure because of antispyware and firewall services. Also, because the code is open sourced everyone is free to examine the code due to which bugs and issues are found and resolved very easily.

➔ **Multitasking-** One of the reasons that Linux servers are used so widely is because they support multitasking capabilities. Multiple users can connect very easily and users can run multiple apps at the same time without crashing or freezing the system.

Disadvantages of Managing the network using LINUX

➔ Linux integration is not so user-friendly. New users may find it difficult to operate a Linux machine.

➔ Many programs which are basically windows friendly won’t run in Linux.

➔ There are very few hardware driver selections in Linux.

Challenges and difficulties faced by the group:

- ➔ Due to the current COVID-19 pandemic, it was very hard for all group members to coordinate with each other. Later things were shifted to virtual software like zoom. It took time to adapt to the new system.

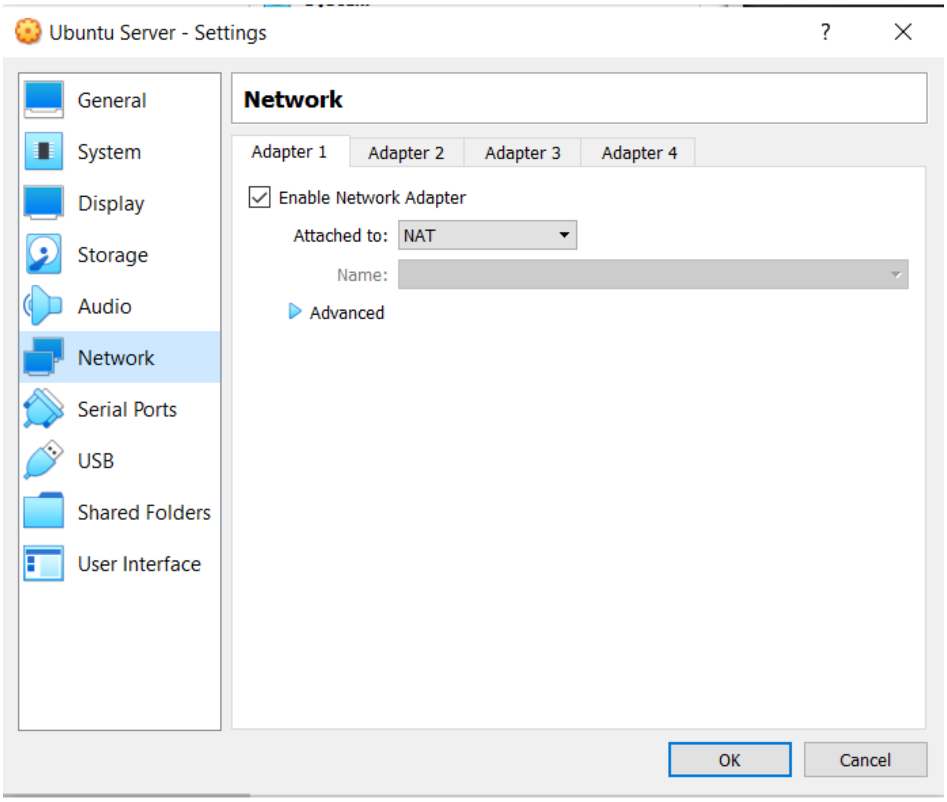
- ➔ Most of us being Windows users found the Linux system a bit hard to understand and confusing. So, a lot of research and hard work was put in by each member of the group in order to understand and implement setting up the server.

- ➔ Many unexpected errors and problems occurred while setting up the server for which we found solutions on the Internet, but they were hard to implement.

- ➔ In Windows, all the setup is done in GUI (Graphical User Interface) while in the case of Linux it's done in the terminal which is a little bit harder. You must take extreme care of command syntax as well as system functionality.

Install and Configure the DHCP server

Step 1: Change the **Network Setting** of your Virtual Machine.
We will add **two adapters**: One for the NAT and other for the Host-Only.



Ubuntu Server - Settings

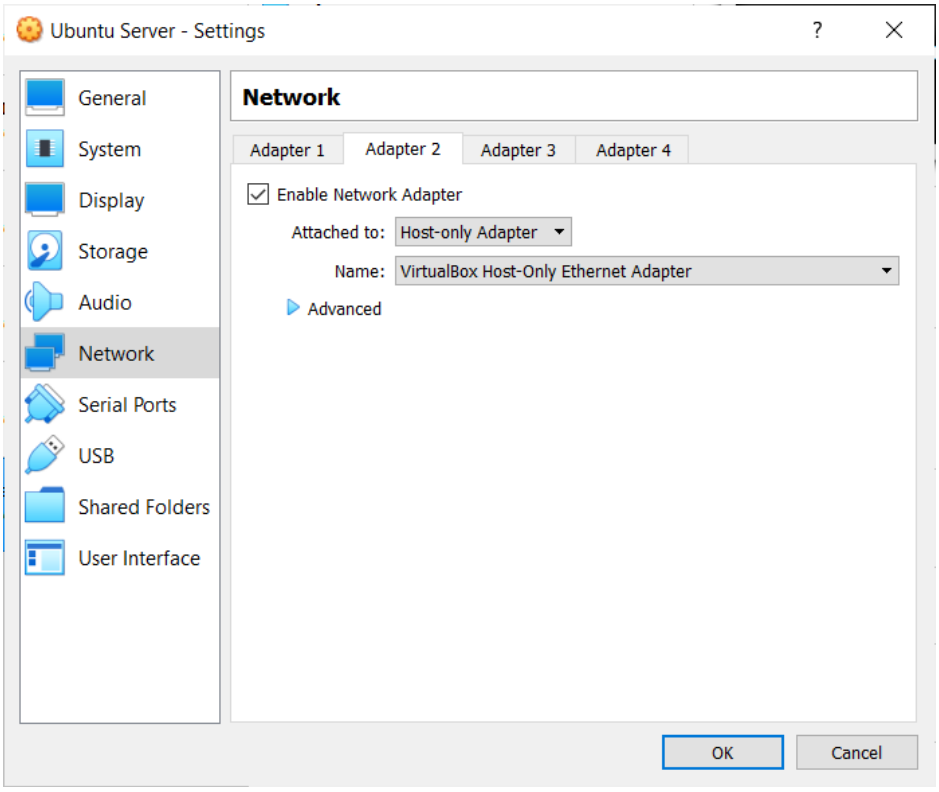
Network

Adapter 1

☑ Enable Network Adapter

Attached to: NAT

Name:



Ubuntu Server - Settings

Network

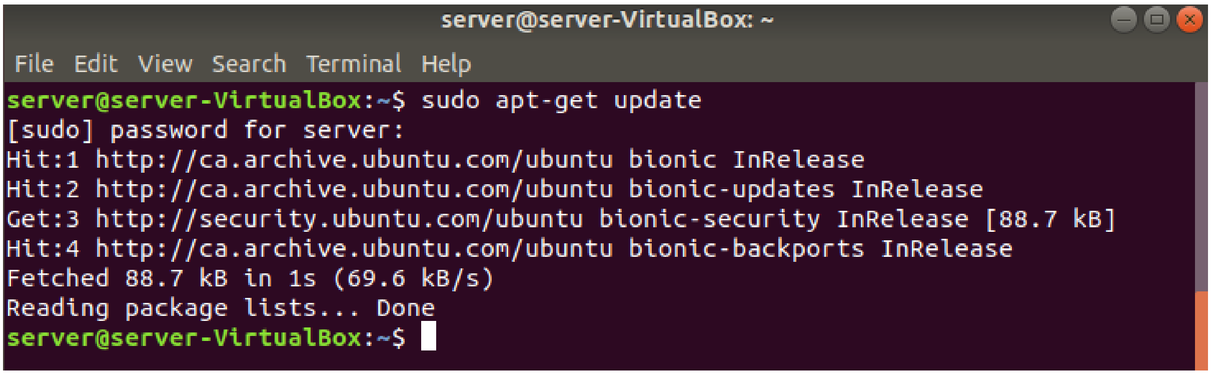
Adapter 2

☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

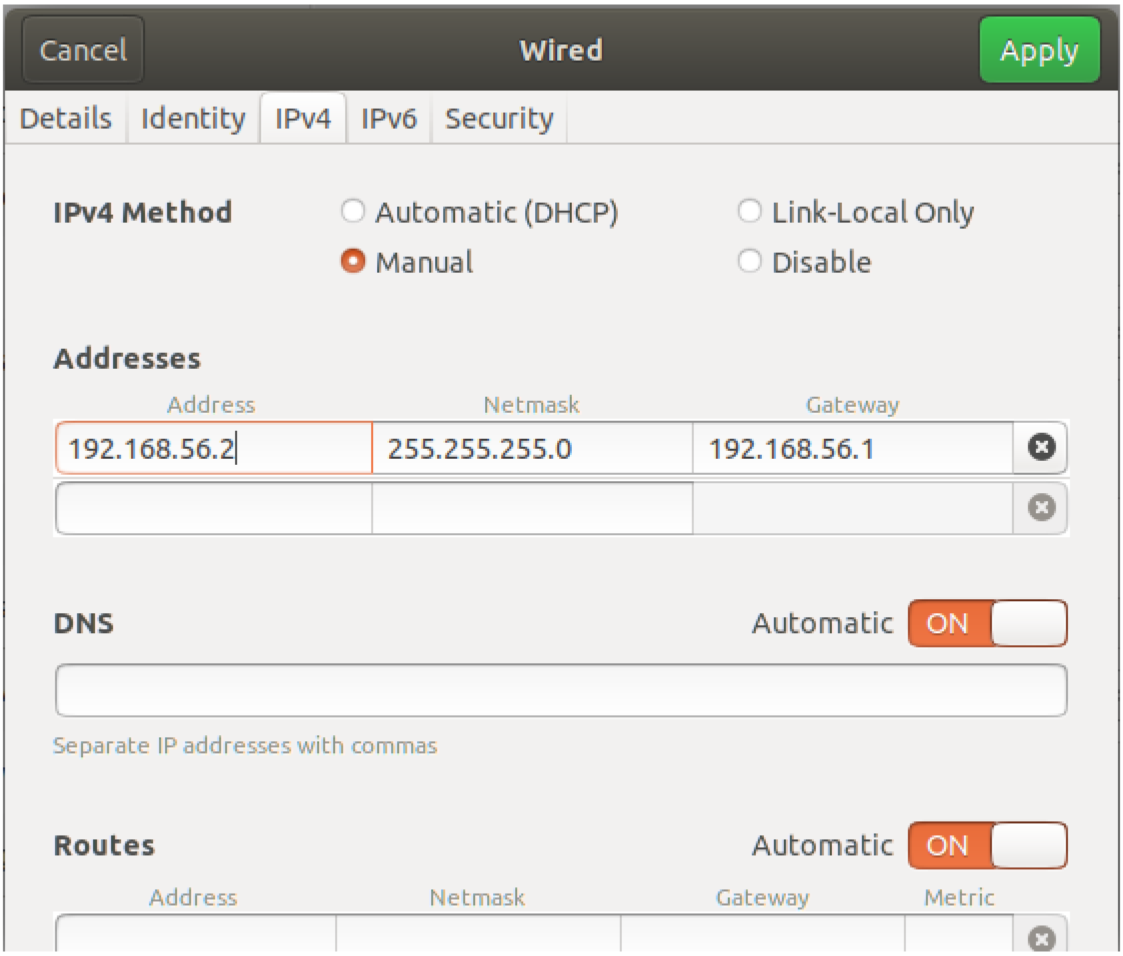
Step 2: Start the VM. First, we will update the Ubuntu repositories by running the following command in the terminal.


server@server-VirtualBox: ~
File Edit View Search Terminal Help

```
server@server-VirtualBox:~$ sudo apt-get update
[sudo] password for server:
Hit:1 http://ca.archive.ubuntu.com/ubuntu bionic InRelease
Hit:2 http://ca.archive.ubuntu.com/ubuntu bionic-updates InRelease
Get:3 http://security.ubuntu.com/ubuntu bionic-security InRelease [88.7 kB]
Hit:4 http://ca.archive.ubuntu.com/ubuntu bionic-backports InRelease
Fetched 88.7 kB in 1s (69.6 kB/s)
Reading package lists... Done
server@server-VirtualBox:~$
```

Step 3: Before we move on, Let's **make sure that the IP address of our server is static** since we don't want the server to assign a random IP address dynamically every time it restarts. In Ubuntu, you can do this by clicking the drop triangle icon ▾ at the top right corner and then choosing **Host-Only Ethernet-> Wired Settings->⚙ icon.**

After you are on the Wired Console go to the IPv4 tab. And choose the manual option under the IPv4 Method. **Put the IPv4 address according to your network ID and subnet.**



IPv4 Method ○ Automatic (DHCP) ○ Link-Local Only ● Manual ○ Disable

Addresses

Address	Netmask	Gateway
192.168.56.2	255.255.255.0	192.168.56.1

DNS Automatic ON

Separate IP addresses with commas

Routes Automatic ON

Address	Netmask	Gateway	Metric

Step 4: Now, run the **$ ifconfig** command to check the interfaces. If the $ ifconfig command is not found, you have to install the net-tools package first as follow

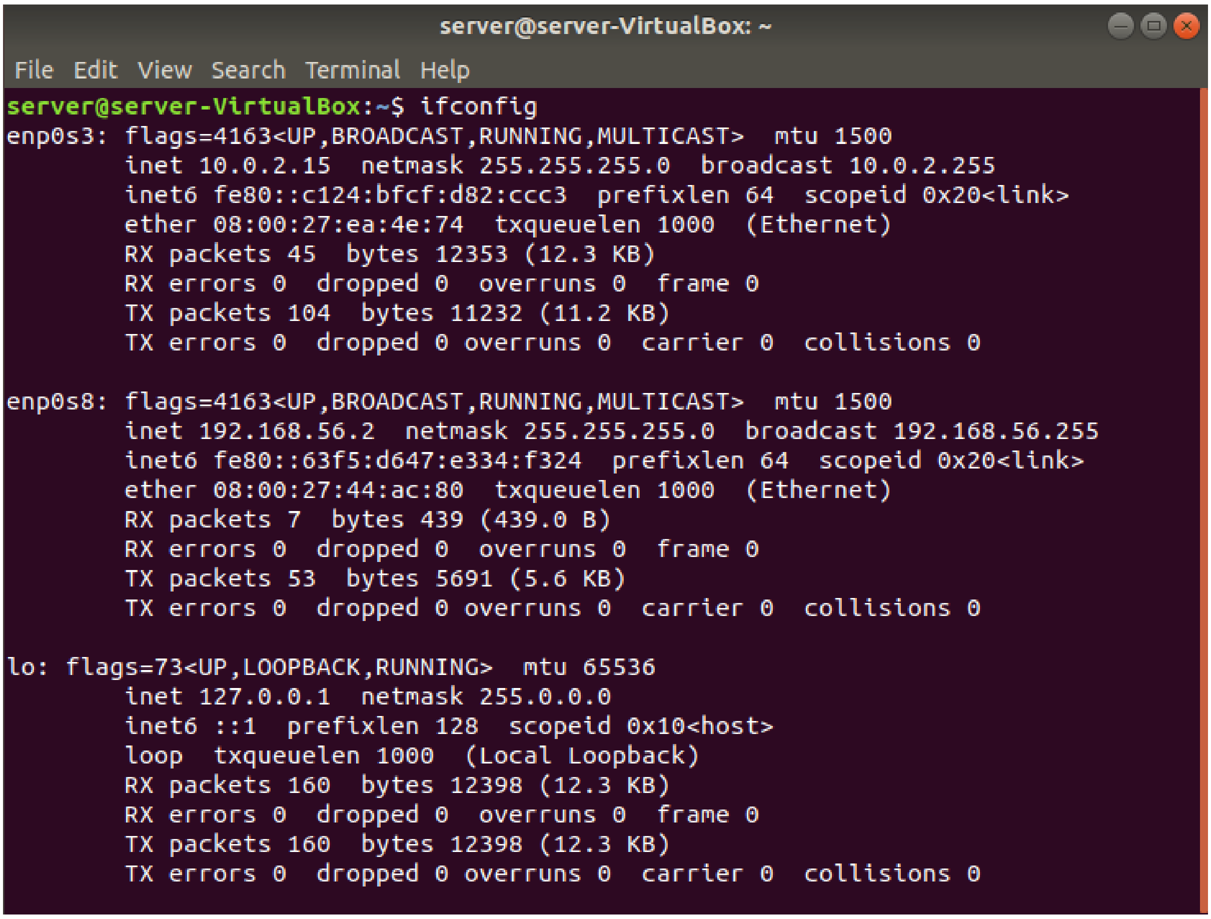
$ sudo apt-get install net-tools

Now, if we run the $ ifconfig, we will see the following:

Here we have two interfaces running: **enp0s3:** NAT Interface:

enp0s8: Host-Only Interface:

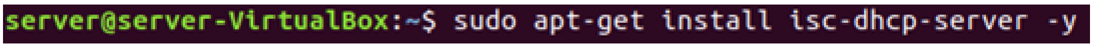
Notice the IP address of the enp0s8 is the same you have configured in the last step.



server@server-VirtualBox: ~
File Edit View Search Terminal Help

```
server@server-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::c124:bfcf:d82:ccc3  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:ea:4e:74  txqueuelen 1000  (Ethernet)
        RX packets 45  bytes 12353 (12.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 104  bytes 11232 (11.2 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.56.2  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::63f5:d647:e334:f324  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:44:ac:80  txqueuelen 1000  (Ethernet)
        RX packets 7  bytes 439 (439.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 53  bytes 5691 (5.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 160  bytes 12398 (12.3 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 160  bytes 12398 (12.3 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Step 5: Further, we will install the DHCP server by running the following command:

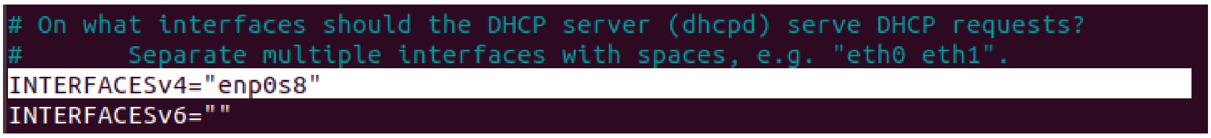


```
server@server-VirtualBox:~$ sudo apt-get install isc-dhcp-server -y
```

Step 6: Next, we will edit the **/etc/default/isc-dhcp-server file** by running the following command:

$ sudo nano /etc/default/isc-dhcp-server

In the file, we will change the value of the **INTERFACESv4 variable to “enp0s8”,** which is the interface for the host-only network.



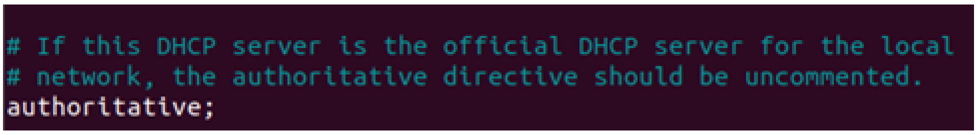
```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACESv4="enp0s8"
INTERFACESv6=""
```

Step 7: Now we will edit the **DHCP configuration file** inside the directory /etc/dhcp/

Run the following command to open the file in the editor:

$ sudo nano /etc/dhcp/dhcpd.conf

In the DHCP configuration file **uncomment the line saying authoritative;** by removing the # sign from the front of the line.



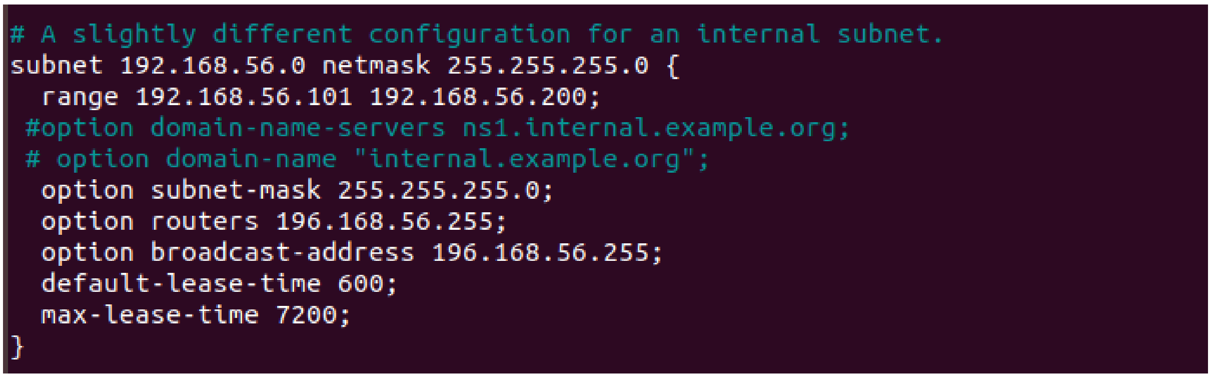
```
# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.
authoritative;
```

First, we set the subnet and netmask to the network Id of the “enp0s8” interface

Then, we will specify the range for the IP addresses that our DHCP server will provide. The range should be according to your usage. If you don’t have the DNS server running yet do not uncomment the domain-name-server lines as shown.

Again, specify the subnet mask. Set the option routers to your broadcast-address.

After the above changes, the code should look like the following except your IP address information might be different than the one shown here.



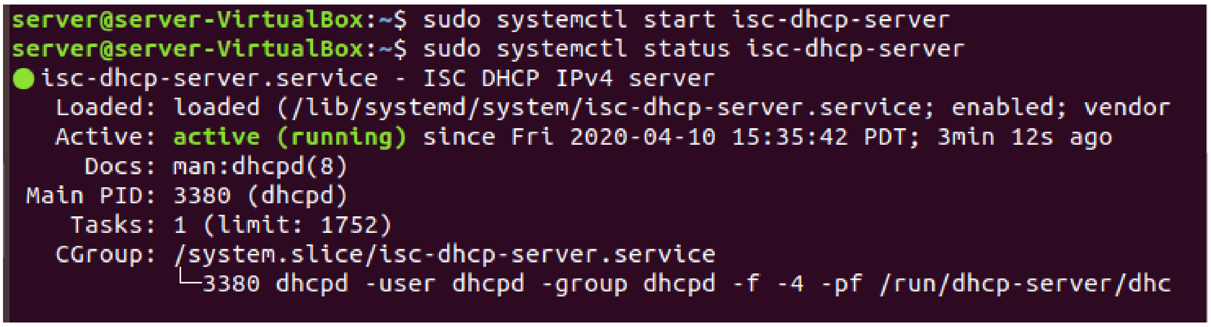
```
# A slightly different configuration for an internal subnet.
subnet 192.168.56.0 netmask 255.255.255.0 {
  range 192.168.56.101 192.168.56.200;
 #option domain-name-servers ns1.internal.example.org;
 # option domain-name "internal.example.org";
  option subnet-mask 255.255.255.0;
  option routers 196.168.56.255;
  option broadcast-address 196.168.56.255;
  default-lease-time 600;
  max-lease-time 7200;
}
```

Step 8: After editing the dhcpd.conf file we will start the DHCP service on the server by running the following command:

$ sudo systemctl start isc-dhcp-server

And to check the status of the DHCP server, run the following command:

$ sudo systemctl status isc-dhcp-server



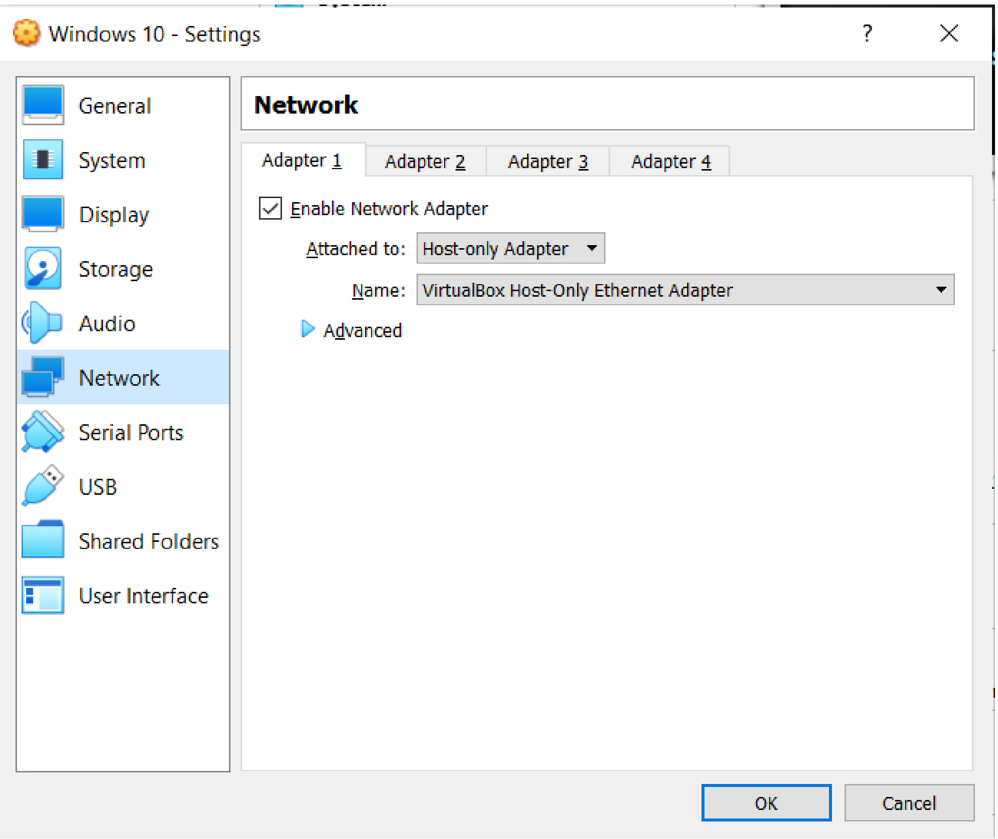
```
server@server-VirtualBox:~$ sudo systemctl start isc-dhcp-server
server@server-VirtualBox:~$ sudo systemctl status isc-dhcp-server
● isc-dhcp-server.service - ISC DHCP IPv4 server
   Loaded: loaded (/lib/systemd/system/isc-dhcp-server.service; enabled; vendor
   Active: active (running) since Fri 2020-04-10 15:35:42 PDT; 3min 12s ago
     Docs: man:dhcpd(8)
 Main PID: 3380 (dhcpd)
    Tasks: 1 (limit: 1752)
   CGroup: /system.slice/isc-dhcp-server.service
           └─3380 dhcpd -user dhcpd -group dhcpd -f -4 -pf /run/dhcp-server/dhc
```

Step 9: To enable the service for the boot time, run the following command.

$ sudo systemctl enable isc-dhcp-server

The DHCP server is configured and running. Now we will configure the client-side.

Before starting, make sure that the Windows 10 VM's network adapter should be set to host-Only.



Windows 10 - Settings

Network

Adapter 1

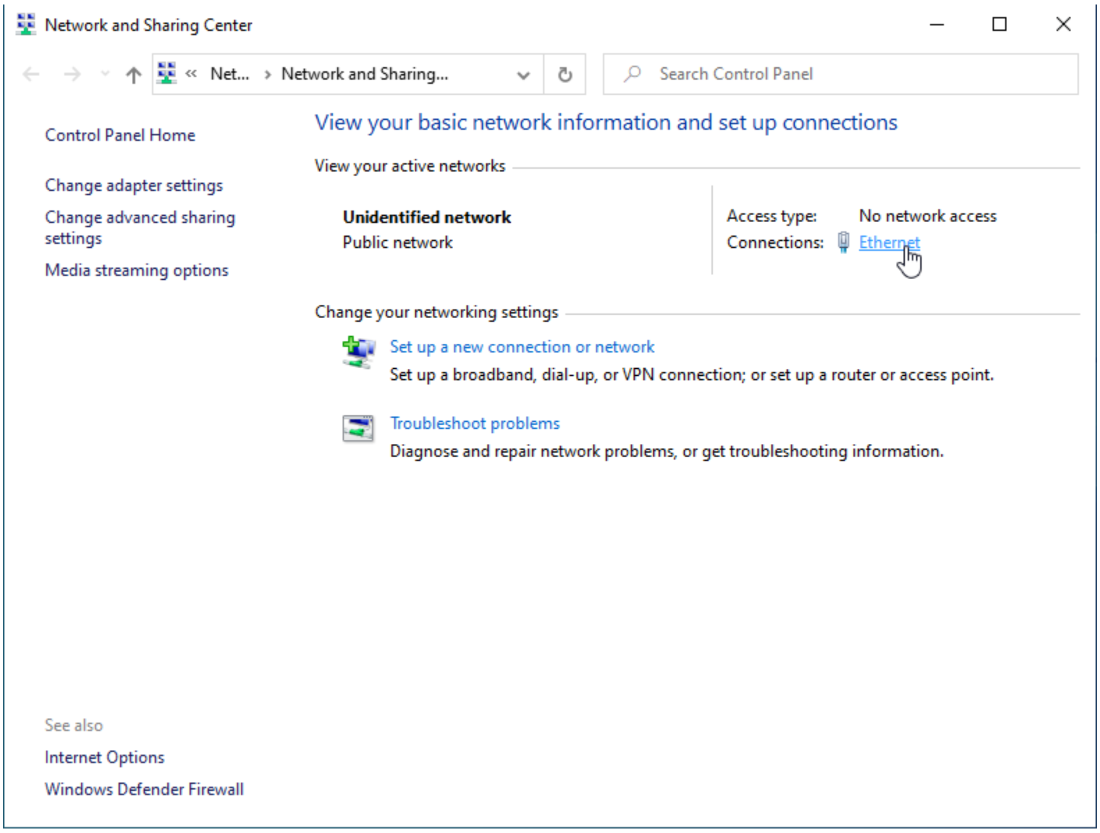
☑ Enable Network Adapter

Attached to: Host-only Adapter

Name: VirtualBox Host-Only Ethernet Adapter

Step 1: Open the windows 10 as a client. Go to the Network and Sharing Center.

Step 2: On the Network and Sharing Center Console, click on the link say Ethernet, as shown below.



Network and Sharing Center
Net... › Network and Sharing...
Search Control Panel
Control Panel Home
Change adapter settings
Change advanced sharing settings
Media streaming options
See also
Internet Options
Windows Defender Firewall

View your basic network information and set up connections

View your active networks

Unidentified network
Public network

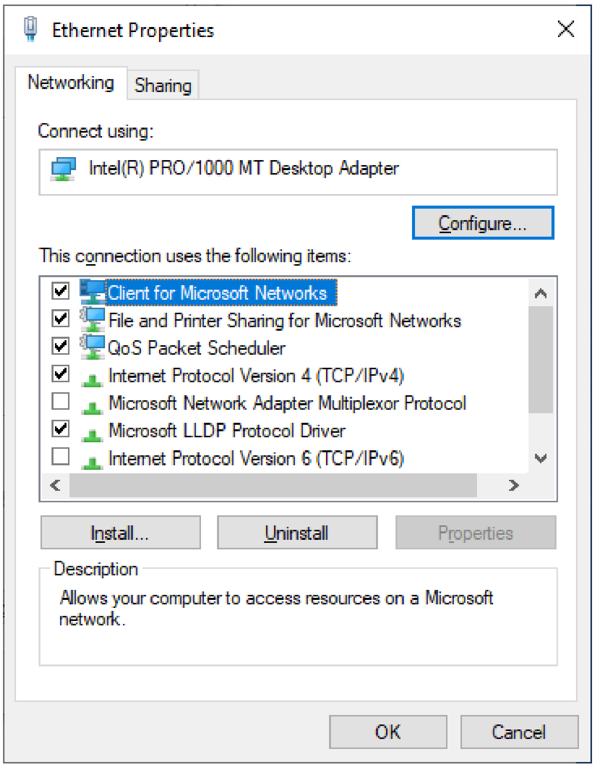
Access type: No network access
Connections: Ethernet

Change your networking settings

Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

Step 3: An ethernet status window shows up. Click on the Properties button. It may require an **administrator privilege**. After putting the administrator credentials, the following window shows up.



Ethernet Properties

Networking | Sharing

Connect using:

Intel(R) PRO/1000 MT Desktop Adapter

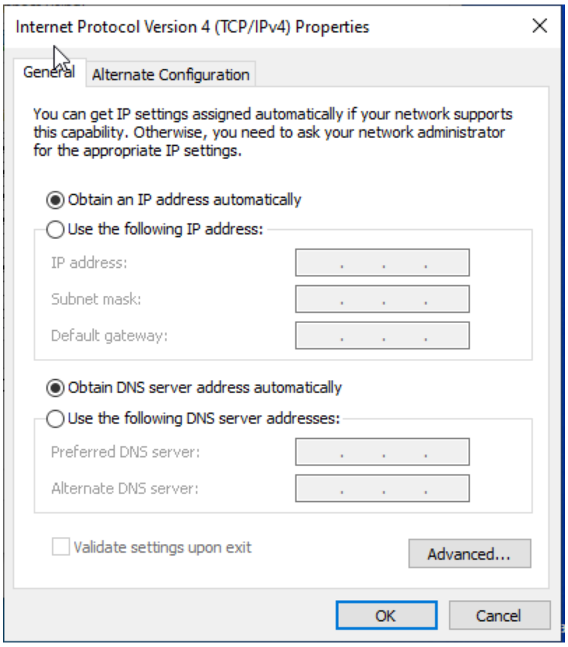
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☐ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☐ Internet Protocol Version 6 (TCP/IPv6)

Description

Allows your computer to access resources on a Microsoft network.

Step 4: Now, double click on the Item saying Internet Protocol Version 4(TCP/ IPv4). Choose *Obtain an IP address automatically* and *Obtain DNS server address automatically*.



Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

● Obtain DNS server address automatically

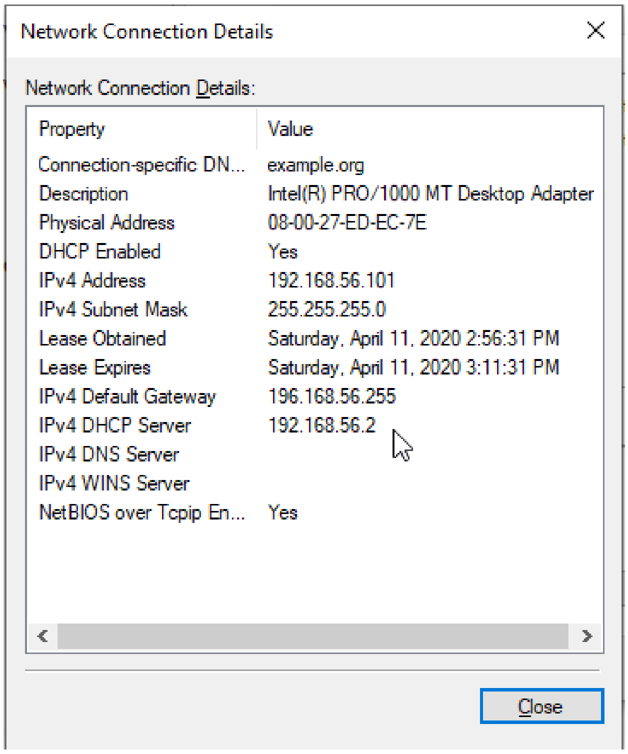
○ Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

☐ Validate settings upon exit

Step 5: Next return to the Ethernet Status window console by clicking OK. Now, click the Details button. As you can see that the IP address of the computer is now within the range, we specify on our DHCP server. And also, the IP address of the DHCP server is the same that we configured before.



Network Connection Details:

Property	Value
Connection-specific DN...	example.org
Description	Intel(R) PRO/1000 MT Desktop Adapter
Physical Address	08-00-27-ED-EC-7E
DHCP Enabled	Yes
IPv4 Address	192.168.56.101
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Saturday, April 11, 2020 2:56:31 PM
Lease Expires	Saturday, April 11, 2020 3:11:31 PM
IPv4 Default Gateway	196.168.56.255
IPv4 DHCP Server	192.168.56.2
IPv4 DNS Server	
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes

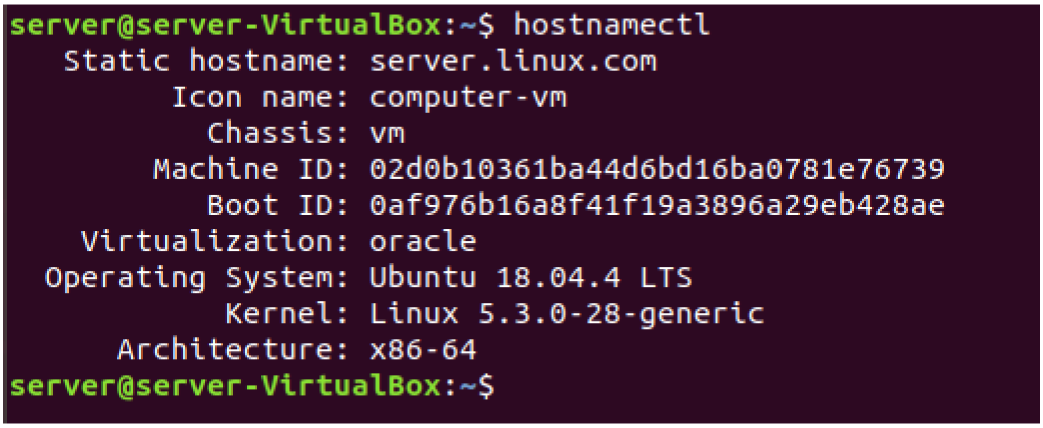
Now, we have configured our client to get the IP address dynamically.

Deploy DNS Name Server

Step 1: Configure the hostname of the server to be static. We are using **“server.linux.com”** as our hostname. To configure the hostname run the following command:

$ hostnamectl set-hostname server.linux.com

To check your current hostname, use the following command:



```
server@server-VirtualBox:~$ hostnamectl
   Static hostname: server.linux.com
         Icon name: computer-vm
           Chassis: vm
        Machine ID: 02d0b10361ba44d6bd16ba0781e76739
           Boot ID: 0af976b16a8f41f19a3896a29eb428ae
    Virtualization: oracle
  Operating System: Ubuntu 18.04.4 LTS
            Kernel: Linux 5.3.0-28-generic
      Architecture: x86-64
server@server-VirtualBox:~$
```

Step 2: Run the following command to install packages **“bind9”** and **“bind9utils”**.

$ sudo apt-get install bind9 bind9utils

Step 3: Now we will start editing the files. First, edit the **/etc/bind/named.conf.local** file by running the following command:

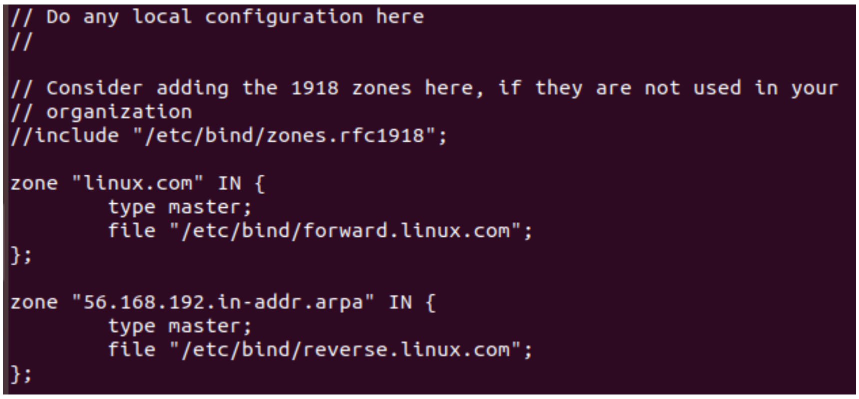
$ sudo nano /etc/bind/named.conf.local

We will create two zones in this file. Forward Zone and Reverse Zone.

Forward Zone: In this, the name will map to the IP address.

Reverse Zone: In this, the IP address will map to the name.

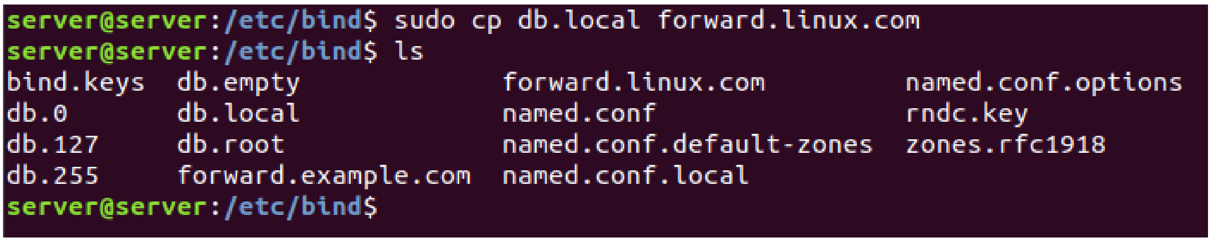
After the edits your file should look as follows:



```
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "linux.com" IN {
        type master;
        file "/etc/bind/forward.linux.com";
};

zone "56.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/reverse.linux.com";
};
```

Note that in the second zone the numbers 56.168.192 correspond to the reverse order of your IP address. We don’t include the number from the last octet.

Step 4: Further, Copy the **db.local** file to **forward.linux.com** by using the following command (we have changed the directory to /etc/bind/):

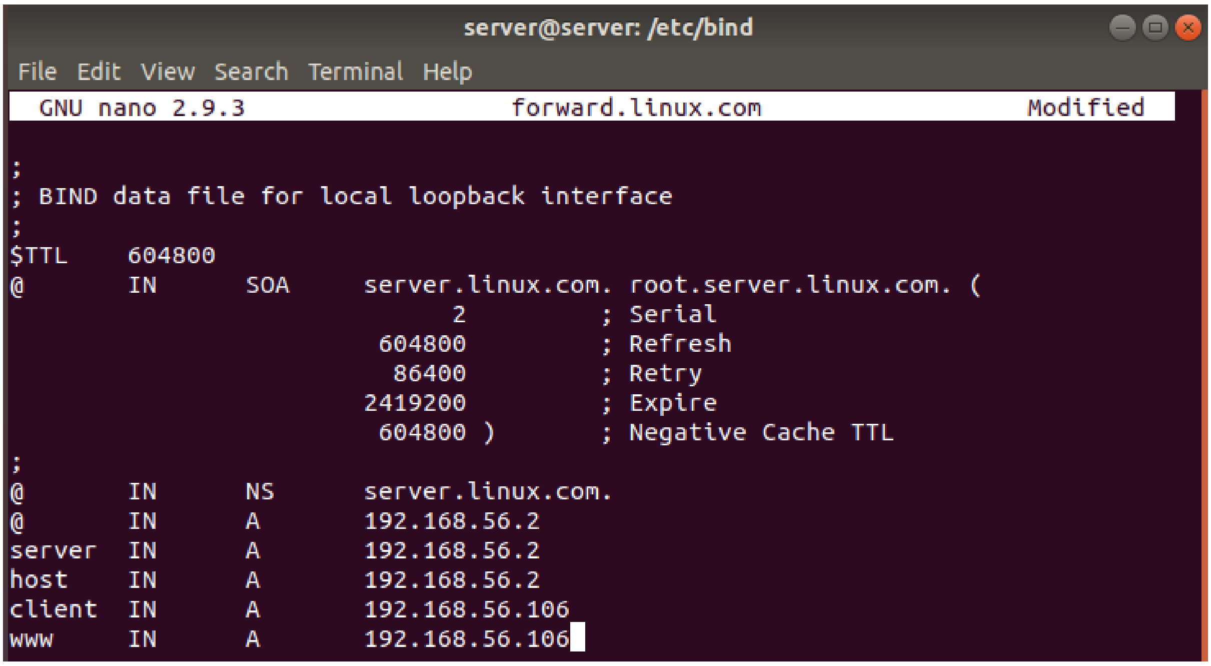


```
server@server:/etc/bind$ sudo cp db.local forward.linux.com
server@server:/etc/bind$ ls
bind.keys  db.empty            forward.linux.com         named.conf.options
db.0       db.local            named.conf                rndc.key
db.127     db.root             named.conf.default-zones  zones.rfc1918
db.255     forward.example.com named.conf.local
server@server:/etc/bind$
```

As you can see, now we have a new **forward.linux.com** file in our bind directory.

Step 5: Next we will edit the **forward.linux.com** file as follow:

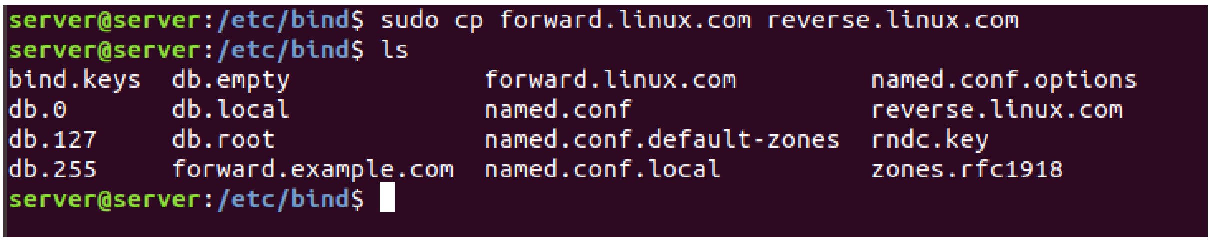
$ sudo nano forward.linux.com



```
  GNU nano 2.9.3               forward.linux.com                    Modified

;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     server.linux.com. root.server.linux.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      server.linux.com.
@       IN      A       192.168.56.2
server  IN      A       192.168.56.2
host    IN      A       192.168.56.2
client  IN      A       192.168.56.106
www     IN      A       192.168.56.106
```

Step 6: Next we will copy **forward.linux.com** file to **reverse.linux.com**. And do some **edits** to reverse.linux.com file

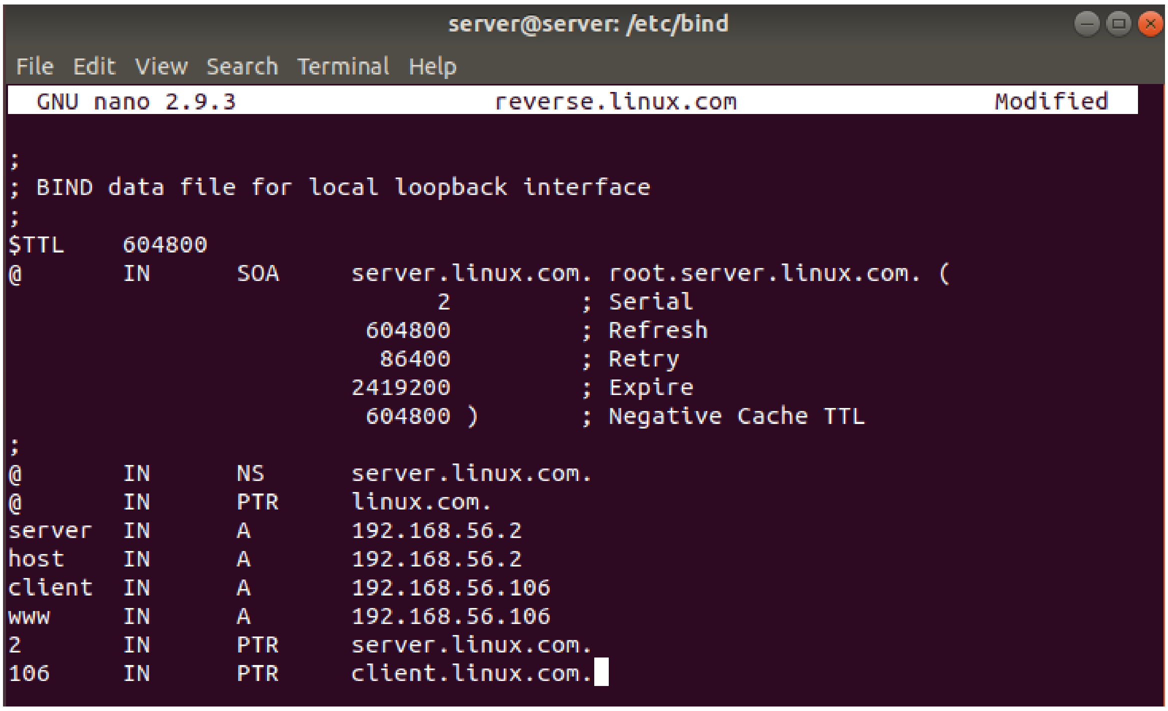
Copy File:



```
server@server:/etc/bind$ sudo cp forward.linux.com reverse.linux.com
server@server:/etc/bind$ ls
bind.keys  db.empty            forward.linux.com         named.conf.options
db.0       db.local            named.conf                reverse.linux.com
db.127     db.root             named.conf.default-zones  rndc.key
db.255     forward.example.com named.conf.local          zones.rfc1918
server@server:/etc/bind$
```

Do the following edits:

$ sudo nano reverse.linux.com

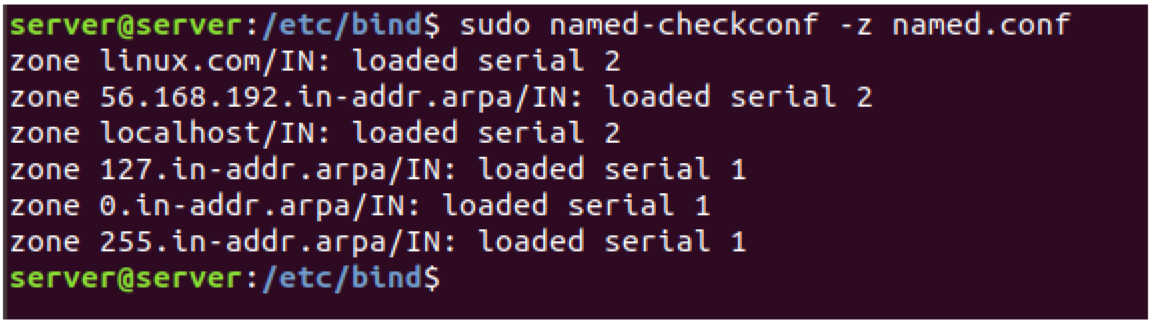


```
  GNU nano 2.9.3                  reverse.linux.com                  Modified

;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     server.linux.com. root.server.linux.com. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      server.linux.com.
@       IN      PTR     linux.com.
server  IN      A       192.168.56.2
host    IN      A       192.168.56.2
client  IN      A       192.168.56.106
www     IN      A       192.168.56.106
2       IN      PTR     server.linux.com.
106     IN      PTR     client.linux.com.
```

Note: we are adding the client just for the testing purpose.

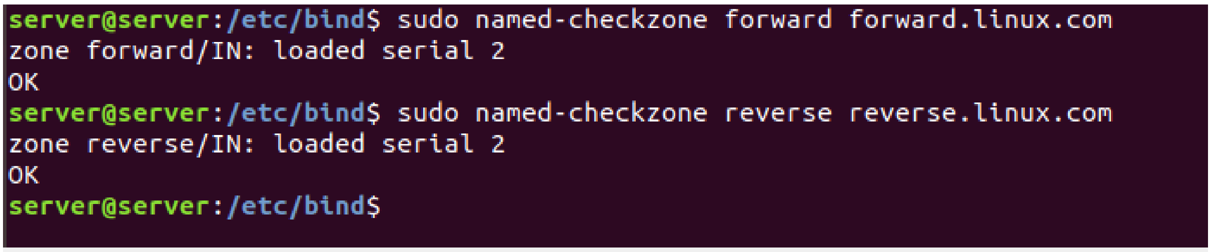
Step 7: Now, we will check our configurations to find if there is any syntax error. Run the following commands:

To check named.conf.local file:



```
server@server:/etc/bind$ sudo named-checkconf -z named.conf
zone linux.com/IN: loaded serial 2
zone 56.168.192.in-addr.arpa/IN: loaded serial 2
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
server@server:/etc/bind$
```

To check the zones:



```
server@server:/etc/bind$ sudo named-checkzone forward forward.linux.com
zone forward/IN: loaded serial 2
OK
server@server:/etc/bind$ sudo named-checkzone reverse reverse.linux.com
zone reverse/IN: loaded serial 2
OK
server@server:/etc/bind$
```

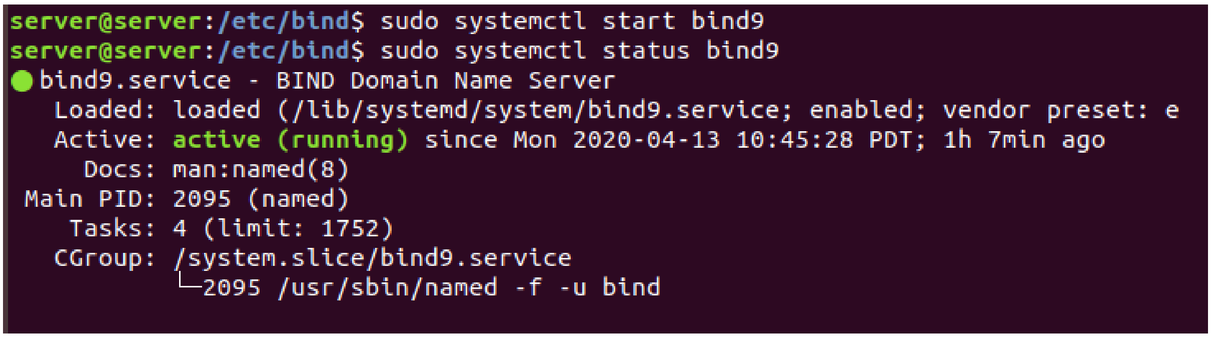
Step 8: Before we start the bind9 service, we will have to change the ownership of the files.

Use the following commands to do so:

$ sudo chown -R bind:bind /etc/bind

$ sudo chmod -R 755 /etc/bind

Step 9: Next we will start the bind9 service by the following command:



```
server@server:/etc/bind$ sudo systemctl start bind9
server@server:/etc/bind$ sudo systemctl status bind9
● bind9.service - BIND Domain Name Server
   Loaded: loaded (/lib/systemd/system/bind9.service; enabled; vendor preset: e
   Active: active (running) since Mon 2020-04-13 10:45:28 PDT; 1h 7min ago
     Docs: man:named(8)
 Main PID: 2095 (named)
    Tasks: 4 (limit: 1752)
   CGroup: /system.slice/bind9.service
           └─2095 /usr/sbin/named -f -u bind
```

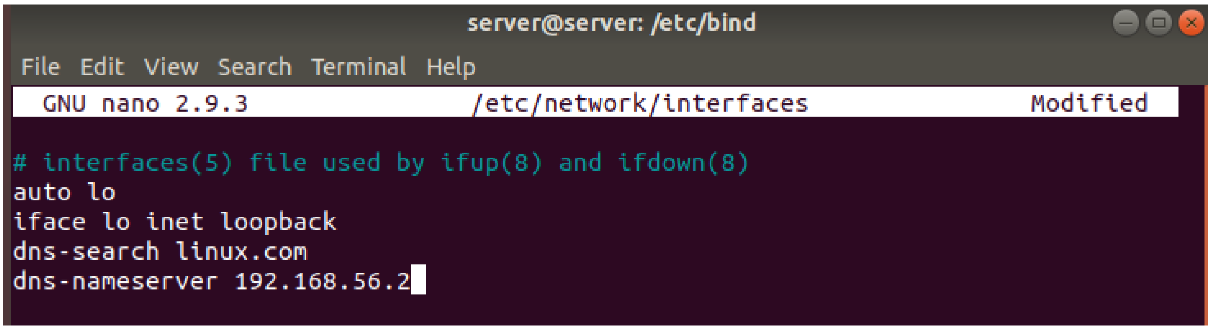
$ sudo systemctl enable bind9

Step 10: Run the following command to allow bind service through the firewall:

$ sudo ufw allow bind9

Step 11: Add the following lines to the interfaces file at /etc/network/interfaces

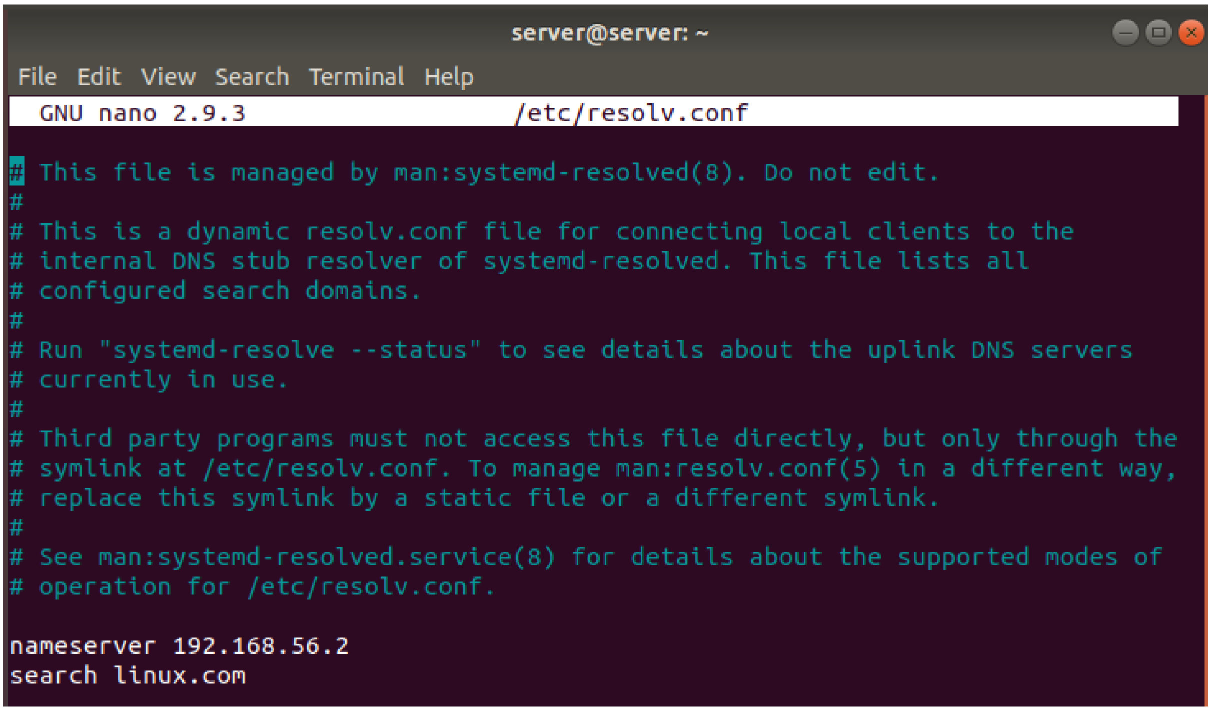
$ sudo nano /etc/network/interfaces



```
  GNU nano 2.9.3              /etc/network/interfaces              Modified

# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
dns-search linux.com
dns-nameserver 192.168.56.2
```

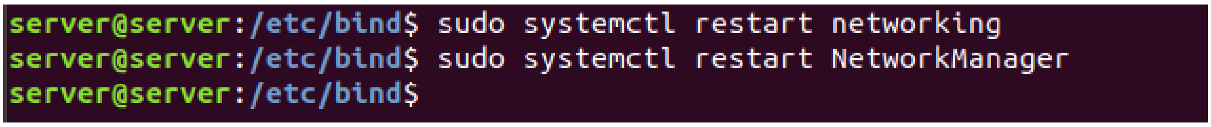
Step 12: Also, edit the **/etc/resolv.conf** as follow:

$ sudo nano /etc/resolv.conf



```
  GNU nano 2.9.3                     /etc/resolv.conf

# This file is managed by man:systemd-resolved(8). Do not edit.
#
# This is a dynamic resolv.conf file for connecting local clients to the
# internal DNS stub resolver of systemd-resolved. This file lists all
# configured search domains.
#
# Run "systemd-resolve --status" to see details about the uplink DNS servers
# currently in use.
#
# Third party programs must not access this file directly, but only through the
# symlink at /etc/resolv.conf. To manage man:resolv.conf(5) in a different way,
# replace this symlink by a static file or a different symlink.
#
# See man:systemd-resolved.service(8) for details about the supported modes of
# operation for /etc/resolv.conf.

nameserver 192.168.56.2
search linux.com
```

Step 13: Now restart the networking and NetworkManager services as shown below:

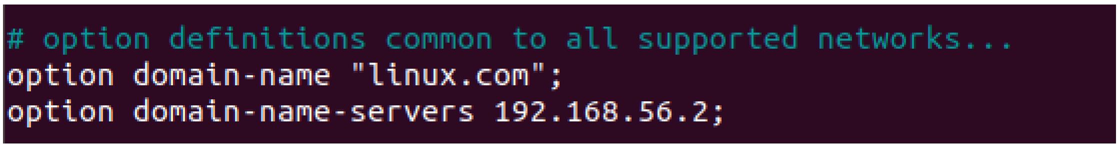


```
server@server:/etc/bind$ sudo systemctl restart networking
server@server:/etc/bind$ sudo systemctl restart NetworkManager
server@server:/etc/bind$
```

Step 14: Now we will edit the **DHCP configuration file** inside the directory /etc/dhcp/ to change the DNS server.

Run the following command to open the file in the editor:

$ sudo nano /etc/dhcp/dhcpd.conf



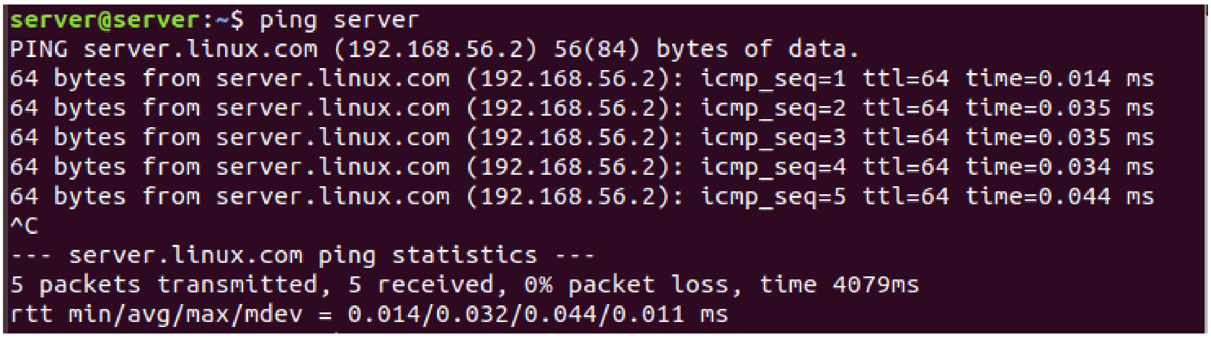
```
# option definitions common to all supported networks...
option domain-name "linux.com";
option domain-name-servers 192.168.56.2;
```

Now restart the DHCP and DNS services:

$ sudo systemctl restart isc-dhcp-server

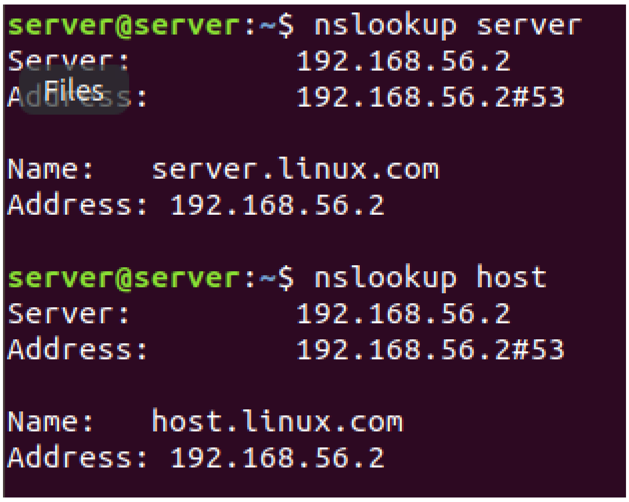
$ sudo systemctl restart bind9

Step 15: Next we will check if our DNS server has been configured right. Run the commands as shown:



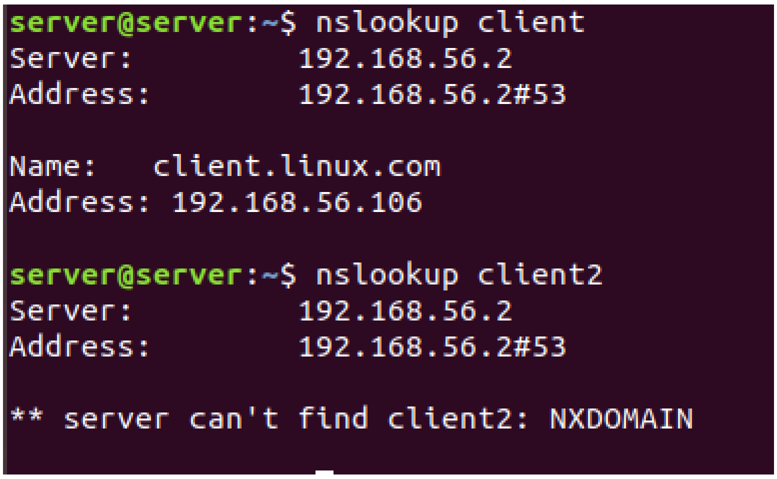
```
server@server:~$ ping server
PING server.linux.com (192.168.56.2) 56(84) bytes of data.
64 bytes from server.linux.com (192.168.56.2): icmp_seq=1 ttl=64 time=0.014 ms
64 bytes from server.linux.com (192.168.56.2): icmp_seq=2 ttl=64 time=0.035 ms
64 bytes from server.linux.com (192.168.56.2): icmp_seq=3 ttl=64 time=0.035 ms
64 bytes from server.linux.com (192.168.56.2): icmp_seq=4 ttl=64 time=0.034 ms
64 bytes from server.linux.com (192.168.56.2): icmp_seq=5 ttl=64 time=0.044 ms
^C
--- server.linux.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 0.014/0.032/0.044/0.011 ms
```

With the **nslookup command** we can see that the server is resolved server.linux.com by the DNS.



```
server@server:~$ nslookup server
Server:         192.168.56.2
Address:        192.168.56.2#53

Name:   server.linux.com
Address: 192.168.56.2

server@server:~$ nslookup host
Server:         192.168.56.2
Address:        192.168.56.2#53

Name:   host.linux.com
Address: 192.168.56.2
```

Testing the client with **nslookup:**



```
server@server:~$ nslookup client
Server:         192.168.56.2
Address:        192.168.56.2#53

Name:   client.linux.com
Address: 192.168.56.106

server@server:~$ nslookup client2
Server:         192.168.56.2
Address:        192.168.56.2#53

** server can't find client2: NXDOMAIN
```

As you can see, we have only set up the **client** so it resolves to client.linux.com but we have not set up the **client2** so our DNS server can't find it

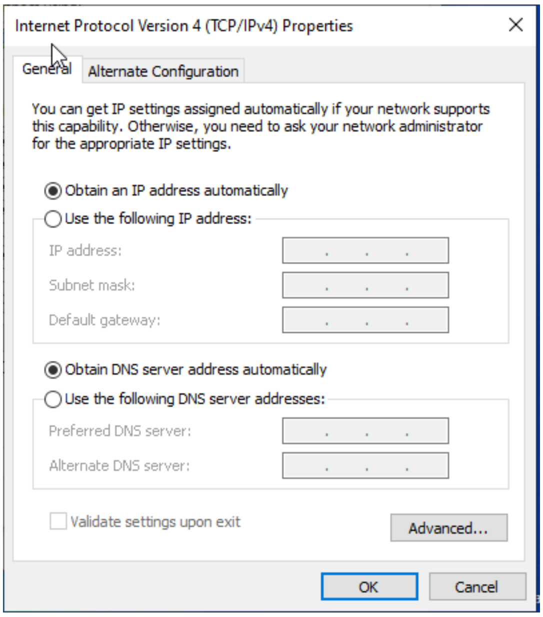
Testing the Windows 10 client

Step 1: Open the Windows 10 VM. Go to the **Network and Sharing Center-> Ethernet-> properties button**. Provide the administrator credentials, if required.

Step 2: Now, double click on the Item saying **Internet Protocol Version 4(TCP/ IPv4)**. Make sure these options are selected:

Choose *Obtain an IP address automatically*

Obtain DNS server address automatically.



Internet Protocol Version 4 (TCP/IPv4) Properties

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

● Obtain DNS server address automatically

○ Use the following DNS server addresses:

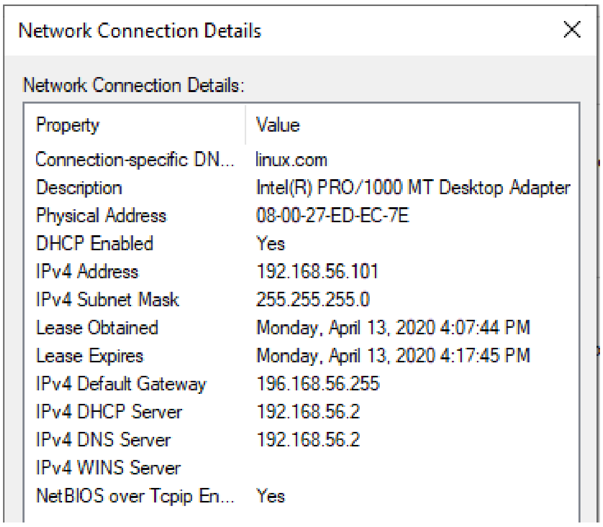
Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Next return to the Ethernet Status window console by clicking OK.

Step 3: Next, click the Details button. As you can see, the IP address of the DNS server is the same that we just configured.

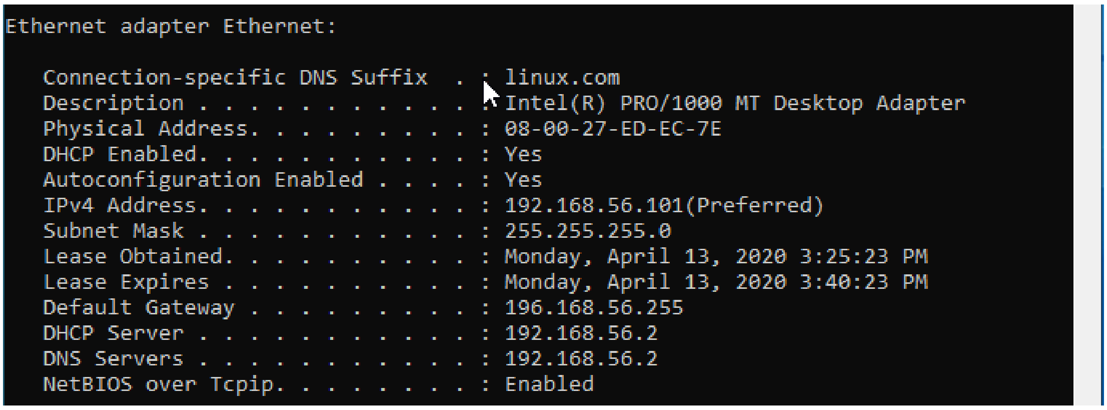


Network Connection Details

Network Connection Details:

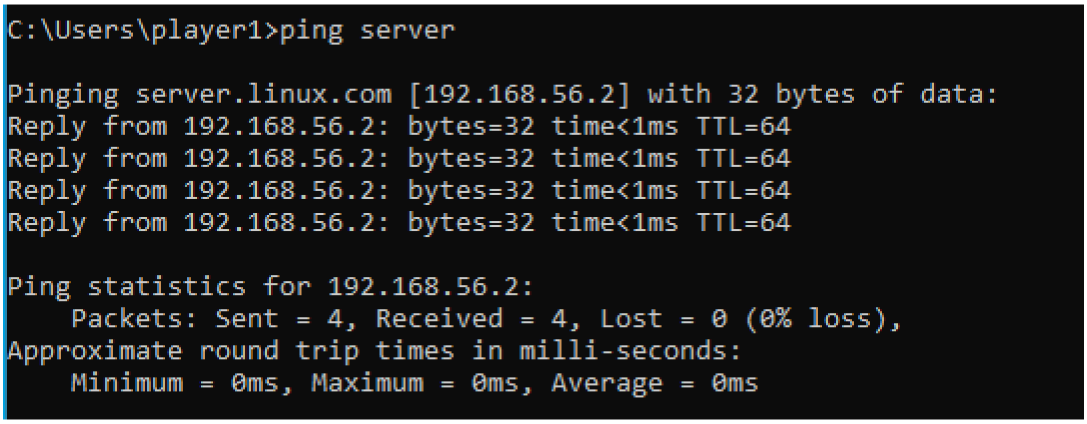
Property	Value
Connection-specific DN...	linux.com
Description	Intel(R) PRO/1000 MT Desktop Adapter
Physical Address	08-00-27-ED-EC-7E
DHCP Enabled	Yes
IPv4 Address	192.168.56.101
IPv4 Subnet Mask	255.255.255.0
Lease Obtained	Monday, April 13, 2020 4:07:44 PM
Lease Expires	Monday, April 13, 2020 4:17:45 PM
IPv4 Default Gateway	196.168.56.255
IPv4 DHCP Server	192.168.56.2
IPv4 DNS Server	192.168.56.2
IPv4 WINS Server	
NetBIOS over Tcpip En...	Yes

Step 4: Now, open the command and run the **ipconfig** command to check if the DNS server is identified.



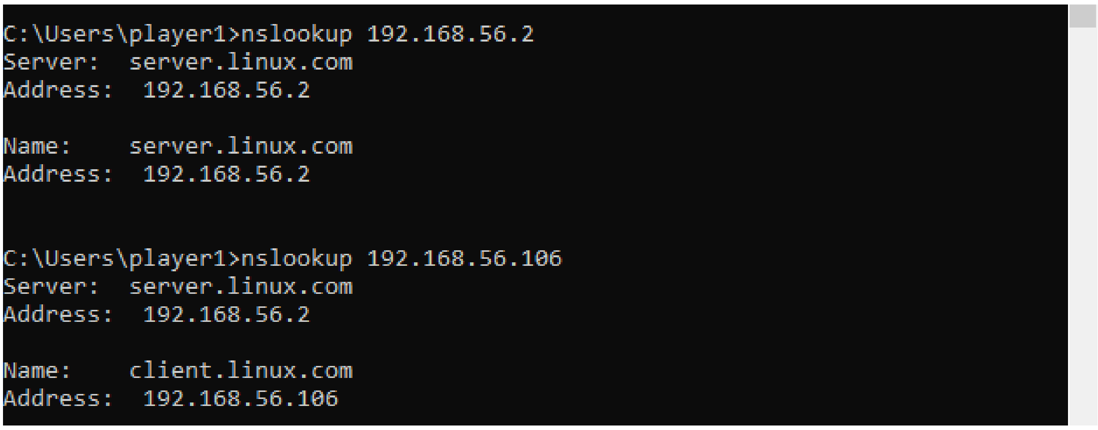
```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : linux.com
   Description . . . . . . . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
   Physical Address. . . . . . . . . : 08-00-27-ED-EC-7E
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv4 Address. . . . . . . . . . . : 192.168.56.101(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Monday, April 13, 2020 3:25:23 PM
   Lease Expires . . . . . . . . . . : Monday, April 13, 2020 3:40:23 PM
   Default Gateway . . . . . . . . . : 196.168.56.255
   DHCP Server . . . . . . . . . . . : 192.168.56.2
   DNS Servers . . . . . . . . . . . : 192.168.56.2
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

Step 5: Ping the server:



```
C:\Users\player1>ping server

Pinging server.linux.com [192.168.56.2] with 32 bytes of data:
Reply from 192.168.56.2: bytes=32 time<1ms TTL=64
Reply from 192.168.56.2: bytes=32 time<1ms TTL=64
Reply from 192.168.56.2: bytes=32 time<1ms TTL=64
Reply from 192.168.56.2: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.56.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Step 6: Test with the **nslookup**:



```
C:\Users\player1>nslookup 192.168.56.2
Server:  server.linux.com
Address:  192.168.56.2

Name:    server.linux.com
Address:  192.168.56.2


C:\Users\player1>nslookup 192.168.56.106
Server:  server.linux.com
Address:  192.168.56.2

Name:    client.linux.com
Address:  192.168.56.106
```

Hence, we can see that the IP address of our server and test client is being resolved to their appropriate domain names.

Install and Configure the Apache Web Server

Step 1: Install the Apache package on Ubuntu by the following command.

$ sudo apt-get install apache2 -y

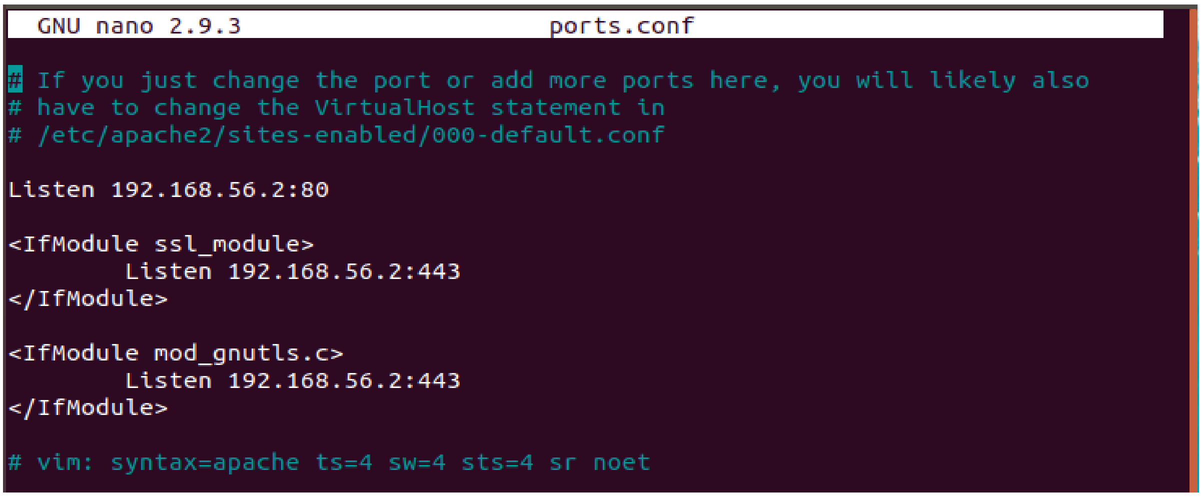
Basics commands for controlling Apache Service:

Stop Apache:	**sudo systemctl stop apache2.service**
Start Apache:	**sudo systemctl start apache2.service**
Restart Apache:	**sudo systemctl restart apache2.service**
Reload Apache:	**sudo systemctl reload apache2.service**

Step 2: Next, we will edit the **ports.conf**. Run the following command to open the file in the editor.

$ sudo nano /etc/apache2/port.conf

Since we want our server to have the host-only IP address, we will do the following edits:



```
  GNU nano 2.9.3                         ports.conf

# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 192.168.56.2:80

<IfModule ssl_module>
        Listen 192.168.56.2:443
</IfModule>

<IfModule mod_gnutls.c>
        Listen 192.168.56.2:443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

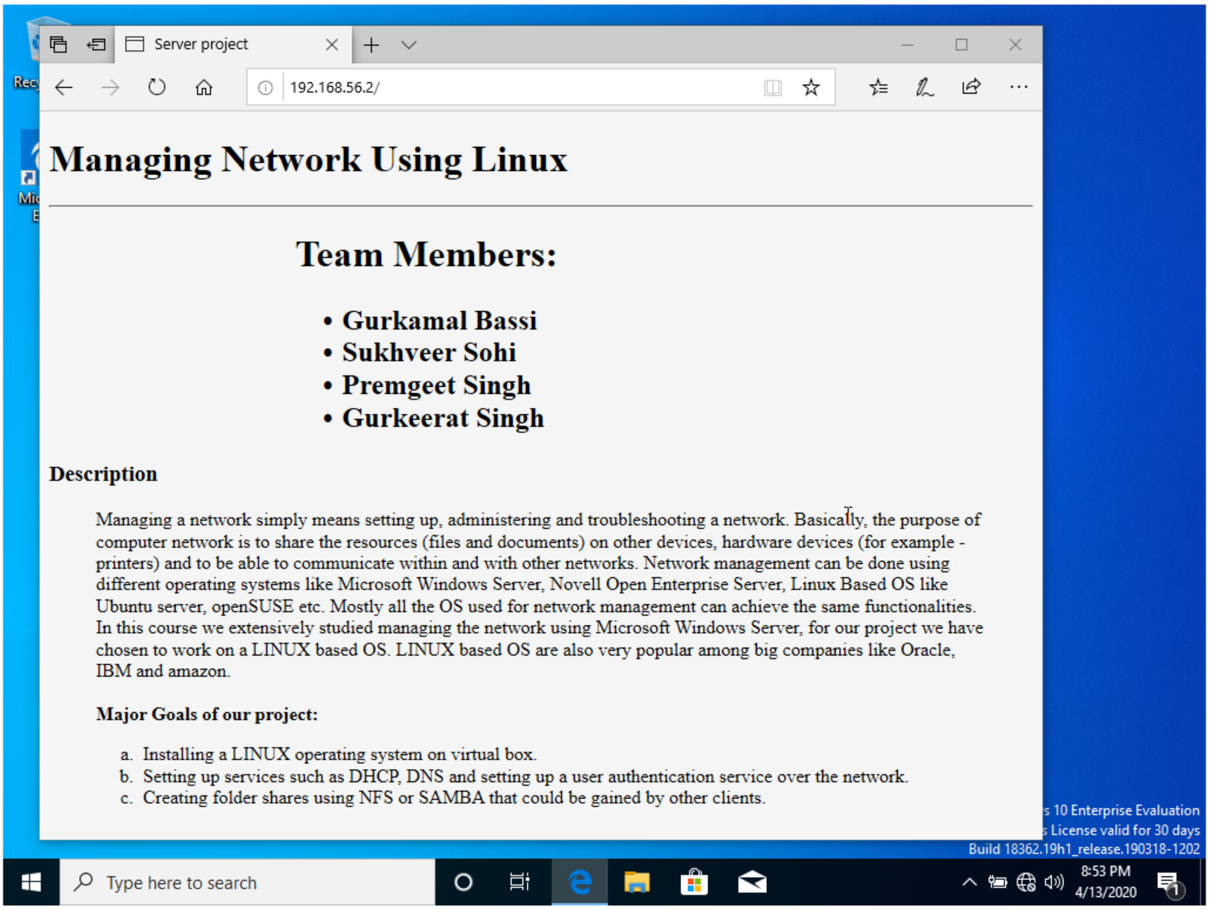
Step 3: Now, let’s restart our server. Run the following:

$ sudo systemctl restart apache2.service

Step 4: Now let’s make sure that our **firewall** is configured to allow traffic on port 80. Use the following command.

$ sudo ufw allow ‘Apache’

Step 5: Now, Open the **windows 10 client machine**. Open the Edge Browser and put the IP address of the Apache Server. The following webpage will show up.



Managing Network Using Linux

Team Members:

- Gurkamal Bassi
- Sukhveer Sohi
- Premgeet Singh
- Gurkeerat Singh

Description

Managing a network simply means setting up, administering and troubleshooting a network. Basically, the purpose of computer network is to share the resources (files and documents) on other devices, hardware devices (for example - printers) and to be able to communicate within and with other networks. Network management can be done using different operating systems like Microsoft Windows Server, Novell Open Enterprise Server, Linux Based OS like Ubuntu server, openSUSE etc. Mostly all the OS used for network management can achieve the same functionalities. In this course we extensively studied managing the network using Microsoft Windows Server, for our project we have chosen to work on a LINUX based OS. LINUX based OS are also very popular among big companies like Oracle, IBM and amazon.

Major Goals of our project:

a. Installing a LINUX operating system on virtual box.
b. Setting up services such as DHCP, DNS and setting up a user authentication service over the network.
c. Creating folder shares using NFS or SAMBA that could be gained by other clients.

Our Apache Server is Up and Running!!

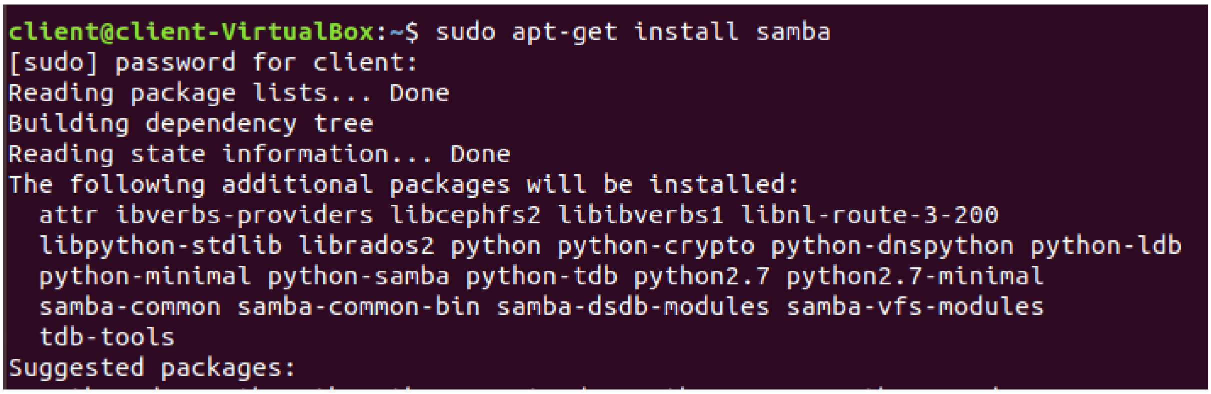
Configure the File Sharing on Ubuntu Server

We are using Samba for implementing File Sharing.

Samba is a free and open-source. It is based on the SMB/CIFS network file sharing protocol which enables the users to access files, as well as other shared resources.

Here are the steps on how to configure File Sharing on the Server.

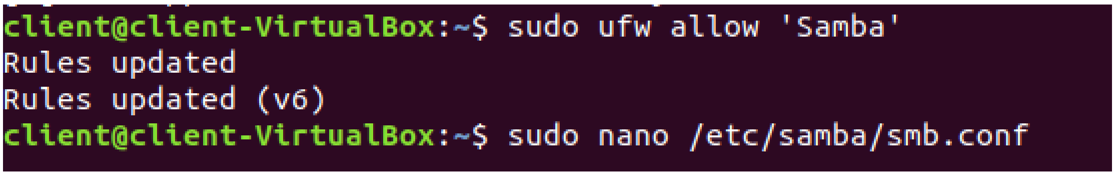
Step 1: Install the samba by using the command as shown below:



```
client@client-VirtualBox:~$ sudo apt-get install samba
[sudo] password for client:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  attr ibverbs-providers libcephfs2 libibverbs1 libnl-route-3-200
  libpython-stdlib librados2 python python-crypto python-dnspython python-ldb
  python-minimal python-samba python-tdb python2.7 python2.7-minimal
  samba-common samba-common-bin samba-dsdb-modules samba-vfs-modules
  tdb-tools
Suggested packages:
```

Step 2: After installation, we will open the ports to allow incoming UDP connections.

Run the command as shown below to configure your firewall.

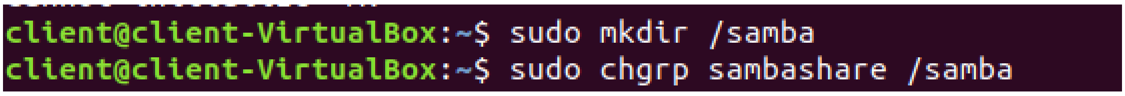


```
client@client-VirtualBox:~$ sudo ufw allow 'Samba'
Rules updated
Rules updated (v6)
client@client-VirtualBox:~$ sudo nano /etc/samba/smb.conf
```

Step 3: Now create a Samba Directory Structure.

First, we will create the /samba directory type and set its group ownership to a samba share. This group is created during the Samba installation.

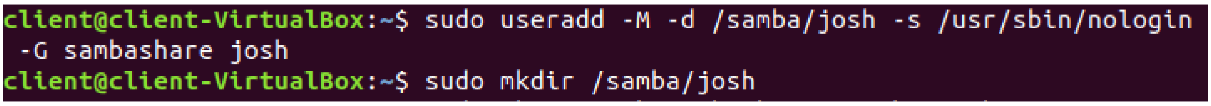
Run the following commands:



```
client@client-VirtualBox:~$ sudo mkdir /samba
client@client-VirtualBox:~$ sudo chgrp sambashare /samba
```

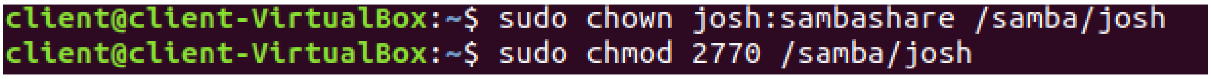
Step 4: Next, create a user using the standard Linux **useradd** tool as well as set the user password with **smbpasswd** utility.

Use the following commands to create a user named ‘josh’ and his home directory /samba/josh:



```
client@client-VirtualBox:~$ sudo useradd -M -d /samba/josh -s /usr/sbin/nologin
 -G sambashare josh
client@client-VirtualBox:~$ sudo mkdir /samba/josh
```

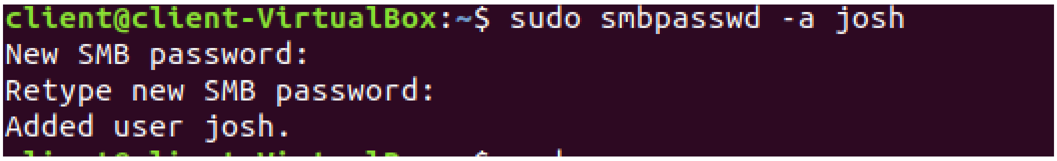
Step 5: Now, set the ownership of the directory to user josh and group **sambashare**:



```
client@client-VirtualBox:~$ sudo chown josh:sambashare /samba/josh
client@client-VirtualBox:~$ sudo chmod 2770 /samba/josh
```

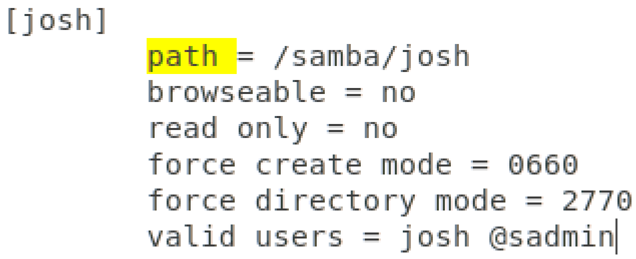
Step 6: Further, set the user password to add the user account to the Samba database:

Enter and confirm the user password.



```
client@client-VirtualBox:~$ sudo smbpasswd -a josh
New SMB password:
Retype new SMB password:
Added user josh.
```

Step 7: Next, open the Samba configuration file and append a section as shown:



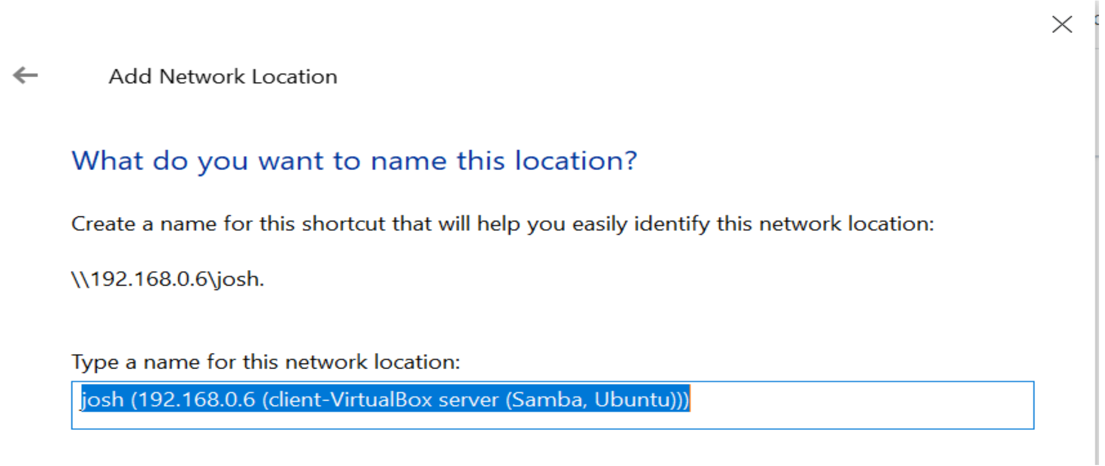
```
[josh]
        path = /samba/josh
        browseable = no
        read only = no
        force create mode = 0660
        force directory mode = 2770
        valid users = josh @sadmin
```

Connecting to the Samba share from Windows 10

Step 1: Open the Windows 10 VM. Then, open the File Explorer. Right-click on "This PC".

Step 2: Select "Choose a custom network location" and then click "Next".

Step 3: In "Internet or network address", enter the address of the Samba share as shown below:



What do you want to name this location?

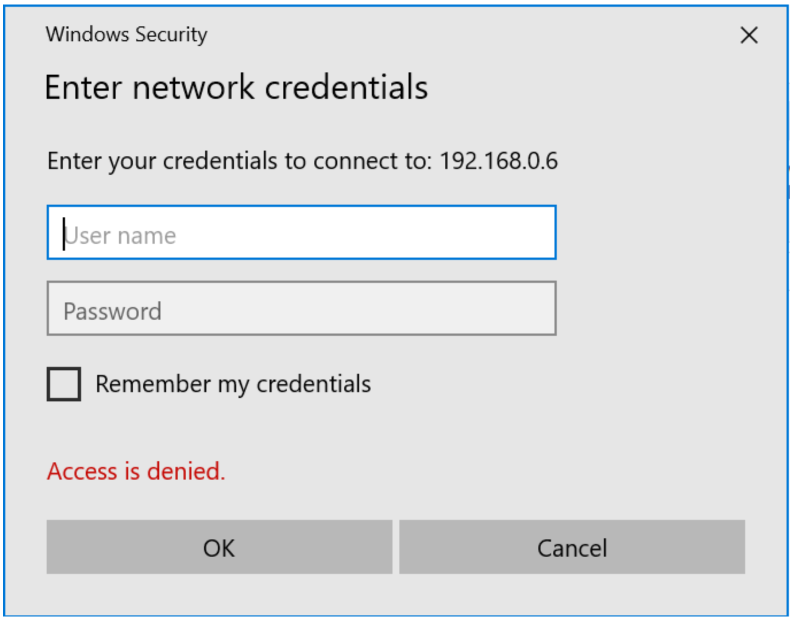
Create a name for this shortcut that will help you easily identify this network location:

\\192.168.0.6\josh.

Type a name for this network location:

josh (192.168.0.6 (client-VirtualBox server (Samba, Ubuntu)))

Step 4: Click "Next". Now, you will be prompted to enter the login credentials of your newly created user as shown below.



Windows Security

Enter network credentials

Enter your credentials to connect to: 192.168.0.6

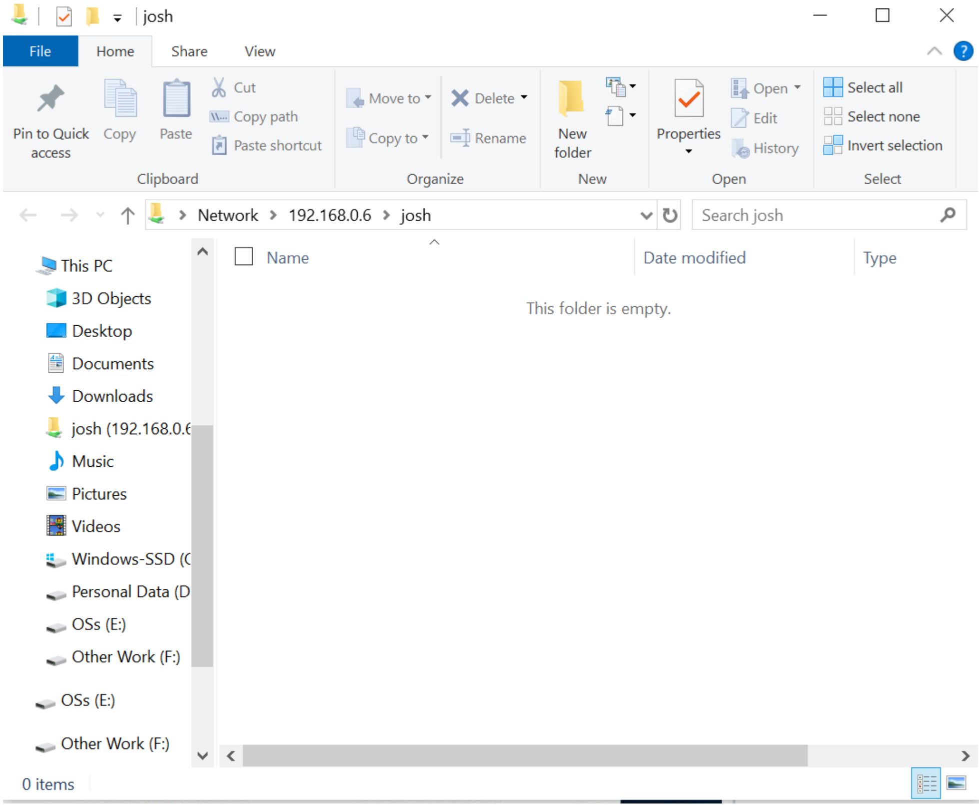
User name

Password

□ Remember my credentials

Access is denied.

Step 5: After entering the credential, the user files will be shown.



josh

Network > 192.168.0.6 > josh

Search josh

This folder is empty.

0 items

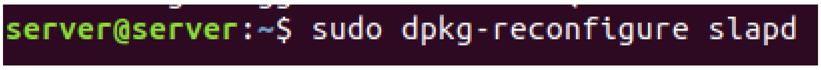
Install and Configure LDAP

Server Configuration

Step 1: Install LDAP and LDAP Utils using

```
sudo apt -y install slapd ldap-utils
```

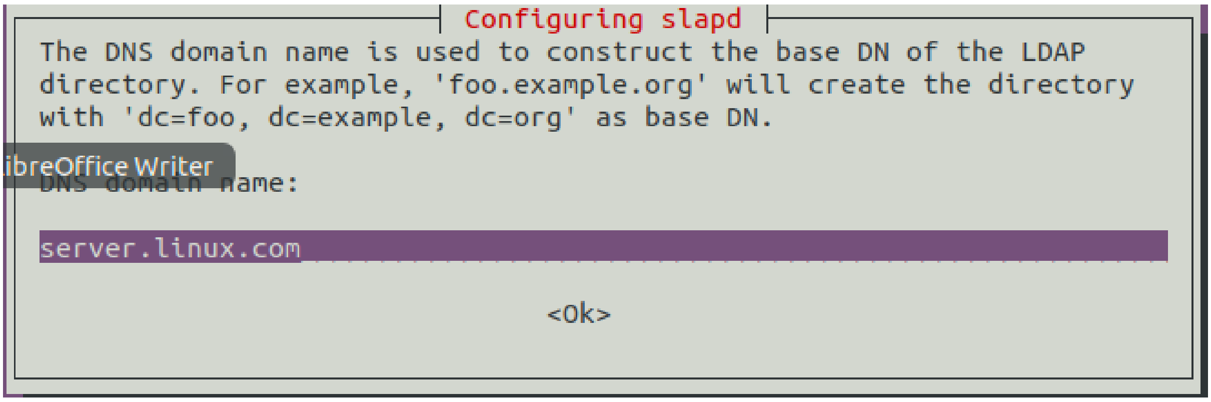
Step 2: Reconfigure the slapd package using



```
server@server:~$ sudo dpkg-reconfigure slapd
```

Step 3: Set up an admin password for LDAP.

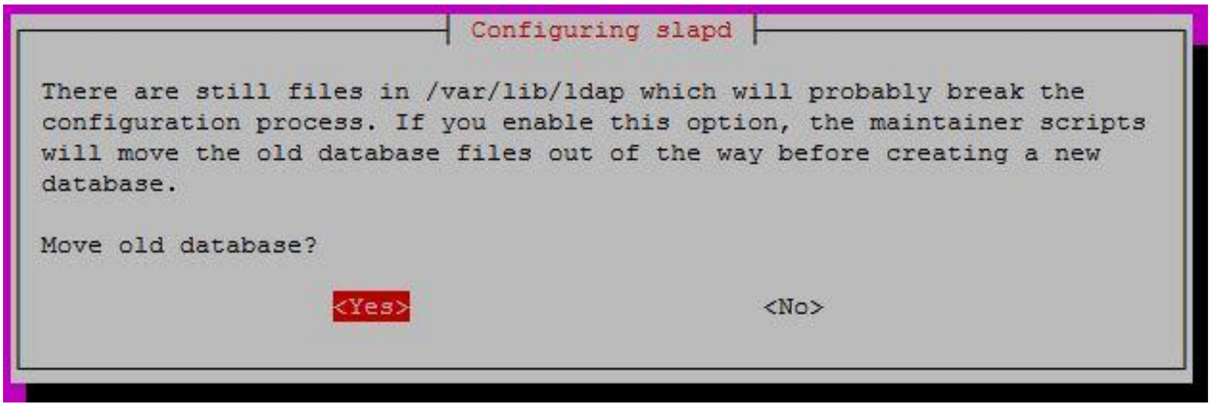
Step 4: Enter the DNS server name.



```
Configuring slapd

The DNS domain name is used to construct the base DN of the LDAP
directory. For example, 'foo.example.org' will create the directory
with 'dc=foo, dc=example, dc=org' as base DN.

DNS domain name:

server.linux.com

<Ok>
```

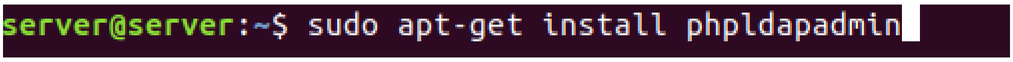
Step 5: Enter the organization’s name (We will just set it to Server Project).

Step 6: Choose MDB as LDAP’s backend format and move the old database when asked.



```
Configuring slapd

There are still files in /var/lib/ldap which will probably break the
configuration process. If you enable this option, the maintainer scripts
will move the old database files out of the way before creating a new
database.

Move old database?

<Yes>                    <No>
```

Step 7: Install phpLDAPadmin for managing the LDAP objects.



```
server@server:~$ sudo apt-get install phpldapadmin
```

Step 8: Configure the LDAP config file using

```
sudo nano /etc/phpldapadmin/config.php
```

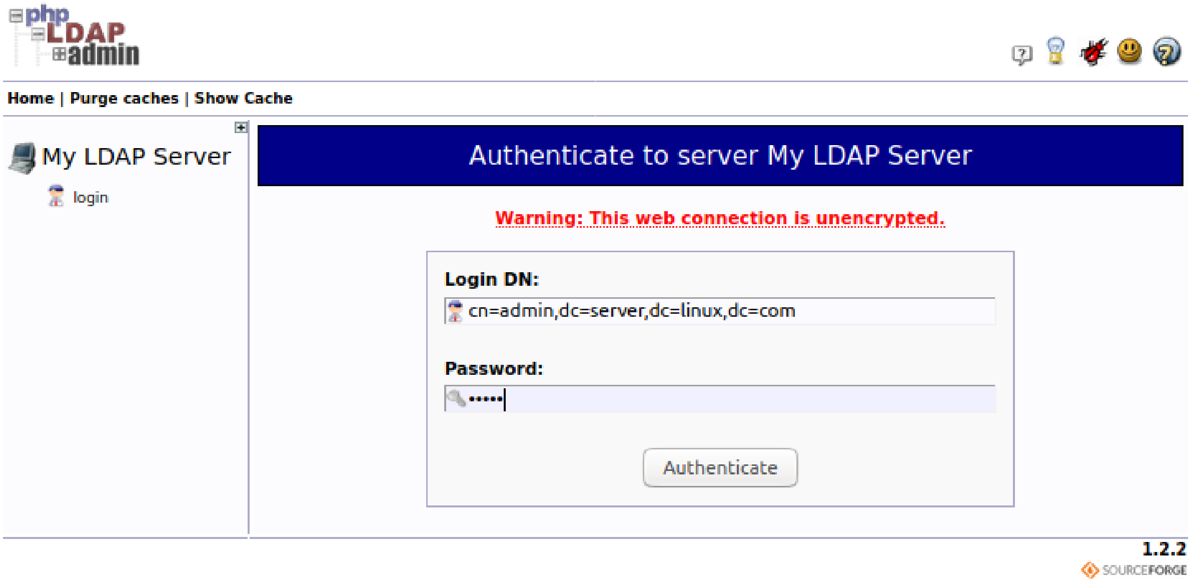
Find the following files and edit them like

```
$servers->setValue(‘server’,’host’,’server.linux.com’);
```

```
$servers-
>setValue('login','bind_id','cn=admin,dc=server,dc=linux,dc=com');
```

Note: linux.com is our DNS server as configured earlier

Step 9: Log into the phpLDAPadmin by visiting Server'sIP/phpldapadmin



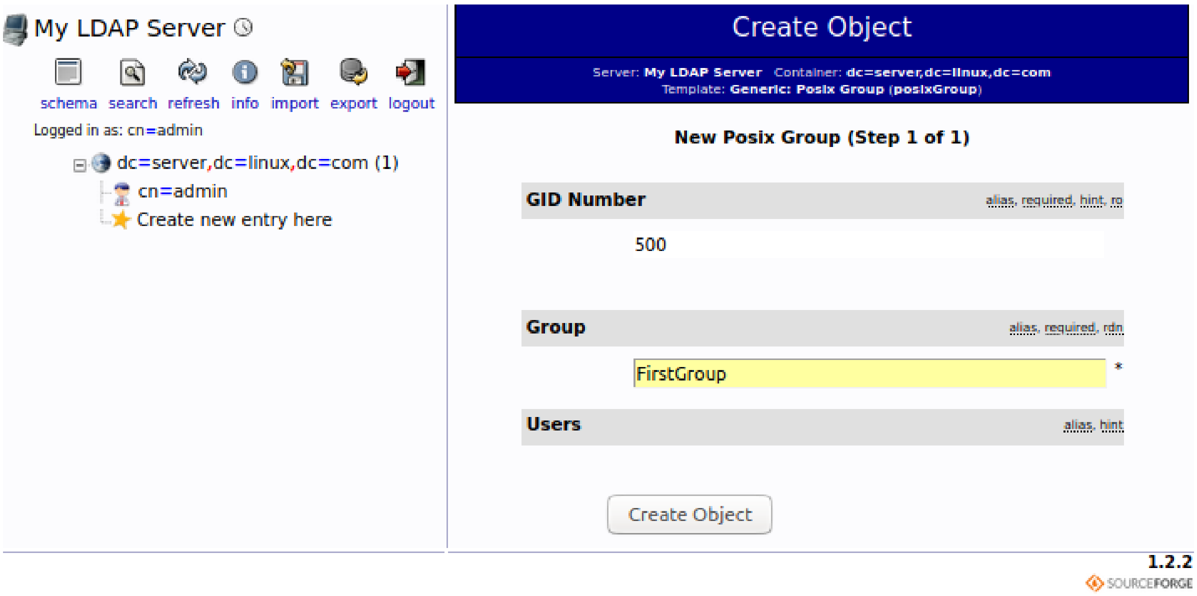
Authenticate to server My LDAP Server

Warning: This web connection is unencrypted.

Login DN: cn=admin,dc=server,dc=linux,dc=com

Password: •••••

Step 10: Create a new entry and create a new Group, then create a new user.



Logged in as: cn=admin

Create Object

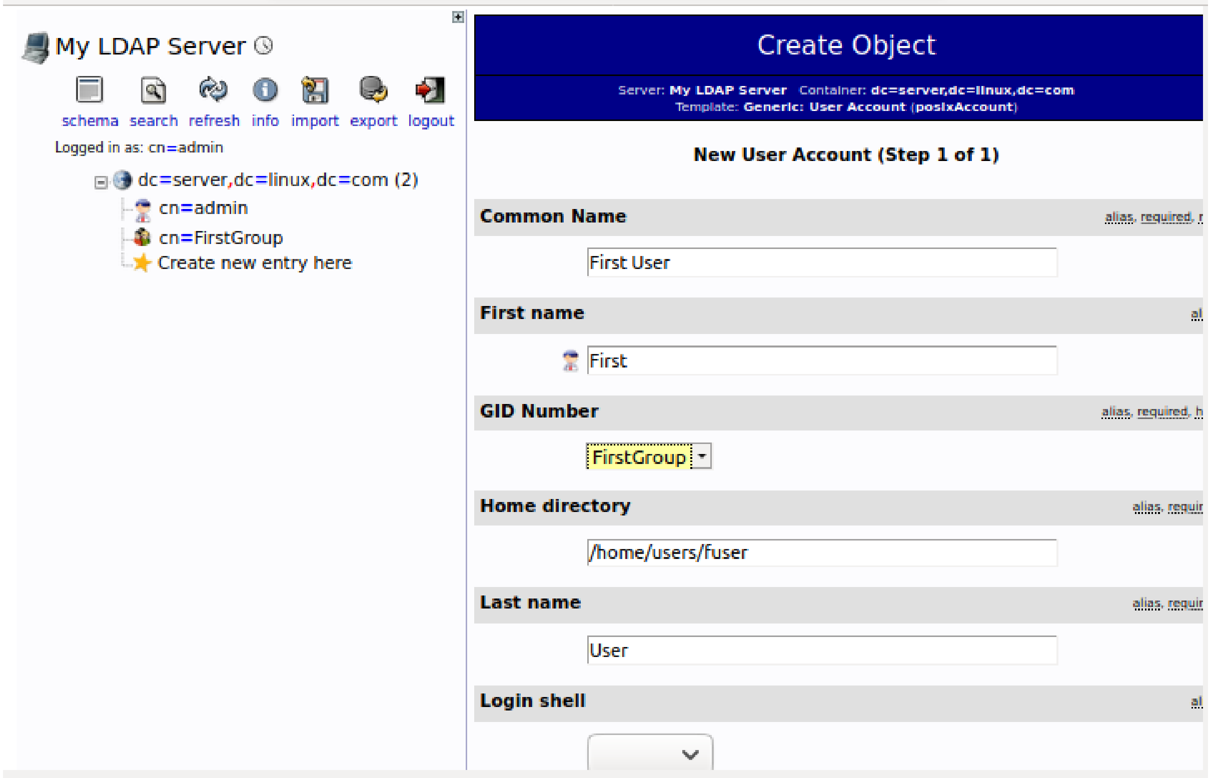
Server: My LDAP Server Container: dc=server,dc=linux,dc=com
Template: Generic: Posix Group (posixGroup)

New Posix Group (Step 1 of 1)

GID Number: 500

Group: FirstGroup *

Users



Logged in as: cn=admin

dc=server,dc=linux,dc=com (2)
cn=admin
cn=FirstGroup
Create new entry here

Create Object

Server: **My LDAP Server** Container: **dc=server,dc=linux,dc=com**
Template: **Generic: User Account (posixAccount)**

New User Account (Step 1 of 1)

Common Name: First User

First name: First

GID Number: FirstGroup

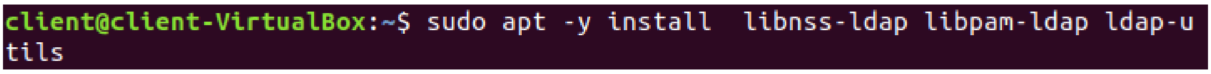
Home directory: /home/users/fuser

Last name: User

Login shell:

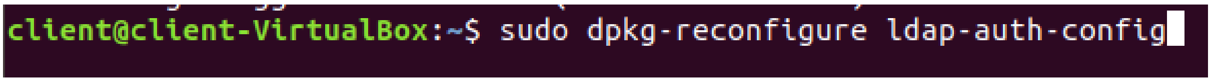
Client Configuration

Step 1: Install client LDAP utilities by using

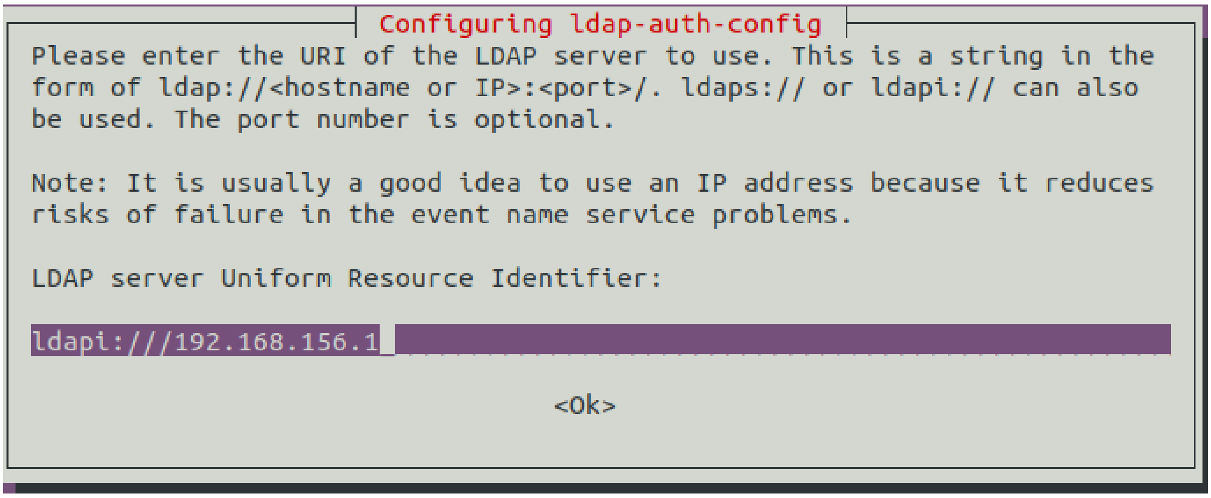


```
client@client-VirtualBox:~$ sudo apt -y install  libnss-ldap libpam-ldap ldap-u
tils
```

Step 2: Configure the LDAP to contact the LDAP Server with the IP address of the server

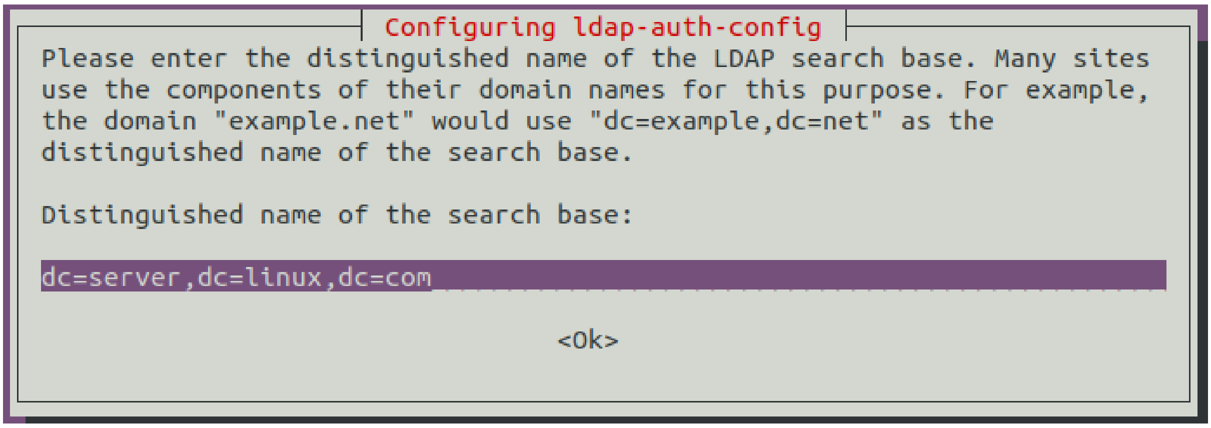


```
client@client-VirtualBox:~$ sudo dpkg-reconfigure ldap-auth-config
```

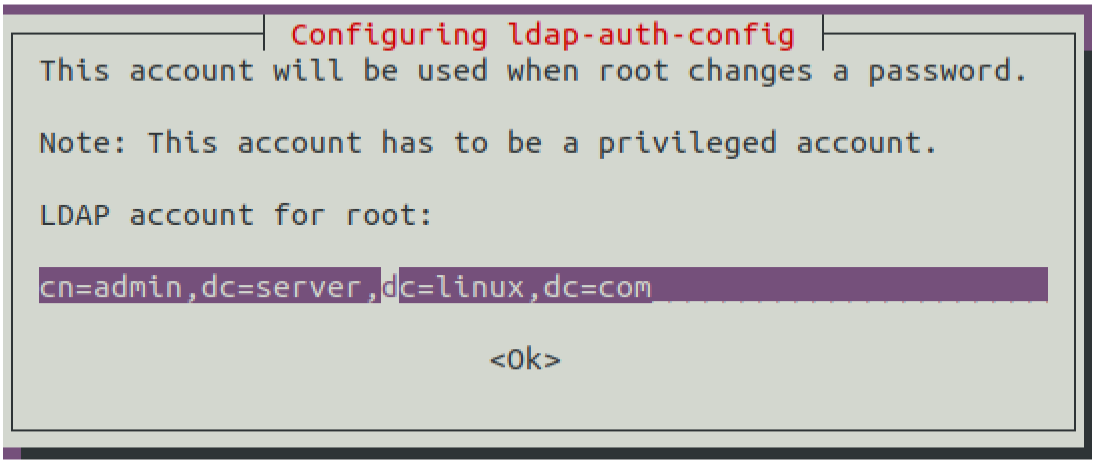


```
Configuring ldap-auth-config

Please enter the URI of the LDAP server to use. This is a string in the
form of ldap://<hostname or IP>:<port>/. ldaps:// or ldapi:// can also
be used. The port number is optional.

Note: It is usually a good idea to use an IP address because it reduces
risks of failure in the event name service problems.

LDAP server Uniform Resource Identifier:

ldapi:///192.168.156.1

                                <Ok>
```

Step 3: Enter the distinguished name of the search base as set up in the server-side configuration. Choose LDAP version 3 in the next step.

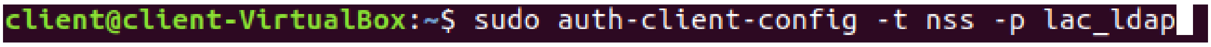


```
Configuring ldap-auth-config
Please enter the distinguished name of the LDAP search base. Many sites
use the components of their domain names for this purpose. For example,
the domain "example.net" would use "dc=example,dc=net" as the
distinguished name of the search base.

Distinguished name of the search base:

dc=server,dc=linux,dc=com

                              <Ok>
```



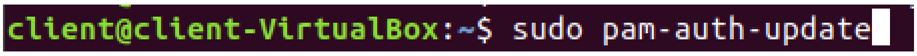
```
Configuring ldap-auth-config
This account will be used when root changes a password.

Note: This account has to be a privileged account.

LDAP account for root:

cn=admin,dc=server,dc=linux,dc=com

                         <Ok>
```

Step 4: Next, configure the LDAP profile for NSS by running.

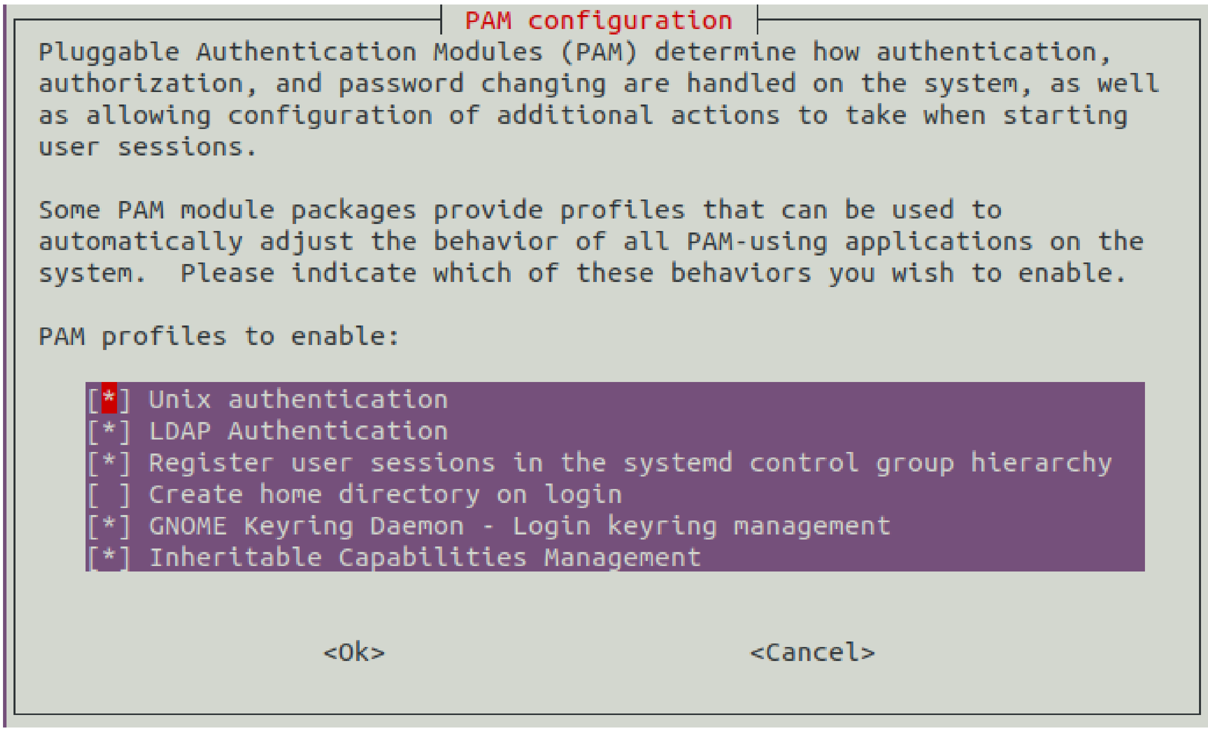


```
client@client-VirtualBox:~$ sudo auth-client-config -t nss -p lac_ldap
```

Step 5: Configure the system to use LDAP for authentication by updating PAM configurations.

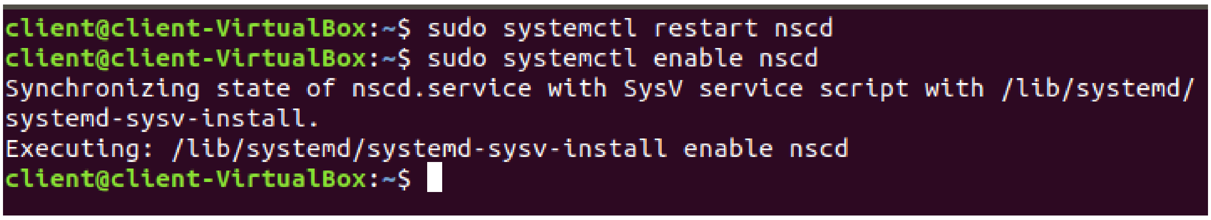


```
client@client-VirtualBox:~$ sudo pam-auth-update
```



```
PAM configuration
Pluggable Authentication Modules (PAM) determine how authentication,
authorization, and password changing are handled on the system, as well
as allowing configuration of additional actions to take when starting
user sessions.

Some PAM module packages provide profiles that can be used to
automatically adjust the behavior of all PAM-using applications on the
system.  Please indicate which of these behaviors you wish to enable.

PAM profiles to enable:

    [*] Unix authentication
    [*] LDAP Authentication
    [*] Register user sessions in the systemd control group hierarchy
    [ ] Create home directory on login
    [*] GNOME Keyring Daemon - Login keyring management
    [*] Inheritable Capabilities Management


                 <Ok>                        <Cancel>
```

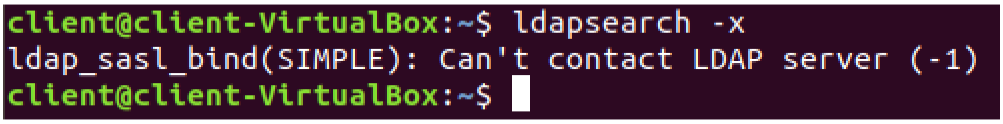
Step 6: Restart the service for these changes to be implemented.



```
client@client-VirtualBox:~$ sudo systemctl restart nscd
client@client-VirtualBox:~$ sudo systemctl enable nscd
Synchronizing state of nscd.service with SysV service script with /lib/systemd/
systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable nscd
client@client-VirtualBox:~$ 
```

Step 7: Check if the client can contact the LDAP server.

Unfortunately, our client cannot contact the LDAP server, we were not able to get the LDAP client to work due to limited meetings and time. But all the required services were installed.



```
client@client-VirtualBox:~$ ldapsearch -x
ldap_sasl_bind(SIMPLE): Can't contact LDAP server (-1)
client@client-VirtualBox:~$ 
```

References:

- ➔ https://renewablepcs.wordpress.com/about-linux/advantages-of-using-linux/
- ➔ https://bloggingtips.guru/advantages-linux-server/
- ➔ https://www.itpro.co.uk/linux/28951/the-benefits-of-linux-servers
- ➔ https://www.vps.net/blog/the-top-five-benefits-of-using-fedora-os-in-the-linux-world/
- ➔ https://help.ubuntu.com/community/Strengths_and_weaknesses
- ➔ https://www.datamation.com/open-source/7-reasons-to-use-debian-and-3-reasons-not-to.html
- ➔ DHCP Server: https://www.youtube.com/watch?v=j3wsYskgdAs
- ➔ DNS Server: https://www.youtube.com/watch?v=P1Kf3rDuhJE
- ➔ Apache Server: https://phoenixnap.com/kb/how-to-install-apache-web-server-on-ubuntu-18-04
- ➔ https://linuxize.com/post/how-to-install-and-configure-samba-on-ubuntu-18-04/
- ➔ LDAP Server: https://computingforgeeks.com/how-to-install-and-configure-openldap-ubuntu-18-04/
- ➔ LDAP Client: https://www.digitalocean.com/community/tutorials/how-to-authenticate-client-computers-using-ldap-on-an-ubuntu-12-04-vps