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

Member Contribution 3

Introduction 4

Project Goals 4

Weakness and Strengths of Linux distributions 5

Advantages of Managing the network using Linux 7

Disadvantages of Managing the network using Linux 7

Challenges and difficulties faced by the group 8

Install and Configure the DHCP Server 9

Deploy the DNS Name Server 17

Install and Configure the Apache Web Server 25

Configure the File Sharing 27

Install and Configure LDAP 30

References 35

**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.

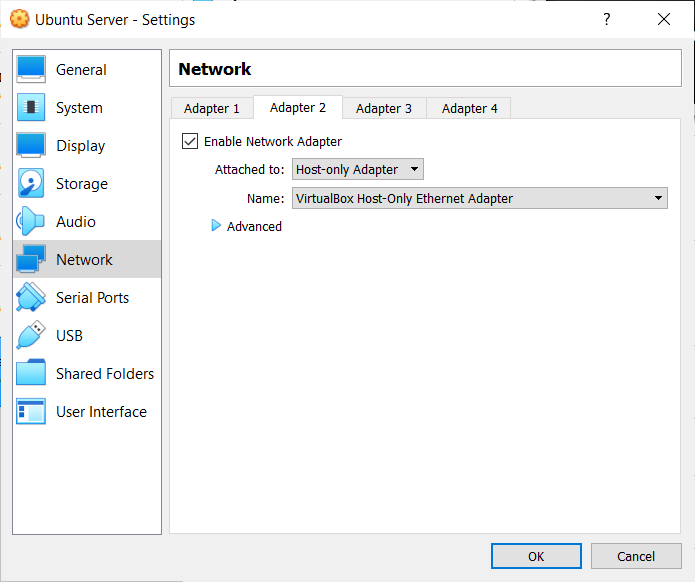
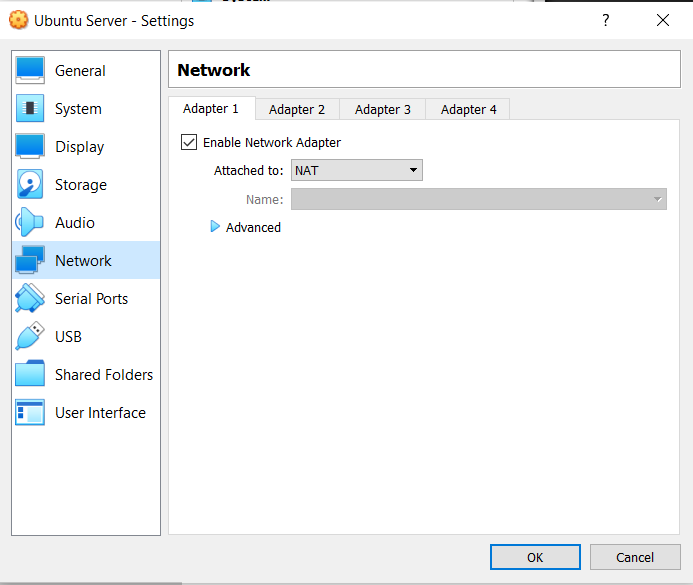
**Challenges and difficulties faced by the group:**

* 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.

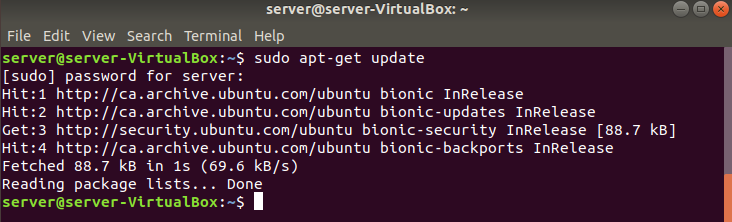
**Install and Configure the DHCP server**

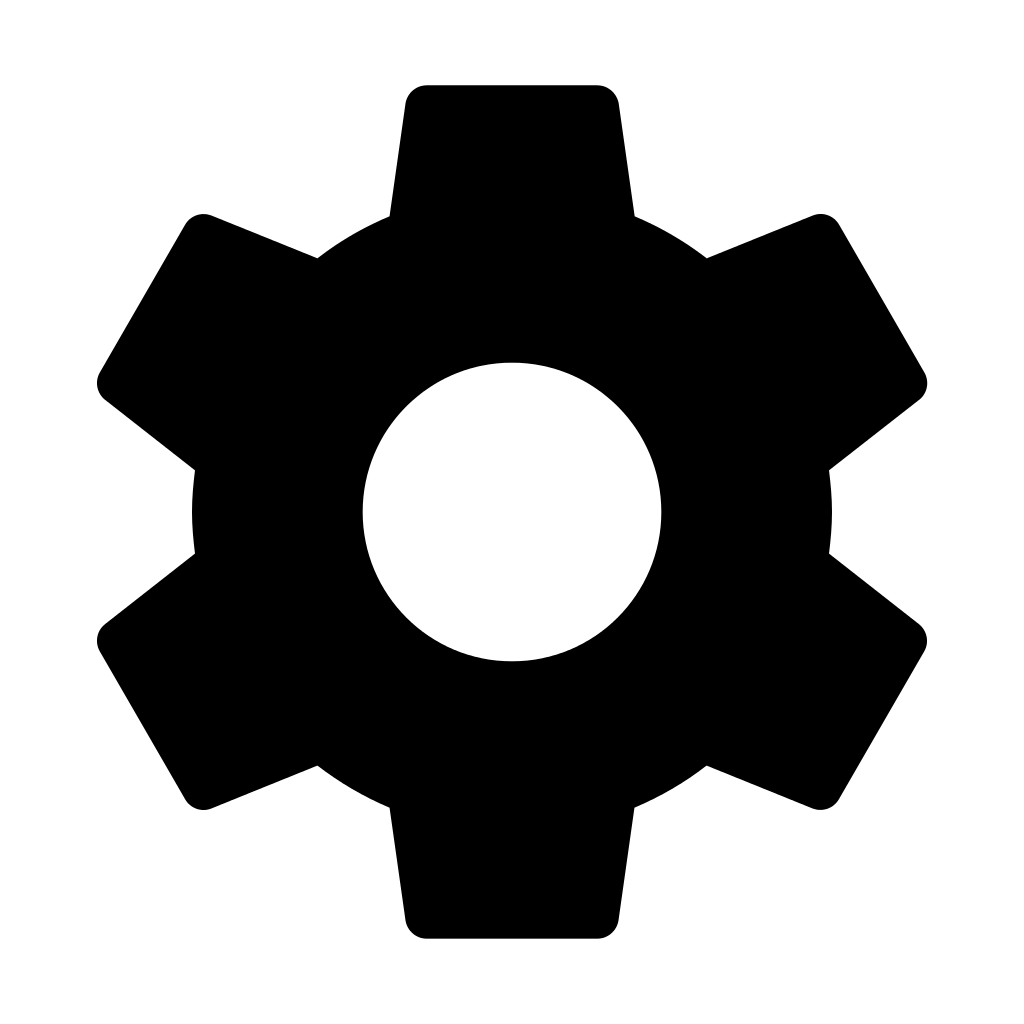
**Step 1:** Change the **Network Setting** of your Virtual Machine.

We will add **two adapters**: One for the NAT and other for the Host-Only.

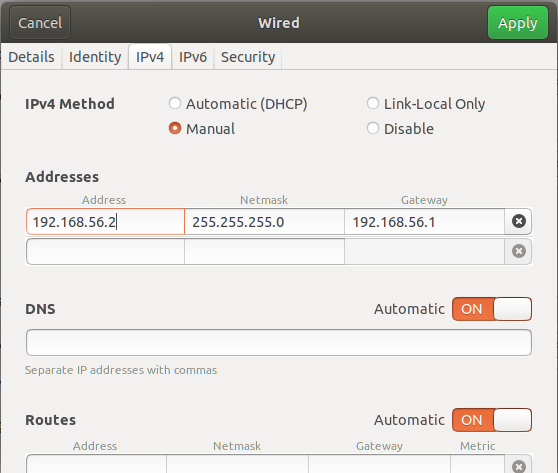


**Step 2:** Start the VM. First, we will update the Ubuntu repositories by running the following command in the terminal.



**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.**



**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

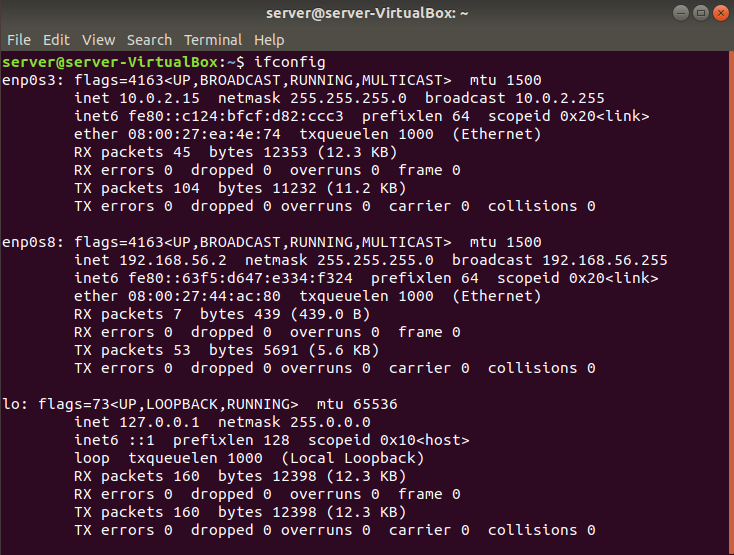
**$ sudo apt-get install net-tools**

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.



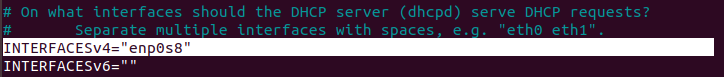
**Step 5:** Further,we will install the DHCP server by running the following command:



**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.



**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.

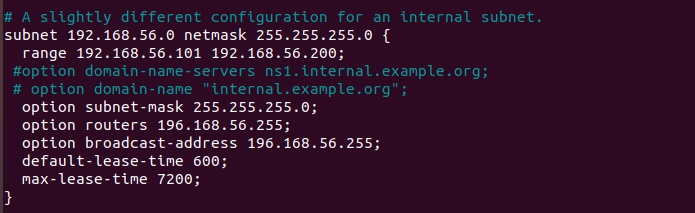


First, we set the subnet and netmask to the network Id of the “enp0s8” interface

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.

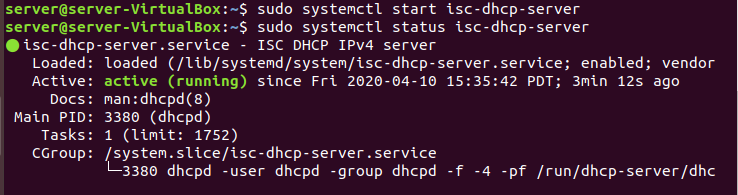


**Step 8:** After editing the dhcpd.conf file we will start the DHCP service on the server by running the following command:

**$ 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**

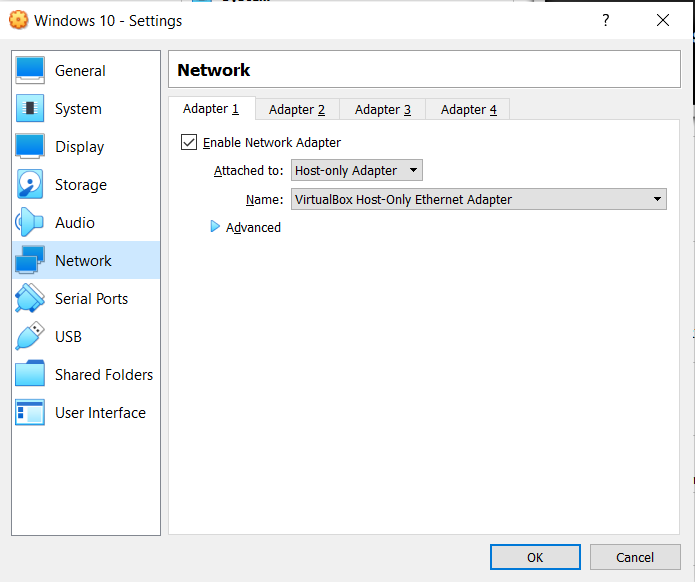


**Step 9:** To enable the service for the boot time, run the following command.

**$ sudo systemctl enable isc-dhcp-server**

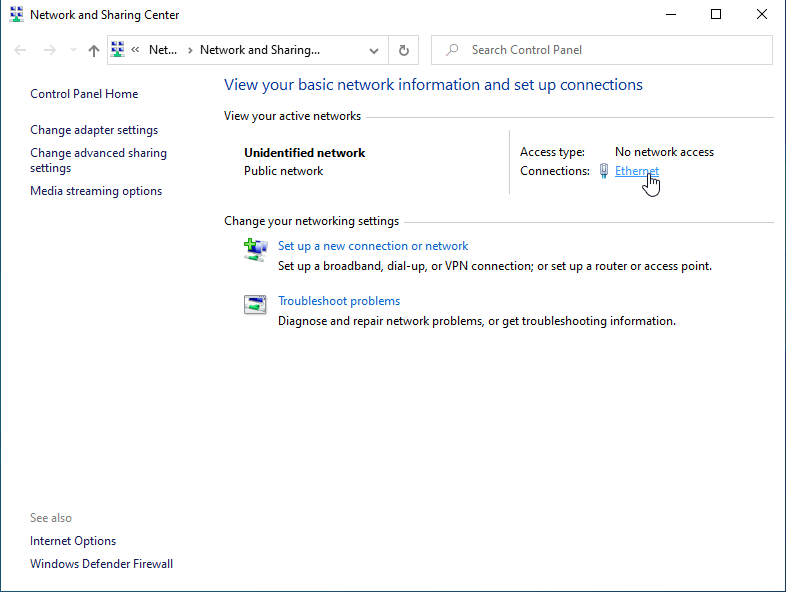
*The DHCP server is configured and running. Now we will configure the client-side.*

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

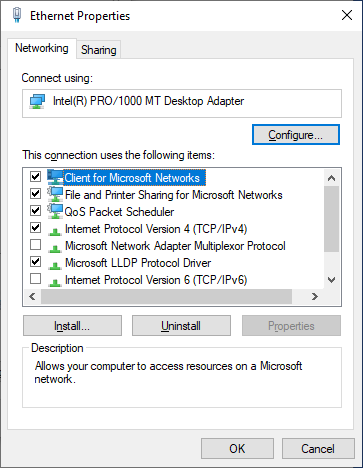


**Step 1:** Open the windows 10 as a client. Go to the Network and Sharing Center.

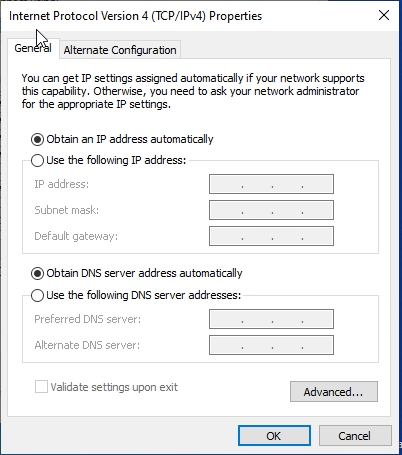
**Step 2:** On the Network and Sharing Center Console, click on the link say Ethernet, as shown below.



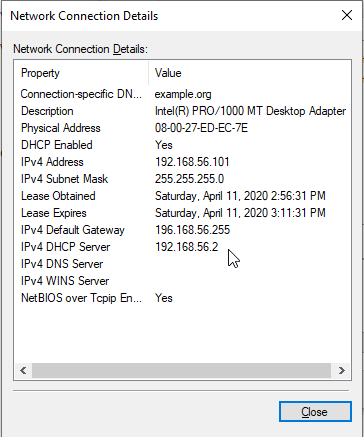
**Step 3:** An ethernet status window shows up. Click on the Properties button. It may require an **administrator privilege**. After putting the administrator credentials, the following window shows up.



**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*.



**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.



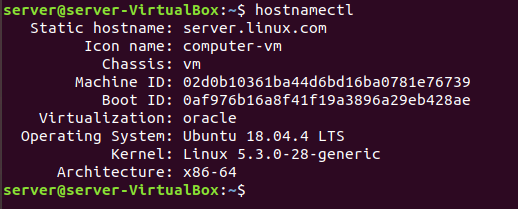
Now, we have configured our client to get the IP address dynamically.

**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:



**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:

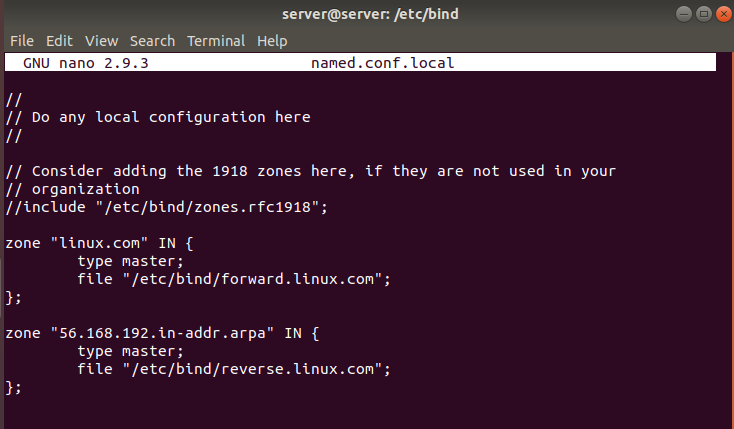
**$ sudo nano /etc/bind/named.conf.local**

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.

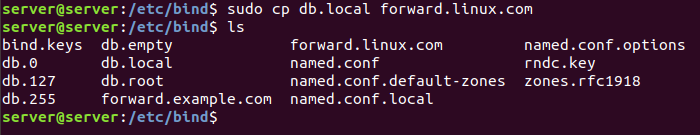
**Reverse Zone:** In this, the IP address will map to the name.

After the edits your file should look as follows:



**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.

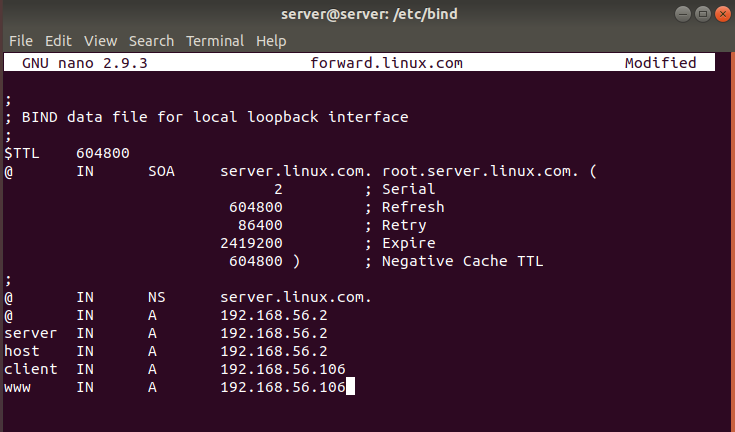
**Step 4:** Further, Copy the **db.local** file to **forward.linux.com** by using the following command (we have changed the directory to /etc/bind/):



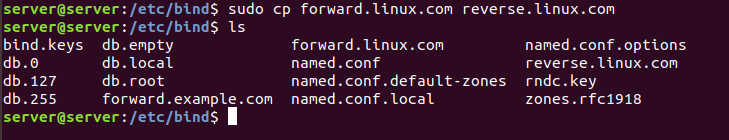
As you can see, now we have a new **forward.linux.com** file in our bind directory.

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

**$ sudo nano forward.linux.com**

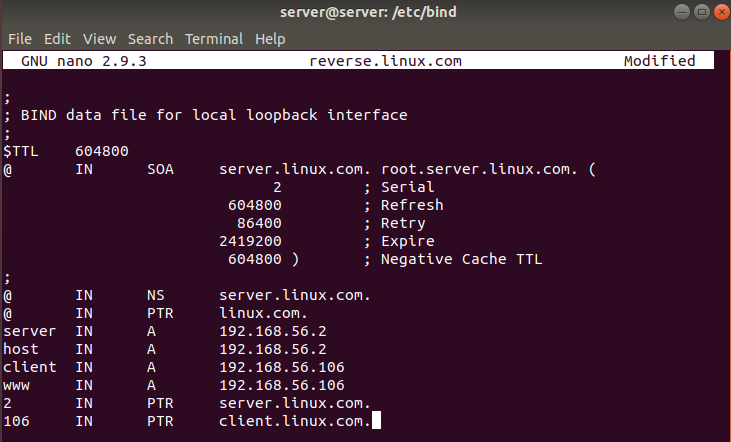


**Step 6:** Next we will copy **forward.linux.com** file to **reverse.linux.com**. And do some **edits** to reverse.linux.com file

Copy File:

Do the following edits:

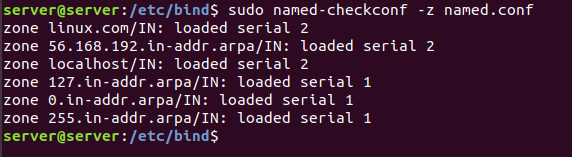
**$ sudo nano reverse.linux.com**

****

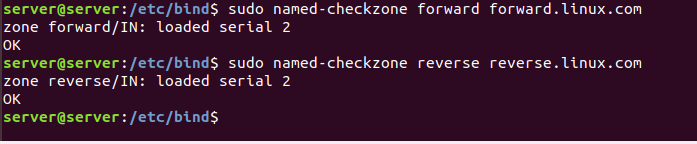
**Note:** we are adding the client just for the testing purpose.

**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:



To check the zones:



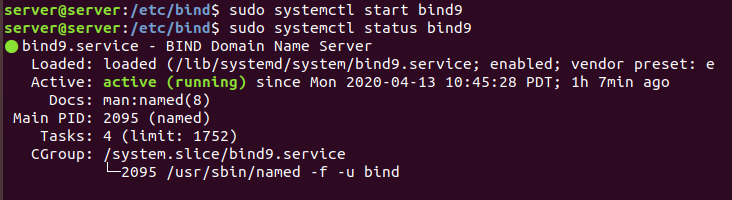
**Step 8:** Before we start the bind9 service, we will have to change the ownership of the files.

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:

****

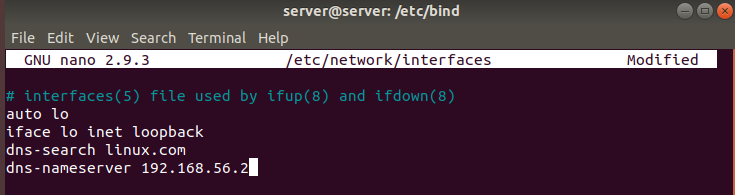
**$ sudo systemctl enable bind9**

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

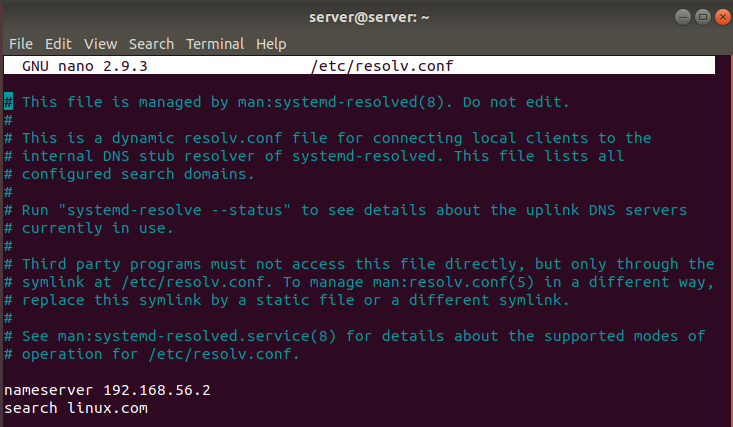
**$ sudo ufw allow bind9**

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

**$ sudo nano /etc/network/interfaces**

****

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

**$ sudo nano /etc/resolv.conf**

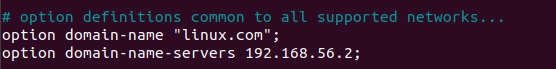
**Step 13:** Now restart the networking and NetworkManager services as shown below:



**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**

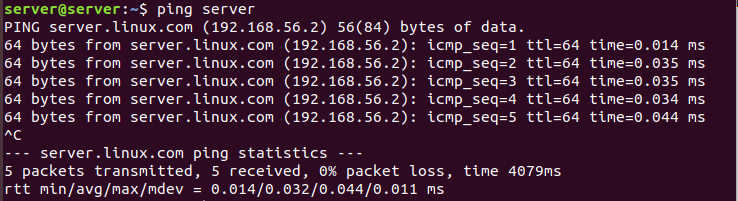


**Now restart the DHCP and DNS services:**

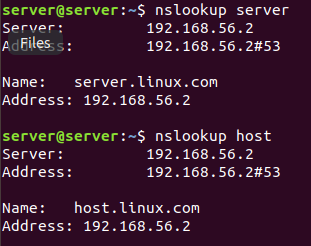
**$ sudo systemctl restart isc-dhcp-server**

**$ sudo systemctl restart bind9**

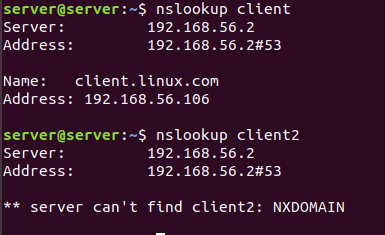
**Step 15:** Next we will check if our DNS server has been configured right. Run the commands as shown:



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



Testing the client with **nslookup:**



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

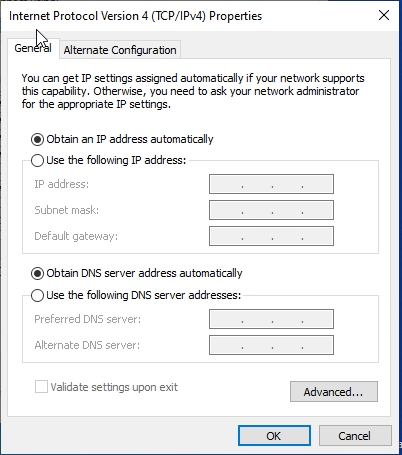
**Testing the Windows 10 client**

**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:

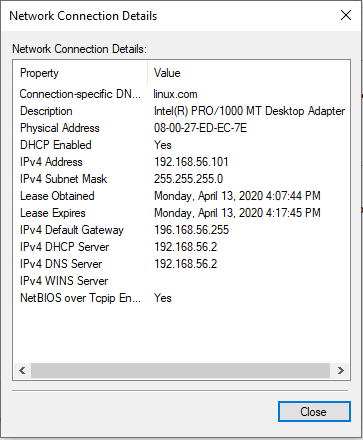
Choose *Obtain an IP address automatically*

*Obtain DNS server address automatically*.

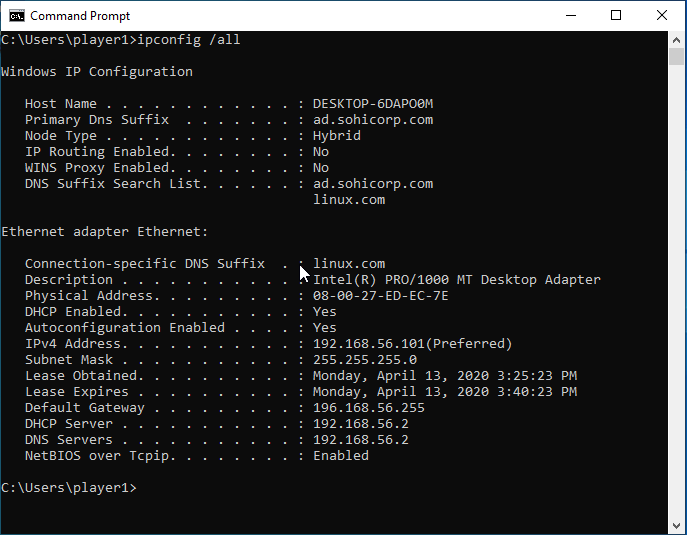


Next return to the Ethernet Status window console by clicking OK.

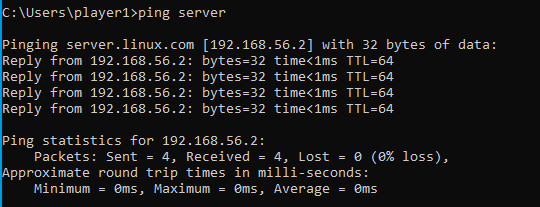
**Step 3:** Next, click the Details button. As you can see, the IP address of the DNS server is the same that we just configured.



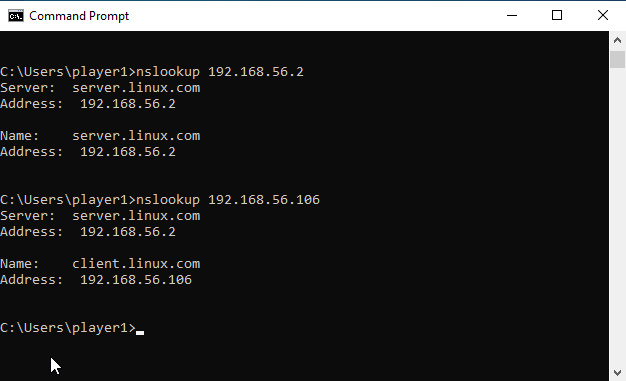
**Step 4:** Now, open the command and run the **ipconfig** command to check if the DNS server is identified.



**Step 5:** Ping the server:



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



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**

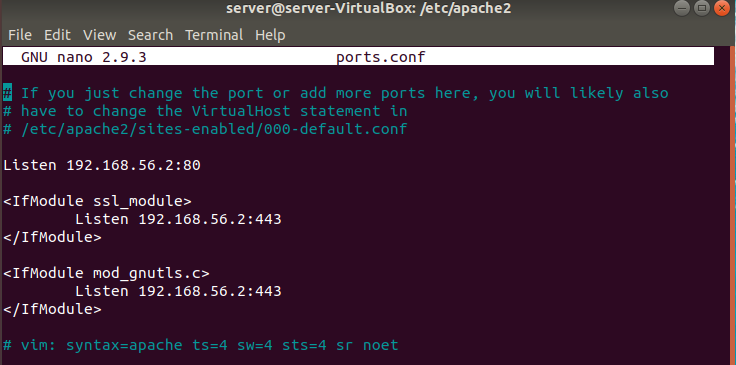
Restart Apache: **sudo systemctl restart apache2.service**

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:



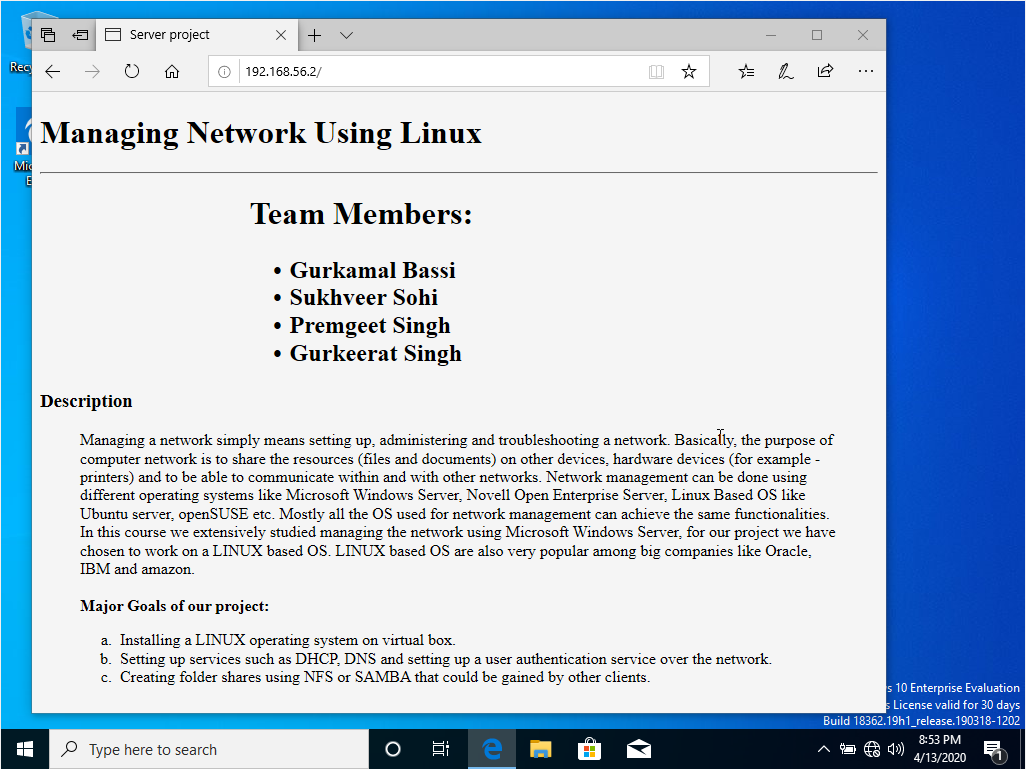
**Step 3:** Now, let’s restart our server. Run the following:

**$ 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.



**Our Apache Server is Up and Running!!**

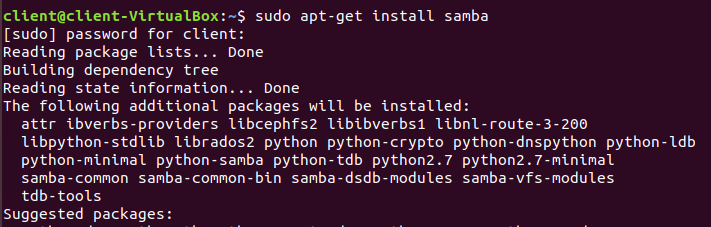
**Configure the File Sharing on Ubuntu Server**

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.

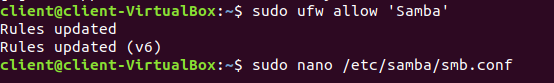
Here are the steps on how to configure File Sharing on the Server.

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



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

Run the command as shown below to configure your firewall.



**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:



**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:

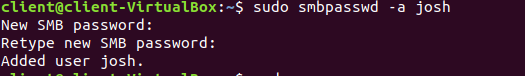


**Step 5:** Now, set the ownership of the directory to user josh and group **sambashare**:

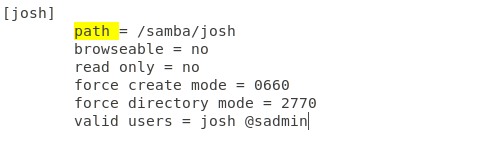


**Step 6:** Further, set the user password to add the user account to the Samba database:

Enter and confirm the user password.



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

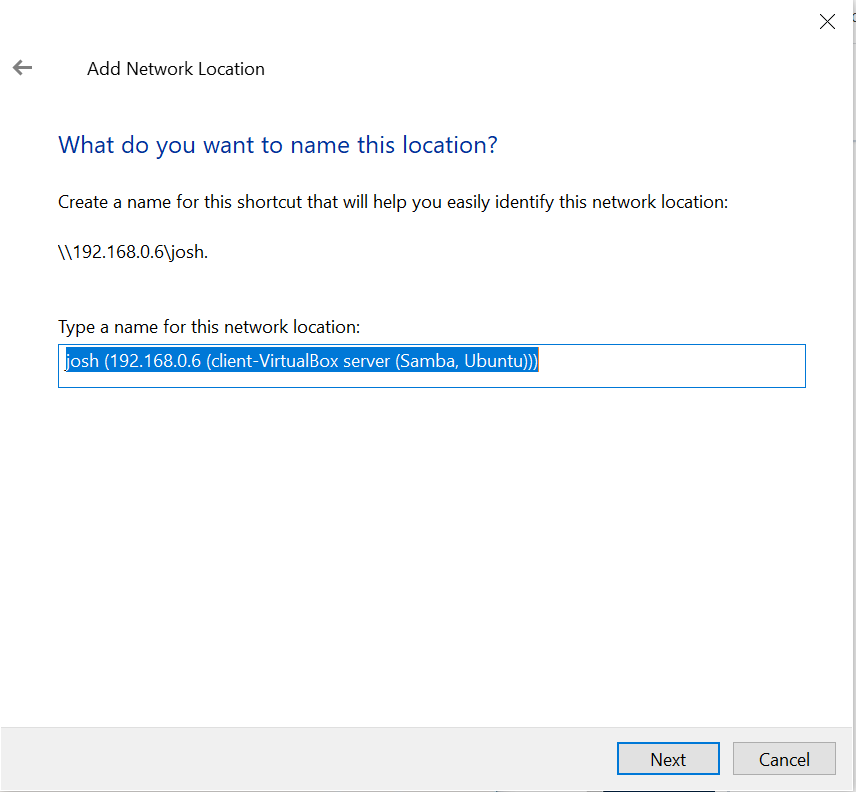


**Connecting to the Samba share from Windows 10**

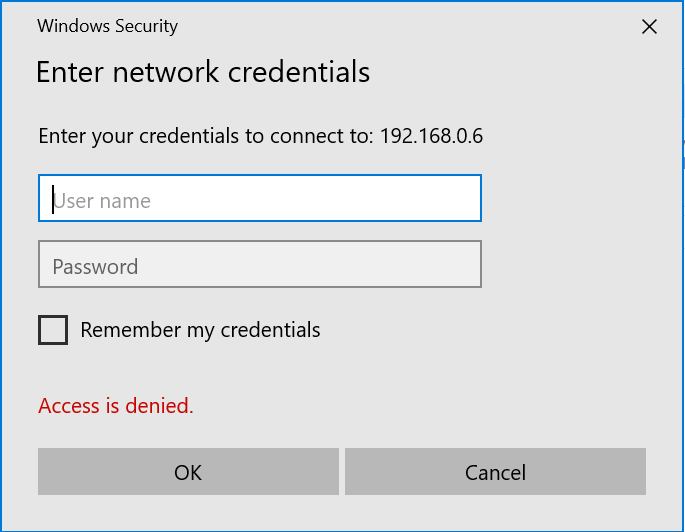
**Step 1:** Open the Windows 10 VM. Then, open the File Explorer. Right-click on “This PC”.

**Step 2:** Select “Choose a custom network location” and then click “Next”.

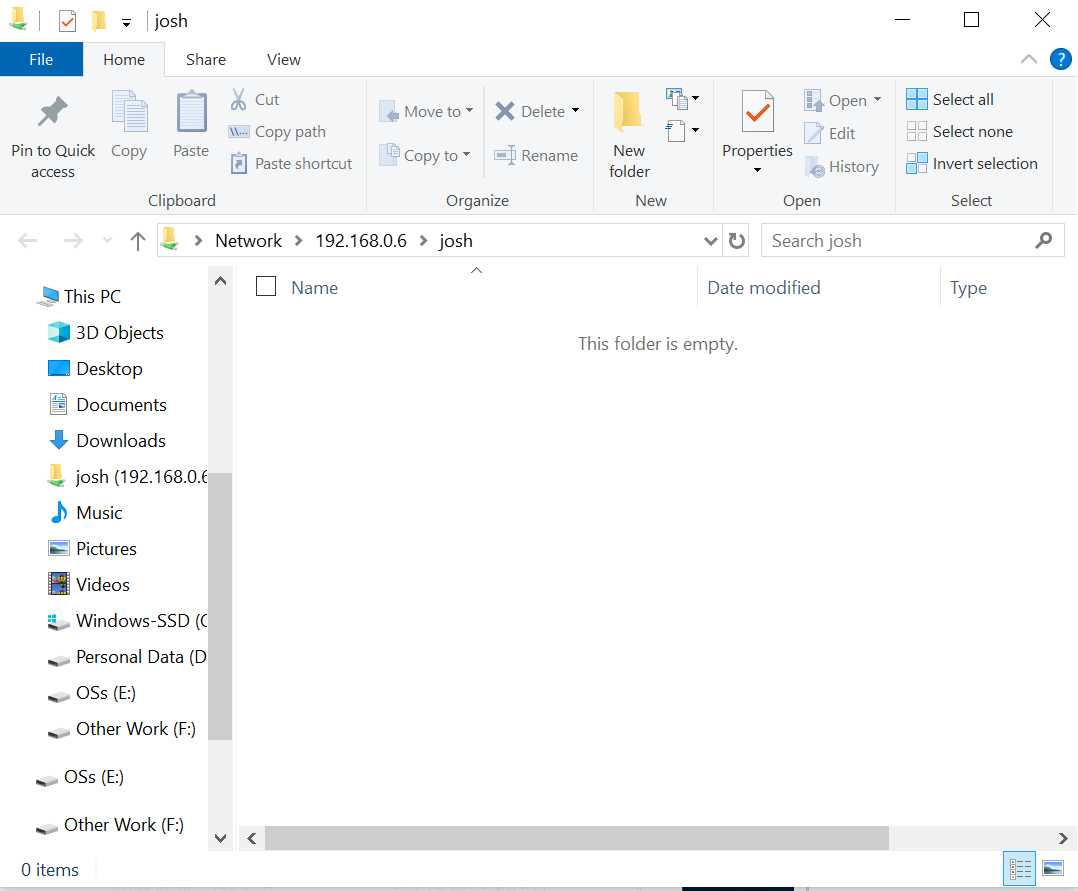
**Step 3:** In “Internet or network address”, enter the address of the Samba share as shown below:



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



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



**Install and Configure LDAP**

**Server Configuration**

**Step 1:** Install LDAP and LDAP Utils using

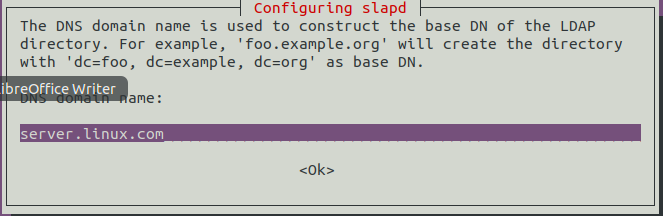
**sudo apt -y install slapd ldap-utils**

**Step 2:** Reconfigure the slapd package using



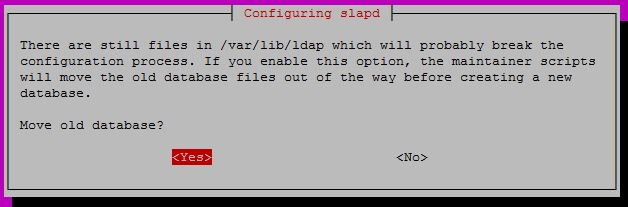
**Step 3:** Set up an admin password for LDAP.

**Step 4:** Enter the DNS server name.



**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.



**Step 7:** Install phpLDAPadmin for managing the LDAP objects.

**Step 8:** Configure the LDAP config file using

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

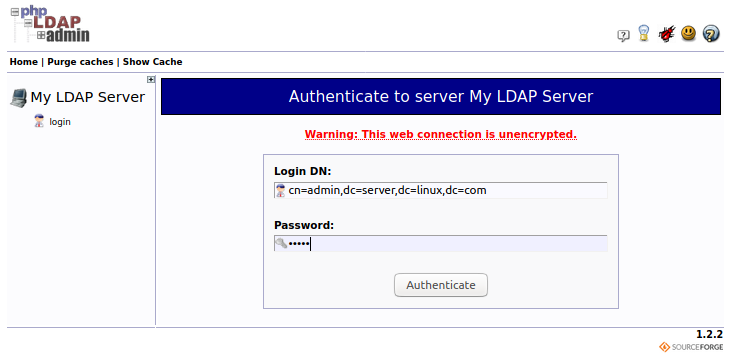
Find the following files and edit them like

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

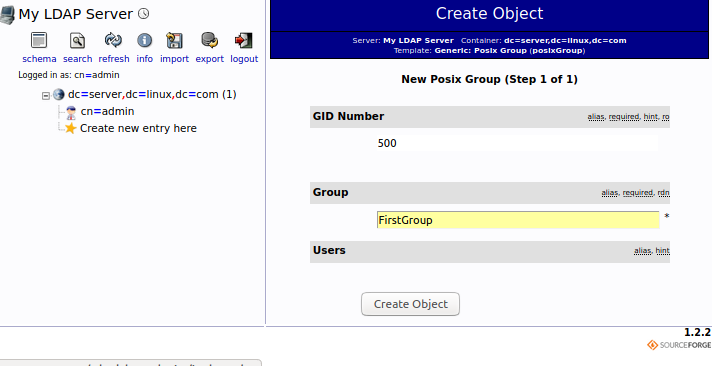
**$servers->setValue(‘login’,’bind\_id’,’cn=admin,dc=server,dc=linux,dc=com’);**

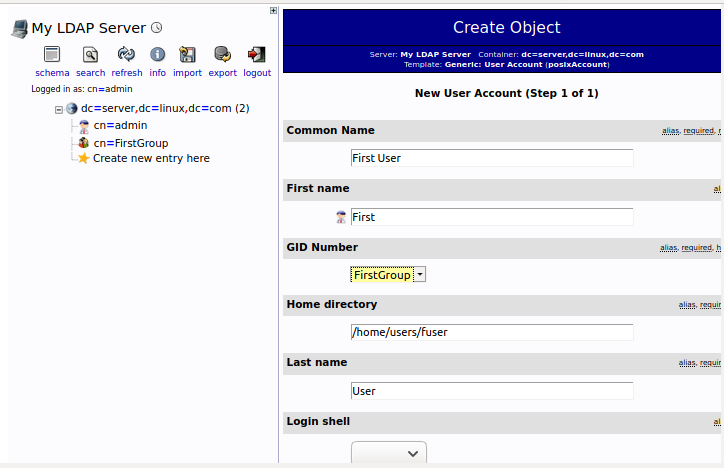
Note: linux.com is our DNS server as configured earlier

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



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





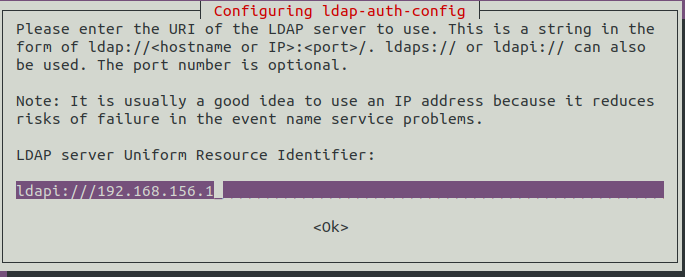
**Client Configuration**

**Step 1:** Install client LDAP utilities by using

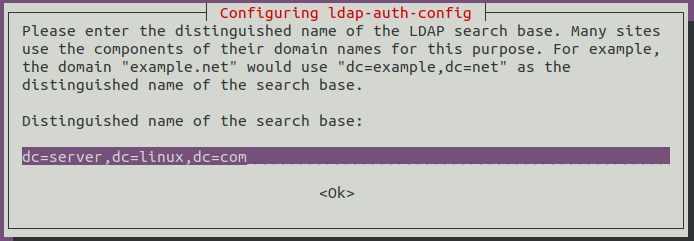


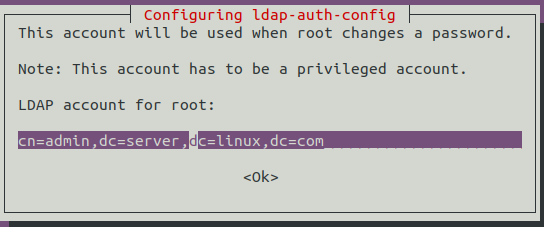
**Step 2:** Configure the LDAP to contact the LDAP Server with the IP address of the server





**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.



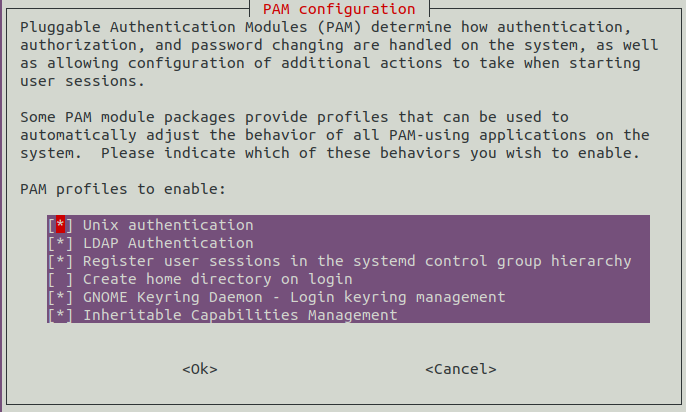


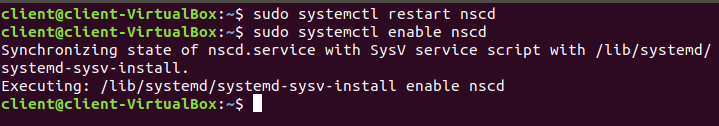
**Step 4:** Next, configure the LDAP profile for NSS by running.



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





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

**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.**



**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>