MGM's

Jawaharlal Nehru Engineering College Aurangabad

Affiliated to Dr.B.A.Technological University , Lonere Maharashtra
ISO 9001:2015,140001:2015 Certified,AICTE Approved

Department of Computer Science & Engineering

LAB MANUAL

Programme(UG/PG) : UG

Year : Final Year

Semester : VIII

Course Code : BTCOL706

Course Title : System Administration

Prepared By
S.A.Kharat
AssistantProfessor
Department of Computer Science & Engineering

FOREWORD

It is my great pleasure to present this laboratory manual for Final year engineering students for the subject of System Administration.

As a student, many of you may be wondering with some of the questions in your mind regarding the subject and exactly what has been tried is to answer through this manual.

As you may be aware that MGM has already been awarded with ISO 9001:2015,140001:2015 certification and it is our endure to technically equip our students taking the advantage of the procedural aspects of ISO Certification.

Faculty members are also advised that covering these aspects in initial stage itself, will greatly relived them in future as much of the load will be taken care by the enthusiasm energies of the students once they are conceptually clear.

Dr. H. H. Shinde
Principal

LABORATORY MANUAL CONTENTS

This manual is intended for the Final year students of Computer Science & Engineering in the subject of System Administration. This manual typically contains practical/Lab Sessions related to System Administration covering various aspects related the subject to enhanced understanding.

System Administration provides students the idea of managing one or more systems, be they software, hardware, servers or workstations. Its goal is ensuring the systems are running efficiently and effectively. It also helps to understands and plan for and responding to service outages and other problems.

Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the books.

Good Luck for your Enjoyable Laboratory Sessions

Mr.S.A.Kharat
Subject Teacher

Dr. Vijaya Musande
HOD

LIST OF EXPERIMENTS

Course Code: BTCOL706

Course Title: System Administration

S.No	Name of the Experiment	Page No
1.	Installations of various Linux flavors (Optionally using Virtual box): Centos (with LVM, without LVM), Ubuntu (with LVM, without LVM), Debian (with LVM, without LVM).	
2.	SSH Server (CentOS and Ubuntu): enable/disable root login.)	
3.	Installation and Configuration of Telnet server (CentOS and Ubuntu)	
4.	Installation and Configuration of FTP Server (CentOS and Ubuntu).	
5.	Using command upload/download files from FTP Server.	
6.	Installation and Configuration of Samba Server (CentOS and Ubuntu).	
7.	Installation and Configuration of HTTP Server (CentOS and Ubuntu)	
8.	Configuration of Proxy Server.	

DOs and DON'Ts in Laboratory:

1. Make entry in the Log Book as soon as you enter the Laboratory.
2. All the students should sit according to their roll numbers starting from their left to right.
3. All the students are supposed to enter the terminal number in the log book.
4. Do not change the terminal on which you are working.
5. All the students are expected to get at least the algorithm of the program/concept to be implemented.
6. Strictly observe the instructions given by the teacher/Lab Instructor.
7. Do not disturb machine Hardware / Software Setup.

Instruction for Laboratory Teachers:

1. Submission related to whatever lab work has been completed should be done during the next lab session along with signing the index.
2. The promptness of submission should be encouraged by way of marking and evaluation patterns that will benefit the sincere students.
3. Continuous assessment in the prescribed format must be followed.

MGM's

Jawaharlal Nehru Engineering College, Aurangabad

Department of Computer Science and Engineering

Vision of CSE Department

To develop computer engineers with necessary analytical ability and human values who can creatively design, implement a wide spectrum of computer systems for welfare of the society.

Mission of the CSE Department:

1. Preparing graduates to work on multidisciplinary platforms associated with their professional position both independently and in a team environment.
2. Preparing graduates for higher education and research in computer science and engineering enabling them to develop systems for society development.

Programme Educational Objectives

Graduates will be able to

I. To analyze, design and provide optimal solution for Computer Science & Engineering and multidisciplinary problems.

II. To pursue higher studies and research by applying knowledge of mathematics and fundamentals of computer science.

III. To exhibit professionalism, communication skills and adapt to current trends by engaging in lifelong learning.

Programme Outcomes (POs):

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems anddesign system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms ofthe engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage independent and life-long learning in the broadest context of technological change.

LABORATORY OUTCOMES

The practical/exercises in this section are psychomotor domain Learning Outcomes (i.e. subcomponents of the COs), to be developed and assessed to lead to the attainment of the competency.

LO-1: Illustrate the role and responsibilities of a Linux system administrator.

LO-2: Make effective use of Linux utilities, and scripting languages.

LO-3: Detect the problem and troubleshoot them.

LO-4: Integrate network services on a Linux system

1. Lab Exercise

Exercise No 1: (2 Hours) – 1 Practical

Aim: - Installations of various Linux flavors (Optionally using Virtual box): Centos (with LVM, without LVM), Ubuntu (with LVM, without LVM) and Debian (with LVM, without LVM).

Objectives:

1. Linux is the foundation of thousands of open source operating systems designed to replace Windows and Mac OS. It is free to download and install on any computer. Because it is open source, there are a variety of different versions, or distributions, available developed by different groups.
2. Students should able to install different flavors of Linux on Computer System and even on virtual environment.

THEORY:

LVM: Logical Volume Management is a storage device management technology that gives users the power to pool and abstract the physical layout of component storage devices for easier and flexible administration. Utilizing the device mapper Linux kernel framework, the current iteration, LVM2, can be used to gather existing storage devices into groups and allocate logical units from the combined space as needed.

Installation steps to be followed:

1) Download the Linux distribution as mentioned in aim of the practical. Linux distributions (known as "distros") are typically available for free to download in ISO format. This format needs to be burned to a CD or USB stick. This will create a Live CD or Live USB. A Live CD or Live USB is a disk that you can boot into, and often contains a preview version of the operating system that can be run directly from the CD or USB stick. Install an image burning program, or use your system's built-in burning tool if you are using Windows 7, 8, or Mac OS X. Pen Drive Linux and UNetBootin are two popular tools for burning ISO files to USB sticks.

2) Boot into the Live CD or Live USB. Most computers are set to boot into the hard drive first, which means you will need to change some settings to boot from your newly-burned CD or USB. Start by rebooting the computer. Once the computer reboots, press the key used to enter the boot menu. If your computer doesn't give you direct access to the boot menu from the manufacturer's splash screen, it's most likely hidden in the BIOS menu. You can access the BIOS menu in the same way that you would get to the boot menu. Once you're in the boot menu, select your live CD or USB. Once you've changed the settings, save and exit the BIOS setup or boot menu. Your computer will continue with the boot process.

3) Try out the Linux distribution before installing. Most Live CDs and USBs can launch a "live environment", giving you the ability to test it out before making the switch. You won't be able to create files, but you can navigate around the interface and decide if it's right for you.

4) Start the installation process. If you're trying out the distro, you can launch the installation from the application on the desktop. If you decided not to try out the distribution, you can start the installation from the boot menu.

5) Create a username and password.

6) Set up the partition. Linux needs to be installed on a separate partition from any other operating systems on your computer if you intend dual booting Linux with another OS. If the installation process does not give you automatic partitions, make sure that the partition you create is formatted as Ext4. If the copy of Linux you are installing is the only operating system on the computer, you will most likely have to manually set your partition size.

7) Boot into Linux. Once the installation is finished, your computer will reboot. You will see a new screen when your computer boots up called ―GNU GRUB‖. This is a boot loader that handles Linux installations. Pick your new Linux distro from the list. This screen may not show up if you only have one operating system on your computer. If this screen isn't being presented to you automatically, then you can get it back by hitting shift right after the manufacturer splash screen. If you install multiple distros on your computer, they will all be listed here.

8) Check your hardware. Most hardware should work out of the box with your Linux distro, though you may need to download some additional drivers to get everything working. Some hardware requires proprietary drivers to work correctly in Linux. This is most common with

graphics cards. There is typically an open source driver that will work, but to get the most out of your graphics cards you will need to download the proprietary drivers from the manufacturer. In Ubuntu, you can download proprietary drivers through the System Settings menu. Select the Additional Drivers option, and then select the graphics driver from the list. Other distros have specific methods for obtaining extra drivers. You can find other drivers from this list as well, such as Wi-Fi drivers.

9) Start using Linux. Once your installation is complete and you've verified that your hardware is working, you're ready to start using Linux. Most distros come with several popular programs installed, and you can download many more from their respective file repositories.

Outcome:

To learn the installation process of Linux operating system with LVM & without LVM.

CONCLUSIONS:

By following These Steps students will able to install Linux Flavors like UBUNTU and CentOs with LVM and Without LVM.

2. Lab Exercise

Exercise No 2: (2 Hours) – 1 Practical

Aim: - Installation and Configuration SSH Server (CentOS and Ubuntu enable/disable root login.)

Objectives:

1. Students should able to install SSH server and configure it on Computer System and even on virtual environment
2. Students should able to Differentiate between secure and unsecure remote system access.

THEORY:

sshd (OpenSSH Daemon or server) is the daemon program for ssh client. It is a free and open source ssh server. ssh replaces insecure rlogin and rsh, and provide secure encrypted communications between two untrusted hosts over an insecure network such as the Internet. Ubuntu Desktop and minimal Ubuntu server do not come with sshd installed.

Installation steps to be followed: (Ubuntu)

1. Open the terminal application for Ubuntu desktop.
2. Type sudo apt-get install openssh-server
3. Enable the ssh service by typing sudo systemctl enable ssh
4. Start the ssh service by typing sudo systemctl start ssh
5. Test it by login into the system using ssh user@server-name

Installation steps to be followed: (CentOs)

Step 1: Install OpenSSH Server Software Package

Enter the following command from your terminal to start the installation process:

sudo yum –y install openssh-server openssh-clients

This command installs both the OpenSSH client applications, as well as the OpenSSH server daemon, **sshd**.

Step 2: Starting SSH Service

To start the SSH daemon on the OpenSSH server:

sudo systemctl start sshd

When active, sshd continuously listens for client connections from any of the client tools. When a connection request occurs, sshd sets up the correct connection.

Step 3: Enable OpenSSH Service

Enable SSH to start automatically after each system reboot by using the systemctl command:

sudo systemctl enable sshd

To disable SSH after reboot enter:

sudo systemctl disable sshd

Step 4: OpenSSH Server Configuration

Properly configuring the sshd configuration file hardens server security. The most common settings to enhance security are changing the port number, disabling root logins, and limiting access to only certain users.

To edit these settings access the /etc/ssh/sshd_config file:

sudo vim /etc/ssh/sshd_config

Once you access the file by using a text editor (in this example we used vim), you can disable root logins and edit the default port number:

- To disable root login:

PermitRootLogin no

CONCLUSIONS:

In this Practical we learned how to enable SSH on a Ubuntu And CentOS server.

Additionally, we configured firewall and SSH rules to limit access.

3. Lab Exercise

Exercise No 3: (2 Hours) – 1 Practical

Aim: - Installation and Configuration SSH Server (CentOS and Ubuntu enable/disable root login.)

Objectives:

1. Students should able to install Telnet server and configure it on Computer System and even on virtual environment
2. Students should able to Differentiate between secure and unsecure remote system access.

THEORY:

Telnet is a terminal emulation program for TCP/IP networks that allows you to access another computer on the Internet or local area network by logging in to the remote system. Telnet is a client-server protocol used to establish a connection to Transmission Control Protocol port number 23. You can also check open ports on a remote system using Telnet.
It's not recommended to use telnet as it is not secure. The passwords are transferred using a plain text and any packet sniffer can easily track you. Nevertheless, it's sometimes required to install telnet anyways.

Installation steps to be followed: (Ubuntu)

Step 1: By default, Telnet server package is available in the Ubuntu 18.04 default repository. You can install it by just running the following command:

sudo apt-get install telnetd -y

Step 2: Once the installation is completed, you can check the status of Telnet service using the following command:

sudo systemctl status inetd

Step 3: Test Telnet Connection from Remote System

Telnet server is now installed and listening on port 23. It's time to connect Telnet server from the remote system.

Now, log in to other Ubuntu system and run the following command:

telnet 192.168.0.100

You will be asked to enter your username and password.

Step 4: Use telnet to Test Open Ports

You can also use Telnet to test open ports on a remote system.

For example, to test port 80 on the remote system (IP 192.168.0.100) run the following command:

telnet 192.168.0.100 80

If the port 80 is open, you should see the following output:

Trying 192.168.0.100...

Connected to 192.168.0.100.

Escape character is '^]'.

If the port 80 is blocked or service is not running. You should see the following output:

Trying 192.168.0.100...

telnet: Unable to connect to remote host: Connection refused

Installation steps to be followed: (CentOs)

Step 1: Install Telnet Server Software Package

Enter the following command from your terminal to start the installation process:

yum install telnet-server telnet

This command installs both the Telnet client applications, as well as the Telnet server daemon, **telnetd**.

Step 2: Starting TelnetService

To start the SSH daemon on the Telnet server:

sudo systemctl start xinetd

When active, xinetd continuously listens for client connections from any of the client tools. When a connection request occurs, xinetd sets up the correct connection.

Step 3: Test Telnet Connection from Remote System

Telnet server is now installed and listening on port 23. It's time to connect Telnet server from the remote system.

Now, log in to other Ubuntu system and run the following command:

telnet 192.168.0.100

You will be asked to enter your username and password.

Step 4: Use telnet to Test Open Ports

You can also use Telnet to test open ports on a remote system.

For example, to test port 80 on the remote system (IP 192.168.0.100) run the following command:

telnet 192.168.0.100 80

If the port 80 is open, you should see the following output:

Trying 192.168.0.100...

Connected to 192.168.0.100.

Escape character is '^]'.

If the port 80 is blocked or service is not running. You should see the following output:

Trying 192.168.0.100...

telnet: Unable to connect to remote host: Connection refused

CONCLUSIONS:

In this Practical we learned how to install and enable Telnet on a Ubuntu And CentOS server. Additionally, we conclude that being insecure avoid the use with root user privileges from remote system.

4. Lab Exercise

Exercise No 4: (2 Hours) – 1 Practical

Aim: - Installation and Configuration of FTP Server (CentOS and Ubuntu).

THEORY:

FTP (**File Transfer Protocol**) is a relatively old and most used standard network protocol used for uploading/downloading files between two computers over a network. However, **FTP** by its original insecure, because it transmits data together with user credentials (username and password) without encryption. FTP is unencrypted by default, so by itself; it is not a good choice for secure transmission of data.

Installation steps to be followed: (Ubuntu)

Step 1: Update System Packages

Start by updating your repositories – enter the following in a terminal window:

sudo apt-get update

The system proceeds to update the repositories.

Step 2: Backup Configuration Files

Before making any changes, make sure to back up your configuration files.

1. Create a backup copy of the default configuration file by entering the following:

sudo cp /etc/vsftpd.conf /etc/vsftpd.conf_default

This command creates a copy of the default configuration file.

2. Create a new vsftpd configuration file /etc/vsftpd.conf using your preferred text editor:

$ sudo gedit /etc/vsftpd.conf

Step 3: Install vsftpd Server on Ubuntu

A common open-source FTP utility used in Ubuntu is vsftpd. It is recommended for its ease of use.

1. To install vsftpd, enter the command:

sudo apt install vsftpd

2. To launch the service and enable it at startup:

sudo systemctl start vsftpd

sudo systemctl enable vsftpd

Step 4: Create FTP User

Create a new FTP user with the following commands:

sudo useradd –m testuser

sudo password testuser

The system should ask you to create a password for the new testuser account. Create a sample file in the new user's home account:

sudo mkdir /home/testuser

Step 5: Configure Firewall to Allow FTP Traffic

If you are using UFW that comes standard with Ubuntu, it will block FTP traffic by default. Enter the following commands to open Ports 20 and 21 for FTP traffic:

sudo ufw allow 20/tcp

sudo ufw allow 21/tcp

Step 6: Connect to Ubuntu FTP Server

Connect to the FTP server with the following command:

sudo ftp ubuntu-ftp

Replace *ubuntu-ftp* with the name of your system (taken from the command line).

Log in using the testuser account and password you just set. You should now be successfully logged in to your FTP server.

Installation steps to be followed: (CentOs)

Step 1: Install FTP Service with VSFTPD

1. Start by updating the package manager:

sudo yum update

Allow the process to complete.

This guide uses the VSFTPD (VSFTPD stands for "Very Secure FTP Daemon software package"). It's a relatively easy software utility to use for creating an FTP server.

2. Install VSFTPD software with the following command:

sudo yum install vsftpd

When prompted, type Y to allow the operation to complete.

3. Start the service and set it to launch when the system boots with the following:

sudo systemctl start vsftpd

sudo systemctl enable vsftpd

4. Next, create a rule for your firewall to allow FTP traffic on Port 21:

sudo firewall-cmd --zone=public --permanent --add-port=21/tcp

sudo firewall-cmd --zone=public --permanent --add-service=ftp

sudo firewall-cmd –-reload

Step 2: Configuring VSFTPD

The behavior of the FTP service on your server is determined by the /etc/vsftpd/vsftpd.conf configuration file.

1. Before starting, create a copy of the default configuration file:

sudo cp /etc/vsftpd/vsftpd.conf /etc/vsftpd/vsftpd.conf.default

This ensures that you have a way to return to the default configuration, in case you change a setting that may cause issues.

2. Next, edit the configuration file with the following command:

sudo nano /etc/vsftpd/vsftpd.conf

3. Set your FTP server to disable anonymous users and allow local users.

Find the following entries in the configuration file, and edit them to match the following:

anonymous_enable=NO

local_enable=YES

This is an important step. Anonymous access is a risky – you should avoid it unless you understand the risks.

4. Next, allow a logged-in user to upload files to your FTP server.

Find the following entry, and edit to match as follows:

write_enable=YES

5. Limit FTP users to their own home directory. This is often called ***jail*** or ***chroot jail***. Find and adjust the entry to match the following:

chroot_local_user=YES

allow_writeable_chroot=YES

6. The **vsftpd** utility provides a way to create an approved user list. To manage users this way, find the **userlist_enable** entry, then edit the file to look as follows:

userlist_enable=YES

userlist_file=/etc/vsftpd/user_list

userlist_deny=NO

You can now edit the **/etc/vsftpd/user_list** file, and add your list of users. (List one per line.) The **userlist_deny** option lets you specify users to be included; setting it to **yes** would change the list to users that are blocked.

7. Once you're finished editing the configuration file, save your changes. Restart the **vsftpd** service to apply changes:

sudo systemctl restart vsftpd

Step 3: Create a New FTP User

1. To create a new FTP user enter the following:

sudo adduser testuser

sudo passwd testuser

The system should prompt you to enter and confirm a password for the new user.

2. Add the new user to the userlist:

echo "testuser" | sudo tee –a /etc/vsftpd/user_list

3. Create a directory for the new user, and adjust permissions:

sudo mkdir –p /home/testuser/ftp/upload

sudo chmod 550 /home/testuser/ftp

sudo chmod 750 /home/testuser/ftp/upload

sudo chown –R testuser: /home/testuser/ftp

This creates a *home/testuser* directory for the new user, with a special directory for uploads. It sets permissions for uploads only to the /uploads directory.

4. Now, you can log in to your FTP server with the user you created:

ftp 192.168.01

Replace this IP address with the one from your system. You can find your IP address in Linux with the ip addr command.

The system should prompt you for a username – enter whatever username you created earlier. Type the password, and the system should log you in.

CONCLUSIONS:

In this Practical we learned how to install and enable FTP Server on a Ubuntu And CentOS. By following above steps you should have installed an FTP server on Ubuntu with **vsftpd**. You should now be able to configure your user lists and accounts, and connect to your new FTP server.

5. Lab Exercise

Exercise No 5: (2 Hours) – 1 Practical

Aim: - Using command upload/download files from FTP Server.

THEORY:

FTP (**File Transfer Protocol**) is a relatively old and most used standard network protocol used for uploading/downloading files between two computers over a network. However, **FTP** by its original insecure, because it transmits data together with user credentials (username and password) without encryption. FTP is unencrypted by default, so by itself; it is not a good choice for secure transmission of data.

Establishing an FTP Connection

1. To open an ftp connection to a remote system, invoke the ftp command followed by the remote server IP address or domain name. For example, to connect to an FTP server at "192.168.42.77" you would type:

   ftp 192.168.42.77
2. If the connection is established, a confirmation message will be displayed, and you will be prompted to enter your FTP username.
3. Once you enter the username you will be prompted to type your password:
4. If the password is correct, the remote server will display a confirmation message and the ftp> prompt.

Common FTP Commands

Most of thr FTP commands are similar or identical to the commands you would type in the Linux shell prompt.

Below are some of the most common FTP commands

- help or ? - list all available FTP commands.
- cd - change directory on the remote machine.
- lcd - change directory on the local machine.
- ls - list the names of the files and directories in the current remote directory.

- mkdir - create a new directory within the current remote directory.
- pwd - print the current working directory on the remote machine.
- delete - remove a file in the current remote directory.
- rmdir- remove a directory in the current remote directory.
- get - copy one file from the remote to the local machine.
- mget - copy multiple files from the remote to the local machine.
- put - copy one file from the local to the remote machine.
- mput - copy one file from the local to the remote machine.

Downloading Files with the ftp Command

1. To download a single file from the remote server, use the get command. For example, to download a file named backup.zip you would use the following command:

   ftp > get backup.zip

2. To download multiple files at once, use the mget command. You can provide a list of individual file names or use wildcard characters:

   ftp> mget backup1.zip backup2.zip

Uploading Files with the FTP Command

1. To upload a file from a local directory to a remote FTP server, use the put command:

   ftp> put image.jpg

2. To upload multiple files from a local directory to a remote FTP server, invoke the mput command:

   ftp> mput image1.jpg image2.jpg

CONCLUSIONS:

In this Practical Session you learned how to use the ftp command to download and upload files to your remote FTP server

6. Lab Exercise

Exercise No 6: (2 Hours) – 1 Practical

Aim: - Installation and Configuration of Samba Server (CentOS and Ubuntu).

THEORY:

Samba is a free and open-source re-implementation of the SMB/CIFS network file sharing protocol that allows end users to access files, printers, and other shared resources.

A Samba file server enables file sharing across different operating systems over a network. It lets you access your desktop files from a laptop and share files with Windows and macOS users.

Samba has both client and server components. Samba uses the SMB protocol, which is necessary when accessing assets on a file server from a Microsoft computer. Samba can also work as a domain controller that is compatible with Microsoft Active Directory.

Installation steps to be followed: (Ubuntu)

Step 1: To install Samba, we run:

sudo apt update

sudo apt install samba

Step 2: Setting up Samba

Now that Samba is installed, we need to create a directory for it to share:

mkdir /home/<username>/sambashare/

The command above creates a new folder sambashare in our home directory which we will share later.

The configuration file for Samba is located at /etc/samba/smb.conf. To add the new directory as a share, we edit the file by running:

sudo nano /etc/samba/smb.conf

At the bottom of the file, add the following lines:

[sambashare]

comment = Samba on Ubuntu

path = /home/username/sambashare

read only = no

browsable = yes

Then press Ctrl-O to save and Ctrl-X to exit from the *nano* text editor.

Now that we have our new share configured, save it and restart Samba for it to take effect:

sudo service smbd restart

Step 3: Update the firewall rules to allow Samba traffic:

sudo ufw allow samba

Step 4: Setting up User Accounts and Connecting to Share

Since Samba doesn't use the system account password, we need to set up a Samba password for our user account:

sudo smbpasswd -a username

Step 5: Connecting to a Samba Share from Windows

Windows users also have an option to connect to the Samba share from both command line and GUI. The steps below show how to access the share using the Windows File Explorer.

1. Open up File Explorer and in the left pane right-click on "This PC".
2. Select "Choose a custom network location" and then click "Next".
3. In "Internet or network address", enter the address of the Samba share in the following format **\\samba_hostname_or_server_ip\sharename.**

Installation steps to be followed: (CentOs)

Step 1: Samba is available from the standard CentOS repositories. To install it on your CentOS system run the following command:

sudo yum install samba samba-client

Step 2: Once the installation is completed, start the Samba services and enable them to start automatically on system boot:

sudo systemctl start smb.service

sudo systemctl start nmb.service

Step 3: Configuring Firewall

Now that Samba is installed and running on your CentOS machine, you'll need to configure your firewall and open the necessary ports. To do so, run the following commands:

firewall-cmd --permanent --zone=public --add-service=samba

firewall-cmd --zone=public --add-service=samba

Step 4: Creating Samba Users

To create a new user named cse, use the following command:

sudo useradd -M -d /samba/cse -s /usr/sbin/nologin -G sambashare cse

Step 5: Set a password and enable the user:

sudo smbpasswd -a sadmin

sudo smbpasswd -e sadmin

Step 6: Configuring Samba Shares

Open the Samba configuration file and append the sections:

sudo nano /etc/samba/smb.conf

[cse]

path = /samba/cse

browseable = no

read only = no

force create mode = 0660

force directory mode = 2770

valid users = cse

Step 7: Connecting to a Samba Share from Linux

The syntax to access a Samba share is as follows:

smbclient //samba_hostname_or_server_ip/share_name -U username

Mount the share using the following command:

sudo mount -t cifs -o username=username //samba_hostname_or_server_ip/sharename /mnt/smbmount

Conclusion:

In this Practical, you have learned how to install a Samba server on Ubuntu & CentOS and create different types of shared and users. We have also learn how to connect to the Samba server from Linux and Windows devices.

7. Lab Exercise

Exercise No 7: (2 Hours) – 1 Practical

Aim: - Installation and Configuration of HTTP Server (CentOS and Ubuntu).

THEORY:

The Apache HTTP server is the most widely-used web server in the world. It provides many powerful features, including dynamically loadable modules, robust media support, and extensive integration with other popular software.

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 25th birthday as a project in February 2020.

Installation steps to be followed: (Ubuntu)

Step 1: Installing Apache

Apache is available within Ubuntu's default software repositories, so you can install it using conventional package management tools.

Update your local package index:

sudo apt update

Install the apache2 package:

sudo apt install apache2

Step 2: Configuration of firewall

sudo ufw allow 'Apache

Step 3: Checking Web Server

Check with the systemd init system to make sure the service is running by typing:

sudo systemctl status apache2

Access the default Apache landing page to confirm that the software is running properly through your IP address:

http://your_server_ip

Installation steps to be followed: (CentOs)

Step 1: Installing Apache

As the non-root sudo user configured in the prerequisites, update the local Apache httpd package index to reflect the latest upstream changes:

sudo yum update httpd

Once the packages are updated, install the Apache package:

sudo yum install httpd

After confirming the installation, yum will install Apache and all required dependencies.

Step 2: Configuration of firewall

sudo firewall-cmd --permanent --add-service=http

If you plan to configure Apache to serve content over HTTPS, you will also want to open up port 443 by enabling the https service:

sudo firewall-cmd --permanent --add-service=https

Next, reload the firewall to put these new rules into effect:

sudo firewall-cmd --reload

After the firewall reloads, you are ready to start the service and check the web server.

Step 3: Checking Web Server

Apache does not automatically start on CentOS once the installation completes. You will need to start the Apache process manually:

sudo systemctl start httpd

Verify that the service is running with the following command:

sudo systemctl status httpd

You will see an active status when the service is running:

Step 4:

Managing the Apache Process

Now that you have your web server up and running, let's go over some basic management commands.

To stop your web server, type:

sudo systemctl stop httpd

To start the web server when it is stopped, type:

sudo systemctl start httpd

To stop and then start the service again, type:

sudo systemctl restart httpd

If you are simply making configuration changes, Apache can often reload without dropping connections. To do this, use this command:

sudo systemctl reload httpd

By default, Apache is configured to start automatically when the server boots. If this is not what you want, disable this behavior by typing:

sudo systemctl disable httpd

To re-enable the service to start up at boot, type:

sudo systemctl enable httpd

Apache will now start automatically when the server boots again.

Conclusion:

In this Practical, we have installed and managed the Apache web server. Now that you have your web server installed, you have many options for the type of content you can serve and the technologies you can use to create a richer experience.

8. Lab Exercise

Exercise No 7: (2 Hours) – 1 Practical

Aim: - Configuration of Proxy Server.

THEORY:

A proxy server is basically another computer which serves as a hub through which internet requests are processed. By connecting through one of these servers, your computer sends your requests to the server which then processes your request and returns what you were wanting. Moreover, in this way it serves as an intermediary between your home machine and the rest of the computers on the internet. Proxies are used for a number of reasons such as to filter web content, to go around restrictions such as parental blocks, to screen downloads and uploads and to provide anonymity when surfing the internet.

Why to Use a Proxy?

If you want to surf the web anonymously then proxies can provide you with a means to hide your home IP address from the rest of the world. By connecting to the internet through proxies, the home IP address of your machine will not be shown but rather the IP of the proxy server will be shown. This can provide you with more privacy then if you were simply connecting directly to the internet. To clarify, there are a number of proxies that can provide you with service. For instance, we searched and found several. Some are free and some charge a small fee, the choice is up to you but we have found that the paid services are more reliable, faster, and more secure.

Installation & configuration of Squid Proxy Server on Ubuntu:

Squid is a Linux-based proxy application. The Squid proxy server is used for filtering traffic, security, and DNS lookups.

Also, Squid can speed up a web server by caching resources. The Squid Proxy allows a server to cache frequently visited web pages. When the user requests a web page or file, the request goes directly to the proxy server — an intermediary device between the user's device and the internet. The proxy server pulls up the resources and relays them to the user.

Step 1: Install Squid Package on Ubuntu

To install Squid, run the command:

sudo apt-get update

sudo apt-get install squid

Step 2: Configuring Squid Proxy Server

The Squid configuration file is found at **/etc/squid/squid.conf.**

1. Open this file in your text editor with the command:

sudo nano /etc/squid/squid.conf

2. Navigate to find the **http_port** option. Typically, this is set to listen on **Port 3218**. This port usually carries TCP traffic. If your system is configured for traffic on another port, change it here.

You may also set the proxy mode to transparent if you'd like to prevent Squid from modifying your requests and responses.

Change it as follows:

http_port 1234 transparent

3. Navigate to the **http_access deny all** option. This is currently configured to block all HTTP traffic. This means no web traffic is allowed.

Change this to the following:

http_access allow all

4. Navigate to the **visible_hostname** option. Add any name you'd like to this entry. This is how the server will appear to anyone trying to connect. Save the changes and exit.

5. Restart the Squid service by entering:

sudo systemctl restart squid

Step 3: Configure Proxy Authentication

This forces users to authenticate to use the proxy.

Start by installing apache2-utils:

sudo apt-get install apache2-utils

Create a passwd file, and change the ownership to the Squid user proxy:

sudo touch /etc/squid/passwd

sudo chown proxy: etc/squid/passwd

Step 4: Add a new user and password

1. To add a new user to Squid, use the command:

sudo htpasswd /etc/squid/passwd newuser

The system will prompt you to enter and confirm a password for newuser.

2. Edit the **/etc/squid/squid.conf** file, and add the following command lines:

auth_param basic program /usr/lib64/squid/basic_ncsa_auth /etc/squid/passwd

auth_param basic children 5

auth_param basic realm Squid Basic Authentication

auth_param basic credentialsttl 2 hours

acl auth_users proxy_auth REQUIRED

http_access allow auth_users

- **Block Websites on Squid Proxy**

1. Create and edit a new text file **/etc/squid/blocked.acl** by entering:

sudo nano /etc/squid/blocked.acl

2. In this file, add the websites to be blocked, starting with a dot:

.facebook.com

.twitter.com

Note: The dot specifies to block all subsites of the main site.

3. Open the **/etc/squid/squid.conf** file again:

sudo nano /etc/squid/squid.conf

4. Add the following lines just above your ACL list:

acl blocked_websites dstdomain "/etc/squid/blocked.acl"

http_access deny blocked_websites

Commands When Working with the Squid Service:

To check the status of your Squid software, enter:

sudo systemctl status squid

This will tell you whether the service is running or not.

To start the service enter:

sudo systemctl start squid

Then set the Squid service to launch when the system starts by entering:

sudo systemctl enable squid

You can re-run the status command now to verify the service is up and running.

To stop the service, use the command:

sudo systemctl stop squid

To prevent Squid from launching at startup, enter:

sudo systemctl disable squid

Conclusion:

In this practical we have learn how Squid works, and how to install and configure Squid Proxy on Ubuntu.

.