Project 1 Topic:
File Server and FTP Server Administration on CentOS

Team Members:

Pratik Bagade

Rutik Borate

Prajwal Chandra

Balaji Dande

Pankaj Kaushik

Dipankar Maharana

Rachana Mangalaram

Under the guidance of:

Zakir Hussain

Introduction

File Server and FTP Server Administration in CentOS involves setting up and managing servers that provide file sharing and transfer services to users and clients.

System Requirements

- **Operating System**: CentOS 7/9
- **Privileges**: Root or sudo access
- **Packages:** nfs-utils, samba, vsftpd

File Server:

A **file server** in CentOS is a server that is designed to store and manage files for multiple clients on a network. It allows users to share files, access them remotely, and manage file permissions. There are several protocols you can use to configure a file server on CentOS, such as **NFS (Network File System)** for Linux-based clients, **Samba (SMB/CIFS)** for Windows-based clients, and **FTP (File Transfer Protocol)** for more general file transfer needs.

Common Protocols for File Sharing:

- NFS (Network File System): Allows remote hosts to mount directories over a network and interact with those directories as though they are mounted locally.
- Samba (SMB/CIFS): Enables file sharing across different OSes (e.g., between Windows and Linux).

Here, we have used **samba** protocol:

Samba is an open-source implementation of the SMB/CIFS protocol, which allows Linux and Windows systems to share files and printers. Given below provides step-by-step guidance on installing and configuring a Samba server on CentOS, allowing Windows clients to access shared directories.

Installation and Setup

yum install samba samba-client samba-common

- Explanation: This command installs three essential Samba packages:
  - samba: The main server package for file and print services.
  - samba-client: The client package for accessing Samba shares from the command line.
  - samba-common: A collection of common files required by both the server and the client.

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

- Explanation: This command allows Samba traffic through the firewall by adding it as a permanent service in the "public" zone. Samba uses specific ports to communicate with clients, and this command opens those ports.
- --permanent: Ensures the change persists after a system reboot.
- --zone=public: Modifies the firewall rule for the "public" zone.
- --add-service=samba: Opens ports used by Samba.

```
sudo firewall-cmd --reload
```

This reloads the firewall rules to apply the changes made above. Without reloading, the new rules would not take effect immediately.

```
systemctl status firewalld.service
```

This command checks the current status of the firewall service (firewalld), ensuring that the firewall is running correctly and that Samba services are allowed through it.

```
mkdir -p /samba/apps
cd /samba/apps/
ls -ltr
```

- Explanation:
    - mkdir -p /samba/apps: Creates the directory /samba/apps where the shared files will be stored. The -p option creates the parent directories as needed.
    - cd /samba/apps/: Navigates to the newly created directory.
    - ls -ltr: Lists the contents of the directory, showing detailed information (though it will be empty initially).

```
chmod a+rwx /samba/
chmod a+rwx /samba/apps/
chmod a+rwx /samba/apps/*
```

- Explanation:
    - chmod a+rwx /samba/: Grants read (r), write (w), and execute (x) permissions to all users (owner, group, and others) on the /samba directory.
    - chmod a+rwx /samba/apps/: Similar permission change for the /samba/apps/ directory.
    - chmod a+rwx /samba/apps/*: Applies read, write, and execute permissions recursively to all files and directories inside /samba/apps.

chcon -t samba_share_t /samba/apps

- Explanation: This command changes the SELinux context of the /samba/apps directory to allow it to be shared via Samba.
  - chcon: Change SELinux security context.
  - -t samba_share_t: Assigns the context samba_share_t (required for Samba shares) to the directory /samba/apps.

vi /etc/samba/smb.conf

This command opens the main Samba configuration file (smb.conf) using the vi text editor. You can use any editor (like nano or vim) based on your preference.

In the smb.conf file, make the following changes:

Global Configuration Section

[global]

```
workgroup = SAMBA
netbios name = centos
security = user
map to guest = bad user
dns proxy = no
```

- Explanation:
  - workgroup = SAMBA: Sets the Windows workgroup name for the server (you can change this to match your Windows network).
  - netbios name = centos: Defines the name that the Samba server will use when appearing on the Windows network.
  - security = user: Specifies that Samba will require a valid user to access the shares.
  - map to guest = bad user: Allows guest access for users who fail to authenticate.
  - dns proxy = no: Disables DNS proxying, as it’s generally not needed for basic file-sharing setups.

Share Configuration Section

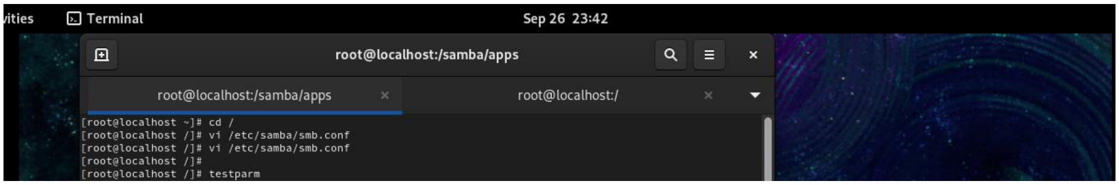
```
[Apps]
```

```
path = /samba/apps
browsable = yes
writable = yes
guest ok = yes
guest only = yes
read only = no
```

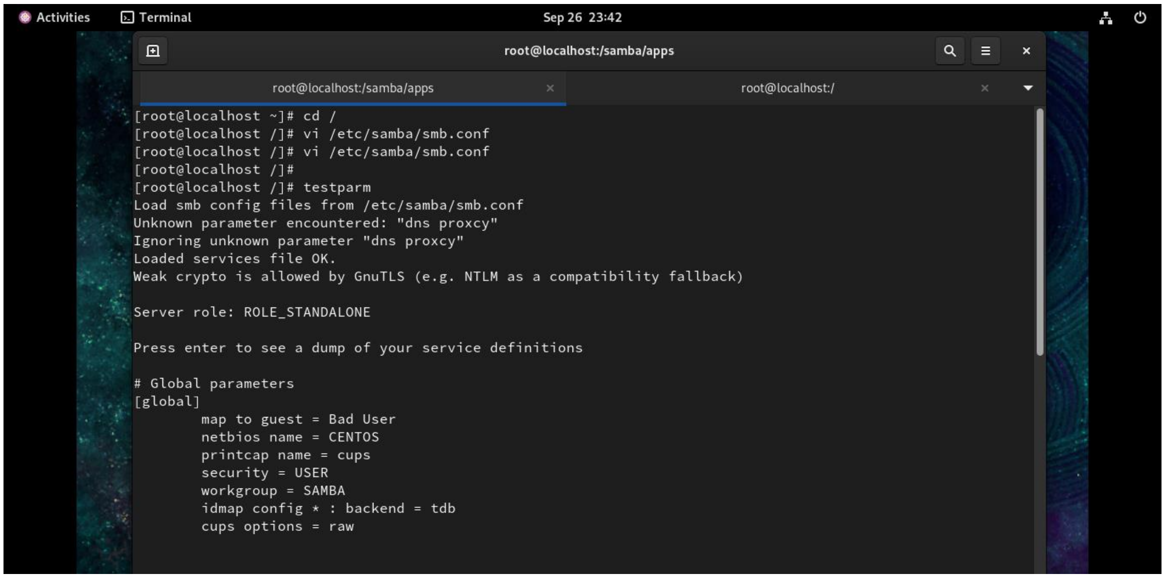
- Explanation:
  - [Apps]: Defines the shared folder name.
  - path = /samba/apps: Specifies the directory to be shared.
  - browsable = yes: Makes the share visible when browsing the network.
  - writable = yes: Allows users to write to the share.
  - guest ok = yes: Allows guest (anonymous) access.
  - guest only = yes: Restricts access to only guest users (no authentication required).
  - read only = no: Specifies that the share is not read-only (i.e., users can write to it).

```
testparm
```

This command tests the Samba configuration file for syntax errors. If any errors are found, it will notify you, allowing you to fix them before starting the service.



```
[root@localhost ~]# cd /
[root@localhost /]# vi /etc/samba/smb.conf
[root@localhost /]# vi /etc/samba/smb.conf
[root@localhost /]#
[root@localhost /]# testparm
```



```
[root@localhost ~]# cd /
[root@localhost /]# vi /etc/samba/smb.conf
[root@localhost /]# vi /etc/samba/smb.conf
[root@localhost /]#
[root@localhost /]# testparm
Load smb config files from /etc/samba/smb.conf
Unknown parameter encountered: "dns proxcy"
Ignoring unknown parameter "dns proxcy"
Loaded services file OK.
Weak crypto is allowed by GnuTLS (e.g. NTLM as a compatibility fallback)

Server role: ROLE_STANDALONE

Press enter to see a dump of your service definitions

# Global parameters
[global]
        map to guest = Bad User
        netbios name = CENTOS
        printcap name = cups
        security = USER
        workgroup = SAMBA
        idmap config * : backend = tdb
        cups options = raw
```

systemctl start smb nmb

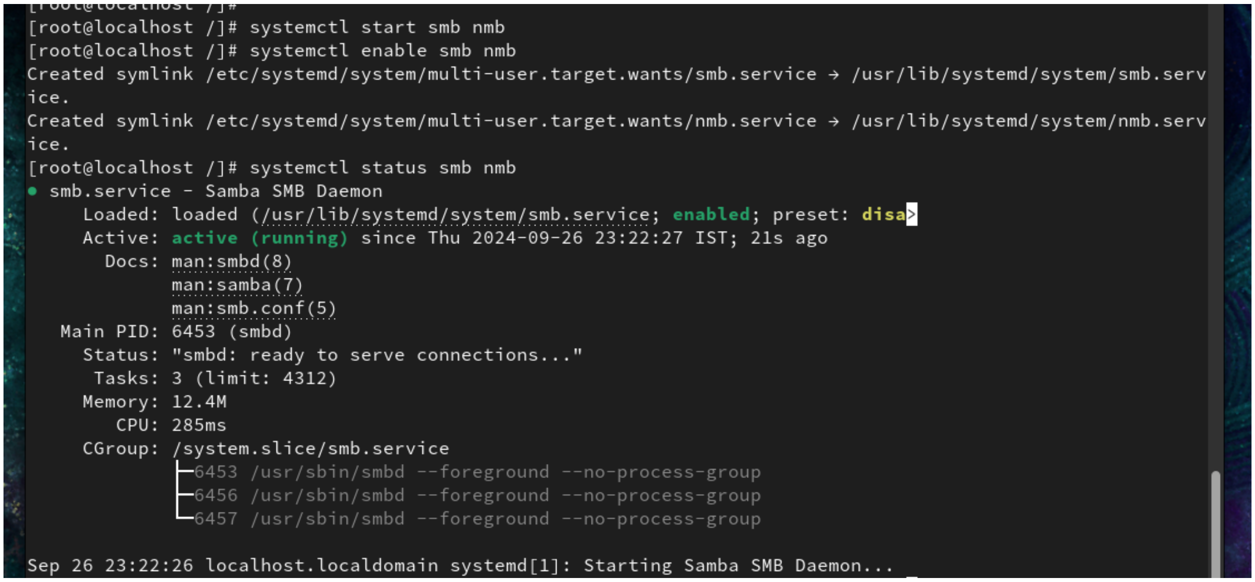
Starts the smb (Samba) and nmb (NetBIOS) services, which are responsible for file sharing and name resolution

systemctl enable smb nmb

Enables the services to start automatically at boot.

systemctl status smb nmb

It used to check the status, i.e active(running)**.**



```
[root@localhost /]#
[root@localhost /]# systemctl start smb nmb
[root@localhost /]# systemctl enable smb nmb
Created symlink /etc/systemd/system/multi-user.target.wants/smb.service → /usr/lib/systemd/system/smb.serv
ice.
Created symlink /etc/systemd/system/multi-user.target.wants/nmb.service → /usr/lib/systemd/system/nmb.serv
ice.
[root@localhost /]# systemctl status smb nmb
● smb.service - Samba SMB Daemon
     Loaded: loaded (/usr/lib/systemd/system/smb.service; enabled; preset: disa>
     Active: active (running) since Thu 2024-09-26 23:22:27 IST; 21s ago
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
   Main PID: 6453 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 3 (limit: 4312)
     Memory: 12.4M
        CPU: 285ms
     CGroup: /system.slice/smb.service
             ├─6453 /usr/sbin/smbd --foreground --no-process-group
             ├─6456 /usr/sbin/smbd --foreground --no-process-group
             └─6457 /usr/sbin/smbd --foreground --no-process-group

Sep 26 23:22:26 localhost.localdomain systemd[1]: Starting Samba SMB Daemon...
```

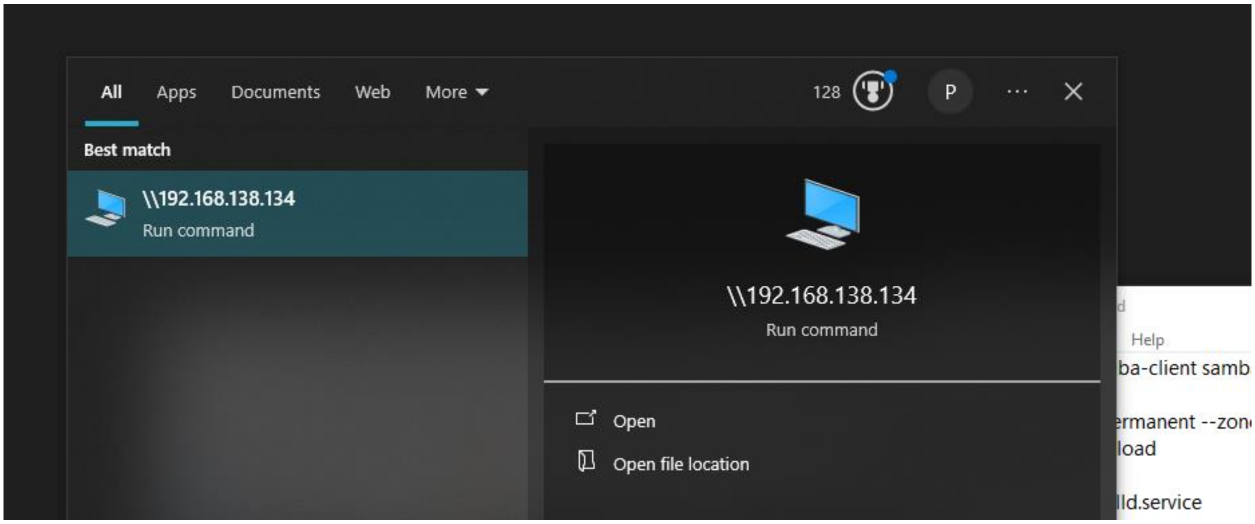
Verifying the Samba Server

Once the Samba server is configured, you can test the connection from a Windows machine by accessing the server using its IP address or hostname.

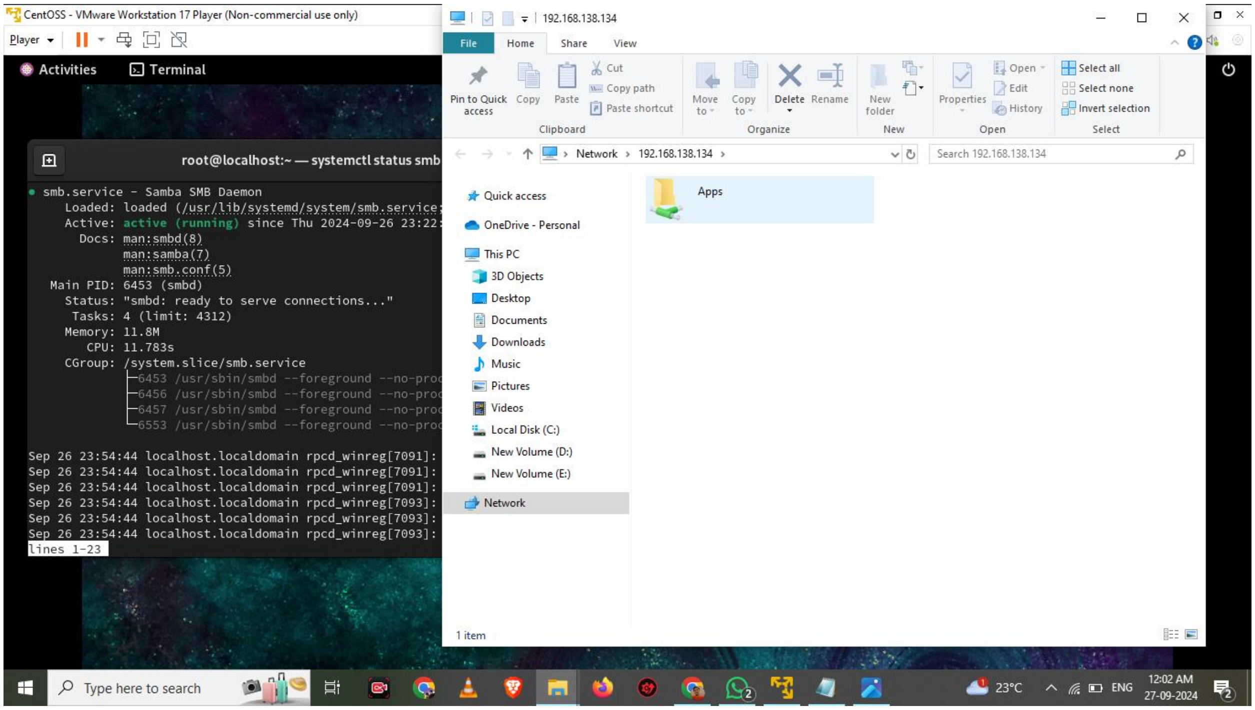
1) Open File Explorer in Windows.
2) Type \\<server-ip-address>\Apps in the address bar and press Enter.
3) The shared folder should appear, and you can interact with the files as configured.

So the IP Address of Our Linux Machine is \\192.168.138.134

A]Accessing Linux Machine in Windows

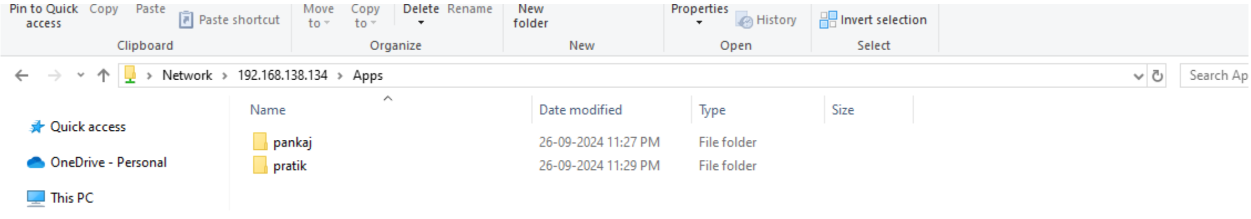


B]Connecting with Linux in Windows



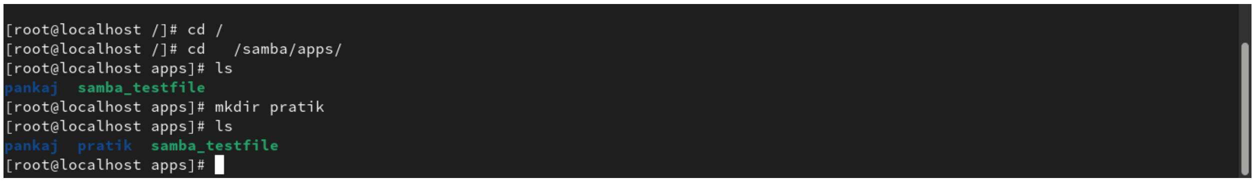
```
● smb.service - Samba SMB Daemon
     Loaded: loaded (/usr/lib/systemd/system/smb.service;
     Active: active (running) since Thu 2024-09-26 23:22:
       Docs: man:smbd(8)
             man:samba(7)
             man:smb.conf(5)
   Main PID: 6453 (smbd)
     Status: "smbd: ready to serve connections..."
      Tasks: 4 (limit: 4312)
     Memory: 11.8M
        CPU: 11.783s
     CGroup: /system.slice/smb.service
             ├─6453 /usr/sbin/smbd --foreground --no-proc
             ├─6456 /usr/sbin/smbd --foreground --no-proc
             ├─6457 /usr/sbin/smbd --foreground --no-proc
             └─6553 /usr/sbin/smbd --foreground --no-proc

Sep 26 23:54:44 localhost.localdomain rpcd_winreg[7091]:
Sep 26 23:54:44 localhost.localdomain rpcd_winreg[7091]:
Sep 26 23:54:44 localhost.localdomain rpcd_winreg[7091]:
Sep 26 23:54:44 localhost.localdomain rpcd_winreg[7093]:
Sep 26 23:54:44 localhost.localdomain rpcd_winreg[7093]:
Sep 26 23:54:44 localhost.localdomain rpcd_winreg[7093]:
lines 1-23
```

C]Creating and Manipulating Files in Linux and Windows



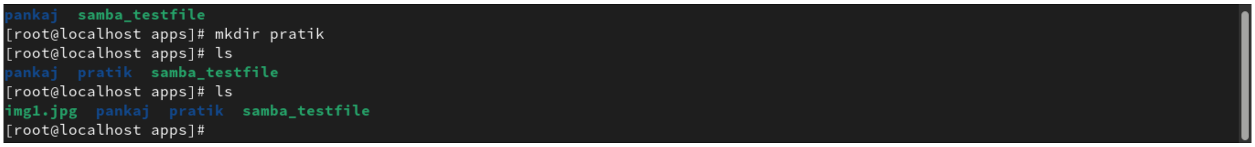
Network › 192.168.138.134 › Apps

Name	Date modified	Type	Size
pankaj	26-09-2024 11:27 PM	File folder	
pratik	26-09-2024 11:29 PM	File folder	

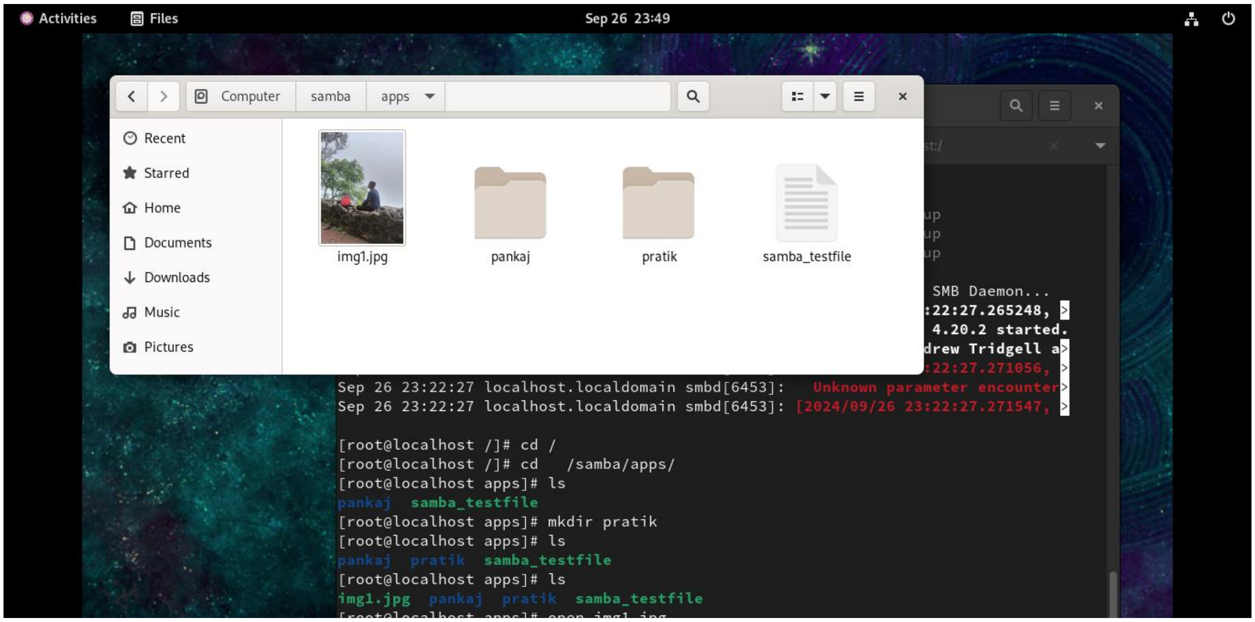


```
[root@localhost /]# cd /
[root@localhost /]# cd   /samba/apps/
[root@localhost apps]# ls
pankaj  samba_testfile
[root@localhost apps]# mkdir pratik
[root@localhost apps]# ls
pankaj  pratik  samba_testfile
[root@localhost apps]# 
```

D]Now Creating files in Linux and accessing in Windows



```
pankaj  samba_testfile
[root@localhost apps]# mkdir pratik
[root@localhost apps]# ls
pankaj  pratik  samba_testfile
[root@localhost apps]# ls
img1.jpg  pankaj  pratik  samba_testfile
[root@localhost apps]#
```



```
up
up
up

 SMB Daemon...
:22:27.265248, >
 4.20.2 started.
drew Tridgell a>
:22:27.271056, >
Sep 26 23:22:27 localhost.localdomain smbd[6453]:   Unknown parameter encounter>
Sep 26 23:22:27 localhost.localdomain smbd[6453]: [2024/09/26 23:22:27.271547, >

[root@localhost /]# cd /
[root@localhost /]# cd   /samba/apps/
[root@localhost apps]# ls
pankaj  samba_testfile
[root@localhost apps]# mkdir pratik
[root@localhost apps]# ls
pankaj  pratik  samba_testfile
[root@localhost apps]# ls
img1.jpg  pankaj  pratik  samba_testfile
[root@localhost apps]# open img1.jpg
```

FTP Server:

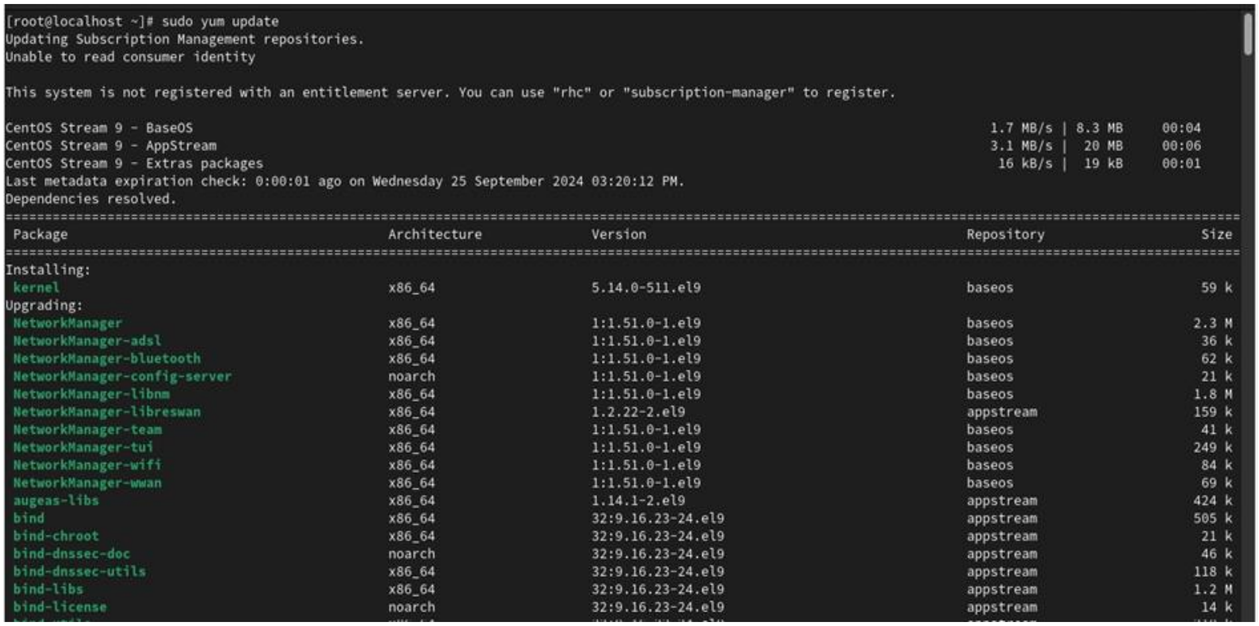
FTP (File Transfer Protocol) is a network protocol that allows users to transfer files between computers. An FTP server is a software solution that uses FTP to enable users to upload and download files.

VSFTPD is a simple way to install an FTP server on CentOS. To install VSFTPD on CentOS 7, you need a user account with sudo privileges, the yum package manager, and a text editor.

Implementation:

sudo yum update

The sudo yum update command updates all installed packages on Red Hat-based systems like CentOS or Amazon Linux. It checks for the latest versions of packages and installs them, ensuring your system is up-to-date with the latest security patches and features.



```
[root@localhost ~]# sudo yum update
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

CentOS Stream 9 - BaseOS                                                                       1.7 MB/s | 8.3 MB     00:04
CentOS Stream 9 - AppStream                                                                    3.1 MB/s |  20 MB     00:06
CentOS Stream 9 - Extras packages                                                               16 kB/s |  19 kB     00:01
Last metadata expiration check: 0:00:01 ago on Wednesday 25 September 2024 03:20:12 PM.
Dependencies resolved.
===========================================================================================================================
 Package                            Architecture        Version                              Repository                Size
===========================================================================================================================
Installing:
 kernel                             x86_64              5.14.0-511.el9                       baseos                    59 k
Upgrading:
 NetworkManager                     x86_64              1:1.51.0-1.el9                       baseos                   2.3 M
 NetworkManager-adsl                x86_64              1:1.51.0-1.el9                       baseos                    36 k
 NetworkManager-bluetooth           x86_64              1:1.51.0-1.el9                       baseos                    62 k
 NetworkManager-config-server       noarch              1:1.51.0-1.el9                       baseos                    21 k
 NetworkManager-libnm               x86_64              1:1.51.0-1.el9                       baseos                   1.8 M
 NetworkManager-libreswan           x86_64              1.2.22-2.el9                         appstream                159 k
 NetworkManager-team                x86_64              1:1.51.0-1.el9                       baseos                    41 k
 NetworkManager-tui                 x86_64              1:1.51.0-1.el9                       baseos                   249 k
 NetworkManager-wifi                x86_64              1:1.51.0-1.el9                       baseos                    84 k
 NetworkManager-wwan                x86_64              1:1.51.0-1.el9                       baseos                    69 k
 augeas-libs                        x86_64              1.14.1-2.el9                         appstream                424 k
 bind                               x86_64              32:9.16.23-24.el9                    appstream                505 k
 bind-chroot                        x86_64              32:9.16.23-24.el9                    appstream                 21 k
 bind-dnssec-doc                    noarch              32:9.16.23-24.el9                    appstream                 46 k
 bind-dnssec-utils                  x86_64              32:9.16.23-24.el9                    appstream                118 k
 bind-libs                          x86_64              32:9.16.23-24.el9                    appstream                1.2 M
 bind-license                       noarch              32:9.16.23-24.el9                    appstream                 14 k
```

sudo yum install vsftpd

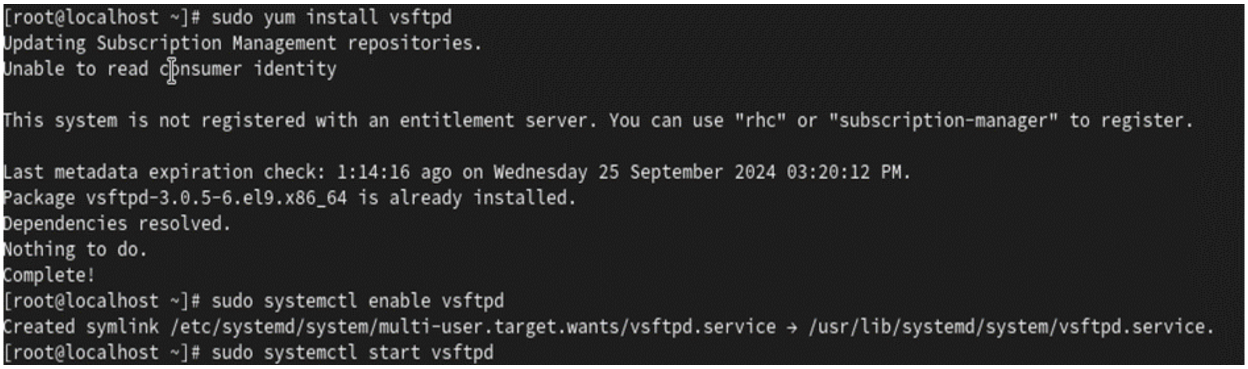
The command sudo yum install vsftpd is used to install the vsftpd (Very Secure File Transfer Protocol Daemon) package on Red Hat-based Linux distributions like CentOS or Amazon Linux. This software allows you to set up and manage an FTP server, enabling users to transfer files to and from the server securely. Running this command will download and install the vsftpd package and its dependencies from the repository.

sudo systemctl enable vsftpd

The command sudo systemctl enable vsftpd enables the vsftpd service to start automatically at boot on systems using systemd (like CentOS, RHEL, or Amazon Linux 2). This ensures that the FTP server starts whenever the system is rebooted or powered on, without needing to manually start it each time. It does not start the service immediately but sets it to start on future boots.

sudo systemctl start vsftpd

The command sudo systemctl start vsftpd is used to start the vsftpd (FTP server) service immediately on a system running systemd. After executing this, the FTP server will be running, allowing users to transfer files to and from the server. To ensure the service starts automatically at boot, you would pair this command with sudo systemctl enable vsftpd.



```
[root@localhost ~]# sudo yum install vsftpd
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Last metadata expiration check: 1:14:16 ago on Wednesday 25 September 2024 03:20:12 PM.
Package vsftpd-3.0.5-6.el9.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@localhost ~]# sudo systemctl enable vsftpd
Created symlink /etc/systemd/system/multi-user.target.wants/vsftpd.service → /usr/lib/systemd/system/vsftpd.service.
[root@localhost ~]# sudo systemctl start vsftpd
```

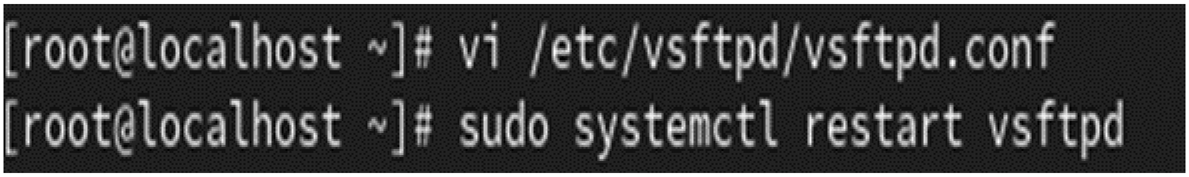
vi etc/vsftpd/vsftpd.conf

The command vi /etc/vsftpd/vsftpd.conf opens the configuration file for vsftpd (Very Secure FTP Daemon) using the vi text editor. This file contains the settings for the FTP server, such as access permissions, security options, anonymous login, and FTP-related parameters.

By editing this file, we can customize the behavior of your FTP server to suit your needs.

sudo systemctl restart vsftpd

It is used to restart the vsftpd service. It stops the currently running FTP service and starts it again, applying any new changes made to the configuration file or clearing any issues that may have occurred. This is commonly used after modifying the /etc/vsftpd/vsftpd.conf file to ensure the changes take effect.



```
[root@localhost ~]# vi /etc/vsftpd/vsftpd.conf
[root@localhost ~]# sudo systemctl restart vsftpd
```

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

It allows FTP traffic through the firewall in the public zone, ensuring the FTP service can be accessed from external networks. The --permanent flag makes this rule persistent across reboots.

sudo firewall-cmd --reload

This command reloads the firewall configuration to apply any new changes made (such as allowing FTP service). It activates the updated rules without needing a system reboot.

sudo adduser ftpuser

This command creates a new user named ftpuser on the system. This user can be given access to the FTP server for file transfers.

sudo passwd ftpuser

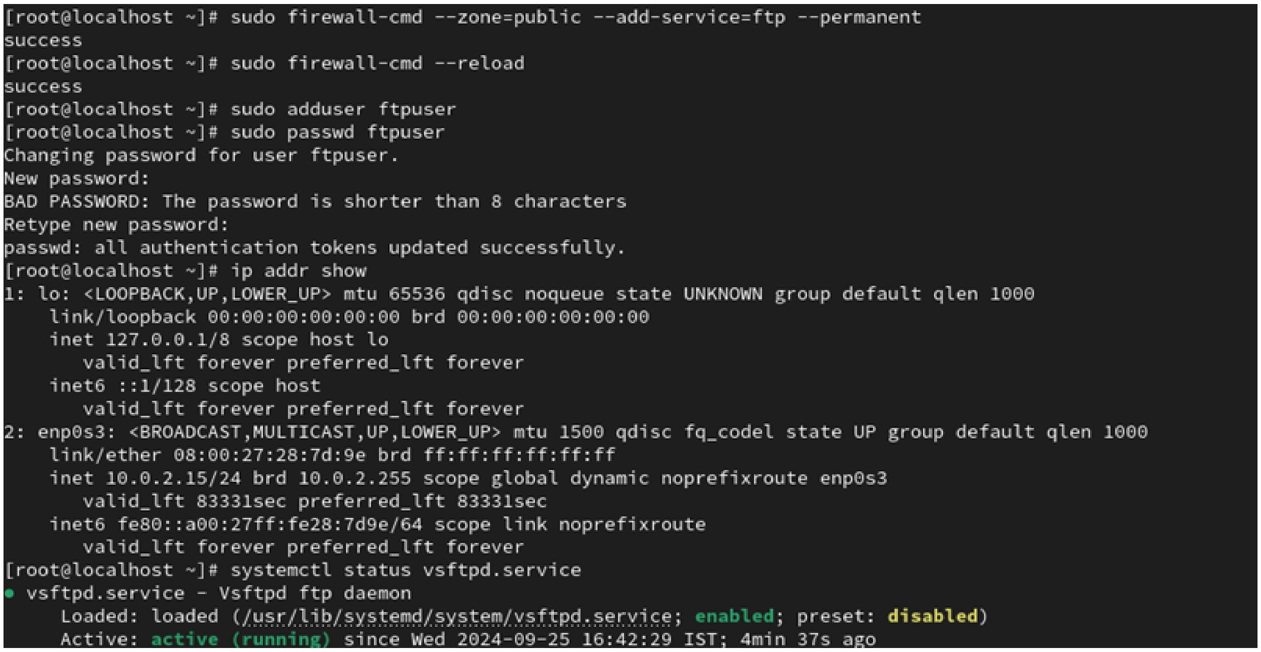
This command sets or changes the password for the ftpuser account. The user will need this password to log in to the FTP server.

ip addr show

This command displays the IP addresses and network interfaces of the system. It's useful for identifying the server's IP address that FTP clients will connect to.

systemctl status vsftpd.service

This command checks the status of the vsftpd service, showing whether it is running, stopped, or experiencing any issues. It also provides logs and details about the service’s current state.



```
[root@localhost ~]# sudo firewall-cmd --zone=public --add-service=ftp --permanent
success
[root@localhost ~]# sudo firewall-cmd --reload
success
[root@localhost ~]# sudo adduser ftpuser
[root@localhost ~]# sudo passwd ftpuser
Changing password for user ftpuser.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@localhost ~]# ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:28:7d:9e brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 83331sec preferred_lft 83331sec
    inet6 fe80::a00:27ff:fe28:7d9e/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[root@localhost ~]# systemctl status vsftpd.service
● vsftpd.service - Vsftpd ftp daemon
     Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; enabled; preset: disabled)
     Active: active (running) since Wed 2024-09-25 16:42:29 IST; 4min 37s ago
```

chmod a-w /home/ftpuser

This command removes write permissions from the /home/ftpuser directory, ensuring that the FTP user cannot upload or modify files within their home directory. It's often done to secure FTP accounts.

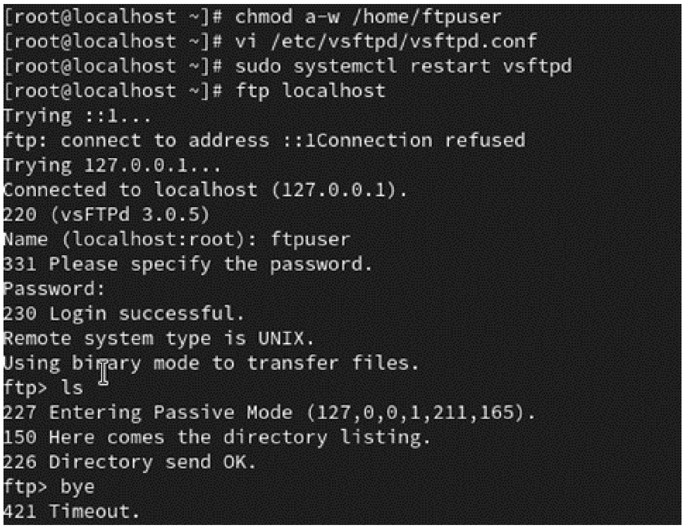
vi /etc/vsftpd/vsftpd.conf

This command opens the vsftpd configuration file for editing using the vi editor. You can make changes to customize FTP settings such as anonymous access, local user permissions, or security options.

sudo systemctl restart

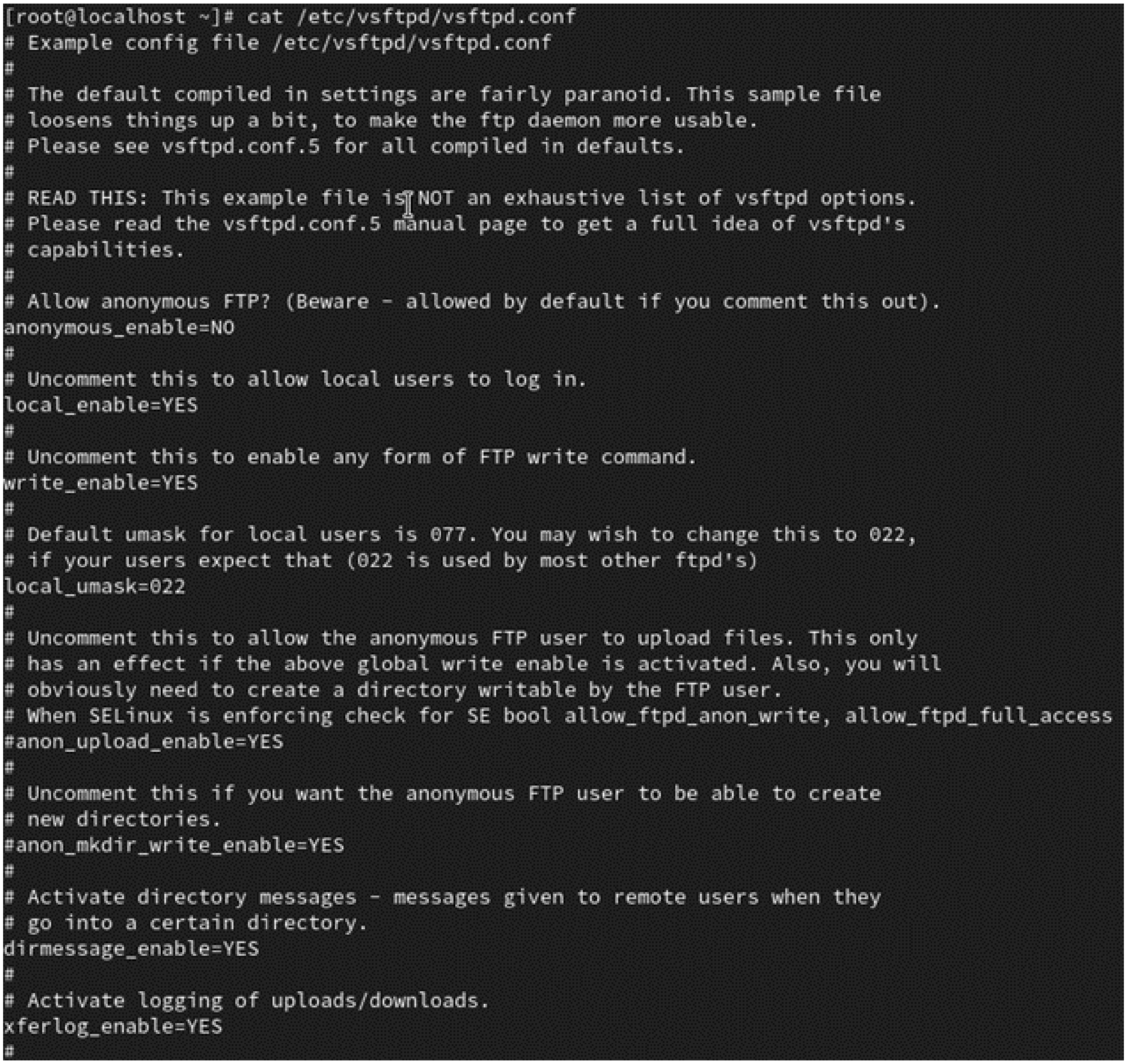
ftp localhost

This command initiates an FTP session with the local machine (localhost). It is typically used to test the FTP server by connecting to it from the same server it's running on.

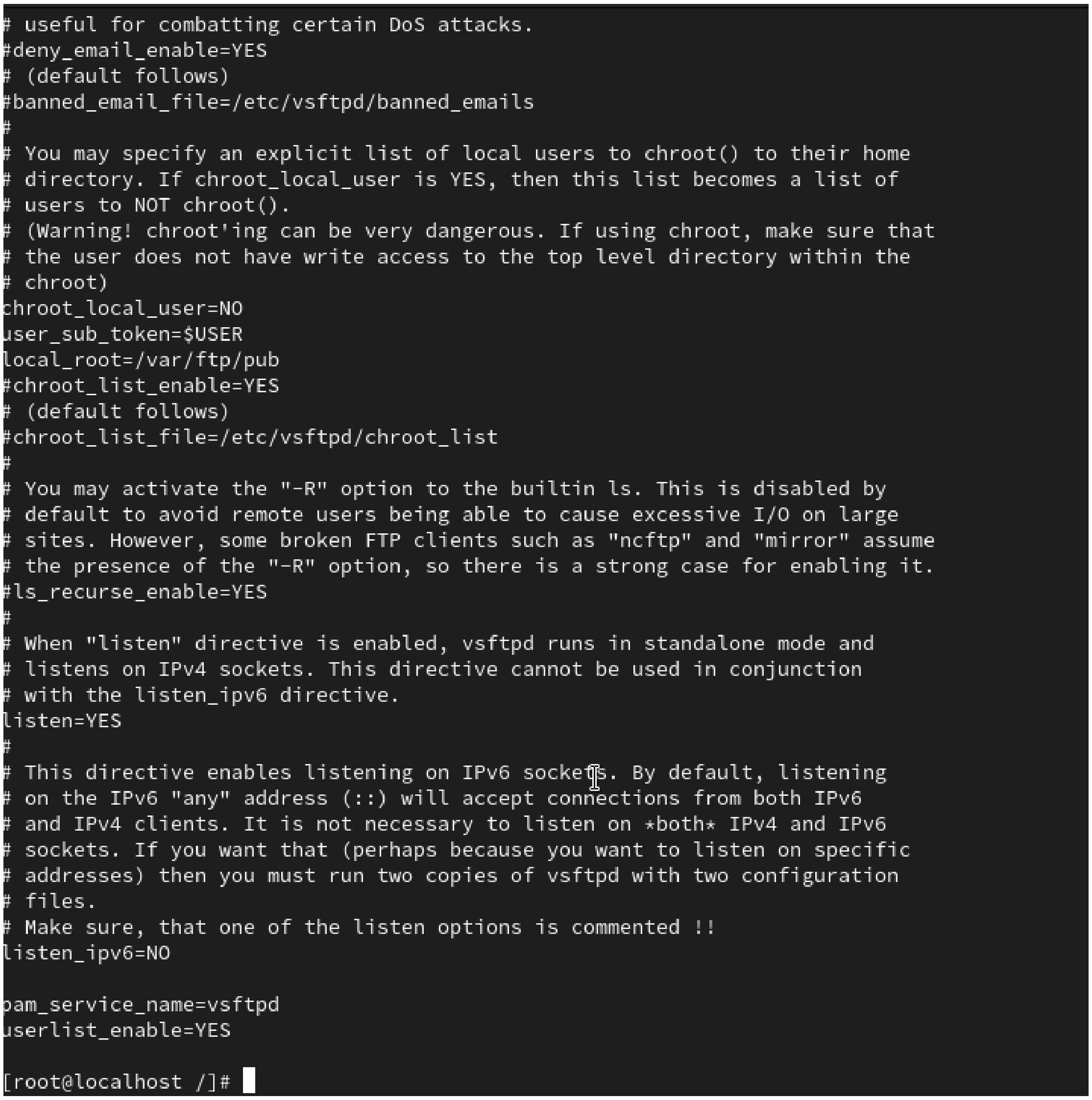


```
[root@localhost ~]# chmod a-w /home/ftpuser
[root@localhost ~]# vi /etc/vsftpd/vsftpd.conf
[root@localhost ~]# sudo systemctl restart vsftpd
[root@localhost ~]# ftp localhost
Trying ::1...
ftp: connect to address ::1Connection refused
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
220 (vsFTPd 3.0.5)
Name (localhost:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode (127,0,0,1,211,165).
150 Here comes the directory listing.
226 Directory send OK.
ftp> bye
421 Timeout.
```

Lets check the contents present in vsftpd.conf :(main part attached)



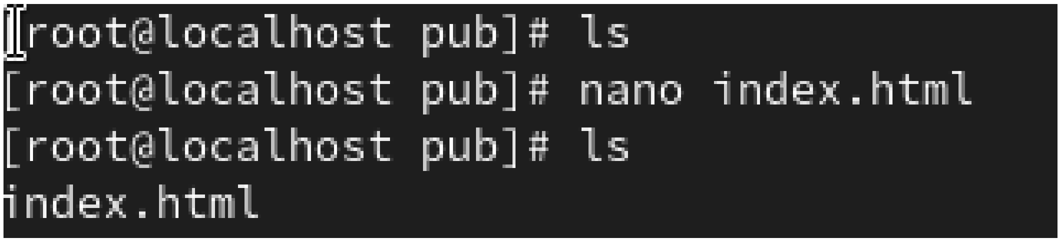
```
[root@localhost ~]# cat /etc/vsftpd/vsftpd.conf
# Example config file /etc/vsftpd/vsftpd.conf
#
# The default compiled in settings are fairly paranoid. This sample file
# loosens things up a bit, to make the ftp daemon more usable.
# Please see vsftpd.conf.5 for all compiled in defaults.
#
# READ THIS: This example file is NOT an exhaustive list of vsftpd options.
# Please read the vsftpd.conf.5 manual page to get a full idea of vsftpd's
# capabilities.
#
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
#
# Uncomment this to allow local users to log in.
local_enable=YES
#
# Uncomment this to enable any form of FTP write command.
write_enable=YES
#
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftpd's)
local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
# When SELinux is enforcing check for SE bool allow_ftpd_anon_write, allow_ftpd_full_access
#anon_upload_enable=YES
#
# Uncomment this if you want the anonymous FTP user to be able to create
# new directories.
#anon_mkdir_write_enable=YES
#
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# Activate logging of uploads/downloads.
xferlog_enable=YES
#
```



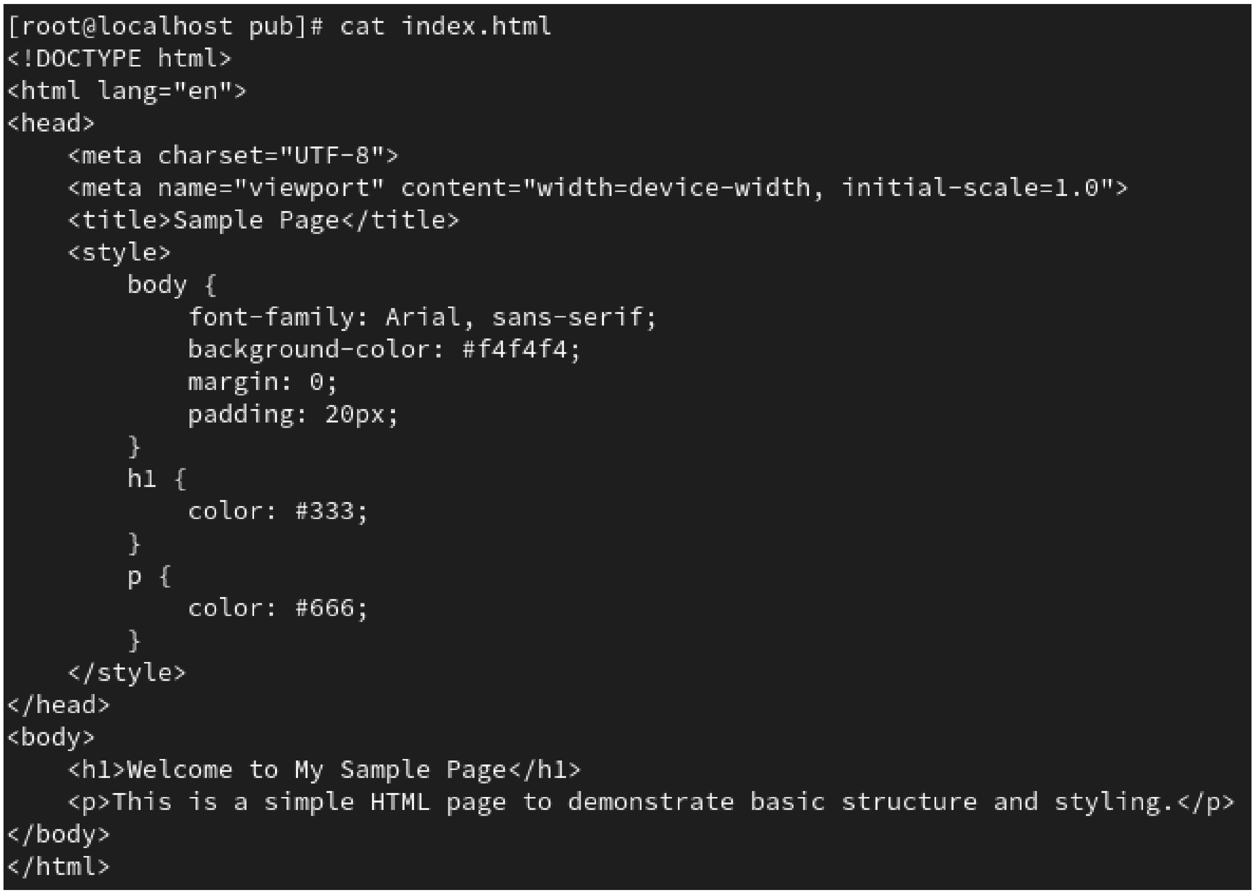
```
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
# (default follows)
#banned_email_file=/etc/vsftpd/banned_emails
#
# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
# (Warning! chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot)
hroot_local_user=NO
user_sub_token=$USER
local_root=/var/ftp/pub
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list
#
# You may activate the "-R" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "mirror" assume
# the presence of the "-R" option, so there is a strong case for enabling it.
#ls_recurse_enable=YES
#
# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the listen_ipv6 directive.
listen=YES
#
# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on *both* IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with two configuration
# files.
# Make sure, that one of the listen options is commented !!
listen_ipv6=NO

pam_service_name=vsftpd
userlist_enable=YES

[root@localhost /]#
```

Now, accessing the files using filezilla:

Navigate to the /var/ftp/pub directory and create a file named index.html. Add the code inside this file save and exit.

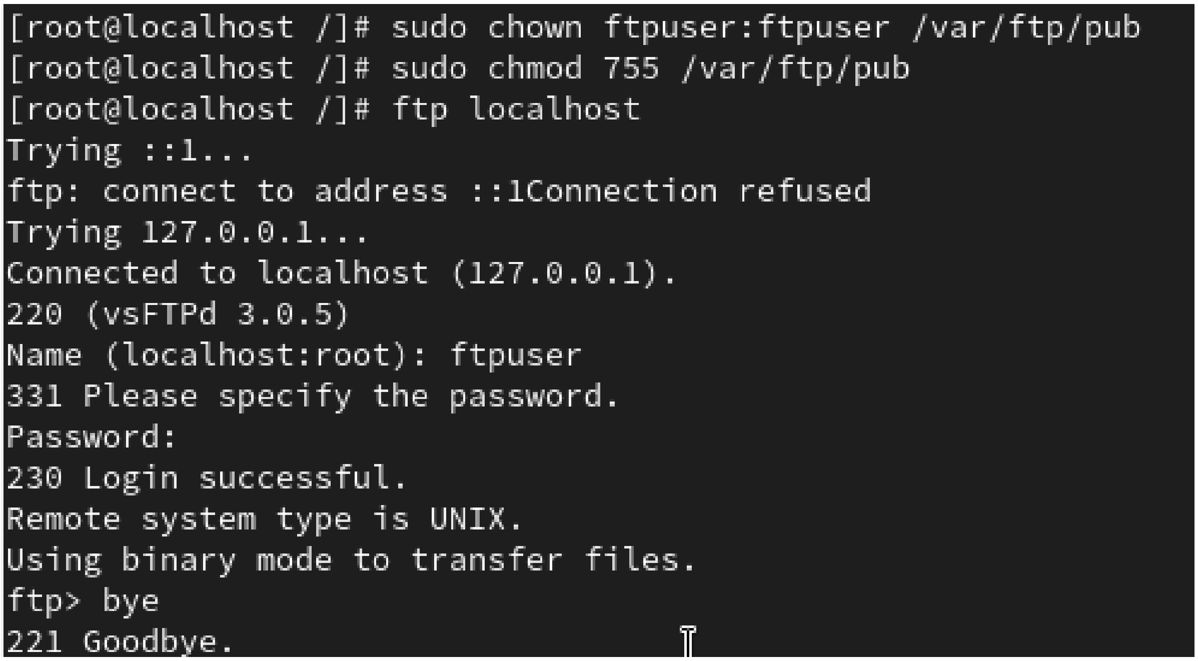


```
[root@localhost pub]# ls
[root@localhost pub]# nano index.html
[root@localhost pub]# ls
index.html
```



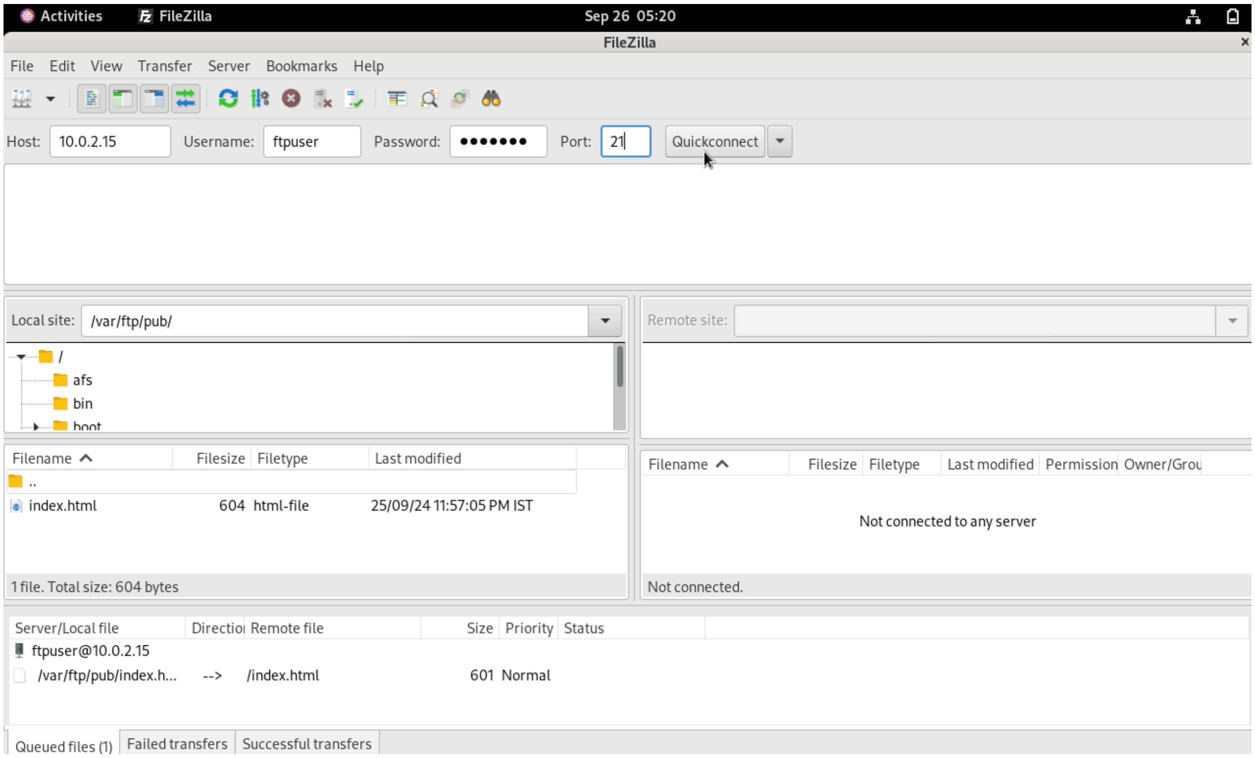
```
[root@localhost pub]# cat index.html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Sample Page</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
        }
        h1 {
            color: #333;
        }
        p {
            color: #666;
        }
    </style>
</head>
<body>
    <h1>Welcome to My Sample Page</h1>
    <p>This is a simple HTML page to demonstrate basic structure and styling.</p>
</body>
</html>
```

Applying permissions in order to access the file.



```
[root@localhost /]# sudo chown ftpuser:ftpuser /var/ftp/pub
[root@localhost /]# sudo chmod 755 /var/ftp/pub
[root@localhost /]# ftp localhost
Trying ::1...
ftp: connect to address ::1Connection refused
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
220 (vsFTPd 3.0.5)
Name (localhost:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> bye
221 Goodbye.
```

Filezilla:



Host: 10.0.2.15 Username: ftpuser Password: ••••••• Port: 21

Local site: /var/ftp/pub/

```
/
  afs
  bin
  boot
```

Filename	Filesize	Filetype	Last modified
..			
index.html	604	html-file	25/09/24 11:57:05 PM IST

1 file. Total size: 604 bytes

Remote site:

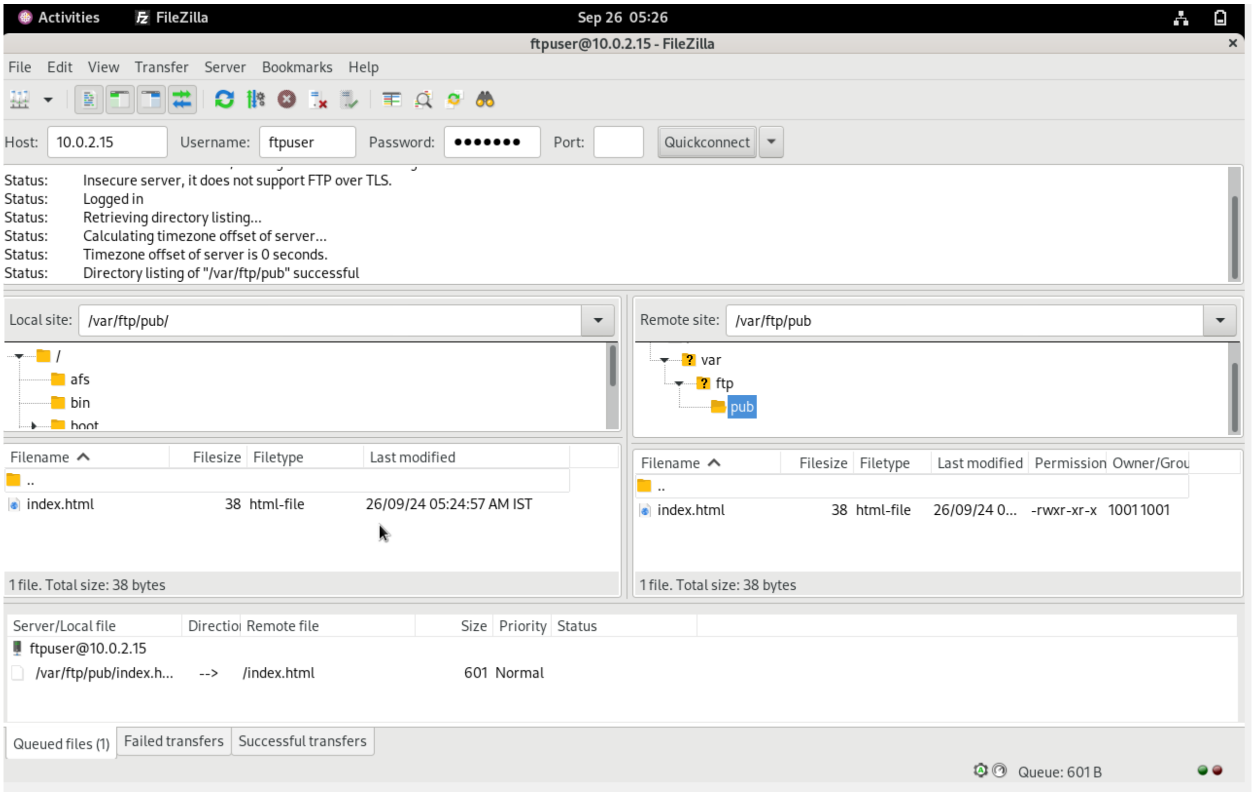
Filename	Filesize	Filetype	Last modified	Permission	Owner/Grou

Not connected to any server

Not connected.

Server/Local file	Directio	Remote file	Size	Priority	Status
ftpuser@10.0.2.15					
/var/ftp/pub/index.h...	-->	/index.html	601	Normal	

Queued files (1) | Failed transfers | Successful transfers


Activities
FileZilla
Sep 26 05:26
ftpuser@10.0.2.15 - FileZilla
File Edit View Transfer Server Bookmarks Help
Quickconnect

Host: 10.0.2.15 Username: ftpuser Password: ••••••• Port:

```
Status:    Insecure server, it does not support FTP over TLS.
Status:    Logged in
Status:    Retrieving directory listing...
Status:    Calculating timezone offset of server...
Status:    Timezone offset of server is 0 seconds.
Status:    Directory listing of "/var/ftp/pub" successful
```

Local site: /var/ftp/pub/

```
/
  afs
  bin
  boot
```

Filename	Filesize	Filetype	Last modified
..			
index.html	38	html-file	26/09/24 05:24:57 AM IST

1 file. Total size: 38 bytes

Remote site: /var/ftp/pub

```
var
  ftp
    pub
```

Filename	Filesize	Filetype	Last modified	Permission	Owner/Grou
..					
index.html	38	html-file	26/09/24 0...	-rwxr-xr-x	1001 1001

1 file. Total size: 38 bytes

Server/Local file	Directio	Remote file	Size	Priority	Status
ftpuser@10.0.2.15					
/var/ftp/pub/index.h...	-->	/index.html	601	Normal	

Queued files (1) | Failed transfers | Successful transfers

Queue: 601 B

Conclusion:

In summary, setting up Samba and vsftpd on CentOS provides a reliable and secure solution for file sharing and file transfer services. Samba allows seamless file sharing across Linux and Windows platforms, making it suitable for environments with mixed operating systems. Its flexibility in controlling user permissions and access ensures secure and organized data sharing.

On the other hand, vsftpd offers a robust and secure FTP solution, supporting features like anonymous access restrictions, user authentication, and encrypted transfers via FTPS. It is lightweight and easy to configure, making it an excellent choice for managing file transfers efficiently.

Regular maintenance, including security updates and log monitoring, is crucial to ensure these services remain secure and performant. By deploying these services, administrators can provide users with reliable, centralized access to shared resources while maintaining control over access permissions and ensuring data security.