CentOS SSH Installation And Configuration

How do I install and configure ssh server and client under CentOS Linux operating systems?

You need to install the following packages (which are installed by default until and unless you removed it or skipped it while installing CentOS)

- openssh-clients : The OpenSSH client applications
- openssh-server : The OpenSSH server daemon

OpenSSH Installations under CentOS Linux

To install the server and client type:

```
# yum -y install openssh-server openssh-clients
```

Start the service:

```
# chkconfig sshd on
# service sshd start


or


#systemctl start sshd
```

Make sure port 22 is opened:

```
# netstat -tulpn | grep :22
```

Firewall Settings

Edit /etc/sysconfig/iptables (IPv4 firewall),

```
# vi /etc/sysconfig/iptables
```

Add the line

```
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
```

If you want to restict access to 192.168.1.0/24, edit it as follows:

```
-A RH-Firewall-1-INPUT -s 192.168.1.0/24 -m state --state NEW -p tcp --dport 22 -j ACCEPT
```

If your site uses IPv6, and you are editing ip6tables, use the line:

```
-A RH-Firewall-1-INPUT -m tcp -p tcp --dport 22 -j ACCEPT
```

Save and close the file. Restart iptables:

```
# service iptables restart
```

OpenSSH Server Configuration

Edit /etc/ssh/sshd_config, enter:

```
# vi /etc/ssh/sshd_config
```

To disable root logins, edit or add as follows:

```
PermitRootLogin no
```

Restrict login to user tom and jerry only over ssh:

```
AllowUsers tom jerry
```

Change ssh port i.e. run it on a non-standard port like 1235

```
Port 1235
```

Save and close the file. Restart sshd:

```
# service sshd restart
```

CentOS / RHEL 7 : How to install and configure telnet

By admin

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. To check if you have telnet package already installed on your system, use :

```
# rpm -qa | grep telnet
telnet-server-0.17-59.el7.x86_64
telnet-0.17-59.el7.x86_64
```

In order to turn Telnet on make sure that you have the packages **telnet-server** and **telnet** installed:

Installing telnet packages

If the 2 required packages are not installed, install it using yum.

```
# yum install telnet-server telnet
```

Configuring/enabling telnet

1. Add the service to firewalld.

The built in firewalld blocks Telnet port 23 by default because the protocol is not considered secure. Please make sure that the port is open or if a non-default port is being used, that the port associated with Telnet is open for telnet traffic to pass through.

```
# firewall-cmd --add-service=telnet --zone=public
```

Run the rule again with the “**–permanent**” flag for it to persist across firewalld restarts.

```
# firewall-cmd --add-service=telnet --zone=public --permanent
```

2. Add the service to selinux.

You will have to also add the service to SELinux. This is required only in the case where SELinux is enabled on the system.

```
# semanage port -a -t telnetd_port_t -p tcp
```

3. Enable and start the telnet service.

Start the service using the systemctl command.

```
# systemctl start telnet.socket
```

Enable the telnet service to start at boot.

```
# systemctl enable telnet.socket
```

4. Verify

Once you are done with the configuration, verify if the telnet to a server works.

```
# telnet <ip address>
Trying 10.10.10.10...
Connected to 10.10.10.10.
Escape character is '^]'.

Kernel 3.10.0-327.el7.x86_64 on an x86_64
<ip addr> login: hpcsa
Password:
Last login: Sat Jan 23 18:19:43 from <ip address>

[hpcsa@<ip address> ~]$ hostname
```

FTP Server

Install FTP Server on CentOS 7

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
```

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
```

Note: If you use a different firewall application, refer to the documentation to configure it correctly for Port 21. Also, some FTP clients use Port 20, so you may wish to include that rule as well. Simply copy the first line, and replace 21 with 20.

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 causes a problem.

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

```
sudo vi /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
```

Note: By default, this line starts with a # sign to indicate it's a comment. Commenting is a useful way to turn commands on and off. The # sign can also be used to make notes in the file without the system interpreting them as instructions.

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
```

Note: for test purposes, the **allow_writeable_chroot=YES** option will create a functioning FTP server that you can test and use. Some administrators advocate the use of the **user_sub_token** option for better security. Refer to the vsftpd documentation for more information on this option.

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 useradd 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 “hpcsa” | 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
```

Note: 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
```

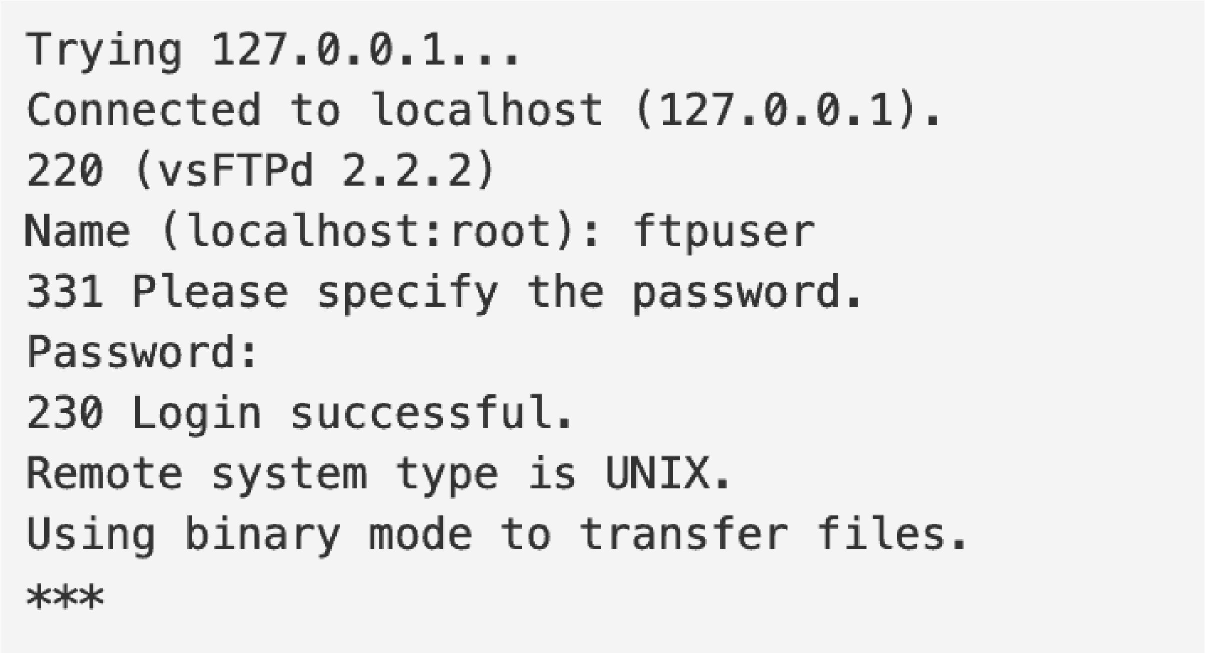
Replace this IP address with the one from your system. You can find your IP address with the `ip addr` command.

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

Step 4: Test the FTP server

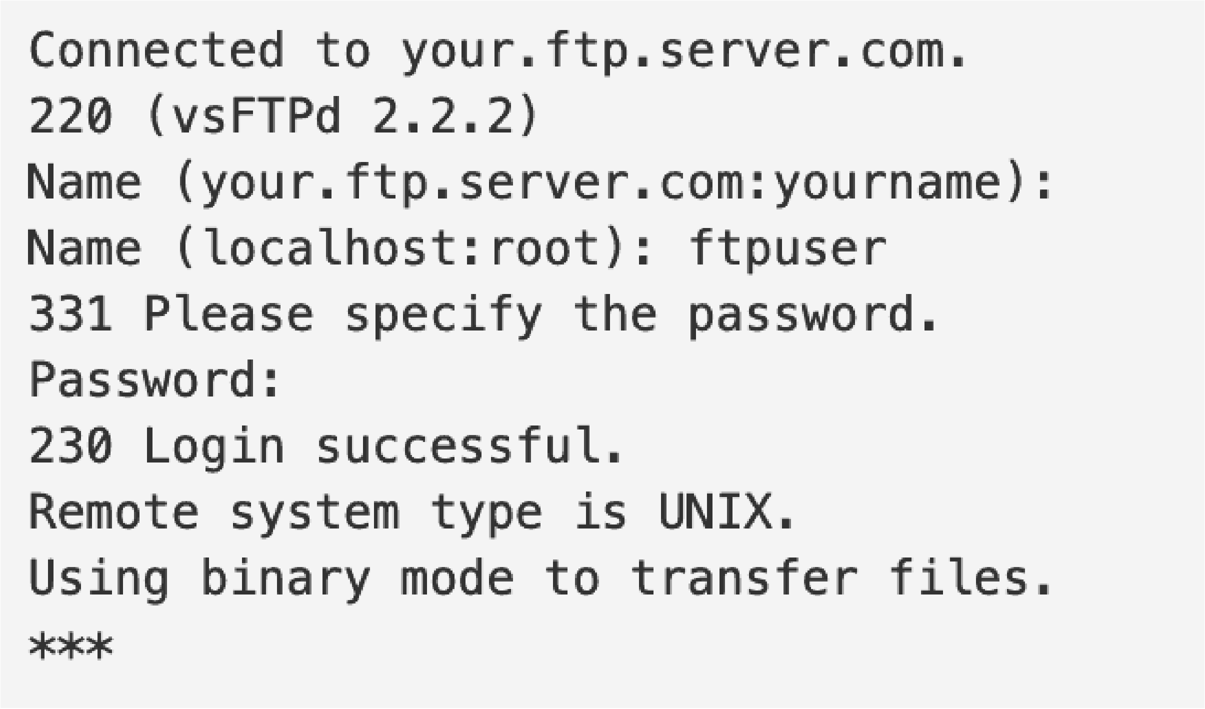
To Test the FTP Server Locally, use the command:

```
ftp localhost
```



```
Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
220 (vsFTPd 2.2.2)
Name (localhost:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
***
```

To Test remotely, use the command:
`ftp your.ftp.server.com`



```
Connected to your.ftp.server.com.
220 (vsFTPd 2.2.2)
Name (your.ftp.server.com:yourname):
Name (localhost:root): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
***
```

Note: While some security measures have been included in this guide, it is strongly recommended that you familiarize yourself with the latest security protocols before implementing an FTP server in a production environment. This is especially important if you’re creating an FTP server that’s open to the internet – many security breaches originate through the FTP protocol.

ftp client

FTP Client : CentOS

2014/07/20

Configure Client computer to connect to FTP Server. The example below is for CentOS.

[1] Install FTP Client.

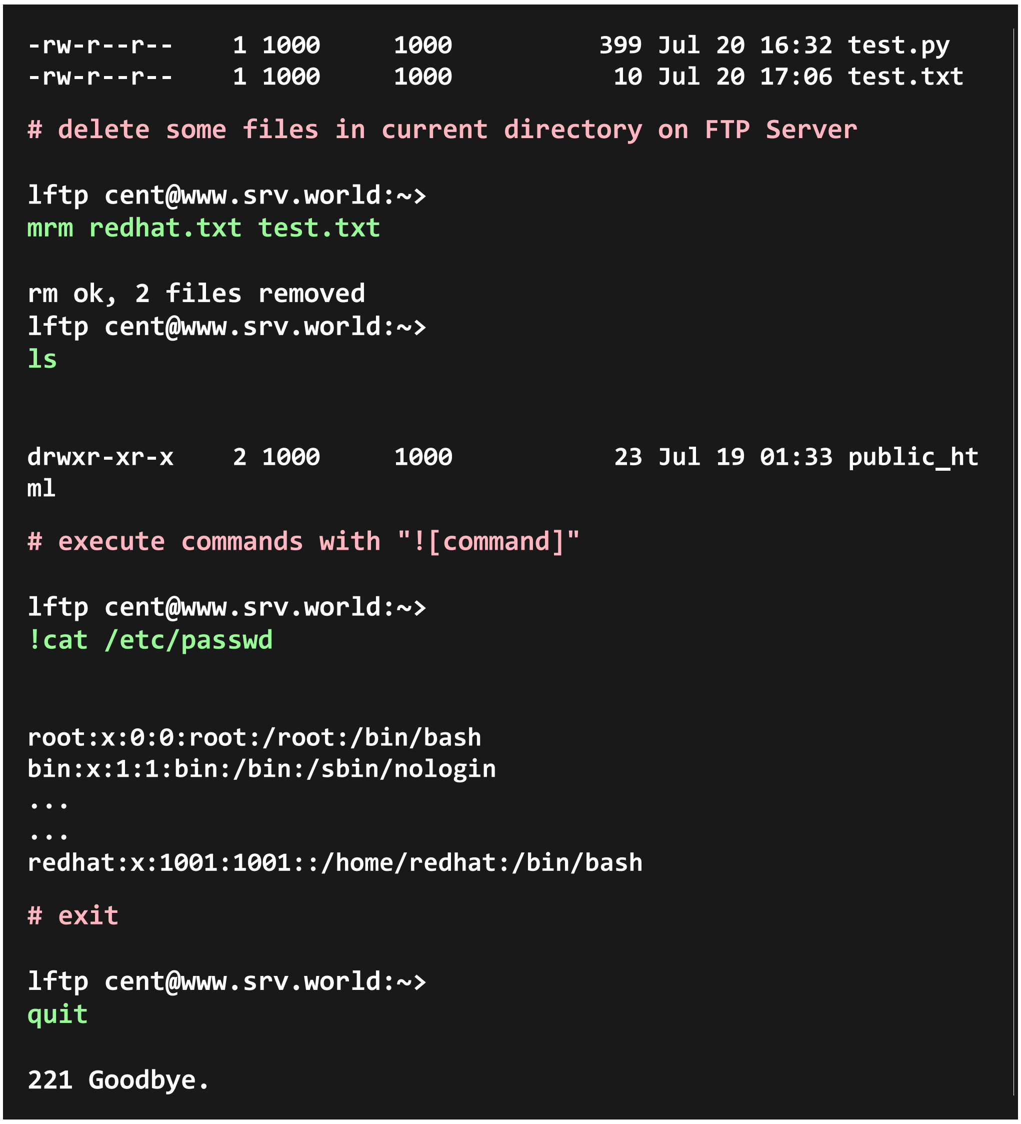
```
[root@dlp ~]#
yum -y install lftp
```

[2] The connection with root account is prohibited by default, so access with an common user to FTP Server.

```
# lftp [option] [hostname]

[redhat@dlp ~]$
lftp -u cent www.srv.world

Password:
# password of the user

lftp cent@www.srv.world:~>

# show current directory on FTP server

lftp cent@www.srv.world:~>
pwd

ftp://cent@www.srv.world

# show current directory on local server

lftp cent@www.srv.world:~>
!pwd

/home/redhat

# show files in current directory on FTP server

lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py

# show files in current directory on local server

lftp cent@www.srv.world:~>
!ls -l


total 12
-rw-rw-r-- 1 redhat redhat 10 Jul 20 14:30 redhat.txt
```

```
-rw-rw-r-- 1 redhat redhat 10 Jul 20 14:59 test2.txt
-rw-rw-r-- 1 redhat redhat 10 Jul 20 14:59 test.txt

# change directory

lftp cent@www.srv.world:~>
cd public_html

lftp cent@www.srv.world:~/public_html>
pwd

ftp://cent@www.srv.world/%2Fhome/cent/public_html

# upload a file to FTP server

# "-a" means ascii mode ( default is binary mode )

lftp cent@www.srv.world:~>
put -a redhat.txt

22 bytes transferred
Total 2 files transferred
lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml
-rw-r--r--    1 1000     1000           10 Jul 20 17:01 redhat.tx
t
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:01 test.txt

# upload some files to FTP server

lftp cent@www.srv.world:~>
mput -a test.txt test2.txt

22 bytes transferred
Total 2 files transferred
lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test2.txt

# download a file from FTP server

# "-a" means ascii mode ( default is binary mode )

lftp cent@www.srv.world:~>
get -a test.py
```

```
416 bytes transferred
# download some files from FTP server

lftp cent@www.srv.world:~>
mget -a test.txt test2.txt

20 bytes transferred
Total 2 files transferred
# create a directory in current directory on FTP Server

lftp cent@www.srv.world:~>
mkdir testdir

mkdir ok, `testdir' created
lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test2.txt
drwxr-xr-x    2 1000     1000            6 Jul 20 17:16 testdir
226 Directory send OK.
# delete a direcroty in current directory on FTP Server

lftp cent@www.srv.world:~>
rmdir testdir

rmdir ok, `testdir' removed
lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test2.txt
# delete a file in current directory on FTP Server

lftp cent@www.srv.world:~>
rm test2.txt

rm ok, `test2.txt' removed
lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml
```



```
-rw-r--r--    1 1000     1000          399 Jul 20 16:32 test.py
-rw-r--r--    1 1000     1000           10 Jul 20 17:06 test.txt

# delete some files in current directory on FTP Server

lftp cent@www.srv.world:~>
mrm redhat.txt test.txt

rm ok, 2 files removed
lftp cent@www.srv.world:~>
ls


drwxr-xr-x    2 1000     1000           23 Jul 19 01:33 public_ht
ml

# execute commands with "![command]"

lftp cent@www.srv.world:~>
!cat /etc/passwd


root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
...
...
redhat:x:1001:1001::/home/redhat:/bin/bash

# exit

lftp cent@www.srv.world:~>
quit

221 Goodbye.
```