

Service Setup

Table of contents

| | | |
|----------|---------------------|----------|
| 1 | Introduction | 1 |
| 2 | HTTP Server | 1 |
| 3 | FTP Server | 2 |
| 4 | SSH Server | 5 |
| 5 | MySQL Server | 6 |

1 Introduction

In this tutorial you are shown how to setup an HTTP server, an FTP server, an SSH server, and a MySQL server.

2 HTTP Server

HTTP stands for Hypertext Transfer Protocol. The HTTP server will be hosted on port 80 (the default http port). For the HTTP server, we will use Apache. For more about the Apache HTTP server project, visit <https://httpd.apache.org/>.

Complete the following steps to set up the server:

1. Install Apache

```
sudo apt-get install apache2
```

2. Check that it is running

```
sudo netstat -alpn | grep ' LISTEN '
```

The output from running the command above should include the following, or something similar to the following. Specifically, you will see a reference to Apache on the right hand side and port 80 in the middle.

```
...
tcp6      0      0 :::80          :::*           LISTEN      29022/apache2
...
```

3. Visit the website that is being hosted by visiting <http://localhost>.
4. Confirm that the code for the website is located at `/var/www/html/index.html` by viewing it with the following command.

```
sudo more /var/www/html/index.html
```

3 FTP Server

FTP stands for File Transfer Protocol. Your FTP server will be hosted on port 21 (the default port). We will use Very Secure FTP Daemon, or vsftpd for short. For more about vsftpd, visit <https://security.appspot.com/vsftpd.html>.

To setup the FTP server, complete the following:

1. Install vsftpd.

```
sudo apt-get install vsftpd
```

2. Confirm that it is running.

```
sudo netstat -alpn | grep ' LISTEN '
```

The output should contain a line similar to the following. Specifically, note that vsftpd is running on port 21.

```
...
tcp       0      0 0.0.0.0:21      0.0.0.0:*      LISTEN      3309/vsftpd
...
```

3. Confirm you can login using the user credentials for your VM. After typing the following command, you'll be prompted for your username and password.

```
ftp localhost
```

If you provide a valid username and password, it will respond with something that looks similar to the following:

```
Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>
```

4. Confirm you are in the home directory for your user by typing `ls -lh` to see the files.
5. Disconnect the ftp connection by typing `control+d`
6. Open the ftp server configuration file.

```
sudo nano /etc/vsftpd.conf
```

Troubleshooting Tip

If the file is blank, you may have mistyped the previous command.

7. Change the configuration file by finding and modifying the variables named `anonymous_enable`, `local_enable`, and `ftpd_banner` and then adding a variable named `anon_root` at the bottom of the file. Note that you can search for each variable in nano using `ctrl+w`.

Note that if any of the variables are commented out, you will need to uncomment them.

```
anonymous_enable=YES  
local_enable=NO  
  
# ftpd_banner probably needs to be uncommented  
ftpd_banner=Whatever you want  
  
# add anon_root var at bottom of the file if it does not exist  
# note that the folder /home/anonymous/ftp doesn't exist yet  
anon_root=/home/anonymous/ftp
```

8. Save and close the config file.
9. Restart the FTP server.

```
sudo service vsftpd restart
```

10. Create the `anon_root` directory and add a test file.
 - a. Navigate to `/home`.

```
cd /home
```

- b. Create a folder named **anonymous**.

```
sudo mkdir anonymous
```

- c. Navigate into the newly created folder and create a folder named **ftp**

```
cd anonymous  
sudo mkdir ftp
```

- d. Navigate into the **ftp** directory and add a file to this folder named **ftp-test**.

```
cd ftp  
sudo touch ftp-test
```

11. Test the ftp server again.

- a. Login to the FTP server.

When prompted for the username, provide **anonymous**.

When prompted for the password, leave it blank and just hit ENTER.

```
ftp localhost
```

- b. Once you are into the ftp server, type **ls -lh** again to confirm that you can see the file named **ftp-test** that you created in a previous step.

12. Test the connection from your host machine.

- a. Find the ip address for you virtual machine with the **ifconfig** command.
- b. From a terminal on your host machine (see tips below), type **ftp X.X.X.X** to access the ftp server. Replace X.X.X.X with the appropriate ip address for your virtual machine.
- c. Type **ls** to make sure you can see the file named **ftp-test**

💡 Tip: Using Windows on your host machine?

If you have a windows host, the ftp command should work by default in powershell.

💡 Tip: Using macOS on your host machine?

You may need to install ftp with **brew install inetutils**. If you do not have homebrew, it is recommended that you install it from <https://brew.sh>

4 SSH Server

SSH stands for Secure Shell. Your ssh server will be hosted on port 22 (the default port for ssh). We will use OpenSSH. For more info about OpenSSH visit <https://www.openssh.com/>.

To setup the ssh server, complete the following:

1. Install OpenSSH Server

```
sudo apt-get install openssh-server
```

2. Confirm that it is running by running the following command. Specifically, confirm `sshd` is running on port 22.

```
sudo netstat -alpn | grep ' LISTEN '
```

The output should contain something similar one of the following text blocks. Specifically, confirm `sshd` is running on port 22. It is possible it may show `init` is running on port 22, which is also fine.

```
...  
tcp        0      0 0.0.0.0:22        0.0.0.0:*        LISTEN      1091/sshd  
...
```

Or it may have this:

```
...  
tcp        0      0 0.0.0.0:22        0.0.0.0:*        LISTEN      1091/init  
...
```

3. Navigate to a directory that isn't your home directory, let's say the desktop. We will be looking for a change in the prompt after completing the next instruction.
4. Log in with your credentials by running the following command. The first time you connect, it will ask if you want to trust the server and prompt if you want to continue. Next, it will prompt you for a password. If successful, a welcome message may or may not appear, however if you were not at the home directory, the prompt should change back to the home directory.

```
ssh localhost
```

5. Test that you are now in your home directory by typing `pwd`.
6. Disconnect from ssh using `ctrl + d`. You will receive a message that the connection has been closed. If you were not connected in the first place, then the terminal will close instead of disconnecting you.
7. Try logging in from your host machine.

- a. First, find your vm's ip address using `ifconfig` on your vm.
 - b. On your host machine, open the Terminal application and use the following command to connect. Replace the word `user` with the name of the user on your vm, and `x.x.x.x` with your vm's ip_address.

```
ssh user@x.x.x.x
```
 - c. Once you are in, run a command like `pwd` to test it out.
 - d. Disconnect from the vm.
8. For Cyber Storm, labs, and challenges you will need to do the following.

- a. Open the ssh config file.

```
sudo nano /etc/ssh/sshd_config
```

- b. Find the line with `UseDNS` and be sure it is set to `UseDNS no`

5 MySQL Server

MySQL is an open-source relational database management system. MySQL is a combination of My, the name of the daughter of MySQL's cofounder, and SQL, the acronym for Structured Query Language. Your MySQL server will be hosted on port 3306 (the default port).

Complete the following to setup the MySQL server.

1. Install MySQL server.

```
sudo apt-get install mysql-server
```

2. Begin securing the MySQL server with the command that follows.

```
sudo mysql_secure_installation
```

During the install, you will be asked about the following:

- Would you like to setup VALIDATE PASSWORD component?

Answering ``y`` will then prompt you to select a validation level.
This will ensure users have strong passwords.
The three validation levels are 0, 1, and 2 with 2 being the strongest.
Choose what you feel is appropriate.

- Remove anonymous users?

Choose y for yes.

- Disallow root login remotely?

Choose y for yes.

- Remove test database and access to it?

Choose y for yes.

- Reload privilege tables now?

Choose y for yes.

3. Check that the server is running. Note that it should be running on port 3306.

```
sudo netstat -alpn | grep ' LISTEN '
```

The output should be similar to the following:

```
...  
tcp      0    0 127.0.0.1:3306      0.0.0.0:*           LISTEN      1389/mysqld  
...
```

Notice that it shows 127.0.0.1 as the ip address. This means it is only going to accept local (not remote) connections.

4. Try to login using root credentials. Note that it may prompt you for a password. Leave it blank and just press ENTER.

```
sudo mysql -uroot -p
```

5. Setup the server to allow a remote user to access it.

- a. Add a new user to MySQL (not your vm). Replace 'new-user-name' with something of your choosing and 'password' with something of your choosing.

```
CREATE user 'new-user-name'@'%' IDENTIFIED BY 'password';
```

The % symbol above means the user can connect from anywhere (including remotely)

- b. Create a new database

```
CREATE DATABASE test;
```

- c. Grant the new user all privileges on the new database.

```
GRANT ALL PRIVILEGES ON test.* TO 'new-user-name'@'%';
FLUSH PRIVILEGES;
```

d. Disconnect from MySQL with `ctrl + d`

e. Login as the new user.

```
mysql -unew-user-name -p
```

f. Test that you can see the database named `test` that you just created.

```
SHOW DATABASES;
```

You should be able to see the database named `test`.

g. Disconnect with `ctrl+d`

6. Setup the server to allow for remote access.

a. Open the mysql config file.

```
sudo nano /etc/mysql/mysql.conf.d/mysqld.cnf
```

b. Find the `bind-address` option and set it to `0.0.0.0`.

```
bind-address = 0.0.0.0
```

c. Restart the MySQL server.

```
sudo service mysql restart
```

d. Confirm that it is listening for remote connections.

```
sudo netstat -alpn | grep ' LISTEN '
```

The output should contain something similar to the following. Specifically note `0.0.0.0:3306` for the ip address and port for `mysqld`.

```
...
tcp      0      0 127.0.0.1:3306      0.0.0.0:*           LISTEN      1389/mysqld
...
```

7. Test a remote login.

a. (Optional) For this, it is recommended that you test with a classmate's virtual machine so that you do not need to install mysql on your host machine. However, if you wanted to install mysql on your host, you can do so by using one of the following options. Feel free to skip this step and move to step b.

- For windows users, visit <https://dev.mysql.com/downloads/installer/> and download either installer. The smaller in size installer will ask which features you want to download once you open it, the other installer includes all of the features.
 - For mac users, use `brew install mysql`.
 - For linux users, use `sudo apt install mysql`.
- b. Connect with the following command. Replace 0.0.0.0 with the ip address of the vm you are trying to connect to, and `username` with the name of the user you created previously.

```
mysql -h 0.0.0.0 -P 3306 -u username -p
```