



Internet & Network Services

Assignment 2

Prepared for : Eoin Reagan

Prepared by : Robert Gabriel ,Tracy Nixon (Captain and Number one) Web Development 3

Date : 3 December 2014

Student Id : R00102430, R00027223

Video : <https://www.youtube.com/watch?v=5mgzgElu8D4&list=UUwE-IEIS0k514M1Q8kDXrdg>

Index

INTERNET & NETWORK SERVICES

ASSIGNMENT 2	1
INDEX	2
ASSIGNMENT REQUIREMENTS	3
IP ADDRESS TABLE	4
SETTING UP CONNECTION	5
VM2 – APACHE SERVER (LINUX , APACHE , MYSQL AND PHP)	7
VIRTUAL INSTANCES	8
INSTALL WORDPRESS (VM2)	14
PRINT SERVER (VM2)	16
EMAIL (VM3)	17
NAME SERVERS (MASTER, SLAVES - VM2, VM3)	18
NETWORK FILE SYSTEM (NFS)	24
SSH SERVER (VM2, VM3)	28
FTP (VM3)	29
DYNAMIC HOST CONFIGURATION PROTOCOL (DHCP)	33
SUMMARY & CONCLUSION	35
REFERENCE	36

Assignment Requirements

As a systems administrator you have been asked to implement the following Internet & Network services using Ubuntu for a company called KhufuNet.

1. Web Server (Apache) with Virtual Hosting two sites.
2. DNS Server (BIND), Primary(Master) & Secondary(Slave)
3. DHCP Server for Ubuntu clients
4. SSH Server
5. eMail Server (Postfix) & POP/IMAP Server (Dovecot)
6. FTP Server
7. NFS
8. Network Printing (CUPS)

The domain name KhufuNet.com has already been registered. Apache will host www.arrowflash.com and a WordPress instance; blog.arrowflash.com

Lab Topology

The lab topology is made up of two PCs running VMware in “Bridged Mode” and connected via a hub. The virtual machines are specified as follows:

VM1 – Ubuntu desktop (DHCP client)

VM2 – Apache Server/Name Server 1(Primary)/Print Server/NFS /SSH Server

VM3 – eMail Server/Name Server 2(Secondary)/DHCP Server /FTP Server/SSH Server

VM4 – Ubuntu desktop (DHCP client)

IP Address Table

This project was to be done in group between Tracey Nixon and Myself we divided the job with me having to handle the installations on VM1 and VM2. Tracey taking VM3 and VM4.

The decision was made to use 192.168.1.0/24 as our network address and we gave each machine there IP addresses as follows:

Device	IP Address
Network	192.168.1.0 /24
VM1	192.168.1.8
VM2	192.168.1.9
VM3	192.168.1.6
VM4	192.168.1.7

After this we did the basic connections and we were able to communicate over the network by issuing the “ping” command. With this done we are now ready to start installation and configuration of packages necessary for this project to work.

VM	Password
vm1	vm1
vm2	vm2
vm3	vm3
vm4	vm4

Setting Up Connection

Server vm2 ,Vm3

This guide is based on Ubuntu 14.04 server, so you should set up a basic 14.04 server installation before you continue with this guide. The system should have a static IP address. Repeat each time for the ip address table above.

You can skip this part if you wish and come back to it later.

- `sudo su`
- `nano /etc/network/interfaces`

Make the necessary changes to the files so it looks like the following and plus ctrl + x:

Vm2

```
# The loopback network interface auto lo
auto lo
iface lo inet loopback
# The primary network interface auto eth0
auto eth0
iface eth0 inet static
address 192.168.1.9
netmask 255.255.255.0
gateway 192.168.10.1
```

Vm3

```
# The loopback network interface auto lo
auto lo
iface lo inet loopback
# The primary network interface auto eth0
auto eth0
iface eth0 inet static
address 192.168.1.6
netmask 255.255.255.0
gateway 192.168.10.1
```

I didn't need to change the hosts settings.

DESKTOP SET UP

On the two desktop vm machines (Desktop) the same settings need to be configured so that the computer can talk to your servers. To set it up as a DHCP clients

First we need to edit the network interfaces settings. Using the following command below.

nano /etc/network/interfaces

Then edit the files below like so

```
# The loopback network interface auto lo
auto lo
iface lo inet loopback
# The primary network interface auto eth0
auto eth0
iface eth0 inet dhcp
```

VM2 – Apache Server (LINUX , APACHE , MYSQL AND PHP)

I will be using root credentials (which give me complete control over the server), if you are doing an installation with any other user just prefix sudo before all commands:

1. First Login into the server , using the username and password.
2. Next type in Sudo Su.
3. It will ask you for the password again.

Now we have root credentials!

```
apt-get -y install apache2 mysql-client mysql-server php5
```

You will be asked to provide a password for the MySQL root user this password is valid for the user root@localhost as well as root@server1.example.com, so we don't have to specify a MySQL root password manually later on:

- **New password for the MySQL "root" user:**
- **Repeat password for the MySQL "root" user:**

Wordpress require some more packages we will install them as follows:

```
apt-get -y install graphviz aspell php5-pspell php5-curl php5-gd php5-intl php5-mysql php5-xmlrpc php5-ldap git-core
```

Virtual instances

Create the Directory Structure

The first step which needs to be taken is to make a directory structure to hold the site data that will be served to visitors.

The document root (the top-level directory that Apache looks at to find content to serve) will be set to individual directories under the `/var/www` directory. A directory will be created here for both of the planned virtual hosts.

Within each of these directories, a `public_html` file will be created to hold the actual files. This allows some flexibility in the hosting.

For instance, for the sites in question, the directories will be made like this:

- **`sudo mkdir -p /var/www/khufunet.com/public_html`**
- **`sudo mkdir -p /var/www/blog.khufunet.com/public_html`**

Grant Permissions

The directory structure for the files is now in place, but they are owned by the root user. If the regular user is to be able to modify files in the web directories, the ownership can be changed by doing this:

- **`sudo chown -R root:root /var/www/khufunet.com/public_html`**
- **`sudo chown -R root:root /var/www/blog.khufunet.com/public_html`**

The permissions should be slightly modified to ensure that read access is permitted to the general web directory and all of the files and folders it contains so that pages can be served correctly:

- **`sudo chmod -R 755 /var/www`**

Your web server should now have the permissions it needs to serve content, and your user should be able to create content within the necessary folders.

Create Demo Pages for Each Virtual Host

The directory structure is now in place. Now some content to serve must be created. As this is just a demonstration, these pages will be very simple. First, Wordpress will be installed twice.

Create New Virtual Host Files

Virtual host files are the files that specify the actual configuration of the virtual hosts and dictate how the Apache web server will respond to various domain requests.

Apache comes with a default virtual host file called **000-default.conf** that can be used as a jumping off point. It must be copied over to create a virtual host file for each of the domains.

Firstly one domain must be configured, copied for the second domain, and then further adjusted as needed. The default Ubuntu configuration requires that each virtual host file end in `.conf`.

Create the First Virtual Host File

Start by copying the file for the first domain:

```
sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/khufunet.com.conf
```

Open the new file in your editor with root privileges:

```
sudo nano /etc/apache2/sites-available/khufunet.com.conf
```

The file will look something like this.

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

The items must now be customised for the first domain and some additional directives must be added. This virtual host section matches any requests that are made on port 80, the default HTTP port.

First, the `ServerAdmin` directive must be changed to an email address that the site administrator can receive emails through.

`ServerAdmin admin@khufunet.com`

After this, two directives must be added. The first, called `ServerName`, establishes the base domain that should match for this virtual host definition. This will most likely be your domain. The second, called `ServerAlias`, defines further names that should match as if they were the base name. This is useful for matching hosts you defined, like **`www:`**

```
ServerName khufunet.com  
ServerAlias www.khufunet.com
```

The only other thing that must be changed for a basic virtual host file is the location of the document root for this domain. The necessary directory has already been created, so all that needs to be done is to simply alter the `DocumentRoot` directive to reflect the that:

```
DocumentRoot /var/www/khufunet.com/public_html
```

At the end, the virtual host file should look like this

```
<VirtualHost *:80>  
  ServerAdmin admin@khufunet.com  
  ServerName khufunet.com  
  ServerAlias www.khufunet.com  
  DocumentRoot /var/www/khufunet.com/public_html  
  ErrorLog ${APACHE_LOG_DIR}/error.log  
  CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

Save and close the file.

Copy First Virtual Host and Customize for blog.khufunet.com

Now that the first virtual host file has been established, the second one can be created by copying that file and performing the necessary adjustments.

Start by copying it:

```
sudo cp /etc/apache2/sites-available/example.com.conf /etc/apache2/sites-available/  
blog.khufunet.com.com.conf
```

Open the new file with root privileges in your editor:

```
sudo nano /etc/apache2/sites-available/blog.khufunet.com.com.conf
```

You now need to modify all of the pieces of information to reference your second domain. When you are finished, it may look something like this:

```
<VirtualHost *:80>  
  ServerAdmin admin@test.com  
  ServerName blog.khufunet.com.com  
  ServerAlias www.blog.khufunet.com.com  
  DocumentRoot /var/www/blog.khufunet.com.com/public_html  
  ErrorLog ${APACHE_LOG_DIR}/error.log  
  CustomLog ${APACHE_LOG_DIR}/access.log combined  
</VirtualHost>
```

Save and close the file when you are finished.

Enable the New Virtual Host Files

Now that the virtual host files have been created, they must be enabled. Apache includes some tools that make this.

The `a2ensite` tool can be used to enable each of the sites like this:

```
sudo a2ensite blog.khufunet.com.com.conf
```

```
sudo a2ensite khufunet.com.com.conf
```

When you are finished, you need to restart Apache to make these changes take effect:

```
sudo service apache2 restart
```

You will most likely receive a message saying something similar to:

```
Restarting web server apache2 AH00558:  
apache2: Could not reliably determine the server's  
fully qualified domain name, using 127.0.0.1. Set  
the 'ServerName' directive globally to suppress  
this message
```

This is a harmless message that does not affect our site.

The creation of the virtual host is now complete.

Install Wordpress (VM2)

Once the virtual hosts have been installed, apache2 must once again be restarted. After this Mysql must be configured, a database called wordpress must be created as well as a user admin and password 'vm2'. Then once all this is entered, the user can view their phpmyadmin.

The wordpress database was created using the following commands

```
mysql -u root -p
```

Then it requested the password vm1

The following sql must be used to create the database

```
CREATE DATABASE wordpress;  
EXIT;
```

Next the Wordpress files were downloaded and installed:

- 1. Apt-get install wordpress**
- 2. Find / -name wordpress**
- 3. Ls /usr/share**
- 4. Mv /usr/share/wordpress /var/www//blog.khufunet.com/public_html**
- 5. Mv /usr/share/wordpress /var/www/khufunet.com/public_html**

This will allow you to view the wordpress site. The default config site was edited to allow the server to point to the wordpress folder.

```
nano /etc/apache2/sites-available/khufunet.com.conf  
nano /etc/apache2/sites-available/blog.khufunet.com.conf
```

and edited the DocumentRoot /var/www/html to /var/www/html/blog.khufunet.com
and edited the DocumentRoot /var/www/html to /var/www/html/khufunet.com

Next go to vm1 (the desktops) and install wordpress. If an error should occur such as a the config file being missing, switch back to vm2 and use the default one which is located in the wordpress folder and edit it.

```
nano /wp-config-sample.php
```

Then add the

```
database : wordpress  
Username : ROOT  
password : vm2
```

Leave the rest as it was and save it as **wp-config.php**.

Return to vm1 and finish the step by step wordpress install.

Print Server (VM2)

To install CUPS on your Ubuntu computer, simply use sudo with the apt-get command and give the packages to install as the first parameter. A complete CUPS install has many package dependencies, but they may all be specified on the same command line. Enter the following at a terminal prompt to install CUPS:

```
sudo apt-get install cups
```

Then configure the **/etc/cups/cupsd.conf** file and add the following.

The first allows for the server.

```
#Only Listen for connections from the local  
Machine .  
Listen localhost:631  
Listen /var/run/cups/cups.sock  
Listen 192.168.1.9:631
```

Edit the following lines so that it resembles the example below.

```
# Web interface setting...  
WebInterface Yes  
  
# Restrict access to the server...  
<Location />  
    Order allow,deny  
    Allow localhost  
Allow 192.168.1.*  
</Location>  
  
# Restrict access to the admin pages...  
<Location /admin>  
    Order allow,deny  
    Allow 192.168.1.*  
</Location>
```

If you go to the any of the dhcp clients you can access the admin page from typing the 192.168.1.9:631 in the web browser.

Email (Vm3)

To install Postfix you must type the command

```
sudo aptitude install postfix
```

This command will bring you back to configuration if you accidentally exit.

```
sudo dpkg-reconfigure postfix
```

In the configuration you will be prompted the following: General type of mail configuration: Internet Site System

mail name: ns2.khufunet.com\

Root and postmaster mail recipient: <robert@mycit.ie>

Other destinations for mail: ns2.khufunet.com, khufunet.com Force synchronous updates on mail queue?: No

```
Local networks: 192.168.1.0/24
```

<https://help.ubuntu.com/community/Postfix>

Due to time constraints, this section had to remain unfinished.

Name Servers (Master, Slaves - Vm2, Vm3)

What is Bind 9

“BIND or named is the most widely used Domain Name System (DNS) software on the Internet. On Unix-like operating systems it is the de facto standard.

The software was originally designed at the University of California Berkeley (UCB) in the early 1980s. The name originates as an acronym of Berkeley Internet Name Domain, reflecting the application's use within UCB. The software consists, most prominently, of the DNS server component, called named, contracted for name daemon. In addition the suite contains various administration tools, and a DNS resolver interface library. The latest version of BIND is BIND 9, first released in 2000.”

Name Servers (VM2 ,VM3)

Install bind9:

First to install it , run the following command.

```
apt-get install bind9
```

The default configuration is setup to act as a caching server. All that is required is simply adding the IP Addresses of your ISP's DNS servers.

Simply uncomment and edit the following in **/etc/bind/named.conf.options**:

```
forwarders {  
    192.168.1.9 ; #vm2 master  
    192.168.1.6; #vm3 slave  
};
```

Now restart the DNS server, to enable the new configuration.

From a terminal prompt:

```
sudo service bind9 restart
```

To add a DNS zone to BIND9, turning BIND9 into a Primary Master server, the first step is to edit **/etc/bind/named.conf.local**:

```
zone "khufunet.com" {  
    type master;  
    file "/etc/bind/db.khufunet.com";  
};
```

Now use an existing zone file as a template to create the **/etc/bind/db.khufunet.com** file:

```
sudo cp /etc/bind/db.local /etc/bind/db.khufunet.com
```

Create an A record for the base domain, khufunet.com. Also, create an A record for ns1.khufunet.com, the name server in this example:

```
;
; BIND data file for khufunet.com
;
$TTL 604800
@ IN SOA khufunet.com. admin.khufunet.com. (
        2      ; Serial
        604800 ; Refresh
        86400  ; Retry
        2419200 ; Expire
        604800 ) ; Negative Cache TTL
;
@ IN NS ns1.khufunet.com.
@ IN NS ns2.khufunet.com.
@ IN A 192.168.1.9
@ IN AAAA ::1
ns2 IN A 192.168.1.9
ns1 IN A 192.168.1.6
www IN CNAME @
blog IN A 192.168.1.9
mail IN A 192.168.1.6
```

You must increment the Serial Number every time you make changes to the zone file. If you make multiple changes before restarting BIND9, simply increment the Serial once.

Once you have made changes to the zone file BIND9 needs to be restarted for the changes to take effect:

```
sudo service bind9 restart
```

Reverse Zone File

Now that the zone is setup and resolving names to IP Addresses a Reverse zone is also required. A Reverse zone allows DNS to resolve an address to a name.

Edit **/etc/bind/named.conf.local** and add the following:

```
zone "1.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192";  
};
```

Now to create the db.127 to match the 192 which is the first octet of your ip address.

Now create the **/etc/bind/db.192** file:

```
sudo cp /etc/bind/db.127 /etc/bind/db.192
```

Next edit **/etc/bind/db.192** changing the basically the same options as **/etc/bind/db.khufunet.com**:

```
;  
; BIND reverse data file for local 192.168.1.9 net  
;  
$TTL 604800  
@ IN SOA ns1.khufunet.com.  
admin.khufunet.com. (  
    2      ; Serial  
    604800 ; Refresh  
    86400  ; Retry  
    2419200 ; Expire  
    604800 ) ; Negative Cache TTL  
;  
@ IN NS ns1.  
10 IN PTR ns1.khufunet.com.
```

For each A record you configure in **/etc/bind/db.example.com**, that is for a different address, you need to create a PTR record in **/etc/bind/db.192**.

After creating the reverse zone file restart BIND9:

```
sudo service bind9 restart
```

Configuring BIND on a Slave (Example)

Once a Primary Master has been configured a Secondary Master is needed in order to maintain the availability of the domain should the Primary become unavailable.

On the Primary Master server, the zone transfer needs to be allowed. Add the allow-transfer option to the example Forward and Reverse zone definitions in **/etc/bind/named.conf.local**:

```
zone "khufunet.com" {  
    type master;  
    file "/etc/bind/db.khufunet.com";  
    allow-transfer { 192.168.1.6; };  
};  
  
zone "1.168.192.in-addr.arpa" {  
    type master;  
    file "/etc/bind/db.192";  
    allow-transfer { 192.168.1.6; };  
};
```

Restart BIND9 on the Primary Master:

```
sudo service bind9 restart
```

Next, on the Secondary Master, install the bind9 package the same way as on the Primary. Then edit the **/etc/bind/named.conf.local** and add the following declarations for the Forward and Reverse zones:

```
zone "khufunet.com" {  
    type slave;  
    file "db.khufunet.com";  
    masters { 192.168.1.9; };  
};  
  
zone "1.168.192.in-addr.arpa" {  
    type slave;  
    file "db.192";  
    masters { 192.168.1.9; };  
};
```

Restart BIND9 on the Secondary Master:

```
sudo service bind9 restart
```

Network File System (NFS)

"NFS allows a system to share directories and files with others over a network. By using NFS, users and programs can access files on remote systems almost as if they were local files."

Setting Up the NFS Server (Master : Vm2)

An NFS mount is set up between at least two virtual servers. The machine hosting the shared network is called the server, while the ones that connect to it are called 'clients'.

The following IP addresses will refer to each one.

Master: 192.168.10.9 (Server)

Client: 192.168.10.8 (Client)

The system should be set up as root. You can access the root user by typing

sudo su-

Download the Required Software

Start off by using apt-get to install the nfs package.

```
apt-get install nfs-kernel-server portmap
```

Export the Shared Directory

The next step is to decide which directory we want to share with the client server. The chosen directory should then be added to the `/etc/exports` file, which specifies both the directory to be shared and the details of how it is shared.

Suppose we wanted to share `/home`

We have to export the directories to the other VPS:

`nano /etc/exports`

Add the following lines to the bottom of the file, sharing both directories with the client:

```
/home 192.168.10.8(rw,sync,no_root_squash,no_subtree_check)
```

These settings

accomplish several tasks:

- `rw`: This option allows the client server to both read and write within the shared directory
- `sync`: Sync confirms requests to the shared directory only once the changes have been committed.
- `no_subtree_check`: This option prevents the subtree checking. When a shared directory is the subdirectory of a larger filesystem, nfs performs scans of every directory above it, in order to verify its permissions and details. Disabling the subtree check may increase the reliability of NFS, but reduce security.
- `no_root_squash`: This phrase allows root to connect to the designated directory"

Once you have entered in the settings for each directory, run the following command to export them:

`exportfs -a`

Setting Up the NFS Client (Slave : Vm2)

Download the Required Software

Start off by using apt-get to install the nfs programs.

apt-get install nfs-common portmap

Mount the Directories

Once the programs have been downloaded to the the client server, create the directories that will contain the NFS shared files

```
mkdir -p /mnt/nfs/home
```

Then go ahead and mount them

```
mount 192.168.10.9:/home /mnt/nfs/home
```

You can use the df -h command to check that the directories have been mounted. You will see them last on the list.

df -h

In the video, it is apparent that a file was created and the server was checked to make sure the file was there.

Testing the NFS Mount

Once you have successfully mounted your NFS directories, you can test that they work by creating files on the Client and checking their availability on the Server.

Create a file in each directory to try it out:

```
touch /mnt/nfs/home/example /mnt/nfs/home/example
```

You should then be able to find the files on the Server in the /home and /var/nfs directories.

ls /home

You can ensure that the mount is always active by adding the directories to the fstab file on the client. This will ensure that the mounts start up after the server reboots.

nano /etc/fstab

```
192.168.10.9:/home /mnt/nfs/home nfs  
auto,noatime,nolock,bg,nfsvers=3,intr,tcp,actimeo=1800 0 0
```

You can check the mounted directories with the two earlier commands:

df -h

mount

SSH Server (vm2, vm3)

“Secure Shell (SSH) is a cryptographic network protocol for secure data communication, remote command-line login, remote command execution, and other secure network services between two networked computers.”

To install ssh on both vm2 and vm3 . I did this both machines .

On the remote machine (ssh server) you do:

```
# apt-get install ssh
```

This installs ssh, sets it to start on boot up, and starts it up right then
Then you do, on the client machine:

```
$ ssh-keygen
```

Answer, the questions when prompted. Then do

```
$ ssh-copy-id <user>@<ssh server>
```

Where <username> is the user you want to authenticate as with your key, and <ssh server> is the ip address/ host/dns of the SSH server you want to add your key too (**192.168.1.9 and 192.168.1.6**). Then test it!

```
$ ssh <user>@<ssh server>
```

Now that you're in you want to get root perms...

```
$ sudo su
```

And, now that you've got root perms, edit /etc/ssh/sshd_config

```
# nano /etc/ssh/sshd_config
```

That's it for setting up SSH on an Ubuntu machine. Now only you can get into it with your key.

FTP (Vm3)

“File Transfer Protocol (FTP) is a TCP protocol for uploading and downloading files between computers. FTP works on a client/server model. The server component is called an FTP daemon. It continuously listens for FTP requests from remote clients. When a request is received, it manages the login and sets up the connection. For the duration of the session it executes any of commands sent by the FTP client.”

Step One—Install vsftpd

You can quickly install vsftpd on your virtual private server in the command line: the ftp is installed under the name vsftpd.

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

Once the file finishes downloading, VSFTP is already configured with a reasonable amount of security. However, it does provide access on your VPS to anonymous users.

Step Two—Configure vsftpd

Once vsftpd is installed, you can adjust the configuration by uncomment the sign #. Open up the configuration file:

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

The biggest change you need to make is to switch the Anonymous_enable from YES to NO:

```
anonymous enable=NO
```

Prior to this change, vsftpd allowed anonymous, unidentified users to access the server's files. This is useful if you are seeking to distribute information widely, but may be considered a serious security issue in most other cases.

After that, uncomment the local_enable option, changing it to yes and, additionally, allow the user to write to the directory.

```
local_enable=YES  
write_enable=YES
```

Finish up by uncommenting command to `chroot_local_user`. When this line is set to Yes, all the local users will be jailed within their chroot and will be denied access to any other part of the server.

```
chroot_local_user=YES
```

Save and Exit that file.

Because of a recent vsftpd upgrade, vsftpd is "refusing to run with writable root inside chroot". A handy way to address this issue is to take the following steps:

- 1. Create a new directory within the user's home directory**
- 2. `mkdir /home/files`**
- 3. Change the ownership of that file to root**
- 4. `chown root:root /home/files`**
- 5. Make all necessary changes within the "files" subdirectory**

Then, as always, restart:

```
sudo service vsftpd restart
```

Securing FTP

There are options in `/etc/vsftpd.conf` to help make vsftpd more secure. For example users can be limited to their home directories by uncommenting:

```
chroot_local_user=YES
```

You can also limit a specific list of users to just their home directories:

```
chroot_list_enable=YES  
chroot_list_file=/etc/vsftpd.chroot_list
```

After uncommenting the above options, create a `/etc/vsftpd.chroot_list` containing a list of users one per line. Then restart vsftpd:

```
#sudo restart vsftpd
```

FTP can also be encrypted using FTPS. Different from SFTP, FTPS is FTP over Secure Socket Layer (SSL). SFTP is a FTP like session over an encrypted SSH connection. A major difference is that users of SFTP need to have a shell account on the system, instead of a nologin shell. Providing all users with a shell may not be ideal for some environments, such as a shared web host.

```
#sudo restart vsftpd
```

Dynamic Host Configuration Protocol (DHCP)

“The Dynamic Host Configuration Protocol (DHCP) is a network service that enables host computers to be automatically assigned settings from a server as opposed to manually configuring each network host. Computers configured to be DHCP clients have no control over the settings they receive from the DHCP server, and the configuration is transparent to the computer's user.”

Installation

At a terminal prompt, enter the following command to install dhcpd:

```
sudo apt-get install isc-dhcp-server
```

You will probably need to change the default configuration by editing `/etc/dhcp3/dhcpd.conf` to suit your needs and particular configuration.

You also need to edit `/etc/default/isc-dhcp-server` to specify the interfaces dhcpd should listen to. By default it listens to `eth0`. and need to change it `eth1`

Configuration

The error message the installation ends with might be a little confusing, but the following steps will help you configure the service:

Most commonly, what you want to do is assign an IP address randomly. This can be done with settings as follows:

```
nano -w /etc/dhcp/dhcpd.conf
# Sample /etc/dhcpd.conf
# (add your comments here)
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.9, 192.168.1.6;
option domain-name "mydomain.example";

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.100;
range 192.168.1.150 192.168.1.200;
}
```

Click save and you're done.

SUMMARY & CONCLUSION

The installation of Ubuntu server and the desktop may look difficult, however it is relatively uncomplicated if you follow our guide or one of the guides which are available online. I have written a simple step by step guide to installing Ubuntu and some of its packages, with definitions for some of the commands so that the basic user can understand in easily understand the various stages and learn during the process.

For security fixes for a basic user, I added more secure ftp connections. I also gave permissions to different logins.

I would also use more snap shots for my VMware as when I was tweaking and changing the server settings , along with the ftp server causing my dhcp server not to work. This was fixed though restarting the service twice.

The ftp was easy no completions when you follow the guide. though the config file was completed as it had a lot of information and data you can change to make it more secure.

The name server for the slave was very easy as after the installing you write one block of code for the file. it was longer on the master as there was more information needed. OAnother problem we had was that when we installed the ftp package it was crashing the dans dhcp server. The way we fixed it was by restarting the vm and it worked perfectly.

The only problem we could not solve was that the email did not work at the end of the project . there was an error the email would not context.

Reference

Nfs Server

<https://help.ubuntu.com/14.04/serverguide/network-file-system.html>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-12-04>

Ssh Server

<https://help.ubuntu.com/community/SSH/OpenSSH/Configuring>

<http://ubuntuhandbook.org/index.php/2014/04/enable-ssh-ubuntu-14-04-trusty-tahr/>

<http://superuser.com/questions/274934/setting-up-ssh-server-on-ubuntu-dektop>

FTP Server

<https://www.digitalocean.com/community/tutorials/how-to-set-up-an-nfs-mount-on-ubuntu-12-04>

<https://www.digitalocean.com/community/tutorials/how-to-set-up-vsftpd-on-ubuntu-12-04>