Systems Integration DT211/4

CA02: Server Configuration

DDNS: DHCP: NFS: FTP: NTP
Server: 192.168.1.11 **Client: 192.168.1.12**

Dean Ryan; C11526797; 17/12/201

Note: To view errors throughout the assignment, use sudo /var/log/syslog
(You can grep | the errors you are looking for if any)

DDNS - https://wiki.debian.org/DDNS

Dynamic DNS (DDNS) is a method of automatically updating a name server in the Domain Name System (DNS), often in real time, with the active DNS configuration of its configured hostnames, addresses or other information.

1) **Install DNS Package:**
   (server)

```
sudo apt-get install bind9
```

2) **Change the server nameservers (Optional):**
   (server)

```
sudo nano /etc/bind/named.conf.options

Forwarders {

        8.8.8.8;

        8.8.4.4;

};
```

3) **Create key to secure the exchange of information between DHCP and DNS server. We do this to allow only our DHCP server perform DNS record updates.**
   (server)

```
dnssec-keygen –a HMAC-MD5 –b 128 –r /dev/urandom –n USER DDNS_UPDATE
```

4) **Two files are now created: Kddns_updater.*.key and Kddns_updater.*.private. Read the Kddns_updater.*.private.**
   (server)

```
cat Kddns_updater.*.private
```

5) **Copy everything after "Key: " including "==" from the .private file.**
   (server)

```
nano ddns.key

key DDNS_UPDATE {

    algorithm HMAC-MD5.SIGALG.REG.INT;
    secret "<key>";
};
```

6) Copy this file to /etc/bind/ and /etc/dhcp and adjust the file permissions as follows: (server)

```
sudo cp ddns.key /etc/bind/

sudo cp ddns.key /etc/dhcp/

sudo chown root:bind /etc/bind/ddns.key

sudo chown root:root /etc/dhcp/ddns.key

sudo chmod 777 /etc/bind/ddns.key

sudo chmod 777 /etc/dhcp/ddns.key
```

**7) Define two zones; one for the forward lookup zone and one for the reverse lookup by adding following to the file /etc/bind/named.conf.local :
(server)**

```
include "/etc/bind/ddns.key";

zone "example.lan" {

        type master;
        file "/etc/bind/db.example.lan";
        allowtransfer { 192.168.1.11; };
        alsonotify { 192.168.1.11; };
        allowupdate { key DDNS_UPDATE; };
};

zone "1.168.192.inaddr.arpa" {

        type master;
        file "/etc/bind/db.192.168.1";
        allowtransfer { 192.168.1.11; };
        alsonotify { 192.168.1.11; };
        allowupdate { key DDNS_UPDATE; };

};
```

8) **Create the two zones declared in the previous step. You can create these from sample file db.empty :**
(server)

```
cd /etc/bind/
cp db.empty db.example.lan
cp db.empty db.192.168.1
```

9) **Edit both the etc/bind/db.example.lan + etc/bind/db.192.168.1 to resemble the following:**
(server)

db.example.lan

```
$TTL 604800
@       IN      SOA     server.example.lan. root.example.lan.  (
                                3                ; Serial

                          604800                 ; Refresh

                           86400                 ; Retry

                        2419200                  ; Expire

                          604800  )               ; Negative Cache TTL
;
@       IN      NS      server.example.lan.
server  IN      A       192.168.1.11
client  IN      A       192.168.1.12
```

db.192.168.1

```
$TTL 604800
@       IN      SOA     server.example.lan. root.example.lan. (
                                3                ; Serial
                          604800                 ; Refresh
                           86400                 ; Retry
                         2419200                 ; Expire
                          604800 )               ; Negative Cache TTL
;
@       IN      NS      server.
@       IN      A       192.168.1.11
11      IN      PTR     server.example.lan
```

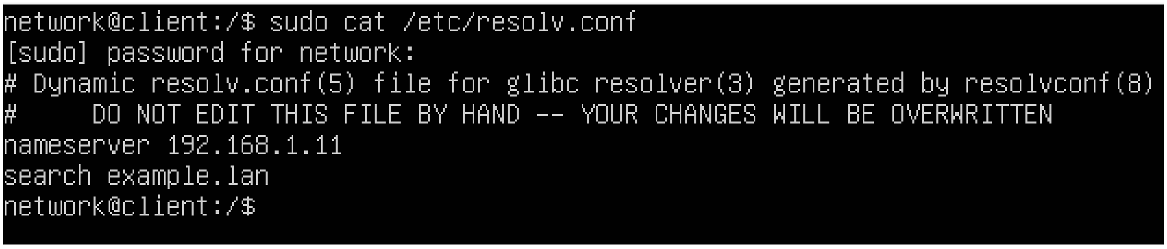
10) **Create symbolic links. This is done due to write permissions on the /etc/bind folder**
(server)

cd /var/cache/bind/
sudo ln -s /etc/bind/db.example.lan .
sudo ln -s /etc/bind/db.192.168.1 .

Confirm your progress

1) **On the client check that the DNS server used is the server**
(client)

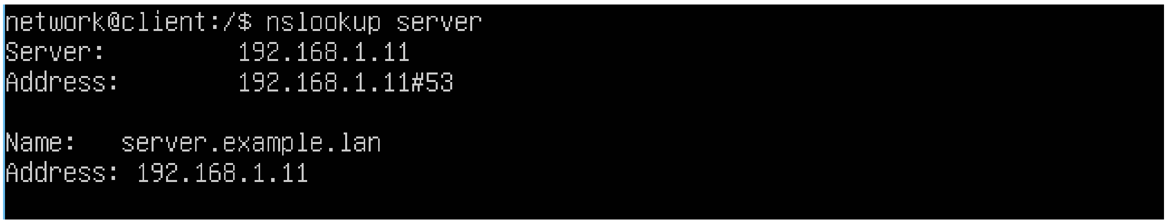
sudo cat /etc/resolv.conf



```
network@client:/$ sudo cat /etc/resolv.conf
[sudo] password for network:
# Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 192.168.1.11
search example.lan
network@client:/$
```

2) **Nslookup of server from client. Nslookup is a tool available for querying the Domain Name System (DNS) to obtain domain name or IP addresss.**
(client)

nslookup server

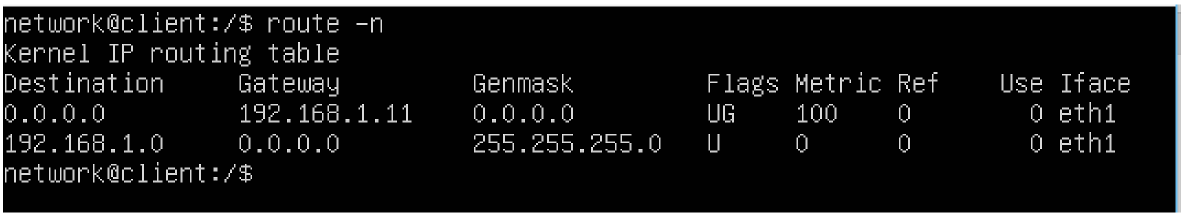


```
network@client:/$ nslookup server
Server:         192.168.1.11
Address:        192.168.1.11#53

Name:   server.example.lan
Address: 192.168.1.11
```

3) **route –n shows the routing table. The –n shows the numerical address instead of a named gateway.**
(client)

route –n



```
network@client:/$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.11    0.0.0.0         UG    100    0        0 eth1
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
network@client:/$
```

DHCP - https://help.ubuntu.com/community/isc-dhcp-server

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

1) On the server install the dhcpd Server (server)

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

2) Edit the file /etc/dhcp/dhcpd.conf to resemble the following: (server)

```
authoritative;
option domain-name "example.lan";
option domain-name-servers 192.168.1.11;

ddns-updates on;
ddns-update-style interim;
ignore client-updates;
update-static-leases on;

default-lease-time 600;
max-lease-time 7200;
log-facility local7;

include "/etc/dhcp/ddns.key";

zone EXAMPLE.LAN. {
        primary 127.0.0.1;
        key DDNS_UPDATE;
}

zone 1.168.192.inaddr.arpa. {
        primary 127.0.0.1;
        key DDNS_UPDATE;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
        range 192.168.1.150 192.168.1.200;
        option routers 192.168.1.11;
}
```

Procedures

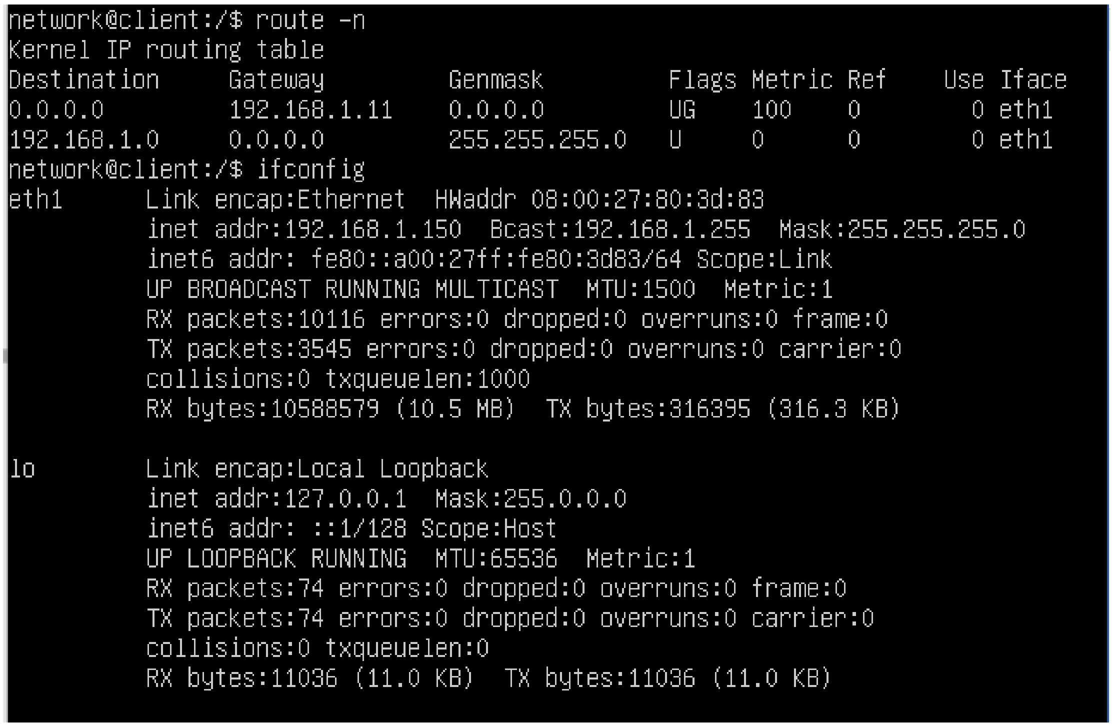
Restart the servers: (server)

```
sudo /etc/init.d/iscdhcpserver restart
sudo /etc/init.d/iscdhcpserver start
sudo/etc/init.d/iscdhcpserver stop
sudo /etc/init.d/bind9 restart
```

Confirm your progress

1) Check if the IP of the client is within the range provided by the DHCP server: (client)

Ifconfig



```
network@client:/$ route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.1.11    0.0.0.0         UG    100    0        0 eth1
192.168.1.0     0.0.0.0         255.255.255.0   U     0      0        0 eth1
network@client:/$ ifconfig
eth1      Link encap:Ethernet  HWaddr 08:00:27:80:3d:83
          inet addr:192.168.1.150  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe80:3d83/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:10116 errors:0 dropped:0 overruns:0 frame:0
          TX packets:3545 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:10588579 (10.5 MB)  TX bytes:316395 (316.3 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:74 errors:0 dropped:0 overruns:0 frame:0
          TX packets:74 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:11036 (11.0 KB)  TX bytes:11036 (11.0 KB)
```

Routing (pinging)

1) **On server, enable packet forwarding for IPv4. To do this edit the file /etc/sysctl.conf and uncomment line.**
   (server)

   sudo nano /etc/sysctl.conf

   #Uncomment the next line to enable packet forawarding for IPv4
   Net.ipv4.ip_forward=1

2) **On server, edit the file /etc/rc.local**

(server)

sudo nano /etc/rc.local

sudo /sbin/iptables –P FORWARD ACCEPT

sudo /sbin/iptables –tables nat –A POSTROUTING –o eth0 –j MASQUERADE

exit 0

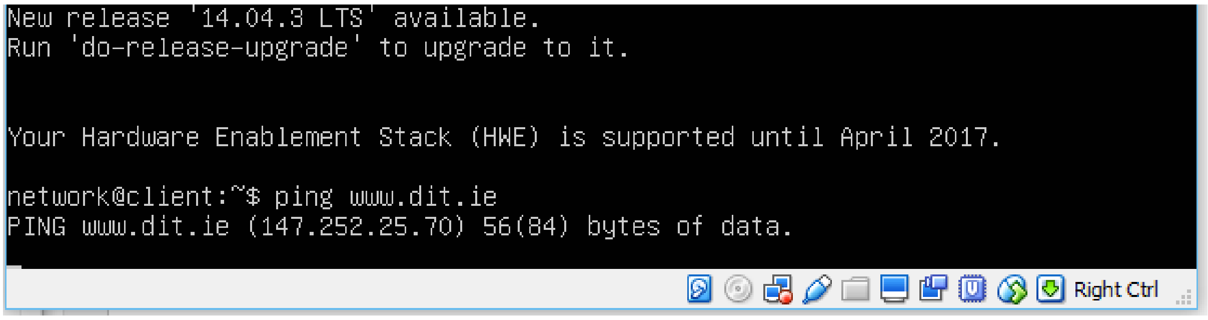
<u>Confirm your progress</u>

1) **Ping is a service which sends packets to a host and checks whether it is reachable across an IP network.**
   (client)

   Ping www.dit.ie

Or

Sudo apt-get update – 'checks if client is connected to internet through the server'



```
New release '14.04.3 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Your Hardware Enablement Stack (HWE) is supported until April 2017.

network@client:~$ ping www.dit.ie
PING www.dit.ie (147.252.25.70) 56(84) bytes of data.
```

Mounting Shared Folder Windows to Virtual machine

1) **Create folder in the virtual machine that will be used for sharing files**
   (server)

   sudo mkdir shared

2) **Run : sudo mount –t vboxsf [Name of Windows Folder] [Path of Linux Folder]**
   (server)

   sudo mount –t vboxsf SHARED_FOLDER
   /home/network/shared

Note: ***IF YOU NEED TO UNDO A MOUNT:*** **run the following:**

```
sudo umount /home/network/shared
```

NFS - https://help.ubuntu.com/12.04/serverguide/network-file-system.html

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.

1) **On the server install the NFS Server**

```
sudo apt-get install nfs-kernal-server
```

2) **Create shared folder on both server and client**

```
sudo mkdir /home/myshare
```

3) **Add shared folder on the server to /etc/exports file**

```
sudo nano /etc/exports

/home/myshare *(rw,sync,no_subtree_check)
```

4) **Start the NFS server**

```
sudo /etc/init.d/nfs-kernel-server start
```

5) **On the client install NFS**

```
sudo apt-get install nfs-common
```

6) **Connect the shared folders. Edit the /etc/fstab file to make a connection between the shared folders each time the client starts.**
(client)

```
sudo nano /etc/fstab

#add the following

node1.example.lan:/home/myshare /home/myshare nfs

rsize=8192,wsize=8192,timeo=14,intr
```

NTP – https://help.ubuntu.com/12.04/serverguide/NTP.html

NTP is a TCP/IP protocol for synchronising time over a network. Basically a client requests the current time from a server, and uses it to set its own clock.

1) **On both the server and host install the ntp**
(server)

sudo apt-get install ntp

2) **Synchronise date and time with the server**
(client)

sudo ntpdate server

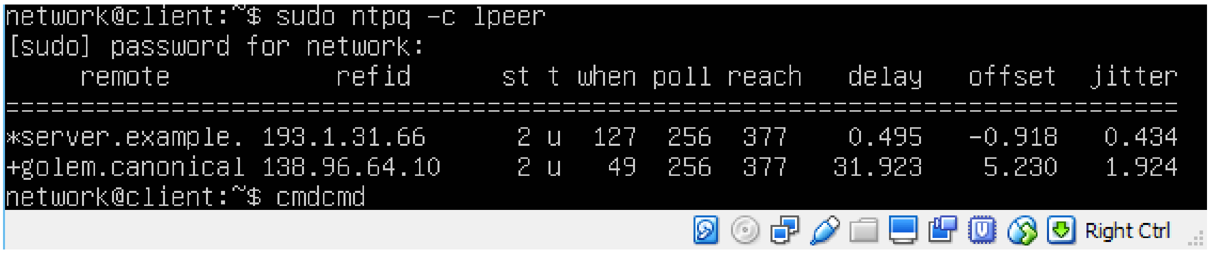
3) **Change the default ntp server on the client to be the server ntp**
(client)

sudo nano /etc/ntp.conf
#Comment out default servers
# eg. server 0.ubuntu.pool.ntp.org
# server 1.ubuntu…
add "server node1.example.lan"

Confirm your progress

1) **Check if server on the client is the node1**

sudo ntpq –c lpeer



```
network@client:~$ sudo ntpq -c lpeer
[sudo] password for network:
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
*server.example. 193.1.31.66      2 u  127  256  377    0.495   -0.918   0.434
+golem.canonical 138.96.64.10     2 u   49  256  377   31.923    5.230   1.924
network@client:~$ cmdcmd
```

2) **Synchronise date and time with the server**

sudo /etc/init.d/ntp stop
sudo ntpdate server
sudo /etc/init.d/ntp start

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

The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

1) **Install vsftpd on the server**

   sudo apt-get install vsftpd

2) **Once vsftpd is installed, you can adjust the configuration. Open the config file.**

   sudo nano /etc/vsftpd.conf

3) **You now need to make a few switches with file:**

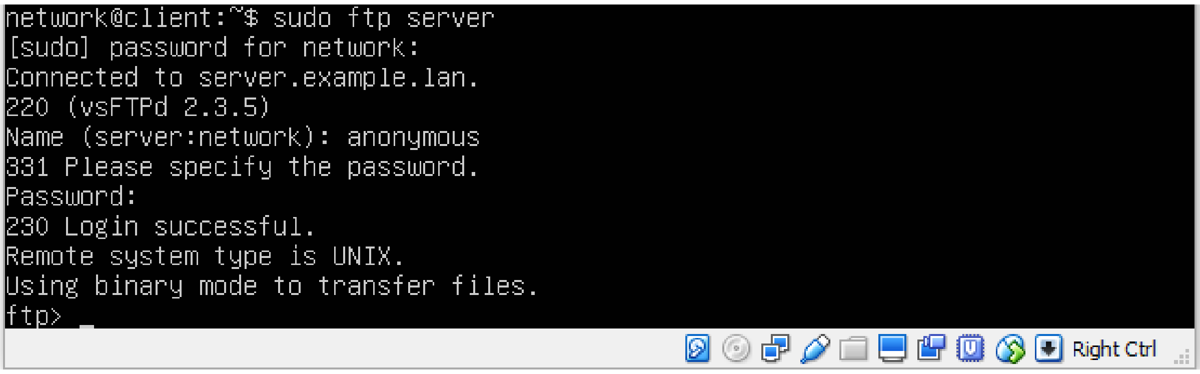
   anonymous_enable=NO
   local_enable=YES
   write_enable=YES

4) **We now need to uncomment the 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

5) **Lastly, navigate to the client and connect to ftp using**

   sudo ftp server



```
network@client:~$ sudo ftp server
[sudo] password for network:
Connected to server.example.lan.
220 (vsFTPd 2.3.5)
Name (server:network): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> _
```