

Student Name: Benjamin Lee
Student ID: 13248113
Tutorial: 1-04 9:00-11:00 Friday
Tutor Name: Yingying Yang
Due Date: 3/11/21

31338 Network Servers

Learning Journal Part 2

Benjamin Lee 13248113



Table of Contents

General Description	2
Intent	2
Week 6	2
Lecture	2
Lab 6a	2
Lab 6b	7
Week 7	9
Lecture	9
Lab 7a	9
Lab 7b	10
Lab 7c	11
Week 8	11
Lecture	11
Lab 8a	11
Lab 8b	14
Week 9	17
Lecture	17
Lab 9a	17
Lab 9b	18
Lab 9c	19
Lab 9d	20
Week 10	22
Lecture	22
Lab 10a	22
Lab 10b	24
Reflection	25
References	25

General Description

In this learning journal, my aim is to note-take my journey to develop a solid understand of networking and improve my skills to manage network servers in both CentOS/Linux and Windows. The purpose of this journal is to keep a log/record of all my system administration tasks, efforts and methods when learning this subject. This will be a continuation from my part 1 learning journal. This journal will try to limit the images since I'm running out of space in my hard drive.

Intent

My intention in this subject is to learn the new methods of administrating these operating systems and update my configuration skills. I had some issues with VMware Fusion on my MAC due to some networking issues so I had to use my old PC for the VM lab work instead. Using the learning materials were very rewarding and I learnt a lot from doing the labs and figuring out what to do by myself. Supplementary materials of the lecture slides were also helpful while I was progressing.

Week 6

Week 6 Topic: DNS

Lecture

The lecture discusses what DNS is, how DNS slaves and caching works, what BIND is when implementing with DNS servers and configurations as well as client configuration for DNS

Lab 6a

Task 1: Design DNS configuration

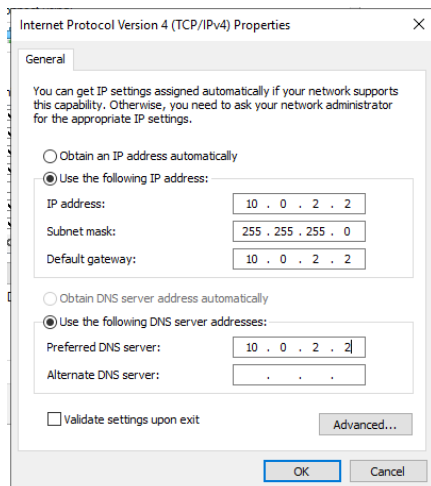
In this task, we will be setting up 2 domain name servers in our network. We will be configuring our DNS configurations via the table below:

DNS Name	IP Address	Type	Server	Comments
netserve.edu.au	10.0.2.2	NS, MX	Windows	Pointer to name & mail server
ns.netserve.edu.au	10.0.2.2	A	Windows	Actual Primary DNS
mail.netserve.edu.au	10.0.2.2	A	Windows	Actual Mail server
site.netserve.edu.au	10.0.2.2	A	Windows	Actual Web server
www.netserve.edu.au	10.0.2.2	CNAME	Windows	Alias to site.netserve.edu.au
it.netserve.edu.au	10.0.2.3	NS, MX	Linux	Point to name server& mail server
ns.it.netserve.edu.au	10.0.2.3	A	Linux	Actual Primary DNS
site.it.netserve.edu.au	10.0.2.3	A	Linux	Actual ftp, web server
www.it.netserve.edu.au	10.0.2.3	CNAME	Linux	Alias to site.it.netserve.edu.au
ftp.it.netserve.edu.au	10.0.2.3	CNAME	Linux	Alias to site.it.netserve.edu.au
mail.it.netserve.edu.au	10.0.2.3	A	Linux	Actual Mail server

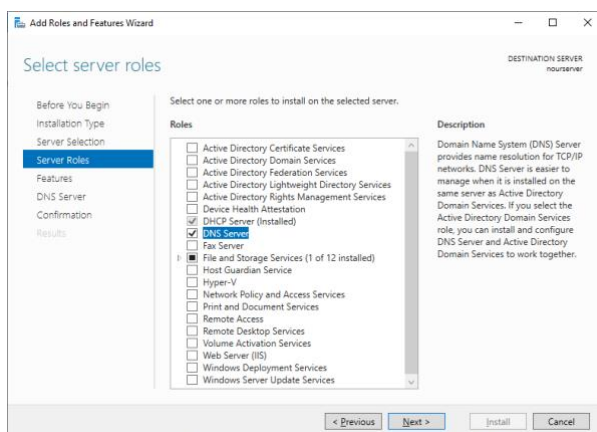
The Default DNS server configuration: 192.168.192.2

Task 2: Set up a DNS Zone (Forward Lookup)

We set static IP of Ethernet 1 to a static address of 10.0.2.2

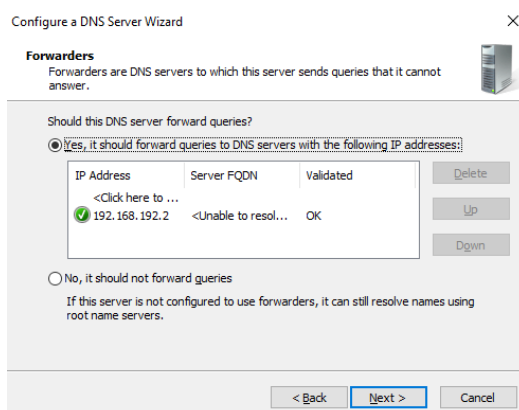


We then add roles and features by installing DNS server role from manager by Manage → Add roles and features → DNS Server

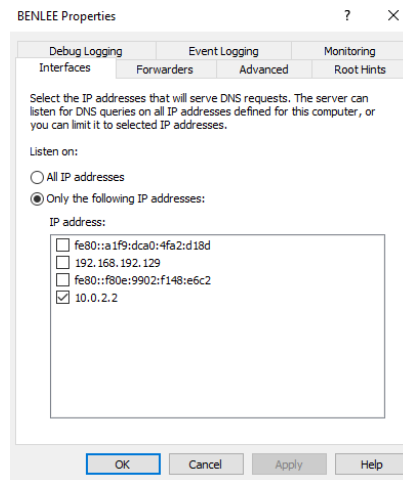


In DNS Manager we right-click hostname and select **Configure DNS Server** and enter the following configuration:

1. Create a forward lookup zone
2. Zone name: *netserv.edu.au*
3. Create a new file with file name: *netserv.edu.au*
4. No dynamic updates
5. Forward queries to DNS server with this IP: *192.168.192.2*
6. *Finish*

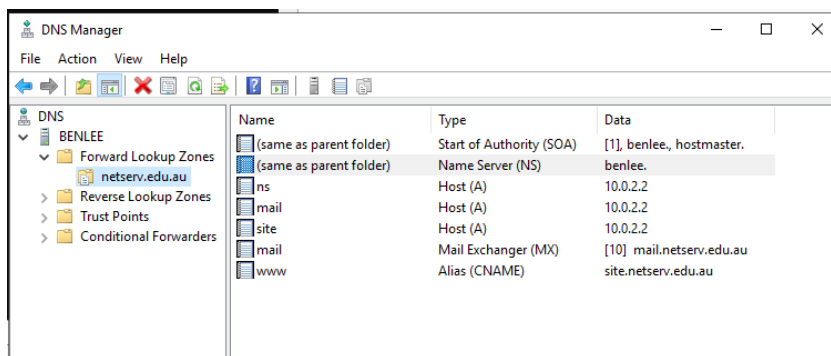


We then right-click on the hostname and select **Properties** and choose “Only the following IP addresses”.

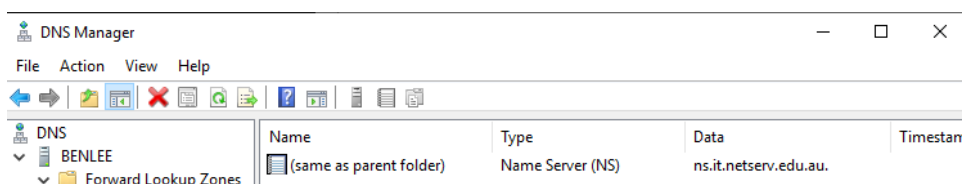


We then go to the Zone Transfers tab and choose “Allow zone transfers to any server”.

We will now create three A records, one MX record and one CNAME record, to do this right-click on **netserv.edu.au** → **(DNS record type)**. We should get the following configuration below:



We now create a delegation for the **it.netserv.edu.au** domain



The DNS file in the file system (C:\Windows\System32\dns\netserv.edu.au.dns) needs to be updated right-click on **(Server Name)** → **Update Server Data Files**. After, we restart the DNS server services in **Server Manager** → **Tools** → **Services** → **DNS Server** → **Restart**

We then test via the cmd line to see if changes were made via nslookup:

```
Default Server: [10.0.2.2]
Address: 10.0.2.2

> set domain=netserver.edu.au
> set type=A
> site.netserver.edu.au
Server: [10.0.2.2]
Address: 10.0.2.2

Name: site.netserver.edu.au
Address: 10.0.2.2

> www
Server: [10.0.2.2]
Address: 10.0.2.2

Name: site.netserver.edu.au
Address: 10.0.2.2
Aliases: www.netserver.edu.au

> www.uts.edu.au
Server: [10.0.2.2]
Address: 10.0.2.2

Non-authoritative answer:
Name: uts.edu.au
Address: 54.79.20.73
Aliases: www.uts.edu.au
```

The **ls.netserver.edu.au** command shows the following:

```
> ls netserver.edu.au
[[10.0.2.2]]
netserver.edu.au.      NS      server = benlee
it                     NS      server = ns.it.netserver.edu.au
ns.it                 A       10.0.2.3
mail                  A       10.0.2.2
ns                    A       10.0.2.2
site                  A       10.0.2.2
```

6c: Configuring the DNS server (BIND) on Linux

In this task, we set up ens37 with a static ip address.

```
TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens37
DEVICE=ens37
ONBOOT=yes
IPADDR=10.0.2.3
PREFIX=24
DEFROUTE=YES
#NETMASK=255.255.255.0
~
~
~
~
<fig/network-scripts/ifcfg-ens37" 17L, 287C written
```

We then use the following commands to check if configuration is right:

```
[root@benjamin ~]# nmcli con reload ens37
[root@benjamin ~]# nmcli con up ens37
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection/5)
[root@benjamin ~]# ifconfig ens37
ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.3 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::dbc6:3151:621:ed84 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:e6:9e:dc txqueuelen 1000 (Ethernet)
    RX packets 11 bytes 1402 (1.3 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 73 bytes 10424 (10.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

We then install BIND through the following command: **dnf install bind**

After, we update the name server configuration file **/etc/named.conf** and add the following:

```
options {
  listen-on port 53 { any; };
  listen-on-v6 port 53 { ::1; };
  directory      "/var/named";
  dump-file       "/var/named/data/cache_dump.db";
  statistics-file "/var/named/data/named_stats.txt";
  memstatistics-file "/var/named/data/named_mem_stats.txt";
  secroots-file  "/var/named/data/named.secroots";
  recursing-file  "/var/named/data/named.recursing";
  allow-query     { any; };
}
```

```
zone "it.netserv.edu.au" IN {
    type master;
    file "it.netserv.edu.au.zone";
};
```

We then create zone file **it.netserv.edu.au.zone** and edit the following

```
$TTL 1D
@      IN SOA    ns.it.netserv.edu.au. root.it.netserv.edu.au. (
                                            0       ; serial
                                           1D      ; refresh
                                           1H      ; retry
                                           1W      ; expire
                                           3H )     ; minimum

      IN NS     ns.it.netserv.edu.au.
      IN MX     10 mail
      IN A      10.0.2.3
site   IN A      10.0.2.3
www    IN CNAME site
ftp    IN CNAME site
mail   IN A      10.0.2.3
NS     IN A      10.0.2.3
```

"it.netserv.edu.au.zone" 15L, 302C written

After this since, we need to change group ownership of the zone file. We can do this via the **chgrp named it.netserv.edu.au.zone** command and also use **ls -l** to check ownership and file permissions after

We check our configuration with the following commands and start our named service:

1. `named-checkconf /etc/named.conf`
2. `named-checkzone it.netserv.edu.au.zone /var/named/it.netserv.edu.au.zone`
 - a. This should show 'OK' sign
3. `systemctl start named`
4. `systemctl enabled named`

After, we can test the server via the 'dig commands'

- Status should have 'no error' and 'answer: 1'

We now add forwarding record by updating the `/etc/named.conf` file and then restart service via the `systemctl restart named` command. Testing with the `dig` command give us an answer of 2 and status of no error. In firewall-config also enable dns.

```

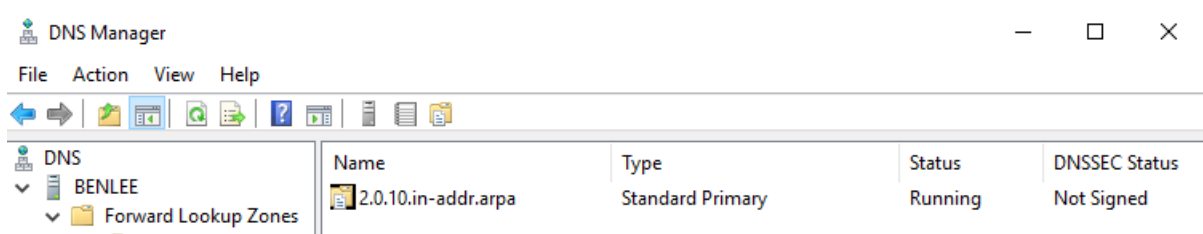
        forwarders{
            10.0.2.2;
        };

        /*
        - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
        - If you are building a RECURSIVE (caching) DNS server, you need to enable
        recursion.
        - If your recursive DNS server has a public IP address, you MUST enable access
        control to limit queries to your legitimate users. Failing to do so will
        cause your server to become part of large scale DNS amplification
        attacks. Implementing BCP38 within your network would greatly
        reduce such attack surface
        */
        recursion yes;

        dnssec-enable no;
        dnssec-validation no;
    }
}

```

We now go back to windows and go to the DNS Manager to create a new Reverse Lookup Zone.



We edit the name server of the reverse lookup to be 'ns' and 10.0.2.2. After we create a new ptr record with host IP address of 10.0.2.2 and Host name: site.netserv.edu.au. Using `nslookup` we can see our configuration is correct.

Lab 6b

Task 1: Examine and understand existing resolv.conf

Disabling the ens33 and ens37 interface and restarting them gives us some entries such as **search uts.edu.au** for ens33 and **search localdomain nameserver 192.168.192.2** for ens37.

We now edit the config files for **ens33** and **ens37** in the `/etc/sysconfig/network-scripts/ifcfg-ens33` and `/etc/sysconfig/network-scripts/ifcfg-ens37` files and restart networkcards via `systemctl`.

```

TYPE=Ethernet
PROXY_METHOD=none
BROWSER_ONLY=no
BOOTPROTO=none
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_FAILURE_FATAL=no
IPV6_ADDR_GEN_MODE=stable-privacy
NAME=ens37
DEVICE=ens37
ONBOOT=yes
IPADDR=10.0.2.3
PREFIX=24
DEFROUTE=YES
DNS1=10.0.2.3
DOMAIN=it.netserv.edu.au

```

This example is from ens37 configuration, the same is done for ens33 but with DNS1=10.0.2.2 and DOMAIN=netserv.edu.au

```

[root@benjamin named]# systemctl restart NetworkManager
[root@benjamin named]# cat /etc/resolv.conf
# Generated by NetworkManager
search localdomain it.netserv.edu.au
nameserver 192.168.192.2
nameserver 10.0.2.3

```


Third nameserver is coming from DHCP but I am not sure which DHCP server.

Running the command: **nmcli conn modify ens33 ipv4.ignore-auto-dns true**, modifies a connection, and in the config file of ifcfg-ens33 has added the line **peerDNS=no**. Restarting the NetworkManager and looking at the **/etc/resolv.conf** file again has the two nameserver entries 10.0.2.2 and 10.02.3 but has removed the third nameserver 192.168.192.2

Running the command: **nmcli conn modify ens37 ipv4.dns-priority 5** will add a new line in the ifcfg-ens37 config file called **IPV4_DNS_PRIORITY=5**

Testing with the **dig** command: **dig www.it.netserve.edu.au** should give us an ANSWER of 2 and status: NOERROR. Pinging www.it.netserve.edu.au has no issues.

Task 2: DNS client setup on windows

We can test if we have the right configurations via **nslookup** and **ping**.

```
Microsoft Windows [Version 10.0.17763.1369]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>nslookup
Default Server: UnKnown
Address: 192.168.192.2

> www.netserve.edu.au
Server: UnKnown
Address: 192.168.192.2

*** UnKnown can't find www.netserve.edu.au: Non-existent domain
> ^C
C:\Users\Administrator>ping 192.168.192.2

Pinging 192.168.192.2 with 32 bytes of data:
Reply from 192.168.192.2: bytes=32 time<1ms TTL=128
Reply from 192.168.192.2: bytes=32 time<1ms TTL=128
Reply from 192.168.192.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.192.2:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\Administrator>_
```

Using powershell and getting the commands we get:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Get-NetIPInterface

ifIndex InterfaceAlias AddressFamily NlMtu(Bytes) InterfaceMetric Dhcp ConnectionState PolicyStore
-----
16 Ethernet1 IPv6 1500 25 Enabled Connected ActiveStore
7 Ethernet0 IPv6 1500 25 Enabled Connected ActiveStore
1 Loopback Pseudo-Interface 1 IPv6 4294967295 75 Disabled Connected ActiveStore
16 Ethernet1 IPv4 1500 25 Disabled Connected ActiveStore
7 Ethernet0 IPv4 1500 25 Enabled Connected ActiveStore
1 Loopback Pseudo-Interface 1 IPv4 4294967295 75 Disabled Connected ActiveStore

PS C:\Users\Administrator> Get-DnsClientServerAddress

InterfaceAlias Interface Index Address Family ServerAddresses
-----
Ethernet0 7 IPv4 {192.168.192.2}
Ethernet0 7 IPv6 {}
Ethernet1 16 IPv4 {10.0.2.2}
Ethernet1 16 IPv6 {}
Loopback Pseudo-Interface 1 1 IPv4 {}
Loopback Pseudo-Interface 1 1 IPv6 {fec0:0:0:ffff::1, fec0:0:0:ffff::2, fec0:0:0:ffff::3}
```

To change order of DNS servers, we can edit the Ethernet1 Properties in Advanced TCP/IP Settings and uncheck **“Automatic Metric”** inputting a number of 10. Using the powershell commands again we can find the Ethernet1 metric is 10 and the ip is 10.0.2.2

Week 7

Week 7 Topic: FileSystems

Lecture

The lecture discusses what Filesystems are, how to create partitions and filesystems via various commands, the problem to document filesystem integrity, how to mount filesystems and enforce disk quotas, the different file permissions and ownership, symbolic and hard link information, how to locate files and different ways to backup files.

Lab 7a

Task 1: Gathering information on the host OS and VM

In Linux Workstation:

- **Cat /proc/partitions** for me shows these partitions. I am not sure what IDE or SCSI is

```
[root@benjamin named]# cat /proc/partitions
major minor #blocks name
8        0    20971520 sda
8        1    1048576 sda1
8        2    19921920 sda2
11       0     1048575 sr0
253      0    17821696 dm-0
253      1     2097152 dm-1
```

- There should be 2 hard disks sda1 and sda2. 2 others are for swap and cl-root
- **Blkid** command shows the UUID's of the partitions
- **Fdisk -l** shows sector size of partitions as well as start and end blocks

Task2: Using file permissions to support file sharing among users

In this task we need to make Stewie and Brian can read and write files in the directory /share/family which we should be creating. Peter should not be able to read and write. Files created by Steve can be read and edited by Brian but shouldn't be world readable

We use these commands in order:

1. **Mkdir -p /share/family**
2. **Chgrp family /share/family**
3. **Chmod g+s /share/family**
4. **Chmod 2770 /share/family**
5. **Su - brian → pwd → vim ~/.bashrc** and add 'umask 0002' → **source ~/.bashrc → umask**

Task3: Making /tmp a separate filesystem, and testing mounting and unmounting

In this task, we want to change the /tmp directory from being part of the root filesystem to using a tmpfs filesystem.

I used these commands in order:

1. **Touch /tmp/mytest**
2. **Vim /etc/fstab → add line: tmpfs /tmp tmpfs defaults 0 0**
3. **Mount /tmp**
4. **Mount.** After this /tmp directory should be empty

Last line should look like this after mount command:

```
tmpfs on /tmp type tmpfs (rw,relatime)
```

5. Comment the 'tmpfs line' and change it back to the way before. Rebooting will have the mytest file back in the /tmp directory again

Lab 7b

Task 1: Setting up disk quotas using ext4 (/opt filesystem)

In this task I used the following commands:

1. **Cp /etc/fstab /etc/fstab.bak**
2. **Vim /etc/fstab → Add line /dev/sdb2 /opt ext4 defaults,usrquota 0 0**
3. **Mount -o remount /opt**
4. **Quotacheck /opt**
5. **Quotaon -auv**

```
/dev/sdb1 [/opt]: user quotas turned on
```

6. **Edquota -u peter**

```
Disk quotas for user peter (uid 1000):  
Filesystem      blocks      soft      hard      inodes      soft      hard  
/dev/sdb1         0         400       500         0         0         0
```

7. **We now test configuration. We create a directory for peter giving write permission**

```
[root@benjamin ~]# mkdir /opt/peter  
[root@benjamin ~]# chmod o+w /opt/peter  
[root@benjamin ~]# ll -d /opt/peter  
drwxr-xrwx 2 root root 4096 Oct 26 13:12 /opt/peter
```

8. **We now switch to peter using dd command to create a file that is more than the quota**

```
[root@benjamin ~]# cd /opt/peter  
[root@benjamin peter]# su peter  
[peter@benjamin]/opt/peter% dd if=/dev/zero of=junk bs=1024 count=600  
sdb1: warning, user block quota exceeded.  
sdb1: write failed, user block limit reached.  
dd: error writing 'junk': Disk quota exceeded  
501+0 records in  
500+0 records out
```

9. **Repquota /opt**

```
[peter@benjamin]/opt/peter% exit  
[root@benjamin peter]# repquota /opt
```

Task 2: Setting up disk quotas using xfs (root filesystem)

In this task follow commands below:

1. **Vim /etc/fstab → add: /dev/mapper/cl-root / xfs defaults,usrquota 0 0**
2. **Vim /etc/default/grub:**
 - a. **Edit line: GRUB_CMDLINE_LINUX="resume=/dev/mapper/cl-swap rd.lvm.lv=cl/root rd.lvm.lv=cl/swap rhgb quiet rootflags=usrquota"**
3. **Use command: Grub2-mkconfig, and reboot**
4. **Use command: xfs_quota -x -c state**
 - a. **Account and enforcement should be ON**
5. **Using the command 'edquota -u peter' we edit the line /dev/mapper/cl-root and change soft to 200 and hard to 300**

6. Log in with peter with su – peter

- Doing the `dd if=/dev/zero of=/junk bs=1024 count=600` command again will show the disk quota has exceeded and error writing 'junk'

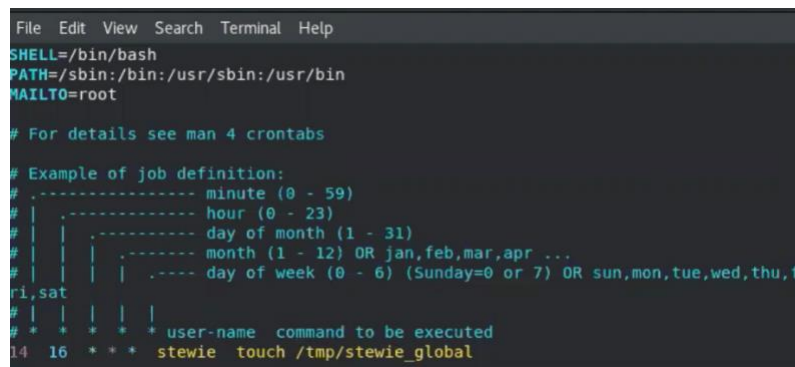
Lab 7c

Task 1: Using cron

In this task, we are using cron to create an empty file in the **/tmp directory** at a predefined time

- Man 5 crontab** shows us some useful information including the time and date fields such as minute, hour, day of month etc. and there allowed values

Using vim, we edit the `/etc/crontab` file via the command : **vim /etc/crontab**



```
File Edit View Search Terminal Help
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# * * * * * user-name command to be executed
14 16 * * * stewie touch /tmp/stewie_global
```

We can then su into our stewie account via the command: **su stewie**. After we use the command **crontab -e** and add the line `14 15 *** touch /tmp/stewie_local`. In the `/tmp` folder we should now have 2 files `stewie_global` and `stewie_local`.

We then create a backup of the `/etc` directory using the command: **tar -cf /tmp/backup-etc.tar etc**

We can check what's in the file by using **tar tf /tmp/backup-etc.tar** to see multiple extra files.

We then use command **cpio -it < /tmp/backup-opt.cpio** to create a backup of the `/opt` directory

We can restore the file `/etc/hosts` into our home dir by using the command: **tar -xvf /tmp/backup-etc.tar -C /root etc/hosts**

Week 8

Week 8 Topic: Networked FileSystems

Lecture

The lecture discusses what NFS are and its history, its states such as stateless and stateful, RPC and portmap, NFS server packages, NFS client information, NFS security and automation, Samba and how to configure it as well as Samba security

Lab 8a

Task 1: Gathering information on the host OS and VM

In this task we setup NFS

Using the command **vim /etc/sysconfig/network-scripts/ifcfg-ens33** we comment and disable the lines `DNS1=10.0.2.2`, `Domain=netsterv.edu.au`, `PeerDNS=no`

We do the same thing for ens37 but change the IPADDR to 10.0.2.1 and disable the DNS, Domain and PEERDNS options. We restart nmcli via the **nmcli** command for both ens33 and ens37

- Nmcli con reload ens33 → nmcli con reload ens37 → nmcli con up ens33 → nmcli con up ens37

We then must ensure the required packages **nfs-utils**, **rpcbind**, **nfs4-acl-tools** are installed. We can do this via the **rpm -q nfs-utils**, **rpm -q rpcbind**, **rpm -q nfs4-acl-tools** commands. All should have a message pop up to show that they are installed.

We then create a directory by the command **mkdir /share/IT_Projects**. We edit the **/etc/exports** file via vim and add the following line: **/share/IT_Projects 10.0.20.0/24(ro)**. We start it now via the commands: **systemctl start rpcbind** → **systemctl start nfs-server** → **systemctl enable rpcbind** → **systemctl enable nfs-server**

Using the **exportfs -v** command we can verify the **/share/IT_Projects** directory is exported as seen below

```
[root@benjamin ~]# exportfs -v
/share/IT_Projects

    10.0.2.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,ro,secure,root_squash,no_all_squash)
```

In another terminal we create **mkdir /mnt/projects** and then use the command **mount -t nfs4 10.0.2.1:/share/IT_Projects /mnt/projects** to create a directory that we can use as a mount point.

We now create a file via the command **touch /share/IT_Projects/file1**. Using the **ls /mnt/projects/** command we can see file1 is in the directory. To unmount the directory we use the command **umount /mnt/projects**. We can now observe that there is nothing else in the **/mnt/projects** directory.

Task 2: Advanced Task –Setup a separate Linux virtual machine to access the remote NFS share

In this task we create a second linked clone of our CentOS VM.

As a result, in our secondVM we set the ens37 IP address to 10.0.2.2/24 via the command **vim /etc/sysconfig/network-scripts/ifcfg-ens37**

- In this we edit the IPADDR line to 10.0.2.3 and use **nmcli con reload ens37** and **nmcli con up ens37** to reload our configuration.
- We then ping the IP addresses to see if they are working correctly which they are

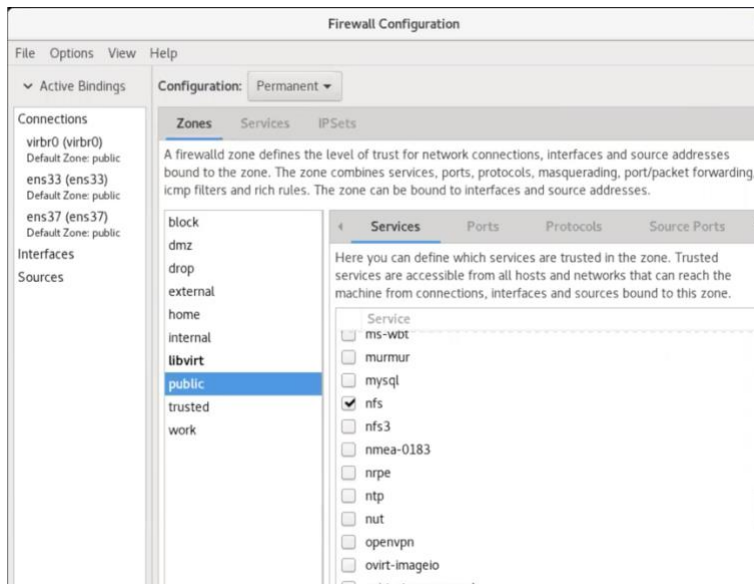
```
PING 10.0.2.3 (10.0.2.3) 56(84) bytes of data:
64 bytes from 10.0.2.3: icmp_seq=1 ttl=64 time=0.280 ms
64 bytes from 10.0.2.3: icmp_seq=2 ttl=64 time=0.755 ms
^C
--- 10.0.2.3 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 15ms
```

We then use the following commands:

```
[root@benjamin ~]# rm -rf /mnt/projects
[root@benjamin ~]# rm -rf /share/IT_Projects/
[root@benjamin ~]# vim /etc/exports (Make sure there is nothing)
```

```
[root@benjamin ~]# systemctl restart rpcbind
[root@benjamin ~]# systemctl restart nfs-server
[root@benjamin ~]# exportfs -v
[root@benjamin ~]# mount -t nfs4 10.0.2.1:/share/IT_Projects
/mnt/projects
```

If **mount -t nfs4 10.0.2.1:/share/IT_Projects /mnt/projects**, does not work we have to go to our other machine and via **firewall-config** enable nfs



Using vim, in the **/etc/fstab** file, we add the line: **10.0.2.1/share/IT_Projects /mnt/projects nfs default 1 1** and reboot our VM.

In our server VM, we rewrite the **/etc/exports** file line to **/share/IT_Projects 10.0.2.0/24(rw)** for read and write.

We now use the commands:

```
[root@benjamin ~]# exportfs -r
[root@benjamin ~]# chmod 777 /share/IT_Projects
[root@benjamin ~]# exportfs -v
/share/IT_Projects
10.0.2.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
```

- We can now see ro has changed to rw now

```
[root@benjamin ~]# mount -o remount /mnt/projects
```

On our client, we now remount the filesystem from the server using the command above. Creating a new file in **/mnt/projects** shows the new file being in the user and group of 'nobody'. If somebody logs in with root and creates files, the privilege will become squash for security reasons.

Lab 8b

Task 1: Samba basics

Before we do anything we must verify if ens37 has a static ip address of 10.0.2.1. We can do this via the command **ipconfig**. We then verify if samba is installed via the command **rpm -q samba**.

After, we do a backup file of **/etc/samba/smb.conf** via the command: **cp /etc/samba/smb.conf /etc/samba/smb.conf_backup**. We then make changes to **/etc/samba/smb.conf** of **workgroup**, **netbios name** to **MYSAMBASERVER1**, **interfaces** **10.0.2.0/24** and **127.0.0.0/8**, and **hosts allow** to **'10.0.2'**. We also set **browseable** & **read-only** to yes in the config file for homes.

```
File Edit View Search Terminal Help
# Run 'testparm' to verify the config is correct after
# you modified it.

[global]
    workgroup = WORKGROUP
    security = user

    passdb backend = tdbsam

    printing = cups
    printcap name = cups
    load printers = yes
    cups options = raw

    netbios name = MYSAMBASERVER1

    interfaces = 10.0.2.0/24 127.0.0.0/8
    hosts allow = 10.0.2.

[homes]
    comment = Home Directories
    valid users = %S, %D%\%S
    browseable = No

-- INSERT --                                     20,1
```

We use the command below to verify our changes:

```
[root@benjamin ~]# testparm /etc/samba/smb.conf
Load smb config files from /etc/samba/smb.conf
Loaded services file OK.
Weak crypto is allowed
Server role: ROLE STANDALONE

Press enter to see a dump of your service definitions
```

We now allow samba via the firewall through the command **firewall-config** in permanent.

After this, we use the **pdbedit** command to set up Samba accounts for existing users such as **pdbedit -a root** and **pdbedit -a peter** for Peter griffin and give them new passwords

```
Unix username:      root
NT username:
Account Flags:      [U          ]
User SID:           S-1-5-21-3529488339-1730825868-2147975435-1000
Primary Group SID:  S-1-5-21-3529488339-1730825868-2147975435-513
Full Name:          root
Home Directory:     \\mysambaserver1\root
HomeDir Drive:
Logon Script:
Profile Path:       \\mysambaserver1\root\profile
Domain:             MYSAMBASERVER1
Account desc:
Workstations:
Munged dial:
Logon time:         0
```

Systemctl restart smb → systemctl restart nmb → systemctl enable smb → systemctl enable nmb are used to enable services at start up.

Task 2: Testing Samba from Linux

Using the command: **smbclient -L 10.0.2.1** will give us these results

```

[root@10.0.2.1 ~]# smbclient -L 10.0.2.1
Enter WORKGROUP\root's password:
Anonymous login successful

        Sharename      Type      Comment
        -----
        homes           Disk      Home Directories
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.11.2)
SMB1 disabled -- no workgroup available

```

When using the samba root password, it will be for the root user, it won't be an anonymous login. We will also get the sharename 'root', which shows the difference in the two lists.

```

        Sharename      Type      Comment
        -----
        homes           Disk      Home Directories
        print$          Disk      Printer Drivers
        IPC$            IPC       IPC Service (Samba 4.11.2)
        root           Disk      Home Directories

```

We can now connect to the home directory of Peter via the command: **smbclient -U peter //10.0.2.1/peter**. Within the home directory and using **dir** we can see many things such as junk, cache and much more.

```

.                D          0
..               D          0
.mozilla         DH         0
.bash_logout     H          18
.bash_profile    H         141
.bashrc          H         312
.zshrc           H         658
.config          DH         0
.esd_auth        H          16
.zcompdump       H        45676
.cache           DH         0
junk             N       208896

```

Task 3: Testing Samba from Windows Server

In windows file explorer, we will try to connect to the home directory from our windows VM. Typing **http://10.0.2.1/peter** we get something like this to see shared files. If we disable read only to **No** in our **/etc/samba/smb.conf** I get a not enough space error.



Task 4: Creating your own shares on linux

We want to now configure Samba to share the /tmp directory of our linux server to our windows clients. We can do this by editing the `/etc/samba/smb.conf` file, and add the following lines:

```
[print$]
comment = Printer Drivers
path = /var/lib/samba/drivers
write list = @printadmin root
force group = @printadmin
create mask = 0664
directory mask = 0775

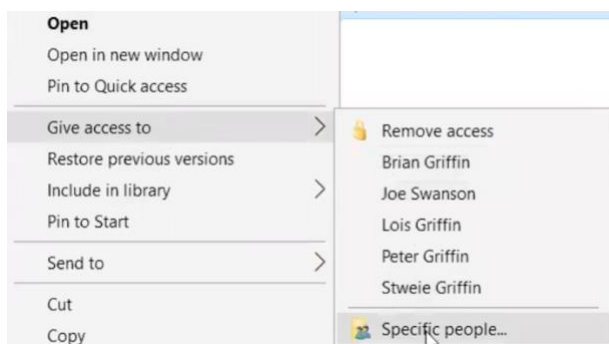
[public]
comment = Public Stuff
path = /tmp
public = yes
writable = yes
browseable = yes
force user = nobody
printable = no
write list = +staff
;
;
[opt]
path=/opt
public=yes
writable=no
browseable = yes
```

Testparm `/etc/samba/smb.conf` shows everything is fine. When browsing [\\10.0.2.1](http://10.0.2.1) in Windows the shares I saw were:

- Peter (Home Dir), homes, opt, public
- Access is denied in 'opt' as we need write permissions
- Files in public were temporary:
 - E.G: systemd-private-a846e433b30e4f85....

Task 5: Creating your own shares on Windows Server

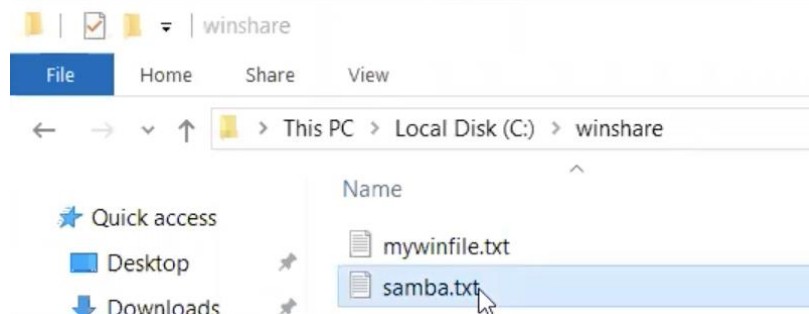
Following the commands by creating a folder in C:\ called 'winshare', we then create a file called 'mywinfile.txt' within that folder. We then right click the folder and give access to specific people.



After this, we add stewie and give him read/write permissions. After this, we just share the folder. On linux we then run the command: **smbclient -U stewie //10.0.2.2/winshare**. We can now see our files we created in our Windows Server in Linux

Using commands below, we can upload files to windows.

```
smb: \> get mywinfile.txt /tmp/win
getting file \mywinfile.txt of size 0 as /tmp/win (0.0 KiloBytes/sec) (average 0
.0 KiloBytes/sec)
smb: \> put /etc/samba/smb.conf samba.txt
```



Week 9

Week 9 Topic: Web Servers

Lecture

The lecture discusses the different type of web servers and there differences such as apache and nginx, Web Service Stack alongside LAMP, apache configurations, virtual hosts, web server security and SSL

Lab 9a

Task 1: Verify Install and start web server

First of all, using the command **systemctl status httpd**, our apache webserver is inactive currently.

rpm -qa | grep mod_ssl, shows mod_ssl being installed.

There are many files installed. We can see some of them via the command **rpm -qs httpd** as seen below as well as **rpm -qs httpd-filesystem** for more succinct results

```
normal /usr/share/httpd/icons/world2.gif
normal /usr/share/httpd/icons/world2.png
normal /usr/share/httpd/icons/xml.png
normal /usr/share/httpd/noindex
normal /usr/share/httpd/noindex/common
normal /usr/share/httpd/noindex/common/css
normal /usr/share/httpd/noindex/common/css/bootstrap.min.css
normal /usr/share/httpd/noindex/common/css/styles.css
normal /usr/share/httpd/noindex/common/fonts
normal /usr/share/httpd/noindex/common/fonts/overpass_bold-web.eot
normal /usr/share/httpd/noindex/common/fonts/overpass_bold-web.svg
normal /usr/share/httpd/noindex/common/fonts/overpass_bold-web.ttf
normal /usr/share/httpd/noindex/common/fonts/overpass_bold-web.woff
normal /usr/share/httpd/noindex/common/images
normal /usr/share/httpd/noindex/common/images/centos-header.png
normal /usr/share/httpd/noindex/common/images/ph-apache.png
```

After this, we use **systemctl start httpd** and **systemctl enable httpd** to enable httpd daemon at boot time.

Everything seems to be fine when we type **localhost** in firefox with an apache startup page.

Task 2: Basic Web server functionality

In this task, we edit the `/etc/sysconfig/network-scripts/ifcfg-ens37` first. We make sure IPADDR is 10.0.2.3 and DNS1 and priority is enabled. We then use `nmcli con reload ens37` and `nmcli con up ens37` to reload configurations. We can use `ifconfig` and `dig www.it.netserv.edu.au` to see if everything is correct, which is good when we get no status errors and answer of 2 from the dig command.

We now edit the `/etc/httpd/conf/httpd.conf` file via vim. In this we change ServerName and DocumentRoot

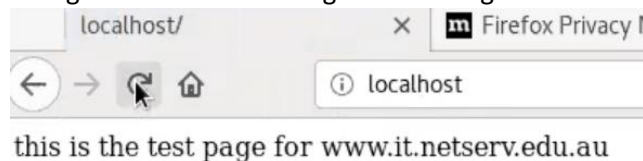
```
#
ServerName www.it.netserv.edu.au:80

DocumentRoot "/var/www/html"
```

We then create a file named `indexhtml` in our `/var/www/html` directory

- `Cd /var/www/html/`
- `Touch index.html`
- `Vim index.html` → Add 'This is the test page for www.it.netserv.edu.au'

Going to web browser we get something like this:



Lab 9b

Task 1: Configure SSL

In this task, we want to remove current keys and existing configuration

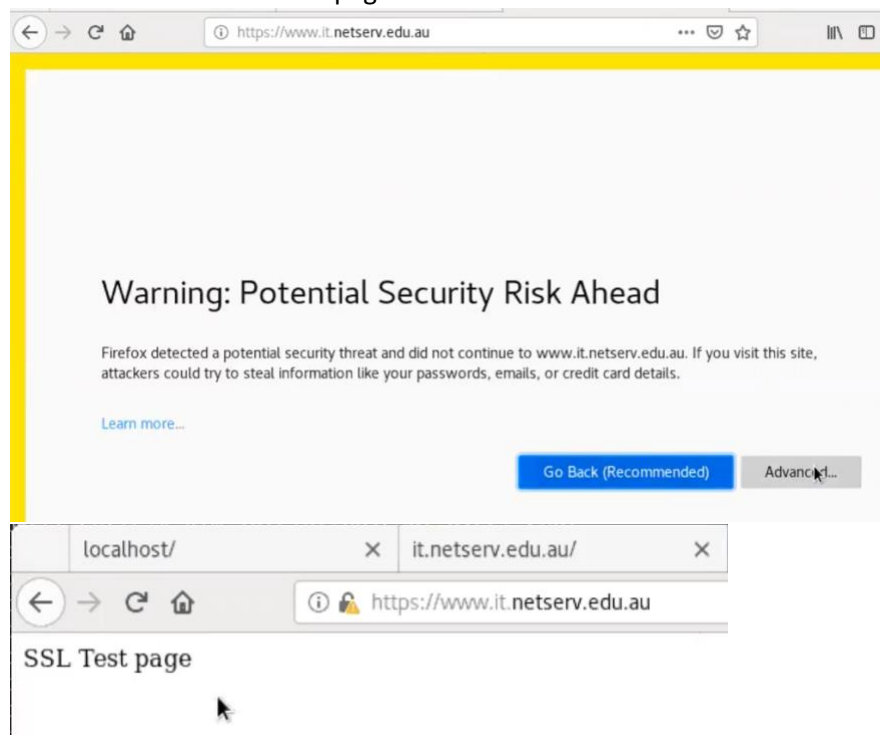
```
root@benjamin certs]# cd /etc/pki/tls/certs
root@benjamin certs]# rm -f ../private/localhost.key
root@benjamin certs]# rm -f localhost.crt
root@benjamin certs]# openssl req -x509 -nodes -days 365 -newkey
rsa:2048 -keyout ../private/localhost.key -out localhost.crt
Generating a RSA private key
.....+++++
.....+++++
Writing new private key to '../private/localhost.key'

You are about to be asked to enter information that will be
incorporated into your certificate request.
What you are about to enter is what is called a Distinguished Name
or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value.
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [XX]:AU
```

```
State or Province Name (full name) []:NSW
Locality Name (eg, city) [Default City]: Sydney
Organization Name (eg, company) [Default Company Ltd]:UTS
Organizational Unit Name (eg, section) []: FEIT
Common Name (eg, your name or your server's host name)
[]:www.it.netserv.edu.au
Email Address []: benjamin.c.lee@student.uts.edu.au
```

After this, the private **localhost.key** and **localhost.crt** should be generated. In **/etc/httpd/conf.d/ssl.conf**, using vim we change the documentroot to "var/www/secure"

We then make a new web page via the commands: **mkdir /var/www/secure → cd /var/www/secure/ → touch index.html → vim index.html** and add line: **SSL Test Page → systemctl restart httpd**. Viewing the page on firefox we get a security risk error. We can then accept the risk in advanced and see the web page.



Lab 9c

Task 1: Virtual hosting a second domain

In this task, we want to create a second domain name in the DNS server. We first change the **/etc/hosts** file with vim and edit/add the following lines within the file:

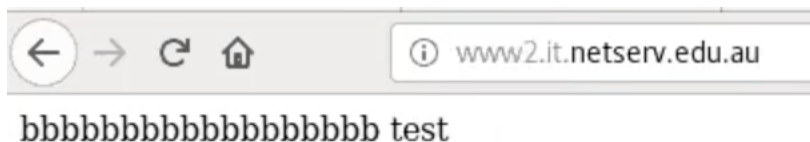
```
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1          localhost localhost.localdomain localhost6 localhost6.localdomain6
10.0.2.3     www2.it.netserv.edu.au
10.0.2.3     www.abc.com
```

In the **/etc/httpd/conf/httpd.conf** file with vim, We add and copy the server configuration given in the lab manual and edit it:

```
# Load config files in the "/etc/httpd/conf.d" directory, if any.
IncludeOptional conf.d/*.conf
<VirtualHost *:80>
DocumentRoot "/var/www/a"
ServerName www.it.netserve.edu.au
# Other directives here
</VirtualHost>
<VirtualHost *:80>
DocumentRoot "/var/www/b"
ServerName www2.it.netserve.edu.au
# Other directives here
</VirtualHost>
```

We then make some new directories via the commands **mkdir /var/www/a** and **mkdir /var/www/b**. We then create an index.html file within the **/var/www/a** directory through the **touch** command. We can add some lines to the index.html file such as 'aaaaaaaaaaaa test'.

We do the same within the **/var/www/b** directory by creating an index.html file with the **touch** command and adding lines such as 'bbbbbbbbbb test'. We should now have 2 web pages when we restart with the command **systemctl restart httpd**.



Task 2: Testing from your Host Operating System

My ens33 is configured as 192.168.192.128. If it doesn't connect we need to build the firewall rules via the command **firewall-config** and add **http** and **https** to the public zone and permanent section.



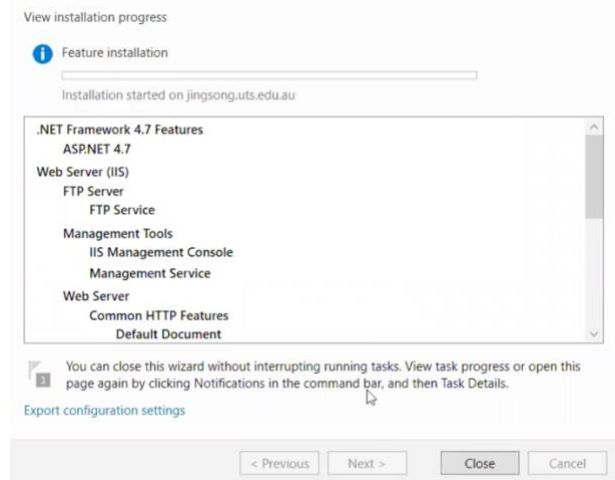
On Windows we then go to: **C:\Windows\System32\drivers\etc\hosts**, and add the following line in the file: **192.168.192.128 www.it.netserve.edu.au www2.it.netserve.edu.au**. We should now be able to have www.it.netserve.edu.au working.

Lab 9d

Task 1: Install the Internet Information Services role

In this task, we go to Server Manager → Local host → Add roles and features → Add web server (IIS) → Enable the following role services: Common HTTP features, Security – Request Filtering, Basic Authentication, IP and Domain Restrictions, URL Authorization, Windows Authentication, FTP

Service, Management tools – IIS Management Console, Management Service



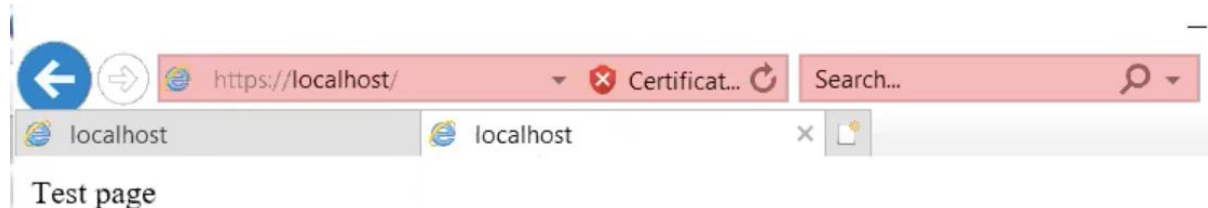
Looking at **http://localhost/** after installation, I see a blue screen with 'Internet Information Services' as a title which means it should be working properly. Unfortunately, I do not know why IIS does not start certain services.

Task 2: Manage the ISS role

In this task we create an index.html file in the default document root directory. We can go to **Tools → IIS Manager → BENJAMIN LEE/Application Pools/Sites/Default Web site → Under Actions, click Explore → Create new index.html page within the directory → Put something in the index.html file like 'test page' → Refresh page for localhost to change the index.html in real time to 'test page'**

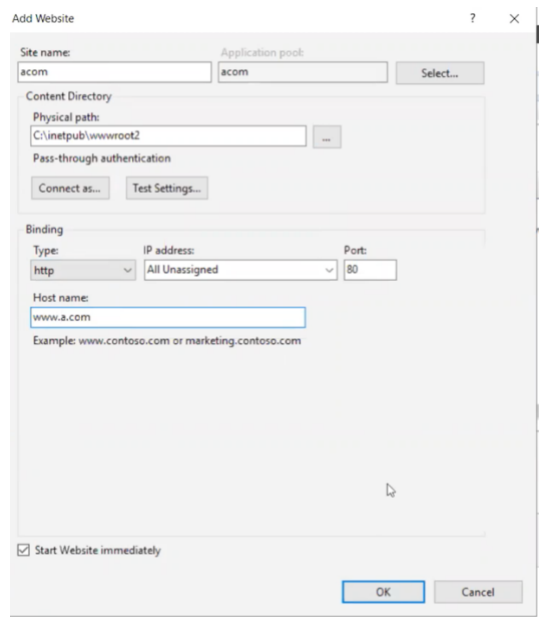
Task 3: SSL web sites

In this task we can go to **Tools → IIS Manager → BENJAMIN LEE/Application Pools/Sites/Default Web site → Under Actions, click Bindings → Add HTTPS with Default certificate: WMSCV-SHA2 → Refresh localhost webpage, we should get a site is not secure error → Go to webpage anyway. We get a certificate error, as we do not have a self-signed certificate.**



Task 4: Virtual Hosting

In this task, we go to **Tools → IIS Manager → BENJAMIN LEE/Application Pools/Sites → Right click and add new website → We will create a website called acom according to the tutorial lab video and create a physical path to c:\inetpub\wwwroot2 → Make Hostname www.a.com**



In our **wwwroot2** directory, we will create a new index.html file with contents 'a.com'. We then change the host file in **C:\Windows\System32\drivers\etc\hosts** and add a new record 10.0.2.2 as seen below.

```
# localhost name resolution is handled within DNS itself.
#       127.0.0.1       localhost
#       ::1            localhost
10.0.2.2       www.a.com
```

We then get a security alert, clicking yes, and we should be directed to our a.com website correctly.

Week 10

Week 10 Topic: Printing and Miscellaneous services

Lecture

The lecture discusses how printing works, different commands to print files, how to manage print queues, how to install and configure printers, discusses window printers and unix security, open ports and file permissions, how to find setuid/setgid files, sudo configurations and security of windows, NTFS and users

Lab 10a

Task 1: Set up printing via cups

Cups is already installed in linux, but we need to install the PDF virtual printer **cups-pdf** and PDF viewer **evince**. We then input the following lines:

1. **dnf -y install cups gcc gcc-c++ cups-devel tar wget evince**
2. **wget https://www.cups-pdf.de/src/cups-pdf_3.0.1.tar.gz**
3. **tar -xvf cups-pdf_3.0.1.tar.gz**
4. **cd cups-pdf-3.0.1/src/**
5. **gcc -O9 -s cups-pdf.c -o cups-pdf -lcups**
6. **chmod 700 cups-pdf**
7. **cp -p cups-pdf /usr/lib/cups/backend/**

8. `cd ../extra`
9. `cp cups-pdf.conf /etc/cups/`
10. `cp CUPS-PDF_opt.ppd /usr/share/cups/model/`

After we restart cups with the commands: **systemctl restart cups → systemctl enable cups**

We now configure our virtual printer via the CUPS administration panel: <http://localhost:631>.

We now select Administration and Add printer. Enter user and password. Select CUPS-PDF (Virtual Printer) and continue. When up to the Add printer part I selected 'generic' with 'generic CUPS-PDF Printer' as the model. After this, we set default options and we should get something like this.

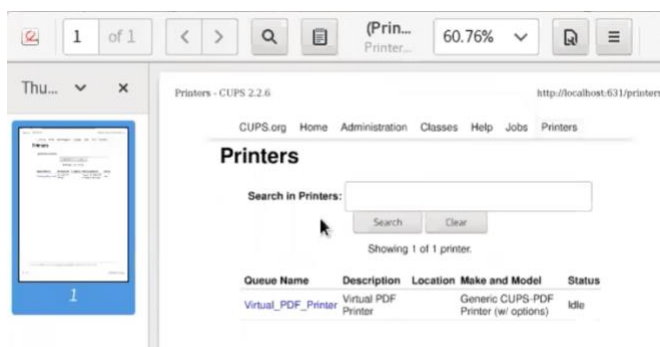
Search in Printers:

Showing 1 of 1 printer.

Queue Name	Description	Location	Make and Model	Status
Virtual_PDF_Printer	Virtual PDF Printer		Generic CUPS-PDF Printer (w/ options)	Idle

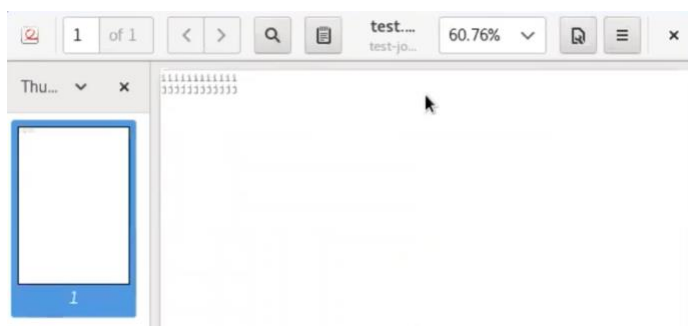
Task 2: Printing on Linux

We can print by pressing Ctrl+P. File result is stored in **/var/spool/cups-pdf/root**



To print on the command line I used the following commands:

1. `export PRINTER=Virtual_PDF_Printer`
2. `lpc status` to see status of printer
3. `touch test.txt`. Added lines `iiiiiiiiii`, `jjjjjjjjjj`
4. `lpr test.txt`



Lab 10b

Task 1: Create and share a printer on Windows

From the lab commands in the lab manual, we go to **start menu → settings → printers and scanners → add printer/scanner → The printer I wanted isn't listed → Add local printer or network printer with manual settings → Create new port → Enter port name: c:\output.prm → Select Generic and Generic Text/Only → Name printer 'generic' → Share printer → Set printer as default → Print a test page.**

Going to our c:/drive we can see a new file called output.prm. We can edit it with notepad and see the following result:

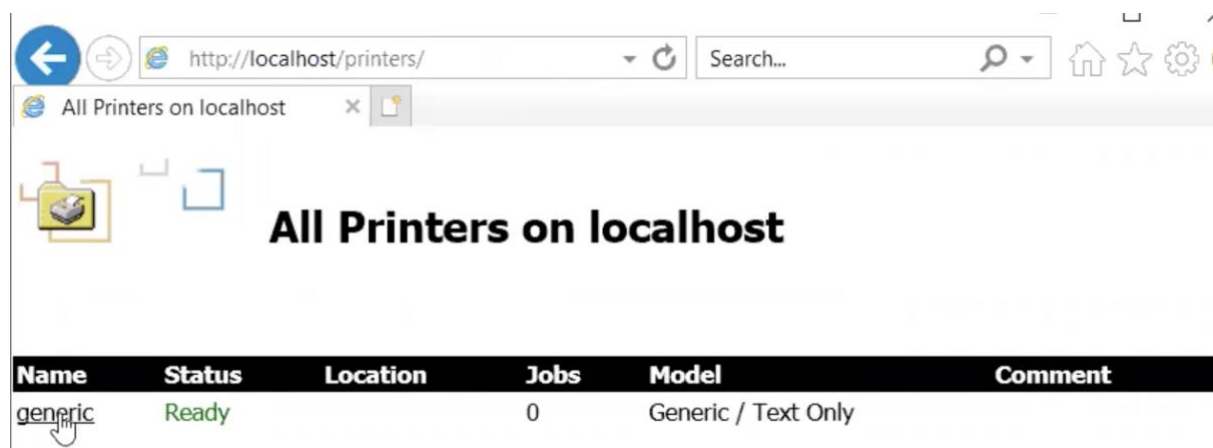
```
Windows Printer Test Page
You have correctly installed your Generic / Text Only on BENLEE
PRINTER PROPERTIES
Submitted      06:52:31 AM
Time
Date           .26/.10/2021
User Name:     BENLEE\Administrator
COMPUTER       BENLEE
Name:
Printer:       generic
Name:
Printer        Generic/ Text Only
MODEL:
COLOR          No
Support
```

Task 2: Set up windows as a print server

In this task, we go to **local server → Add roles and features → When you reach 'Server Roles' select 'Print and Document Service Tools' → When you reach 'Role and Services' select all options → Install**

Task 3: Managing Printers

Going to localhost and finishing installing from task 2 we should get the picture below:

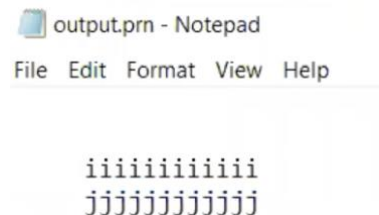


Task 4: Printing from Linux

In this task, in **localhost:631/admin → Add Printer → Select LPD/LPR Host or Printer → In connection, write: lpd://10.0.2.2/generic → Name is 'winprint' and share with printer → Check make is 'generic' and model is 'Generic Text-Only printer (en) → Add printer**

Now with the command line and our test.txt file from beforehand, we can check if we can print via the command **export PRINTER=winprint**. Print via **lpr test.txt**

We can now check that the print occurred by going back to our Windows Server VM and viewing the output file **c:\output.prn**



Reflection

The learning part for the second part journal was much tougher than the 1st. There were some questions that I could not answer from the lab manual, but I enjoyed and knew most of the content.

Unfortunately, my VM machine decided to kill itself when I accidentally mistyped some of the grub commands I think in Week7, and that killed my CentOS machine entirely. It was really frustrating as my VMware license is free and does not support snapshots so I had to basically redo everything from scratch and that took a lot of my time again. I also tried to stop with the overload of pictures to shorten the page count to 25 pages. Overall, I think I did ok when writing this journal, if I had more time and made fewer mistakes I feel like my report could have been better.

The labs were rewarding and I greatly learnt a lot. Other than that, everything else was good.

Overall, I am satisfied with the amount of effort I put into this learning journal.

References

Canvas Lab 6 Materials 2021, Canvas. viewed 2 Nov 2021, <
https://canvas.uts.edu.au/courses/18126/pages/lecture-6-materials?module_item_id=463836>.

Canvas Lab 7 Materials 2021, Canvas. viewed 2 Nov 2021, <
https://canvas.uts.edu.au/courses/18126/pages/lecture-7-materials?module_item_id=463838>.

Canvas Lab 8 Materials 2021, Canvas. viewed 2 Nov 2021, <
https://canvas.uts.edu.au/courses/18126/pages/lecture-8-materials?module_item_id=463840>.

Canvas Lab 9 Materials 2021, Canvas. viewed 2 Nov 2021, <
https://canvas.uts.edu.au/courses/18126/pages/lecture-9-materials?module_item_id=463842>.

Canvas Lab 10 Materials 2021, Canvas. viewed 2 Nov 2021, <
https://canvas.uts.edu.au/courses/18126/pages/lecture-10-materials?module_item_id=463844>.