**31338 Network Servers**

**Learning Journal for:**

**General** **Description**:

The learning journal 2 covers all the Network Servers lectures and labs that is compulsory to be finished. Specifically covering the commands and processes that are required to complete the related tasks.

**Intent**:

Journal 2 has intent to produce a learning portfolio that allows me to remember every action and information needed to complete the process. Due to the tasks being extensive and complicated, a detailed journal will need to be made to ensure that I could complete the tasks by referring to this journal. It will include all the content learnt throughout the semester from week 6 to week 12 for 31338 Network Servers.

**Week 6**

**Lecture 6 Domain Name System (DNS)**

Before DNS

* Centrally held list of hostnames and addresses (Called **HOSTS.TXT**)

Modern equivalent lives on Unix

* */etc/hosts*
* *C:\Windows\System32\drivers\etc\hosts*

DNS

* Name Space

Root Domain (null): represented by “”

Top Level Domain (TLD): first level

* **Generic***: .org; .gov; .com; .com*
* **Country code:** *.au; .uk; .us*

Second-Level Domain

* **2nd level***: edu.au; ns.edu.au*
* **3rd level***: uts.edu.au; ns.uts.edu.au*
* Basic DNS

**FUNCTION**

* resolving hostnames to IP addresses
* IP addresses to hostnames
* specifying name server(s) for a domain
* specifying mail server(s) for a domain
* see */etc/named.conf*
* Slaves & caching
* Primary DNS (**Master**) – (Required, authoritative server)
* Secondary DNS (**Slave**) – (Optional, improve performance)
* Caching-only DNS – (Caching to up performance)

**BIND** (Berkeley Internet Name Daemon)

* **Two** **steps** configuration
  + /etc/named.conf
  + /var/named/chroot/var/named/*myplace.com.zone*

Client

* DNS Client (/etc/resolve.conf) -> set by DHCP, determined by /etc/nsswitch.conf
* Client **Configuration**
  + Traditional ways -> change in /etc/resolve.conf
  + NetworkManager -> interface is separated in /etc/sysconfig/network-scripts/ifcfg-ens ens\*
* **Steps** for **one** **NIC** enabled
  + nmcli general hostname myhostname.mydomain
  + vim /etc/sysconfig/network-scripts/ifcfg-ens ens37
  + nmcli c reload ens37
  + nmcli c up ens37

**Lab 6a DNS Server Configuration**

**Task 1: Design DNS Configuration**

Designing several decisions before configuring DNS Server by:

1. On Windows, choose to set up using the “Standard Zone” configuration (static).
2. Set up 2 domain name servers for our network. (Primary DNS = Windows Server || secondary server = Linux server) (Works contrary for the sub-domain)
3. Requests to the primary server, forward to the university server (ns.uts.edu.au) and thence the internet.

Suggested configuration (assuming Windows is 10.0.2.2 and Linux is 10.0.2.3)

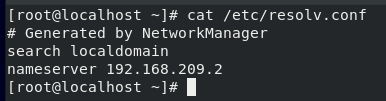
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DNS name | IP address | Type | Which server? | Comments |
|  |  |  |  |  |
| netserv.edu.au | 10.0.2.2 | NS, MX | Windows | Pointer to name & mail server |
| ns.netserv.edu.au | 10.0.2.2 | A | Windows | Actual Primary DNS |
| mail.netserv.edu.au | 10.0.2.2 | A | Windows | Actual Mail server |
| site.netserv.edu.au | 10.0.2.2 | A | Windows | Actual Web server |
| [www.netserv.edu.au](http://www.netserv.edu.au/) | 10.0.2.2 | CNAME | Windows | Alias to site.netserv.edu.au |
|  |  |  |  |  |
| it.netserv.edu.au | 10.0.2.3 | NS, MX | Linux | Point to name server & mail server |
| ns.it.netserv.edu.au | 10.0.2.3 | A | Linux | Actual Primary DNS |
| site.it.netserv.edu.au | 10.0.2.3 | A | Linux | Actual ftp,www server |
| [www.it.netserv.edu.au](http://www.it.netserv.edu.au/) | 10.0.2.3 | CNAME | Linux | Alias to site.it.netserv.edu.au |
| ftp.it.netserv.edu.au | 10.0.2.3 | CNAME | Linux | Alias to site.it.netserv.edu.au |
| mail.it.netserv.edu.au | 10.0.2.3 | A | Linux | Actual Mail server |

The default DNS server configuration on windows is checked by using ipconfig /all, which the default is 192.168.3.2.





However, the current config for windows is different (192.168.209.2) since it is changed for the previous lab experiment.



It is works the same as on Linux (checked by *cat /etc/resolv.conf*) due to previous lab experiment.

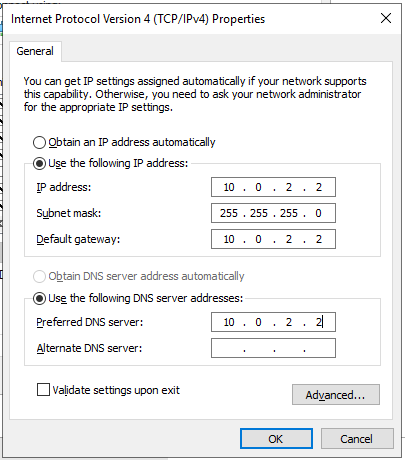
Put a forwarding record in Windows to the UTS name and forward all requests to ns.netserv.edu.au for Linux server.

**Task 2: Set Up a DNS Zone (Forward Lookup)**

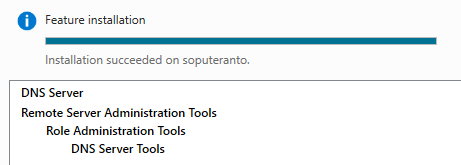
Designing several decisions before configuring DNS Server by:

1. **Configuring DNS role on Windows**

**Step 1**: Set up Ethernet1 adapter to a static address 10.0.2.2

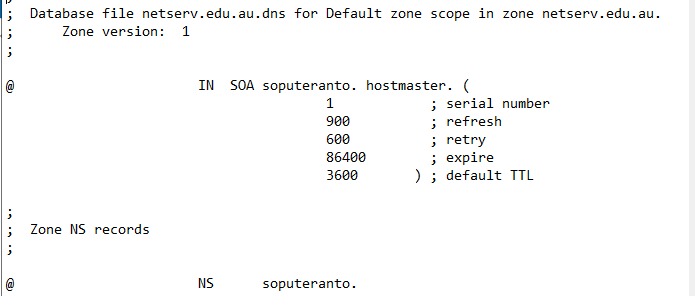


**Step 2**: Install the DNS role from the server manager



**Step 3**: Set up a zone for the DNS server

Primary Zone name will be netserv.edu.au.



Created a file called “netserv.edu.au.dns” in *C:\Windows\System32\dns directory*.

**Step 4**: Set up DNS server properties

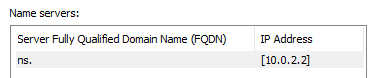
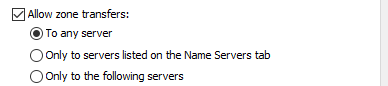
Turn off the Ethernet0 and just keep listening to only 10.0.2.2.

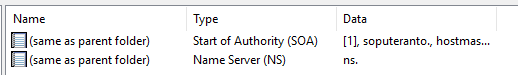


Adjusting the forwarders and other parameters.

**Step 5**: Create the Forward Lookup zone file

Edit the name server entry to be: ns with IP address of 10.0.2.2 and configure the Zone Transfers to “Allow zone transfers to Any server”.



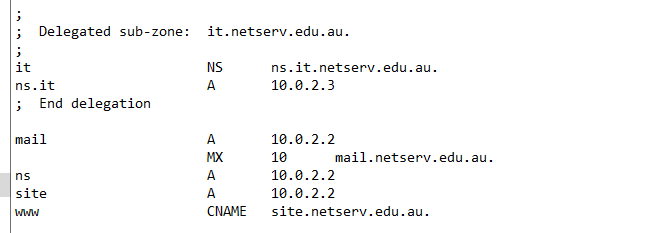


**Step 6**: Add more entries to the Zone file

Enter several records in the zone file:

* New host (A): ns with IP address 10.0.2.2
* New host (A): mail with IP address 10.0.2.2
* New Mail Exchanger (MX): leave host blank and the FQDN should be mail
* New host (A): site with IP address 10.0.2.2
* New Alias (CNAME): www and FQDN site

Then, create a delegation for the it.netserv.edu.au domain, with IP address as 10.0.2.3 (IP address of Linux machine)



Check the updated file in *C:\Windows\System32\dns\netserv.edu.au.dns.*

Then, restart the DNS Server via the server manager.

1. **Testing the configuration**

Set up the default DNS for the windows machine to localhost or 10.0.2.2 by:

**Step 1**: Using the *nslookup* tool on Windows From a command prompt, and input command:

server 10.0.2.2 (use local machine as default server)

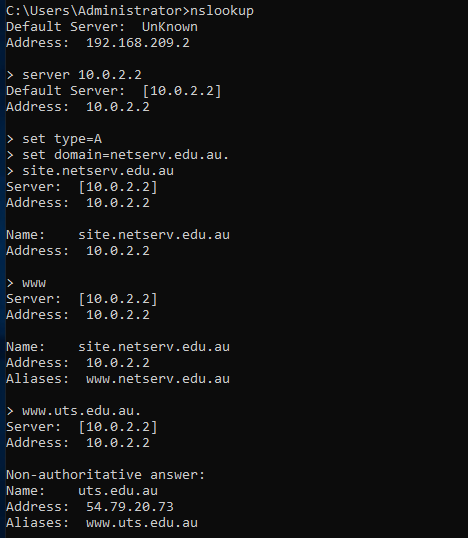
set type=A (only look up A records)

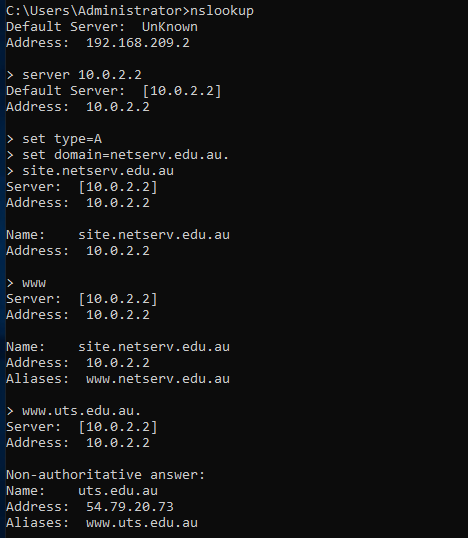
set domain=netserv.edu.au. (set default domain to netserv.edu.au)

site.netserv.edu.au. (lookup site.netserv.edu.au)

www (lookup the IP address of www.netserv.edu.au)

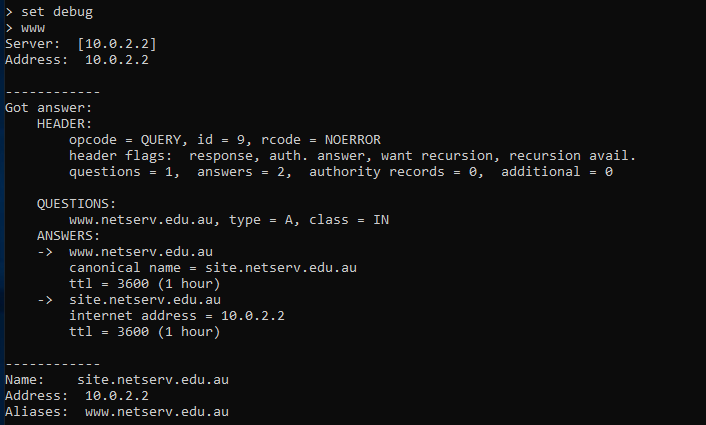
www.uts.edu.au. (lookup UTS main web server)





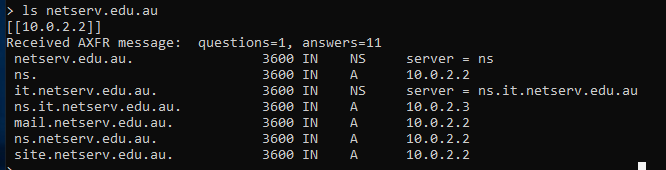
These commands setting up the default DNS and looking up to the site.

Then running *nslookup* command in debug mode: *set debug*, and try the commands again.



This time, the debug command give more information about the sequence for the debugging.

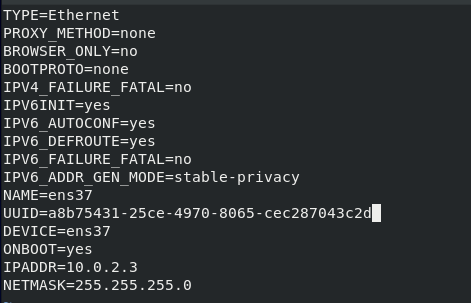
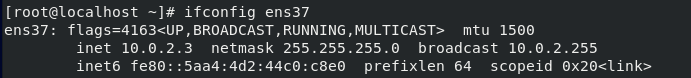
***What about the*** *ls netserv.edu.au.* ***command?***



The *ls netserv.edu.au*. command show more information regarding the A & NS records.

1. **Configuring the DNS server (BIND) on Linux**

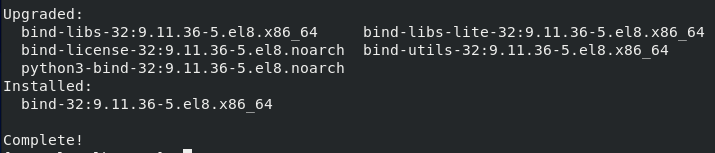
**Step 1**: Set up ens37 with static ip address



Checking and setting up the ens37 with static ip address.  
(*vim /etc/sysconfig/network-scripts/ifcfg-ens37*)

**Step 2:** Install name server

Install the BIND name server by using: *dnf install bind*, and accept all extra libraries.



**Step 3:** Update the name server configuration file

Edit the existing configuration file *on /etc/named.conf* as:

1. Find “*listen-on port 53 {127.0.0.1; };*” and replace “*127.0.0.1*” with “*any*”.

2. Find “*allow-query { localhost; };*” and replace “*localhost*” with “*any*”.

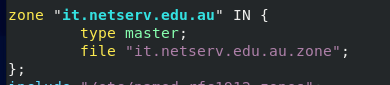
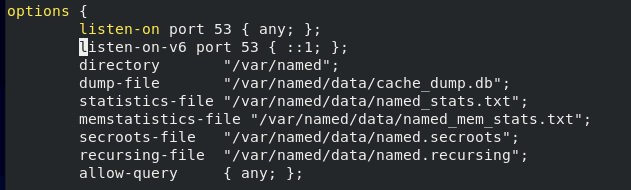
3. Add the configuration as:

*zone "it.netserv.edu.au" IN {*

*type master;*

*file "it.netserv.edu.au.zone";*

*};*

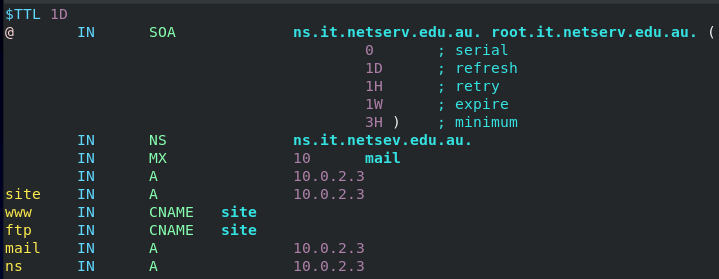
After updating several configuration, save and quit.

**Step 4:** Create Zone file

Create new zone file for the domain by inserting on the directory */var/named/directory* (default location in system). - The name of the zone file must be the same as specified in *named.conf* file.



Copy *named.localhost* and fix the configuration for the SOA record entry and add the rest record.



The added record are:

ns IN A 10.0.2.3  
 site IN A 10.0.2.3  
 ftp IN CNAME site  
 www IN CNAME site

**Step 5:** Change group ownership of file (file permissions)

Change the file permissions so the name server process can read it and also change the group ownership of the zone file.



Using *chgrp* named *it.netserv.edu.au.zone* command to change the group to be ‘named’ group.





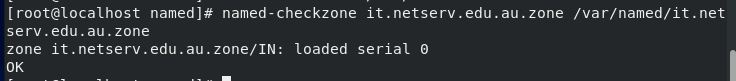
1. **Starting the DNS server on Linux**

**Step 1:** Check configuration

Check configuration by: *named-checkconf /etc/named.conf*



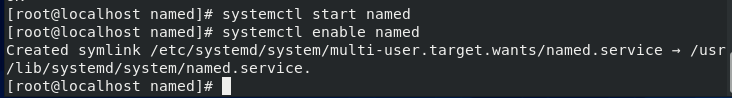
Check your zone file with: *named-checkzone it.netserv.edu.au /var/named/it.netserv.edu.au.zone*



The configuration seems to be OK without any error.

**Step 2:** Start named service

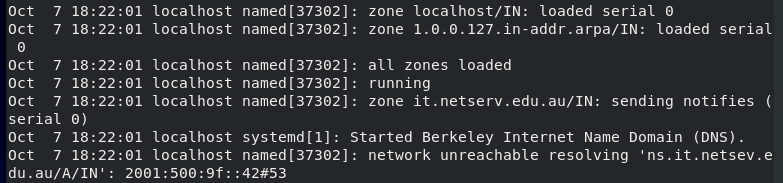
Start named service by: *systemctl start named* and enable it by: *systemctl enable named*



1. **Testing the DNS server on Linux**

**Step 1:** Check system log

Check the end of the */var/log/messages* file and look for line “*zone it.netserv.edu.au/IN*” then find the word “loaded”. There may some issues about “network unreachable”.



**Step 2:** Test server

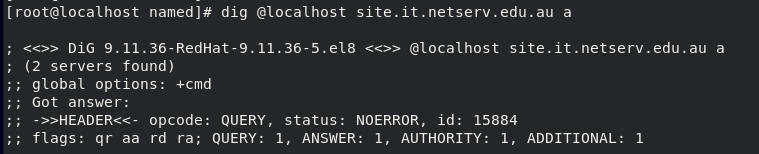
Run test queries using the dig command:

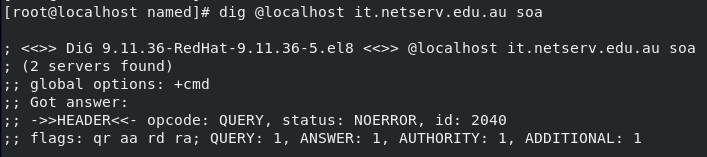
Look up A record: dig @localhost site.it.netserv.edu.au a (use with hostname)

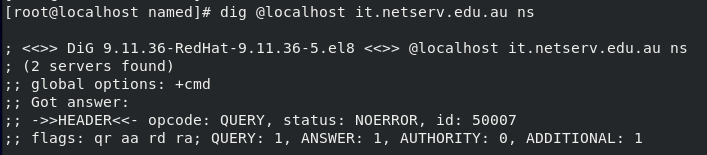
Look up SOA record: dig @localhost it.netserv.edu.au soa (use with domain name)

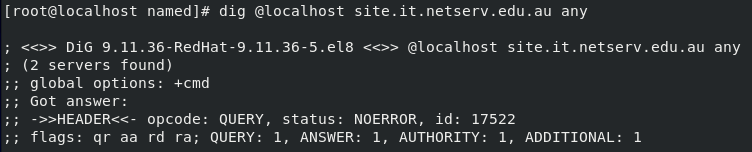
Look up NS record: dig @localhost it.netserv.edu.au ns (use with domain name)

Look up “any” records: dig @localhost site.it.netserv.edu.au any (use with host or domain)

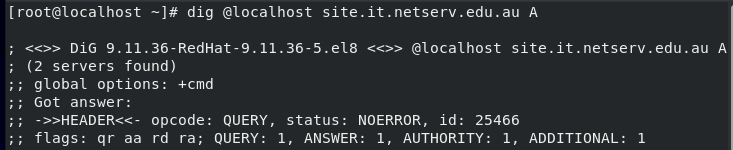
  
Shown the A record with result NOERROR and ANSWER=1 (queries are correct)

  
Shown the SOA record with result NOERROR and ANSWER=1 (queries are correct)

  
Shown the NS record with result NOERROR and ANSWER=1 (queries are correct)

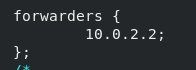
  
Shown any record with result NOERROR and ANSWER=1 (queries are correct)

Then reboot machine and test if the DNS server starts up automatically.

  
The DNS server automatically starts, seen by run the queries to test the server.

**Step 3:** Add forwarding record

Update the /etc/named.conf file and add:

*forwarders {*

*10.0.2.2;*

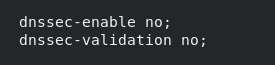
*};*

The DNSSEC is enabled by default to check whether the information from the forwarder has a digitally signed zone file to verify its authenticity. Then, break it by disabling Linux bind server from checking DNSSEC data.

Update “yes” to “no” for the following lines in the */etc/named.conf* file:

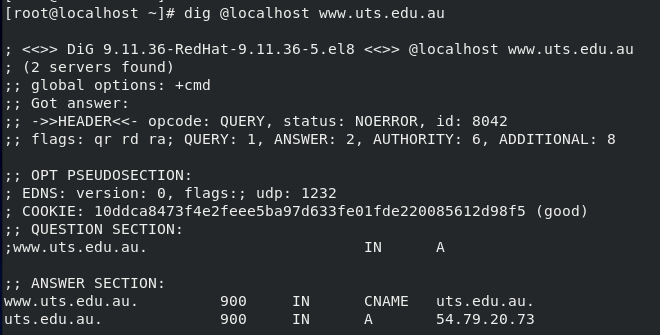
*dnssec-enable no;*

*dnssec-validation no;*



Restart named and re-test it.





It successfully reaches *www.uts.edu.au* shown above image.

**Task 3: Set Up Reverse Lookup Zone File**

The set up a zone file for forward lookup works by mapping from hostnames into IP addresses. However, sometimes the services need to perform a reverse lookup, which works by mapping from given IP address as input, find out what hostname corresponds to it. This could be resolved by setting up another zone file with the reverse lookup information.

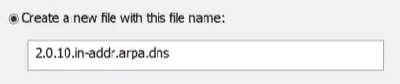
Reverse lookups are based on subnets, so, configure the reverse lookup information for the 10.0.2.0 subnet. There is a special DNS suffix of “in-addr.arpa” used for reverse lookup information. The sequence of octets is also reversed. As a result, set up a DNS zone named “2.0.10.in-addr.arpa” to set up a reverse lookup domain for 10.0.2.0 subnet.

1. **On Windows server**

**Step 1:** Use the DNS manager to create a new Reverse Lookup Zone

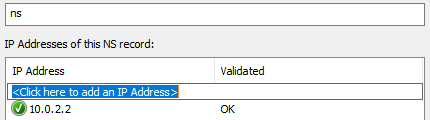
(Reverse Lookup Zones -> Action -> New Zone -> Primary -> IPv4 Reverse Lookup Zone)

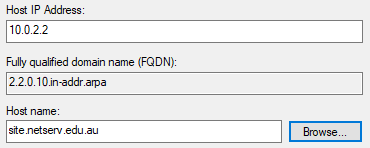
Our network id is: 10.0.2

The wizard chosen filename: 2.0.10.in-addr.arpa.dns



Update the correct SOA and NS entries.



**Step 2:** Create Pointer records

Create new PTR records as :

Host IP address: 10.0.2.2

Hostname: site

**Step 3:** Testing

Test by using *nslookup* command, then set the server and query type as before:

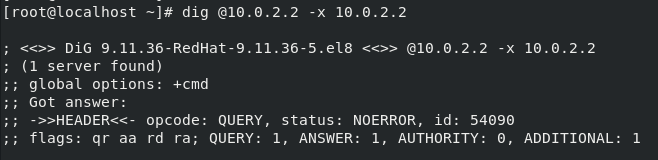
server 10.0.2.2 (local machine default server)

set type=PTR (only look up PTR records)

2.2.0.10.in-addr.arpa. (reverse lookup query)

  
Shown proved that the queries are correct.

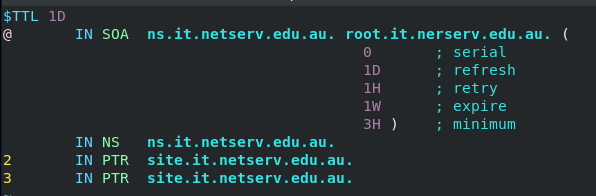
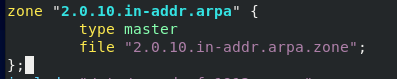
From Linux run command: *dig @10.0.2.2 –x 10.0.2.2*

  
Shown the record with result NOERROR and ANSWER=1 (queries are correct)

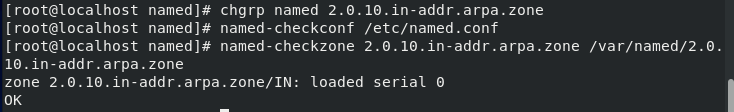
1. **On Linux server (optional)**

**Step 1:** Create Zone file

Edit */etc/named.conf* and add entry for “2.0.10.in-addr.arpa” zone. Then create the zone file in the */var/named/ directory* and add the entry “2 IN PTR site.netserv.edu.au.”.



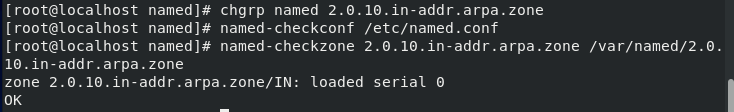
The first entry on the line is the host’s address (“2” comes from the host part of 10.0.2.2).  
The “.” (dot) at the end of the line used to terminate the domain name (essential when use a FQDN in a zone file).



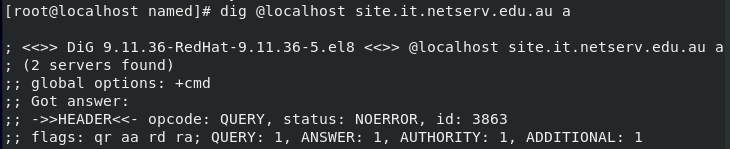
Then change permissions on the zone file.

**Step 2:** Restart named and test

Check configuration using the *named-checkconf* and *named-checkzone* command



Restart your DNS server, check the log file for errors, and test the forward lookups.

  
Shown A record with result NOERROR and ANSWER=1 (queries are correct)

Then try a reverse lookup test, using dig: *dig @localhost –x 10.0.2.2*

  
Shown the record with result NOERROR and ANSWER=1 (queries are correct)

**Lab 6b DNS Client Configuration**

**Task 1: DNS Client Configuration on Linux**

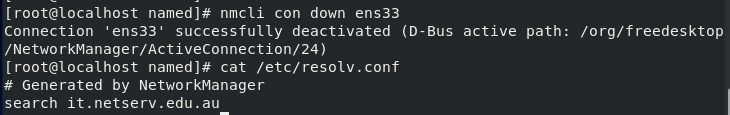
**Step 1:** Examine and understand existing *resolv.conf*

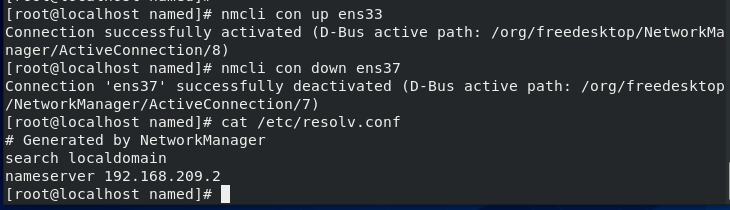
Check and analyze */etc/resolv.conf* file, then determine which network interface (ens33 or ens37) is responsible for each entry by:

• Disabling ens33 interface (set ONBOOT=no in /ifcfg-ens33) then restart NetworkManager (systemctl restart NetworkManager). Then check the resolv.conf file again to see what entries are there.

• Repeat by disabling ens37 and enabling ens33 and restarting NetworkManager.

• Finish by re-enabling both interfaces.

  
Disabling the ens33 interface (enabling ens37) and turns out that the hostname is appeared as it.netserv.edu.au and name server didn’t return anything.

  
Disabling the ens37 interface (enabling ens33) and returns that the localhost is generated by the NetworkManager (has no domain) and name server return as stated in the *resolv.conf* file.

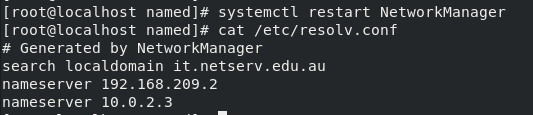
**Step 2:** Add new nameservers and search domains

Add the configuration to the ifcfg-ens37 as:

DNS1=10.0.2.3 #IP address of Linux VM

DOMAIN=it.netserv.edu.au

Restart NetworkManager and check */etc/resolv.conf*.



**What has it done? Which nameserver will be searched first?**

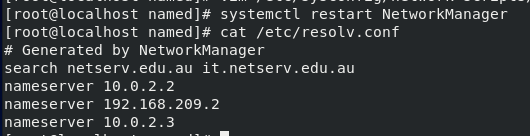
After the new configuration, it will search first the DHCP domain then the network card. It works the same for the nameserver as it searches the DHCP IP first before the network card IP.

Add similar configuration in the ifcfg-ens33 as:

DNS1=10.0.2.2 #IP address of Windows VM

DOMAIN=netserv.edu.au

Restart NetworkManager and check */etc/resolv.conf*



**You’ve probably noticed that there are three nameserver lines rather than just two. Where is the third coming from?**

After the second change for ens33, it will search the etc domain first. It will also show the nameserver from the etc first before the DHCP and the windows nameserver.

**Step 3:** Ignore DHCP nameservers on ens33 interface

Run the command to ignore the DNS server provided by DHCP by:

*nmcli conn modify ens33 ipv4.ignore-auto-dns true*

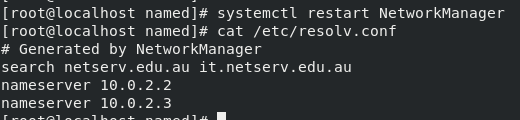




**What new line has the command above just added (ifcfg-ens33)?**

The new command added to *ifcfg-ens33* is *PEERDNS=no*. This command will ignore DNS server from DHCP.

Then restart NetworkManager and view */etc/resolv.conf*. **What has this done?**

  
After restarting NetworkManager, the DHCP nameserver is removed as it is ignored by above command.

**Step 4:** Change the order of nameservers / search domains

The order of search domains and nameservers in *resolv.conf* is determined by NetworkManager (default priority rules).

To change the order of the interface, the priority need to be changed by:

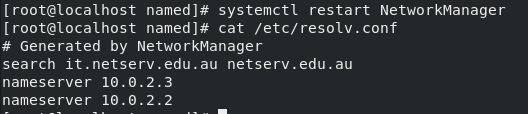
*nmcli conn modify ens37 ipv4.dns-priority 5*



The default priority for normal connections is 100. Check the added configuration in ifcfg-ens37

  
It shows that the command *IPV4\_DNS\_PRIORITY=5*.

Then restart NetworkManager and re-check */etc/resolv.conf.*

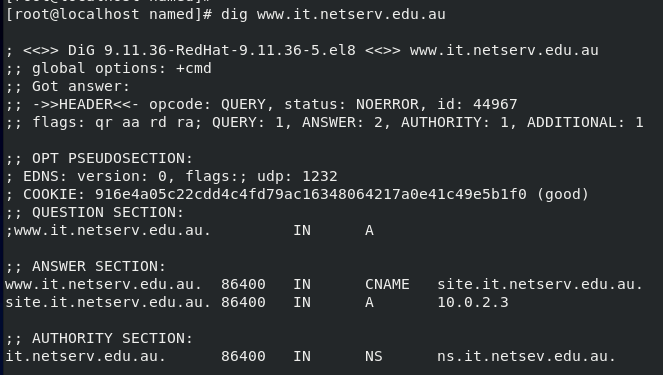
  
The order is now fixed by the ens37 show first.

**Step 5:** Run DNS queries

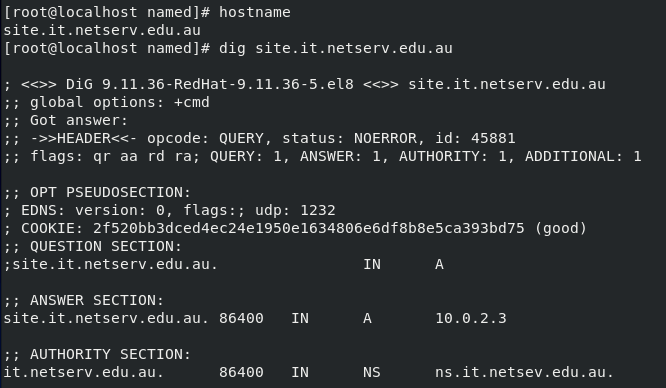
Set DNS queries (by default) to go to Linux server first by configured with the first nameserver entry showing the IP address of Linux VM, and the search domain having “it.netserv.edu.au” as the first entry.

Test the queries using the dig command without using the @localhost statement:

*dig www.it.netserv.edu.au A*

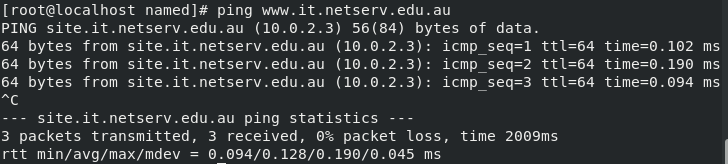
  
It shows that the answer is 2 (correct queries) with the answer section having correct domain.

Replace “*www.netserv.edu.au*” with the hostname of Linux DNS server.

  
It shows that the answer is 1 (correct queries) with the answer section having correct domain.

Then, test by using the ping command:

*ping www.it.netserv.edu.au*

  
It will automatically call on the resolver library to map the supplied hostname into an IP address.

**Task 2: DNS Client Setup on Windows**

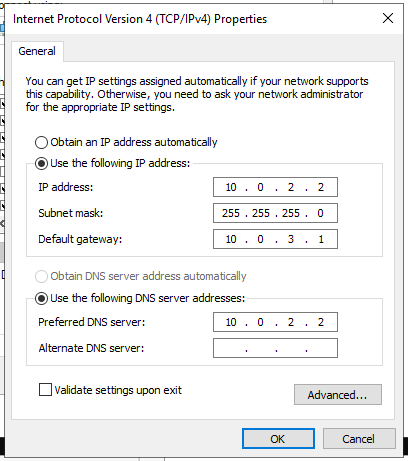
**Step 1:** Set DNS servers in Windows

Override the DNS settings by opening up the Network Connections panel

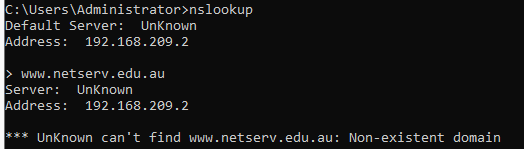
Server manager → Local Server → Ethernet1 → View Network Connections panel

Right-click “Ethernet1” → Properties → Internet Protocol version 4 (TCP/IPv4) → Properties.

The DNS should be static and it can be overridden the “Preferred DNS server” with 10.0.2.2.

  
The preferred DNS server is overridden with 10.0.2.2.

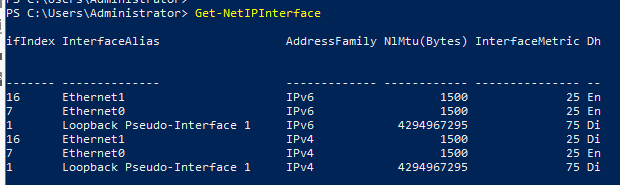
Test that this works correctly using nslookup and ping on Windows.

  
It shows that it could not access to www.netserv.edu.au

**Step 2:** Check the order of DNS servers used by Windows Server

Use PowerShell to view all network parameters by running the command:

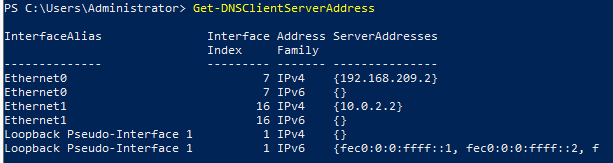
*Get-NetIPInterface*



Check the AddressFamily to look at IPv4/6 interfaces and the InterfaceMetric to see the default metric (Eth0 and Eth1 probably have the same metric). The ifIndex will show the interface index order which probably the order of priority of the interfaces.

Verify the DNS server search order with the command:

*Get-DNSClientServerAddress*



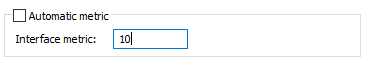
Most likely the DNS server configured on Ethernet0 has priority over the DNS server configured on Ethernet1.

**Step 3:** Change the order of DNS servers used by Windows Server by changing the interface metric

Back to IPv4 properties of Ethernet1, through:

Ethernet1 → … → Properties → “Advanced” → “Automatic metric”.

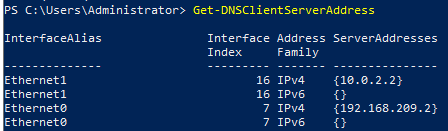
Uncheck “Automatic metric” and enter 10 metric value (lower than default metric - higher priority)



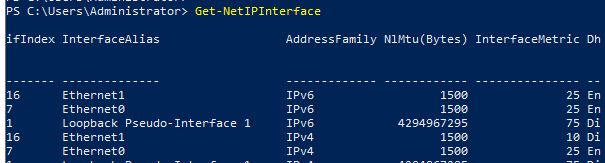
See the changes to the interface metric and DNS server search order:

*Get-NetIPInterface*

*Get-DNSClientServerAddress*

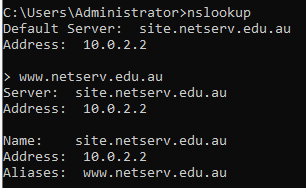


It shows that now the ethernet1 interface is at the top of ethernet0.



It also shows that the InterfaceMetric of ethernet1 is changed to 10.

Re-test DNS *lookups* using nslookup and ping on Windows.

  
The DNS is now working.

**Week 7**

**Lecture 7 Filesystems and Backups**

Create filesystems and partitions

* Create – edit partition table
  + fdisk (MBR only)
  + gdisk (GPT indexing method)
  + parted (GNU program)
* Create filesystem -> mkfs
* Swap space -> mkswap

Filesystem integrity

* Check / repair filesystem -> fsck
* Commands to display info -> blkid, dumpe2fs -h, tune2fs, debugfs

Mounting filesystem

* Filesystem table that able to be mounted -> /etc/fstab
  + mount /media/cdrom
  + mount -t vfat /dev/sdb1 /media/usbdisk
  + unmount /media/cdrom

Monitoring disk usage

* Quota report
  + Repquota -a
* By partition (df)
  + df -k (display all filesystem and disk usage in 1k block size)
  + df -h
* By directory (du)
  + du /home/brookes (See disk usage summary of its directory and sub directories)
  + du -sh /home/brookes (See the summary of total disk usage size of its directory)
  + du -sh \*.txt

Enforcing disk quota

* To set limits on user/group
  + Create separate partition -> /dev/sdb1
  + Mount /dev/sdb1 to /home
  + Set limit on /dev/sdb1 or /home for user peter
  + Generate files in /home/peter
* Edquota -> edit user’s quota

File permission

* Using chmod command
* umask for the default file creation mode
* Change file attributes by chattr
* chown to change owner and its group
* chgrp to change group

Locating files

* find command -> flexible search
* locate command -> based on database
* whereis command -> locates source and manual page file for a command
* which command -> search executable file

**Backups**

Backup tools

* tar (Tape ARchive) -> Unix-style option, BSD-style, GNU style
* cpio -> copy in and out
* dd -> low-level tool for copying data byte by byte

Automation tools

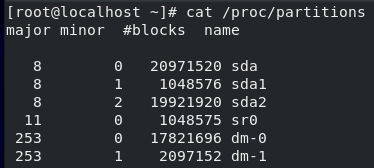
* cron -> run commands at regular intervals
  + controlled with /etc/cron.allow and / etc/cron.deny
* anacron -> catch up on jobs missed by cron
  + configured in /etc/anacrontab
* at -> run command once in the future
  + controlled with /etc/at.allow and /etc/at.deny

**Lab 7a: Managing Filesystems, Including Mounting and Unmounting**

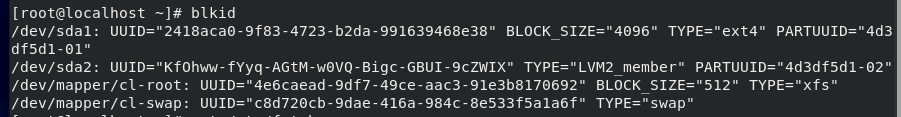
**Task 1: Gathering Information on the Host OS and Your Virtual Machine**

Document the structure of the filesystem including all the currently mounted filesystems.

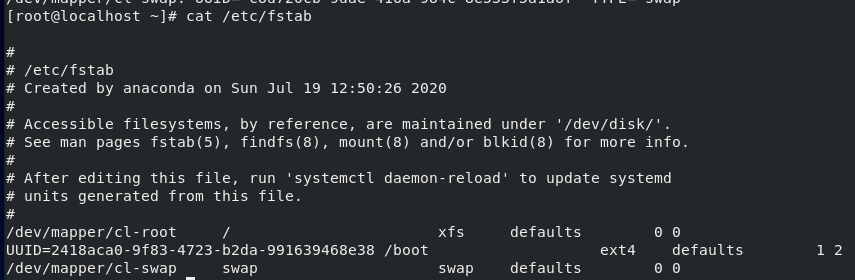
Look at both */proc/partitions* and the *fstab* file. Use the */sbin/blkid* command to figure out the mapping of UUIDs to partitions.



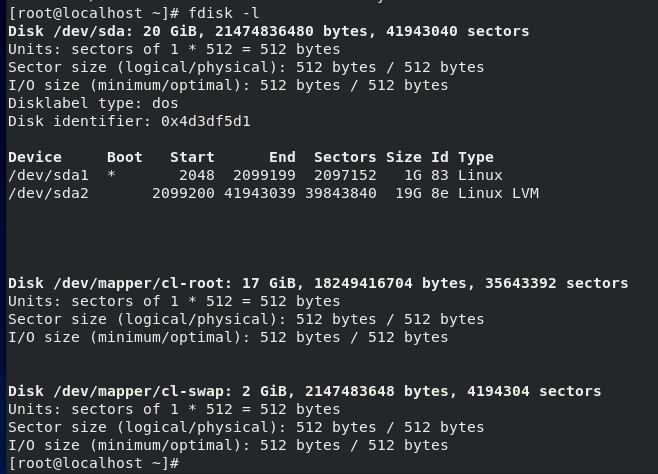
The command */proc/partitions* shows all the partition created in the Linux VM.



*blkid* command will display the UUID of all the disk partition



The *fstab* file shows the UUID of the system.



While the *fdisk -l* command will give more complete information about the partition including the start and end proc.

**How many (virtual) hard disks? Can you tell from the device file names whether the disks are SATA, IDE or SCSI? How many partitions? Which partitions are primary, extended and logical? Where are home directories stored?**

There are 6 (virtual) hard disks on the VM. It could be seen that there are four SCSI disk since sda and sr are the filenames for SCSI. While for dm, it is most likely a SATA filename. The primary partition is the SCSI partition with the sda filenames with the home directories stored in the sda1 device.

**Task 2: Using File Permissions to Support File Sharing Among Users**

In earlier lab, there are four users named Peter, Stewie, Brian and Lois, and assigned two of those users, Stewie and Brian, to a group called family.

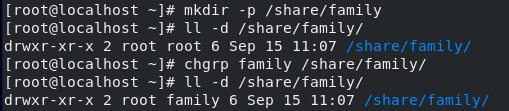
Assume that members of the family group need to share files with each other. Make a directory called */share/family* and make Stewie and Brian can read and create files in this directory, but Peter cannot. Files created by Stewie can be read and edited by Brian and vice versa.

Solve this problem by:

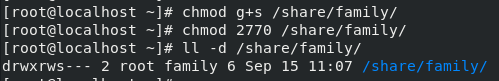
• File permissions on the */share/family* directory, including the *SetGID* *bit* (*chmod* command)

• Group ownership of the */share/family* directory (*chgrp* command)

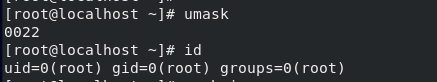
• Umask settings of Stewie and Brian (*umask* command)



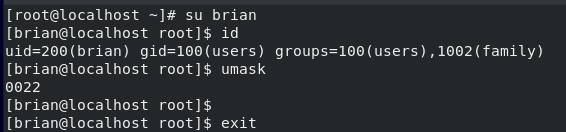
Create */share/family* directory and change the group ownership using *chgrp* command.



Set the file permission on the directory with the GID and 2770 as 2 is the group id and 770 is read write and execute permission for the group member.



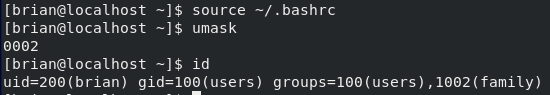
Checking the umask and id for the root.

****

Checking the umask and id for the brian.

****

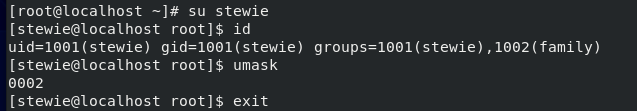
Add umask on the brian user by editing *~/.bashrc*

****

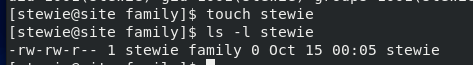
Now the brian user has the umask 0002 not the 0022.

****

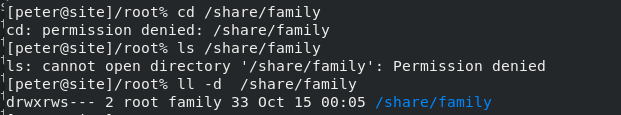
Touch brian to change the ownership group to family not user.

****

Checking the umask and id for the stewie.

****

Stewie already has the group ownership belong to family.



Shown the peter user that peter could not access the */share/family* group.

**Task 3: Making /tmp a Separate Filesystem, and Testing Mounting and Unmounting**

Change */tmp* directory from being part of the root filesystem to using a *tmpfs* filesystem.

Create an empty file in */tmp* directory ("*touch /tmp/mytest*"), then make change to */etc/fstab* file so */tmp* uses *tmpfs*, as:

*Tmpfs /tmp tmpfs defaults 0 0*



The file is created on the directory.



Add configuration to the *fstab* file

Mount filesystem with ‘*mount /tmp*’ and use the mount command to show the current mounted filesystems and verify that */tmp* is now using *tmpfs*.





Shown mount command proved that the *fstab* file is working.

Verify the file (*/tmp/mytest*) is not in the *tmpfs* filesystem **(where is it?).**



The */tmp* file is not in the *tmpfs* filesystem anymore. It is not removed but is hidden.

Change */tmp* back to the way it was before (to not degrade the performance of VM).



Commenting the command will prevent it from running.



Test the command again and it shows that the file is back.

**Lab 7b: Implementing Disk Quotas**

**Task 1: Setting Up Disk Quotas Using Ext4 (/opt filesystem)**

The basic steps are:

1. Edit */etc/fstab* and add the option to turn on user quotas (*usrquota*) for the opt directory filesystem (*/opt*). Make a copy of */etc/fstab*, calling it */etc/fstab.bak* for recovery (corrupted *fstab* file will prevent system from booting normally).

2. Remount filesystem so the new options from *fstab* take effect. To remount a filesystem when having changes:

*mount –o remount /opt*

3. Run *quotacheck* command for */opt* filesystem.

4. Turn quotas on with *quotaon* command (needs an argument – read man entry)

5. Use *edquota* to edit peter's disk quota. Set a soft limit (warning) at 400KB and hard limit at 500KB for block usage. No need to set limits on inodes (which is a limit on the number of files a user can create).



Create a recovery *fstab* file as *fstab.bak*



Add the *usrquota* for the */opt* to add the option to turn on user quotas on *fstab* file.



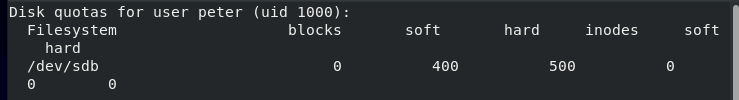
Remount the filesystem by the new option.



Check the */opt* by running *df -h* command.

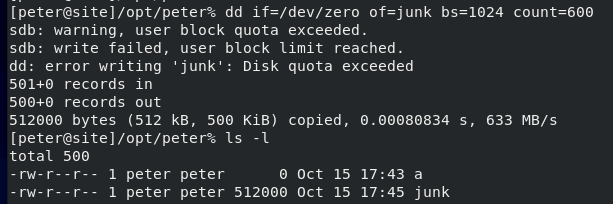


Run *quotacheck* for */opt* then turn on the user by quotas command.



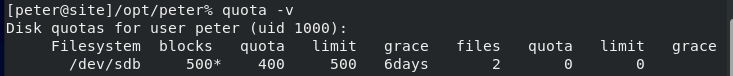
Change the soft limit to 400 and hard limit to 500 on peter’s disk quota using *edquota* command.

Test whether disk quotas is working by logging in as the user peter and trying to store more than 500KB in the */opt* directory. Create a directory under */opt* and make peter able to write to that directory (Create a large file using *dd* command - *dd if=/dev/zero of=junk bs=1024 count=600*).



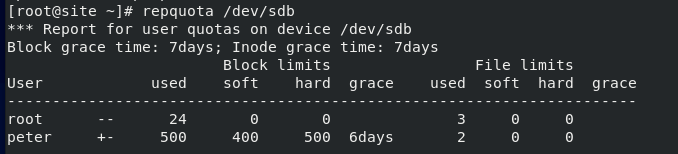
The created junk file cannot be created since it exceeded the limit on it stop on 512K.

Run *quota –v* command to show the quota values for peter.



It shows the quota values for peter that blocks at 500.

Run *repquota* to generate a quota report for the */opt* filesystem as a root.



The report generated with the information regarding soft limit and hard limit on */dev/sdb*, using *repquota* command.

**Task 2: Setting Up Disk Quotas Using Xfs (Root Filesystem)**

Enabling user quotas on root filesystem need to tell GRUB2 bootloader to load filesystem with quota support. This is only needed for xfs quotas on the root filesystem (any other filesystem doesn’t need this step).

The steps are:

1. Edit */etc/fstab* and add the option to turn on user quotas (*usrquota*) for the root filesystem (*/*) (Make */etc/fstab.bak* as recovery).

2. (Only for root filesystem) Edit */etc/default/grub* to add the root filesystem quota options in the GRUB2 bootloader config. Look for “*GRUB\_CMDLINE\_LINUX*” and add extra entry at the end of the line (within double quotes) as:

*rootflags=usrquota*

3. (Only for root filesystem) To read */etc/default/grub* file and rebuild GRUB2 bootloader configuration with the new options by running command:

*grub2-mkconfig*

4. (Only for root filesystem) Reboot system, then verify the new filesystem options by running mount command (xfs quotas doesn’t need to run quotacheck – it manages automatically).

5. Check quota accounting and enforcement are both enabled using (use *quotaon* to turn enforcement on):

*xfs\_quota -x -c state*

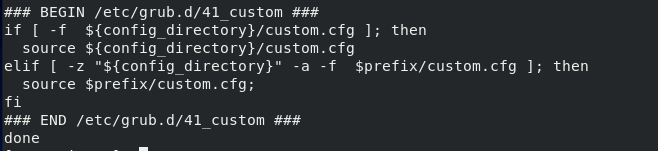
6. Use *edquota* to edit peter's disk quota. Check current block then set soft a little higher than the current usage and a hard limit higher again (If usage was 11223 blocks, set soft quota at 12MB and the hard quota at 13MB).



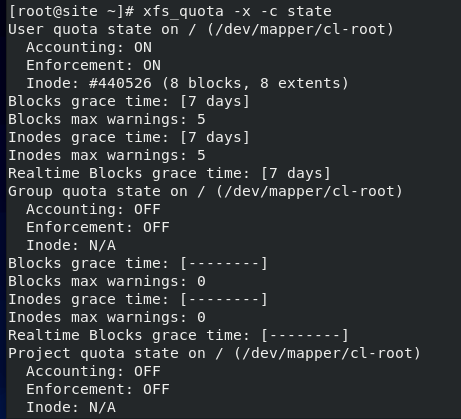
Add the *usrquota* for the root filesystem to add the option to turn on user quotas on *fstab* file.



Add *rootflags=usrquota* configuration inside *GRUB\_CMD\_LINE* on the grub file.



Run the *grub2-mkconfig* command to rebuild the bootloader.

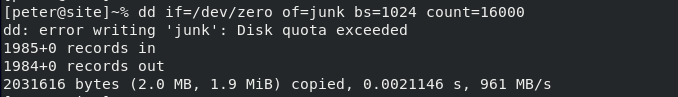


Both quota accounting and enforcement are enabled.



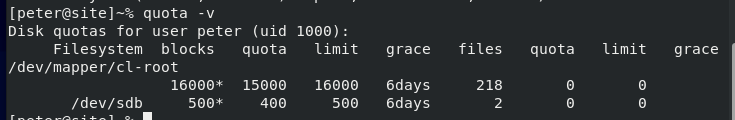
Set the soft limit to 15000 and hard limit to 16000 on peter’s disk quota using *edquota* command.

Test whether disk quotas are working by logging in as the user peter, and trying to store more than the hard limit. (Create a large file using dd command - *dd if=/dev/zero of=junk bs=1024 count=600*)



The junk file cannot be created since it exceeded the limit on the disk quota.

Show the quota values for peter by running *quota –v* command as peter.



Shows the quota values for peter that blocks at 16000.

Generate a quota report for the */* filesystem by command *repquota* as a root.



The generated information regarding soft and hard limit on root (*/*), using *repquota* command.

With xfs, generate a report with: *xfs\_quota -x -c report*



Similar report generate using *xfs\_quota -x -c* report command

**Lab 7c: Implementing Backups with Cron**

**Task 1: Using Cron**

Creating simple *cron* that will create empty file in */tmp* directory at a predefined time (touch command). Find further information in crontab file by using man 5 crontab (the "5" searched for section 5 - descriptions of system file contents).



The further information about the descriptions of system file contents.

Schedule cron job to execute a few minutes later than the current time (at least 2 minutes). Test whether cron job is working by looking for the empty file in the */tmp* directory.

There are two places to put cron job – */etc/crontab* (needs to specified the username under which the job will be executed) or in root user's own personal cron table (username is not used).

The crontab command is used to edit/view/remove a user's personal cron table:

*crontab –e (edit)*

*crontab –l (list)*

*crontab –r (remove)*

The editor used by default is *vi*, however, for the nano editor enter the command:

*export EDITOR=nano*

to set the value of the EDITOR environment variable equal to the name of the editor.

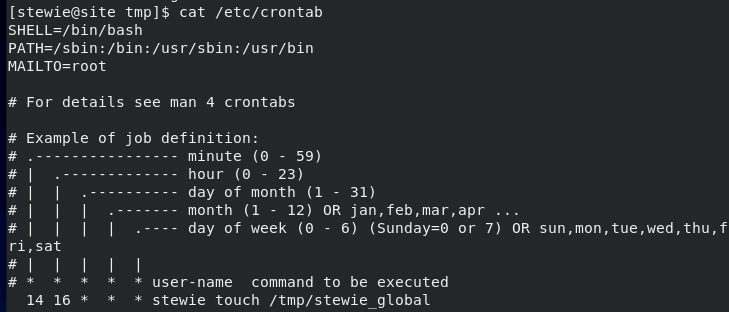
Create cron jobs in both using */etc/crontab* and root's personal crontab file.



Adding the configuration time on the */etc/crontab* file.



Adding the configuration time on root user’s personal cron table.

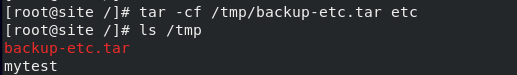


The content information of the crontab file.

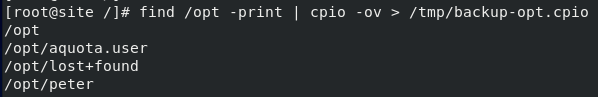
**Task 2: Backup and restore**

Backup two directories on system using two different backup tools.

1. Create backup of */etc* directory using *tar* and store in file named */tmp/backup-etc.tar*
2. Create backup of */opt* directory using *cpio*, and store in file named */tmp/backup-opt.cpio*

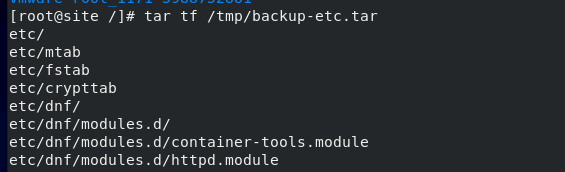


Creating backup using *tar* command for */etc* directory.

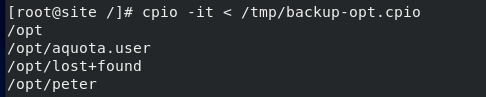


Creating backup using *cpio* command for */etc* directory.

Verify that backups created successfully by viewing a table of contents of each of the archives.

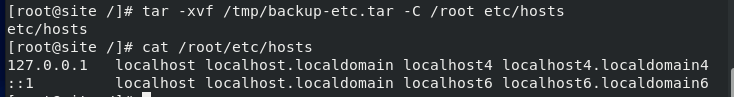


Shown the archives of the backup file using tar command to be verified.



Shown the archives of the backup file using *cpio* command to be verified.

Restore the file */etc/hosts* into your (root's) home directory (from tar backup) to test it.



Test the restoration into root’s home directory from tar backup.

**Task 3: Automating backups**

**Finally, combine what you have done in Tasks 1 and 2. Schedule your two backups (of** */etc* **and** */opt***) to be performed once a week using** *cron***. Check next week to verify that your backup is still occurring.**



Creating the configuration time on the */etc/crontab file.*



Creating the configuration time on root user’s personal cron table.

**Week 8**

**Lecture 8 Networked Filesystems: NFS & Samba**

NFS

* NFS is a **distributed** filesystem protocol which allowing user on client device to access files over the network.
* Build on the **ONC** **RPC** (Open Network Computing Remote Procedure Cell)
* IETF standard defined in a RFC
* **Stateless** NFS
  + Simplicity (simpler recovery)
  + Client keeps info about files and need to check its state
  + No consistency guarantees
* **Stateful** NFS
  + All state info stored on both client and server (when active)

NFS Server

* Packages -> nfs-ytils, rpcbind, and nfs4-acl-tools
* Start NFS sever daemons (nfsd, mountd) -> systemctl start nfs-server
* Configurationfiles
  + /etc/nfs.conf (main config file for NFS daemons and tools)
  + /etc/nfsmount.conf (NFS mount config file)
* Export system -> exportfs command
* Display current export list -> exportfs -s or -v
* Enable NFS in firewall
  + CUI -> firewall-config (enable NFS permanently)
  + CMD -> firewall-cmd –permanent –add-service=nfs; …-service=rpc-bind; …-service=mountd

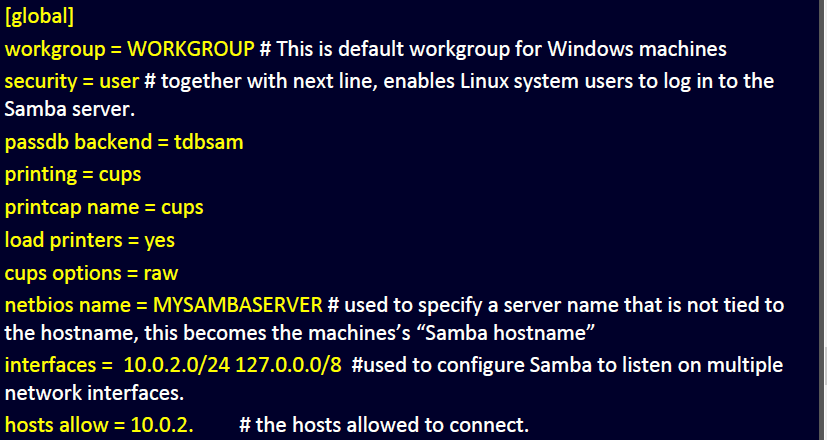
NFS Client

* Packages -> dnf install nfs-utils
* Client mount filesystem on a server -> showmount -e 10.0.2.1
* Create local directory for mounting filesystem as an ntf file
  + mkdir -p /media/perl
  + mount -t nfs4 10.0.2.1:/opt/per /mnt/perl
* To access remote directory -> add entry to /etc/fstab

SMB

* Server Message Block
* Is a client-server communication protocol for sharing access files and devices on a network.
* Open-source SMB implementation -> **Samba** (allow sharing across windows and Linux)

Samba Server

* Packages -> dnf install samba
* Service-Daemon
  + smb (d) - (samba server, user authen, share locking)
  + nmb (d) - (NetBIOS over TCP/IP)
  + systemctl start/restart/enable smb (nmb)
* Config file
  + /etc/samba/smb.conf
* Config global (as in left pict)
* Log and troubleshooting
  + testparm smb.conf
  + /var/log/samba/\*
* Firewall
  + firewall-config
* Samba security is user-based
* Maintain own authentication database by using tools to keep in sync
  + pdbedit -a -u newuser # (add user to database)
  + pdbedit -x -u olduser # (delete an account from database)
  + pdbedit -L # (list all user accounts)

Samba Client

* Typically, Windows machine (access by server’s IP)
* Sometimes Unix/Linux
  + If kernel has smbfs/cifs support
  + Permanent access in /etc/fstab

**Lab 8a: NFS**

**Task 1: Basic NFS Client and Server Functionality**

To setup NFSv4, several guidance to be followed are:

1. Ensure ens37 interface uses static network address (10.0.2.1/24).
2. Ensure the required packages - *nfs-utils*, *rpcbind*, and *nfs4-acl-tools* (install using *dnf* install *packagename*)
3. Create directory */share/IT\_Projects*. Modify the */etc/exports* file to export the directory as read-only. Only hosts on local subnet (10.0.2.0/24) should be able to mount this file system using NFS.
4. Turn on the *rpcbind* and NFS server service (daemons) using *systemctl*.

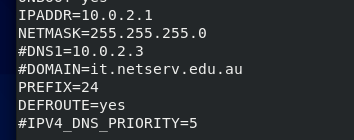
*systemctl start rpcbind*

*systemctl start nfs-server*

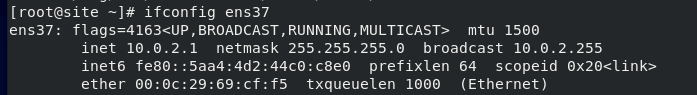
1. Use *systemctl* command to make sure these services for NFS are enabled.



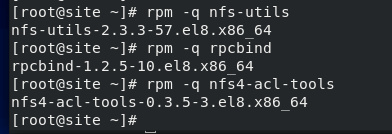
Set the static network by turning off the DNS on ens33 ethernet.



Set the static network address by 10.0.2.1 and turning off the DNS on ens37 ethernet.



Check the changes on ens 37.

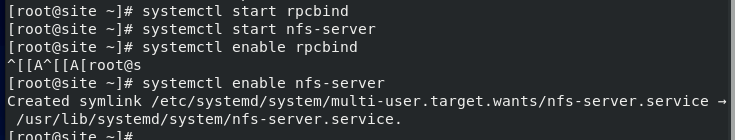


Check whether all required packages (*nfs-utils*, *rpcbind*, *nfs4-acl-tools*) are installed on system.



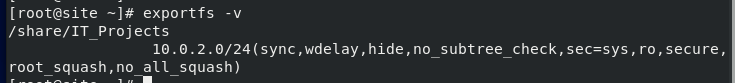


Create directory and export it as read-only by editing thet */etc/exports* file.



Turn on the *rpcbind* and NFS server services and enable both services.

Verify the */share/IT\_Projects* directory is exported (read-only|ro) using command *exportfs -v.*



Check whether the directory already exported as read-only using *exportfs -v* command.

***Test 1 – local mount***

Use same VM as an NFS client to access the exported directory. In this case, use the same machine to act as both client and server. Having two shell windows and assume one of them as server and client would be helpful.

Create directory called */mnt/projects* that will be used as a mount point. Then try mounting the exported */share/IT\_Projects* directory onto the */mnt/projects* mount point (client shell) by using command:

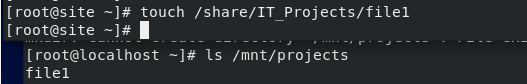
*mount -t nfs4 10.0.2.1:/share/IT\_Projects /mnt/projects*



Create a new directory as the mount point, and mount the first directory onto the new directory.

For testing:

1. Create file in the */share/IT\_Projects* directory on server shell (*touch /share/IT\_Projects/file*). Verify the “file1” is inside the */mnt/projects directory*.



Creating file1 as a new file on the IT\_Projects directory and it’s verified that the file1 is inside */mnt/projects* directory.

2. Create file in the */mnt/projects* directory on client shell (*touch /mnt/projects/file2*) and it shouldn’t be allowed, as the filesystem was read-only.



It’s also verified that file cannot be created on the */mnt/project* as it is only read-only file system.

Finish the testing by unmounting the directory by using command:

*umount /mnt/projects*



The directory is umounted and contain nothing on it.

**Task 2: Advanced Task – Setup a Separate Linux Virtual Machine to Access the Remote NFS Share**

Setup second Linux VM by either:

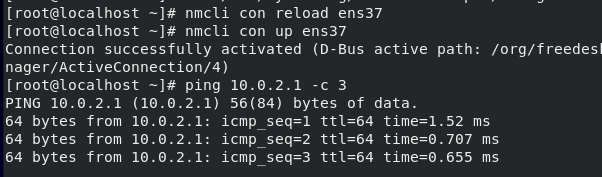
1. Create a second “linked clone” of the Linux VM (labs)
2. Make a complete copy of the VM folder (own device).

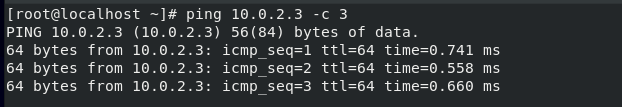
The second VM will be configured as an NFS client.

1. At start, set ens37 IP address to 10.0.2.3/24.



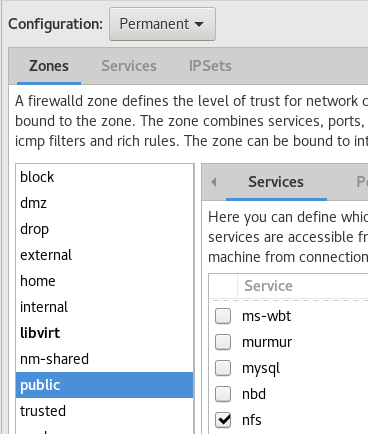
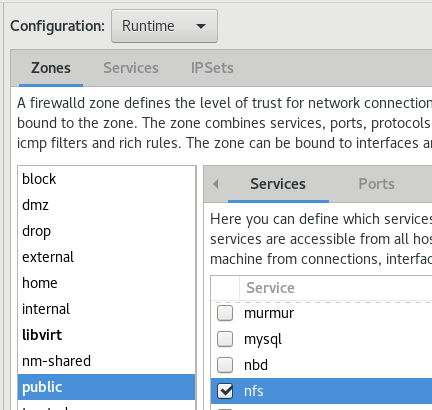
Set the IP to 10.0.2.3 since 10.0.2.2 already used.



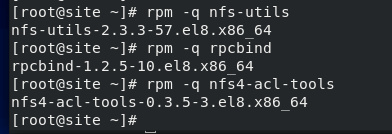


Then check the connectivity by pinging each VM.

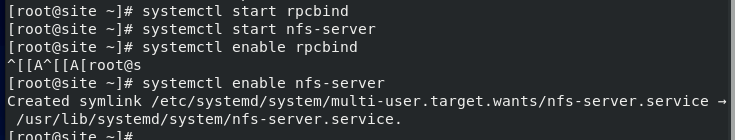
1. Turn on the nfs services on firewall on the server VM



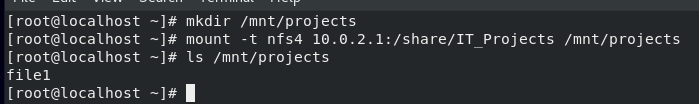
1. Install required NFS packages as before.



1. Start up the NFS related services using *systemctl* (NFS related services are enabled).



1. Create */mnt/projects* directory to use as a mount point. Mount the remote NFS exported directory */share/IT\_Projects* from server 10.0.2.1 onto */mnt/projects*. Verify whether the remote directory is accessible from client machine.

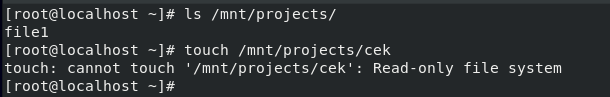


1. Append something to */etc/fstab* to automatically have the client machine mount the filesystem from the server during startup. Explore the fields in the */etc/fstab* file on the client.

*10.0.2.1/share/IT\_Projects /mnt/projects nfs default 1 1*



Reboot second VM and verify that it automatically mounts the remote NFS filesystem on boot and is accessible to remote NFS exported directory (read-only|ro).

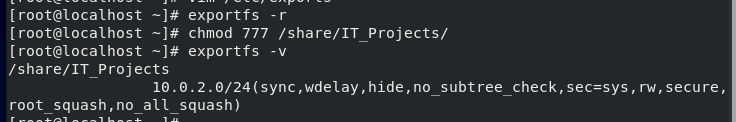


1. On NFS server, change filesystem to read-write by editing */etc/exports*. Tell server to reread the exports file (re-export) with the *-r* option (change UNIX filesystem permissions on */share/IT\_Projects* directory).

*exportfs -r (on NFS server)*

*chmod 777 /share/IT\_Projects (on NFS server)*

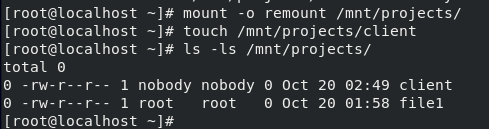




On client, ask to remount filesystem from server using remount option to mount:

*mount -o remount /mnt/projects (on NFS client)*

Verify that client machine able to create and edit files in */mnt/projects*.



It’s verified that client able to create and edit file in */mnt/projects*.

**If you are logged in as the root user, and create a file, what happens? Make sure you use “ls -l” to see the long directory listing. Who owns the new file you created?**

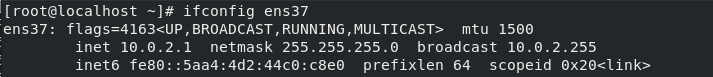
If logged in as root user, the file will state as nobody for both the user and group. It is happened because the directory has the *root\_squash* parameter that will increase security by preventing ownership of the root account. By this, the file created still owned by the root.

**Lab 8b: Samba**

**Task 1: Samba Basic**

Install and create a Samba share by:

1. Ensure ens37 interface uses a static network address (10.0.2.1/24).



The ens37 ethernet already set to 10.0.2.1/24.

1. Verify whether Samba package is installed (Use *dnf* to install the samba package)

*rpm –q samba*



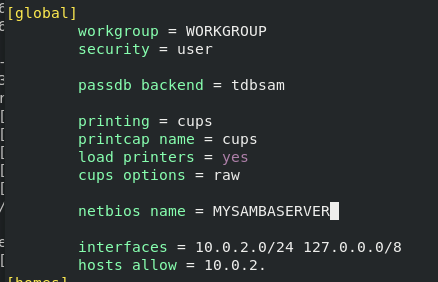
Samba package is already installed.

1. Backup the file */etc/samba/smb.conf* as */etc/samba/smb.conf\_backup* (Recover *smb.conf* if needed to).

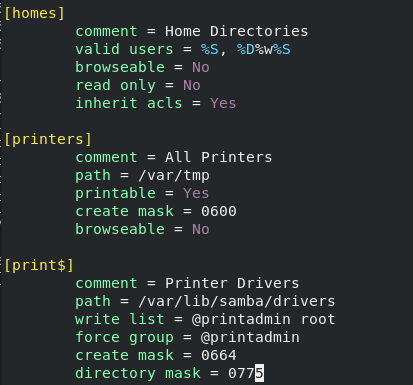


Make changes and configures to the file by:

* Workgroup (set to “WORKGROUP” - default for Windows machines)
* Netbios name (set to “MYSAMBASERVER” – machine's "Samba hostname")
* Interfaces (use “10.0.2.0/24” and “127.0.0.0/8”)
* Hosts allowance (set to “10.0.2.”)

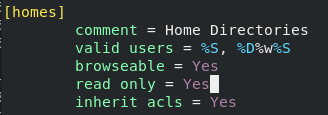


1. Check the different directories that going to be shared with Samba. Stick with *[homes]*, *[printers]* and *[print$]* (*[homes]* is default share of home directories).



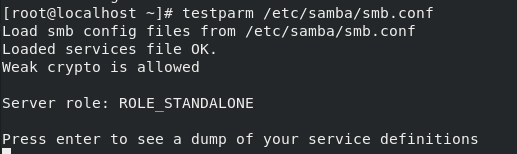
Shown the default directories shared with Samba.

1. Set browseable option to Yes and read only option to Yes (under the *[homes]*).



1. Verify the changes using testparm utility:

*testparm smb.conf*



All the changes are changed well without giving any error.

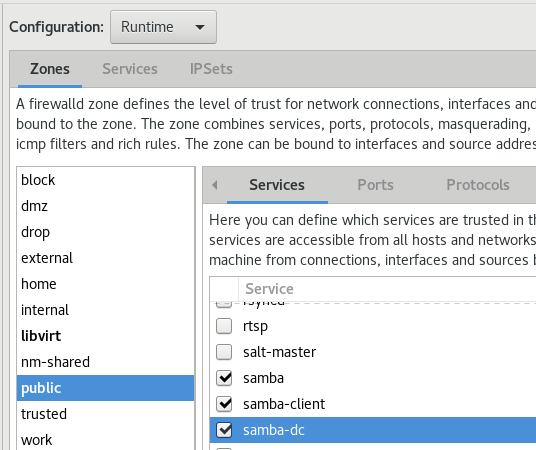
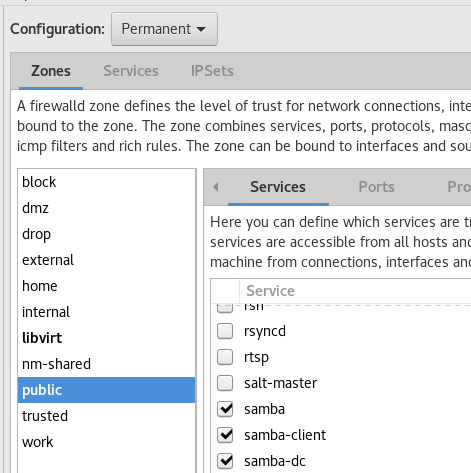
1. Allow Samba traffic by configuring the firewall and SELinux (currently protecting Linux system). Configure the firewall (since SELinux is disabled) by running:

*firewall-config*

Check location of ens37 interface (Default Zone – public).

Click on public zone -> Find samba -> Turn on checkbox beside samba -> See an option of “Configuration: runtime” (changed firewall for current runtime - not permanently) -> Change drop-down to “Configuration: permanent” -> make the change (turn on samba in public zone).

**\*\*Note: If this was a production system, change ens37 interface to “trusted” firewall zone and keep ens33 in the “public” zone.**

****

Allow Samba configuration on the Linux firewall.

1. User that requires access to Samba’s shared resource must be configured as a Samba user and assigned a password. Use *pdbedit* to set up Samba accounts, such as:

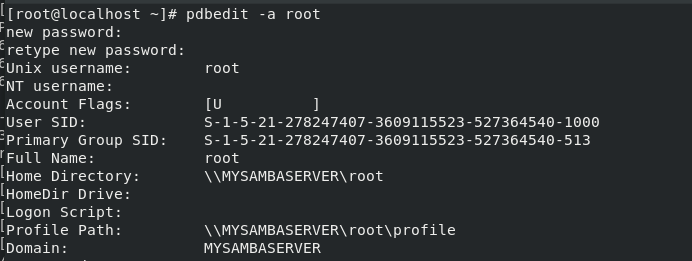
*pdbedit –a root*

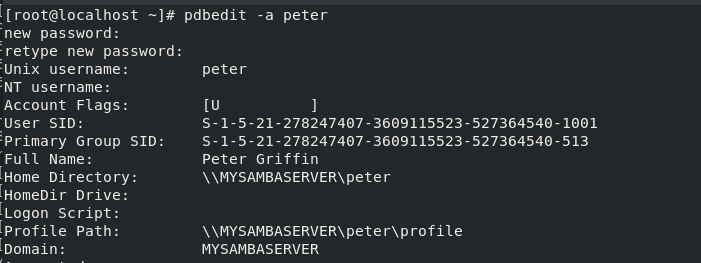
*pdbedit –a peter*

*pdbedit -L* (show list of current users)

*pdbedit -L -v* (show more verbose listing)

*pdbedit -x* (delete user from Samba password database)

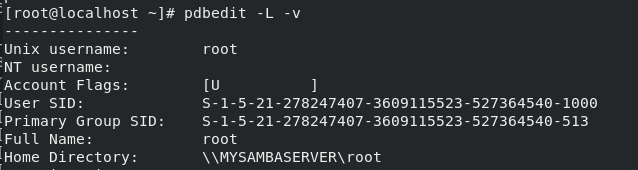




Create both user root and peter to give it access to Samba’s shared resources.

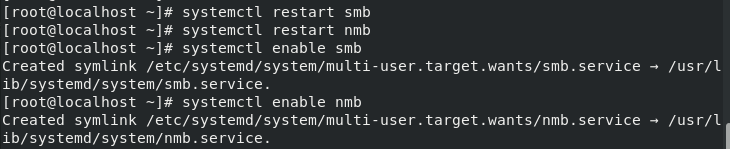


Check the current created user.



Use *pdbedit -L -v* to see more verbose listing.

1. Start up the smb and nmb services with *systemctl* (use systemctl to enable services).

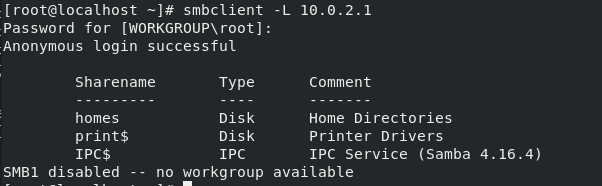


**Task 2: Testing Samba from Linux**

Test Samba from Linux VM by viewing list of shares with:

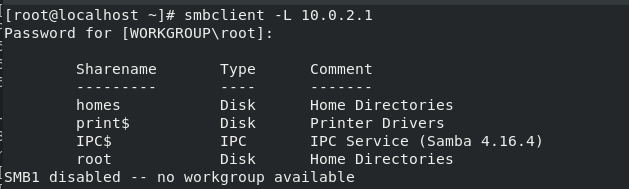
*smbclient -L 10.0.2.1*

There will be a password prompt, and try without any password (just Enter), to see list of shares that publicly (anonymously) visible.



Shown the Samba login anonymously that only show home directories.

Then try with Samba root password to see list of shares available to the root user.



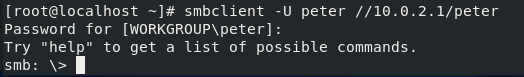
Shown the Samba login as root and it also shows the root directories.

**What is different between the two lists?**

Samba will show the root directories only if user input the root password. If there is no password given, it will login anonymously and only show home directories.

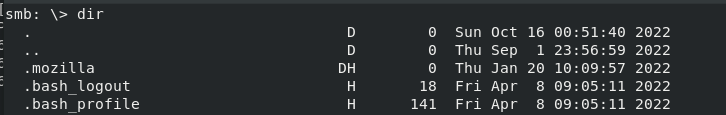
Try connecting to home directory for one of users (e.g. peter) by running:

*smbclient -U peter //10.0.2.1/peter*

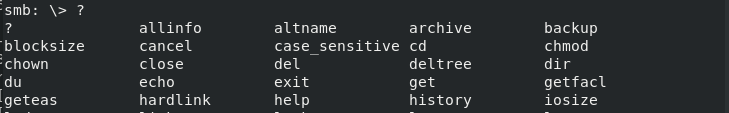


smbclient provides interface similar to command-line FTP.

Verify whether the contents of the user’s home directory could be seen through smbclient (*dir* command).Type ‘*?*’ to see list of smbclient commands.



The *dir* command shows the contents of the current directories only.

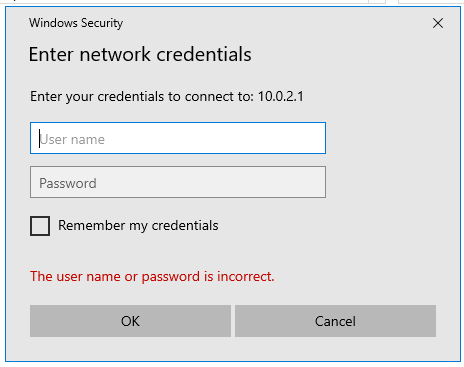
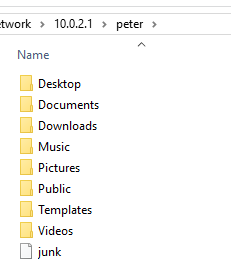


The ‘*?*’ command will shows help on each command.

**Task 3: Testing Samba from Windows Server**

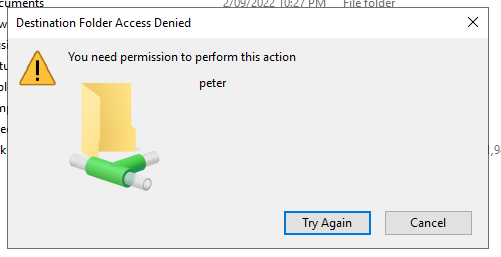
Try connect to home directory from Windows Server VM by opening Windows File Explorer and typing in \\10.0.2.1\peter in the address bar.

\*If it is cannot connect (error), troubleshoot and check by pinging the server. If successful, then it could be on Linux server, that the firewall is blocking incoming samba connections.



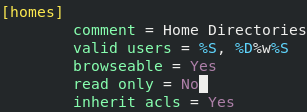
Connected to user peter’ home directory from Windows VM.

Once connected to the share, try creating a file or directory.



The access is denied since the permission before was set to read-only.

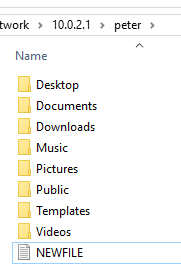
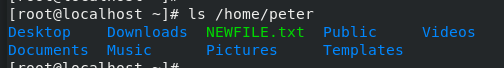
If “Access is denied” message appeared, make change in *smb.conf* (Linux server) to give users write access. Use *systemctl* to restart the smb service.



Change the read-only permission in the *smb.conf* file.



Restart both smb and nmb to update the permission.



New file successfully created after the update and it could be seen through Linux as well.

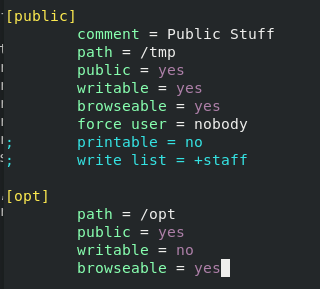
**Task 4: Creating Your Own Shares on Linux**

Once the Samba works, try configuring:

1. Share */tmp* directory of Linux server to Windows clients. The tmp share should be browseable, writeable and public, but files created by Windows users in */tmp* should be owned by the UNIX user “*nobody*” (not the actual user who is logged in). Read about Samba “*force user*” option by:

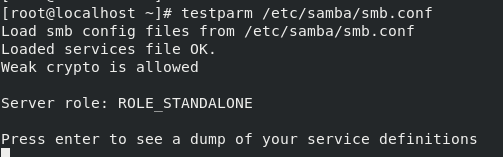
*force user = nobody*

1. Share */opt* directory of Linux server to Windows clients (should be read-only but publicly available and browseable).



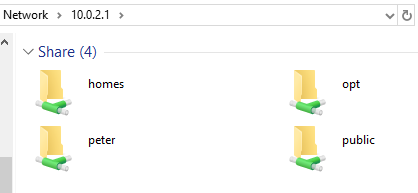
Add the configuration to able to share the */tmp* and */opt* directory.

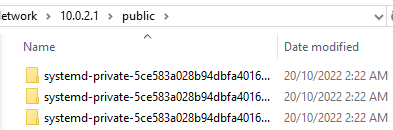
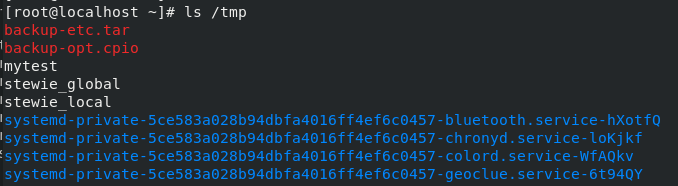
Use *testparm* to check the configuration.

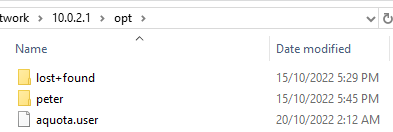


The new configurations are changed well without giving any error.

Test the access of both shares on Windows by browsing just: *\\10.0.2.1*







Both */opt* and */tmp* shares could be accessed on Windows and have the same file while checked on Linux.

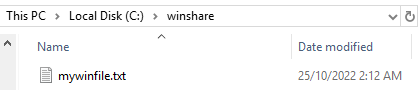
**Which shares do you see? Document all of this in your journal.**

Both */tmp* and */opt* shares could be seen on Windows VM. The */opt* shares also won’t allowed to create any new file (read-only) while the */tmp* shares has the user stated as nobody.

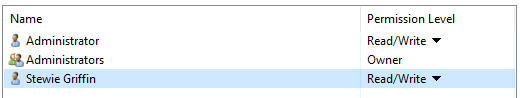
**Task 5: Creating Your Own Shares on Windows Server**

Since SMB is the native Windows approach for file sharing, it is just called as file sharing (Samba is name of Linux software package which implements Linux SMB server).

1. Open File Explorer on Windows.
2. Create folder c:\winshare -> create a file (e.g., “mywinfile.txt”).



1. Right-click and choose Give access to -> Specific people.
2. Type “stewie” as username of the person (username that previously created), then choose either “Read/write” or “Read”.



1. Click “Share” button to save changes -> see confirmation that folder is shared.
2. On Linux, run the command:

*smbclient -U stewie //10.0.2.2/winshare*

The IP address is for the Windows Server (Ethernet1).

Enter stewie’s password on the Windows Server not the Linux password.

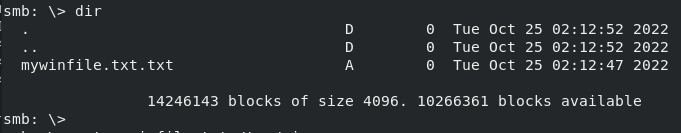
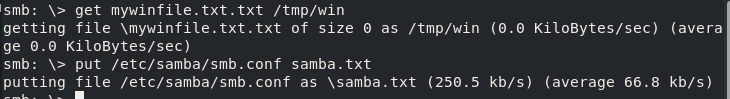


1. In Linux, check the file created on the Windows Server and try running these commands inside smbclient.

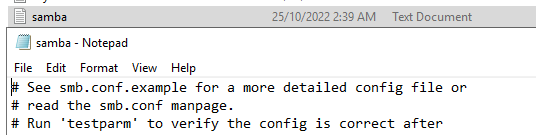
*dir*

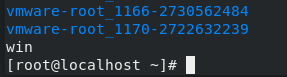
*get mywinfile.txt /tmp/win*

*put /etc/samba/smb.conf samba.txt*

Check the created file then copy the file on Windows to Linux and vice versa.





Both the copied file appeared on both Windows and Linux directory.

**Week 9**

**Lecture 9 Web Servers**

Scheme (URL in browser)

* DNS
  + Check cache
  + Query
* TCP/IP
  + Three-way handshake
* HTTP
  + Browser -> GET/POST request
  + Server -> Apache, IIS
  + Browser -> HTML Content

Apache vs Nginx

* Basic Architecture
  + A: Process driven (new thread each request)
  + N: Event driven (handle multiple request w/ one thread)
* Performance
  + A: Support dynamic content (server)
  + N: Cannot process dynamic content
* OS Support
  + A: All Unix-like, fully Windows
  + N: Almost all Unix-like, partially Windows

Web Services Stack

* LAMP stack
  + Linux (OS)
  + Apache (Web server)
  + Mysql (database server)
  + Php (application and dynamic web pages)
* LNMP -> Nginx/Tengine/OpenResty
* LNMPA -> MySQL/MariaDB/Percona
* LTMP -> PHP/Perl/Python

Apache Config

* Config directory
  + /etc/httpd/conf/httpd.conf (main)
  + /etc/httpd/conf.d/\*.conf (module config file)
  + /var/log/httpd/\* (log files)
* Web documents -> *DocumentRoot* directory
  + A: Support dynamic content (server)
  + N: Cannot process dynamic content

SSL (Secure Sockets Layer)

* Essential for encrypting network traffic
* Basic Idea
  + Server sends certify to client
  + Client optionally send certify to server
  + Client and server agree upon a ‘session key’ for data encryption

IIS vs Apache

* Internet Information Server (IIS) – (Windows’s own web server)
* Both offer same services
* IIS has windows authentication
* Provides application backend (ASP.NET, Azure)

**Lab 9a: Basic Web Server Configuration**

**Task 1: Verify Installation and Start the Web Server**

Check whether Apache web server is installed using command:

*systemctl status httpd*



Apache web server is installed and inactive at the moment.

Check that mod\_ssl is installed using command:

*rpm -qa | grep mod\_ssl*



*Mod\_ssl* is already installed.

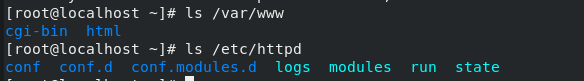
Check to see what files were installed using:

*rpm –qs httpd*

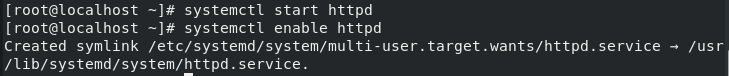
*rpm -qs httpd-filesystem*



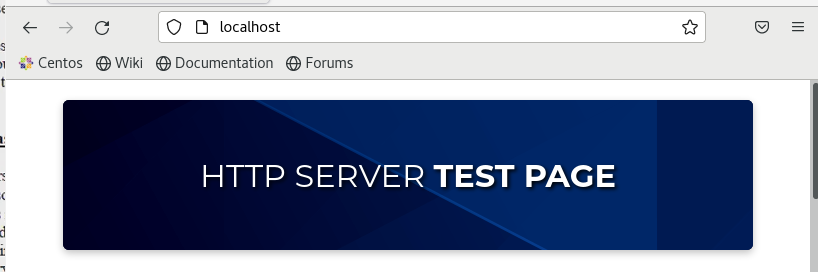
Both main web pages and configuration files are installed in */var/www* and *in /etc/httpd* respectively.



Start and enable httpd daemon by using *systemctl*.



Test server by opening Firefox and browse *http://localhost*

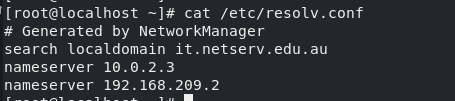


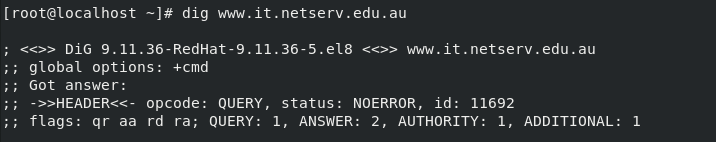
Apache server is already installed correctly.

**Task 2: Basic Web Server Functionality**

Ensure DNS is working for the domain (e.g., it.netserv.edu.au), and also the consistency in what IP address appears in the zone file, and on ens37 interface.

Confirm that the top “nameserver” line in */etc/resolv.conf* points to the nameserver.





Shown that DNS is working fine for the domain.

Edit */etc/httpd/conf/httpd.conf* file by modifying ServerName and note down the *DocumentRoot* directory (*DocumentRoot* is where HTML documents for website is placed).





Then restart web server by:

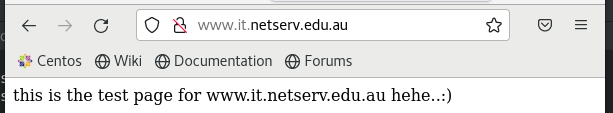
*systemctl restart httpd*



Create file named index.html in *DocumentRoot* directory (by default is */var/www/html*).



Test web server using *http://www.it.netserv.edu.au* in the browser.



The content successfully appears using above url.

**Common issues:**

* When attempting to start/restart httpd daemon, appear a message about unable to bind to port 80 -> probably because need to be root to start the web server.
* Cannot view index.html (404 error or 403 error) -> Check that file exists in the correct *DocumentRoot* and/or it has public read permission and directory has have public read/execute permissions.

Try execute *ps -ef | grep httpd*. Web service runs under userid apache -> all files in */var/www* directory need to be able to be read by userid apache.

* Getting FAILED error on startup or error 500 from the browser -> configuration error -> use *apachectl –t* to check configuration.

**Lab 9b: Configuring Apache SSL**

**Task 1: Configuring SSL**

The Apache installation already comes with SSL enabled -> remove some of configuration to understand the SSL configuration.

Set up both public and private encryption keys used for SSL.

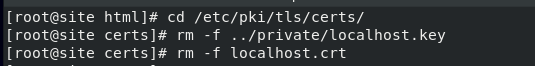
Create a private key for web server, including public certificate that web server will send out to clients (Public key will be contained inside a certificate).

Start by removing existing SSL server key and certificate using command:

*cd /etc/pki/tls/certs*

*rm ../private/localhost.key* (Server’s private key -> to sign certificate)

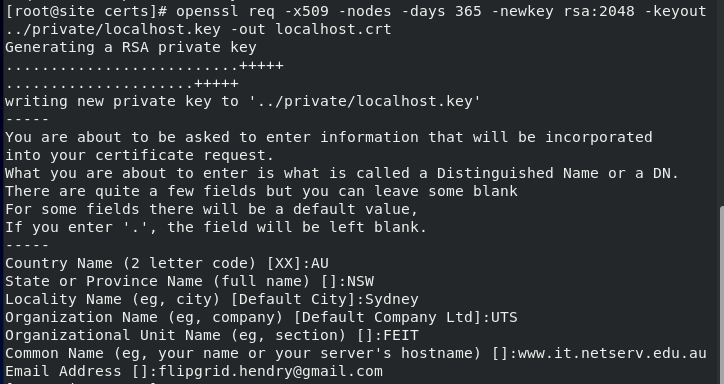
*rm localhost.crt* (Web server’s certificate)



Then generate new ones by inputting command (ensure located in */etc/pki/tls/certs* and enter the command in **one line**):

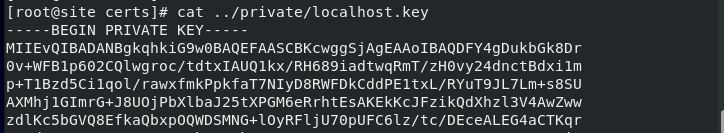
*openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout ../private/localhost.key -out localhost.crt*

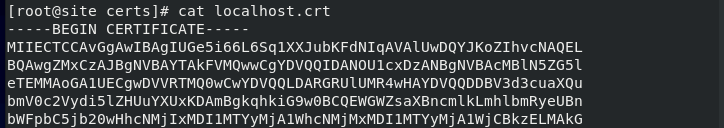
Answer all the questions and enter the web server’s DNS name for the Common Name (e.g., www.it.netserv.edu.au), else use a wildcard entry (e.g., \*.it.netserv.edu.au).



Successfully created new certificates.

It results two generated new files –> ../private/localhost.key (new private key for web server) and localhost.crt (new certificate). Filenames configurable in SSL configuration file */etc/httpd/conf.d/ssl.conf*.





The generated keys and certificate.

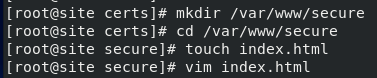
After generating the keys/certificates, specify new *DocumentRoot* for SSL web server (SSL web pages come from different location). SSL in Apache is configured as a module in */etc/httpd/conf.d* -> change to that directory and edit ssl.conf file.

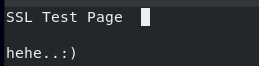
Find ssl.conf file for DocumentRoot directive (should initially be commented out -> SSL web server by default serve documents from same directory as normal web server).

Test by uncommenting the DocumentRoot directive in ssl.conf and make it to point to new directory (e.g., /var/www/secure). Create this new directory and put a simple index.html file inside.



Changing thte directive into /secure.





Creating index.html in the new /secure directory.

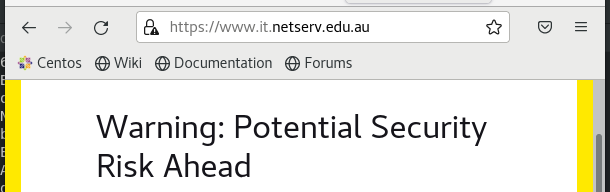
Then restart httpd using systemctl.



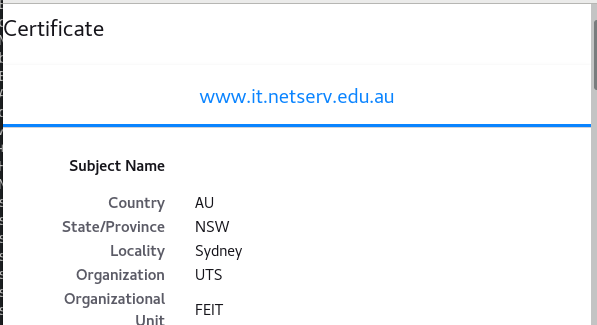
Test web server by accessing “https” URL:

*https://www.it.netserv.edu.au/*

It will appear the warning about the certificate (self-signed).



Choose “Advanced” -> View the certificate -> “Accept the risk and continue” -> see the “secure” web page show up.



The created certificate before.



The page successfully showed up

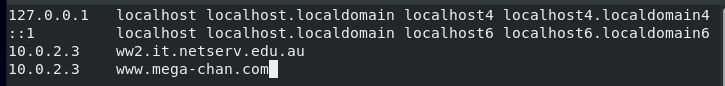
**Lab 9c: Virtual Hosting with Apache**

**Task 1: Virtual Hosting a Second Domain**

Set up virtual hosting to be able to host multiple web domains on the same computer (to save resources). Virtual hosting can be implemented in three different ways, IP-based, name-based and dynamic.

Name-based virtual hosting uses a feature introduced in HTTP version 1.1 to distinguish the target website. A client using HTTP/1.1 sends a Host header to identify the hostname that should respond to the request. Apache serves based on the contents of the Host header which arrived with the request.

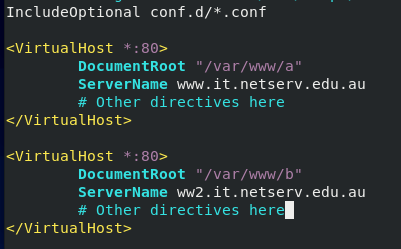
Create a second domain name in DNS server (need separate entry in named.conf file + create separate DNS zone file). An alternative is to define the web servers qualified domain names in /etc/hosts file of VM (will be uses to send HTTP requests to VM running Apache).



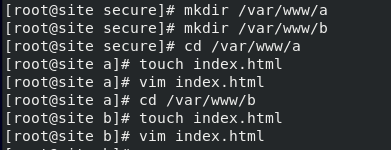
Add two web servers domain in /hosts file.

Change Apache configuration file to accept web requests for second (virtual) domain (config entries for virtual hosting are at the end of httpd.conf). Specify different DocumentRoot for virtual host (new directory with new index.html file).

Turn on NameVirtualHost directive in httpd.conf file (have <VirtualHost> entries for both main hostname and virtual hostname).

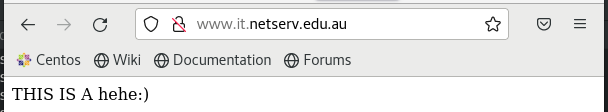


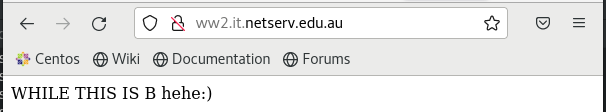
Set Apache to accept web request for second virtual domain.



Creating both new directories including its index.html files.

Verify that web requests to the original host, by using Firefox to retrieve one web page (from original DocumentRoot), while web requests to the second (virtual) host retrieve a different web page (from a different DocumentRoot).

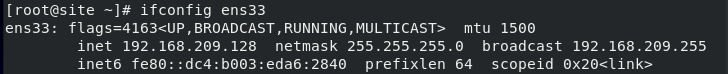




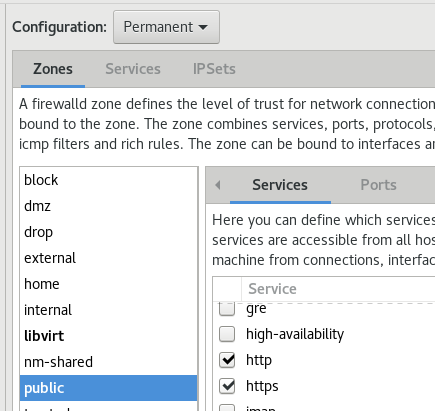
Both web requests to original host and virtual host successfully retrieved on the web page.

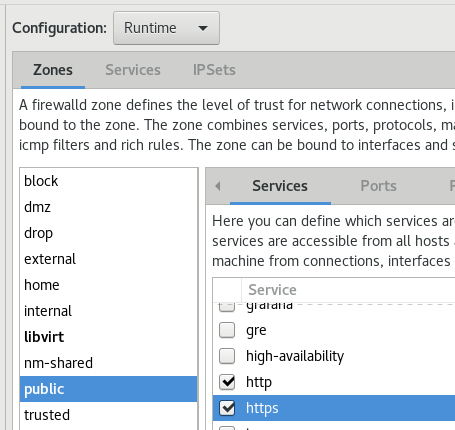
**Task 2: Testing from Your Host Operating System**

Try further testing by connecting from host machine (not VM) (need to know IP address configured on ens33 -> ifconfig or nmcli).

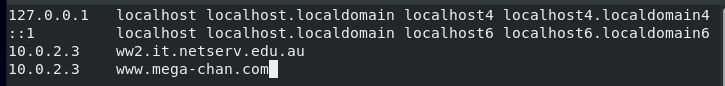


**Why doesn’t it connect?** (Hint: use firewall-config to add http and https into public zone).

It does need to set the firewall to allow http and https.



For superuser/Administrator privileges on host machine, add entry into “hosts” file for virtual servers (mapped to ens33 IP of virtual machine -> e.g., on Linux, edit /etc/hosts).



Already created on previous task.

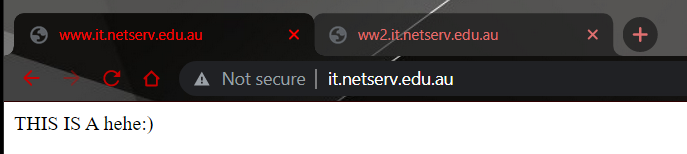
On Windows, edit C:\Windows\System32\drivers\etc\hosts (open Notepad as “Run as administrator”). The added line will look (change IP address and names):

192.168.209.128 www.it.netserv.edu.au www2.it.netserv.edu.au



Created the new line using Linux VM IP address and the following url.

From host OS, try to type those hostnames in the web browser and see the different virtual host sites.





\*\*Remember to delete the line above from hosts file on host OS when finished the testing.

\*\*Note:

Creating virtual host configurations on Apache server does not magically creating DNS entries for the host names (must have the names in DNS)

Server configuration:

# Ensure that Apache listens on port 80

Listen 80

<VirtualHost \*:80>

DocumentRoot "/www/example1"

ServerName www.example.com

# Other directives here

</VirtualHost>

<VirtualHost \*:80>

DocumentRoot "/www/example2"

ServerName www.example.org

# Other directives here

</VirtualHost>

**Lab 9d: Web Server (IIS) on Windows**

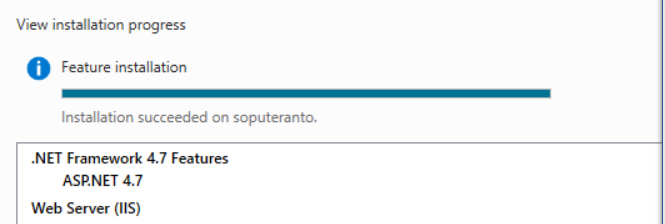
**Task 1: Install the Internet Information Services Role**

Windows has an integrated web server called Internet Information Services (IIS).

Add the “Web Server (IIS)” role on Server Manager -> Choose “Local Server” -> Find “Roles and Features->, choose “Add Roles and Features” (“Tasks” dropdown).

Add “Web Server (IIS)” role and its associated features (IIS management console), then specify that the following role services:

* 1. Common HTTP features
  2. Security –> Request Filtering, Basic Authentication, IP and Domain Restrictions, URL Authorization, Windows Authentication
  3. FTP Service
  4. Management tools –> IIS Management Console & Management Service

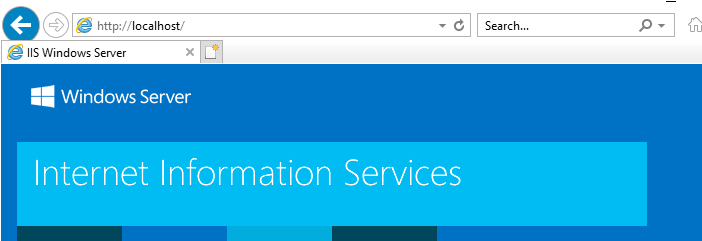


Installation succeeded.

Try check the status of the server.

Browse the configuration.

Test that the IIS server is running by opening http://localhost



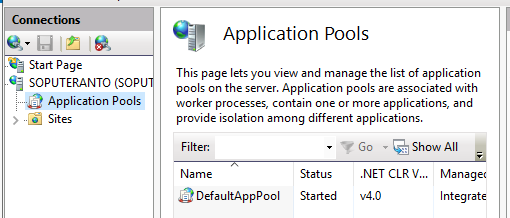
**Task 2: Manage the IIS Role**

IIS has a separate management panel from the default Server Manager panel.

Use Tools menu (on Server Manager) to choose “Internet Information Services (IIS) Manager”. Else, use the Start menu → Windows Administrative Tools → Internet Information Services (IIS) Manager.

View the start page and a “folder” labelled after the Windows hostname on IIS manager. Expand the folder and see the 2 sub-folders: Application Pools and Sites.

The DefaultAppPool allows to isolate and tune web server

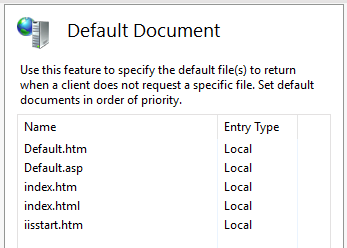


**Sites are obvious – these are FTP or Web sites, and we will modify the “Default Web Site” in this task.**

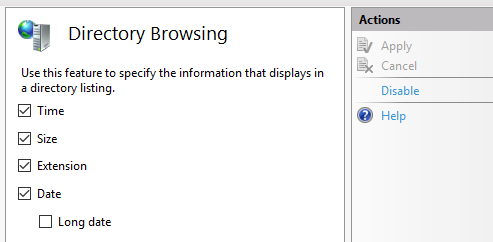
**Overall settings**

Overall settings are controlled by top-level server home. In IIS group, the icons by default will apply to all the sites controlled by this server.

Double Click on Default Document (list of filenames which server will display first).

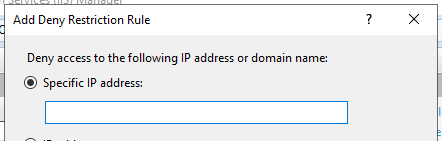


Allow directories to be browsed via Directory Browsing applet (will turn on/off the option to generate a default directory list if Default Document is not found).

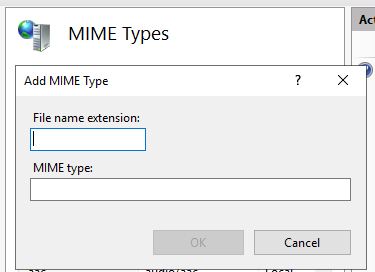


When getting certain HTTP errors, override the default page generated through the error (404 page not found is preferred).

Add restrictions based on clients IP address via the IP Address and Domain Restrictions applet.



Create and update MIME types.



Then, enable the web management service feature and add users to authorized manager list.

**Site specific settings**

See a similar control panel to the overall panel on the Default Web Site (overrides the server defaults).

Some of the actions available on the panel:

• Explore –> opens folder which contains default document root directory

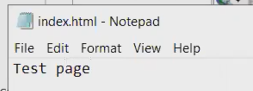
• Bindings –> allowing to choose the protocol (e.g. http), IP address and Port (80) to bind to.

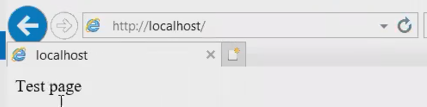
• Basic Settings –> shows default document root (%SYSTEMDRIVE%\inetpub\wwwroot)

• Advanced Settings –> allowing to change the site name, physical path etc

**Creating web pages:**

Create an index.html file in default document root directory and test it.





Index file is successfully created.

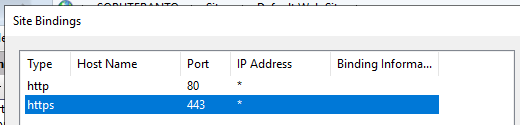
\*\*Windows ‘File Explorer’ hides file extensions by default. Unhide them by clicking ‘This PC’ icon -> choose View → File name extensions (turn on the checkbox).

**Task 3: SSL Web Sites**

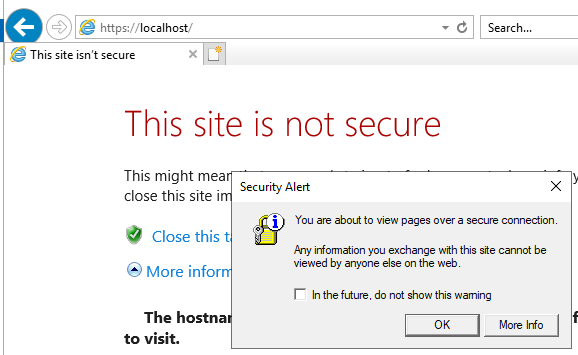
Enable SSL for the web site by selecting bindings action on IIS management panel (default web site).

Add a Site Binding -> choose https: type -> port will change to 443.

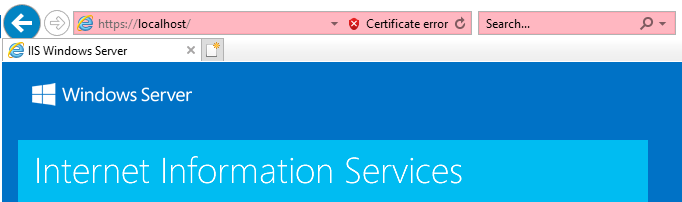
Add an SSL certificate –>choose the default one (dropdown of available certificates -> IIS installer is the one generated).



Test via https://localhost



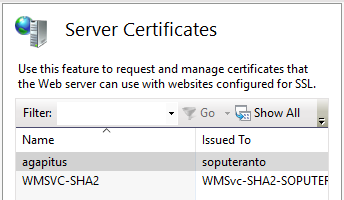
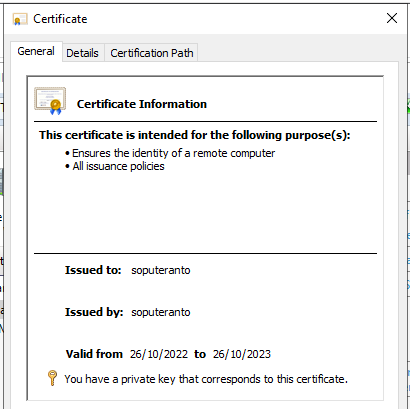
Accept the accept/deny the server certificate prompt (“This site is not secure”).



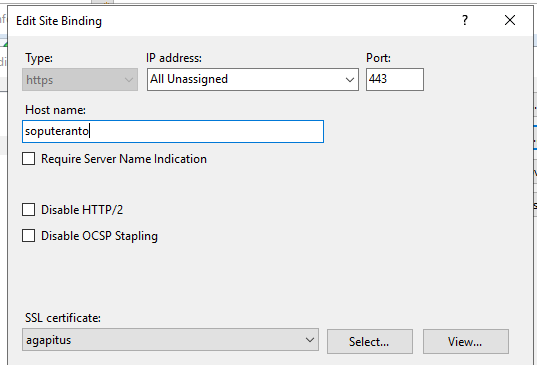
**Self-Signed Certificate**

Create own server certificate by locating the Server Certificates icon (return to IIS server home management panel).

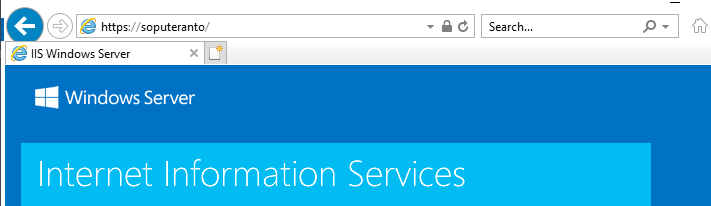
Select “Create a Self-Signed Certificate” -> follow the wizard -> store in the “web hosting” certificate store.



Return to default web site menu -> edit the existing https binding -> choose new certificate -> test again.



Enter the DNS hostname as in the certificate (e.g., https://www.netserv.edu.au).

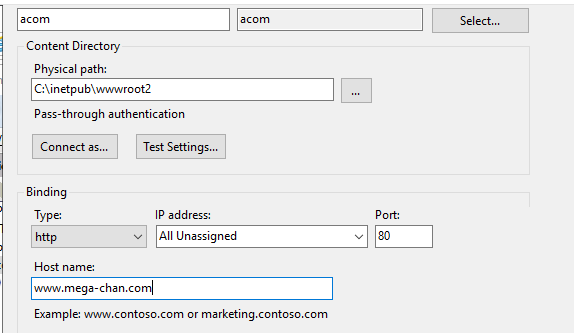


To create a real SSL server certificate, visit a certification authority (CA) such as Verisign.

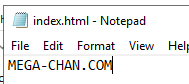
**Task 4: Virtual Hosting**

Create virtual hosting (similar to Apache) by adding a new Site from the Sites menu in IIS Manager.

Choose Sites → Add Web Site action -> give a name and a physical directory path (create new directory e.g., c:\inetpub\wwwroot2)



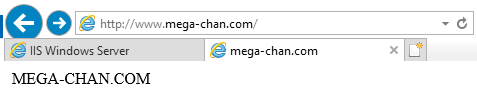
Then assign a host name (an Alias in DNS | e.g., add www2.netserv.edu.au to DNS -> this alias would be a CNAME that points to real server A record such as site.netserv.edu.au).



Created website index.html file.



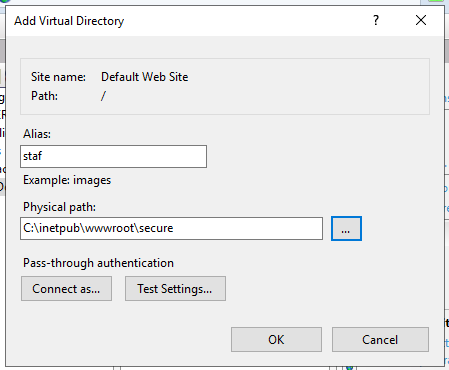
Add the host address to the hosts drivers.

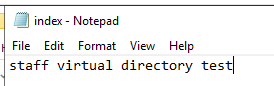


**Other tasks**

Right click mouse on the site (e.g., Default Web Site) and add virtual directory (a mapping of real directory to a “Alias” or virtual directory).

Create a virtual directory called “staff” which maps to “c:\inetpub\wwwroot\secure” and create a different index.html in this physical directory.





Test this by opening http://localhost/staff (should open up c:\inetpub\wwwroot\secure\index.html).



**Week 10**

**Lecture 10 Printing and Miscellaneous Security Services**

Unix Printing

* One of three different packages
  + BSD LPD printing system
  + LPRng printing system
  + CUPS (Common Unix Printing System)
* Printing methods
  + Print using lpr
  + Document is spooled in /var/spool/cups
  + Printing daemon (cupsd) sees new doc to print
* Printing files
  + Use lpr command
    - lpr -Pblue-draft myfile.txt
    - lpr -Pblue-clean -# 5 myfile.ps
  + Non-BSD system use lp/cups
    - lp -d blue-draft myfile.txt
  + Check the queue
    - lpq -Pblue-draft
* Printing configuration
  + LPD best for older systems or mixed Unix envi
  + IPP (Internet Printing Protocol) best for Linux-to-Linux printing
  + Samba best if large number of Windows client access printer

Unix Security

* Security administration tasks
  + Discover open ports
  + Find setuid/setgid files
  + Setup pwd aging (chage)
  + Verify packages (rpm -Va)
  + Update packages (yum update)
* Host security
  + Turn off network service
  + Configure syslog (/etc/rsyslog.conf)
  + Setup shadow password
  + Setup email alias for root’s email (/etc/aliases)
* Discovering open ports
  + netstat -ap (show all ports in use and the process ID)
  + nmap -sT localhost (network scan on localhost for TCP ports)
* su
  + used to launch superuser login shell
  + asks for root password
  + ‘all or nothing’ access
* sudo (/etc/sudoers)
  + run individual command as root
  + asks for user’s own password
  + can restrict

Windows Security

* Host security
  + physical
* Operating system
  + Strong authentication
  + File and folder permissions (local access rights)
  + Remove default administrator access (UAC and elevation)
  + Audit via event viewer
* Network security
  + IPSec tunnelling support (using industry standard IPSec)
  + Windows Firewall
  + Network Access Protection

**Lab 10a: Printing in Linux**

**Task 1: Set Up Printing on Linux Using CUPS**

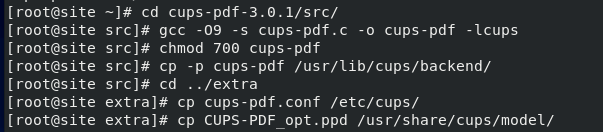
**Install cups-pdf**

Install the Common Unix Printing System (CUPS) which replaces the older LPD daemon in Unix, and also install PDF virtual printer cups-pdf and use the PDF viewer, evince.

Install cups-pdf package for virtual PDF printer function.

Run all ten commands one after another. Each line is a new command.

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/



Restart cups: systemctl restart cups

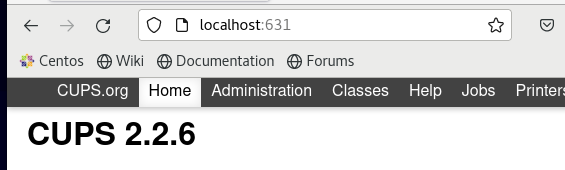


**Creating a virtual printer queue**

Use VMWare Virtual printer setting to link the existing printers on host to Linux virtual machine (just create a virtual PDF printer for now).

1. Enter CUPS administration panel via starting a web browser on Linux server

http://localhost:631



\*\*Note that there are 3 main tabs to be aware of:

* + 1. Administration -> manage printers, classes and jobs; configure and view server settings
    2. Jobs –> view print jobs
    3. Printers –> view printer queues

1. Select the Administration tab to set up our printer.
2. Choose Add Printer

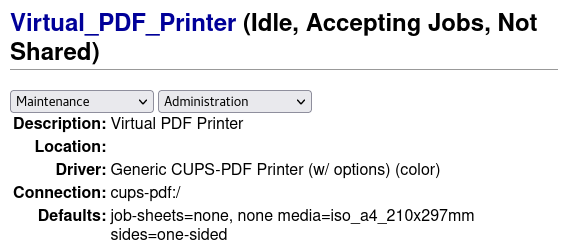
There should be a local printer called “CUPS-PDF (Virtual PDF Printer)”. Other types are e.g., LPT #1, Serial port #1

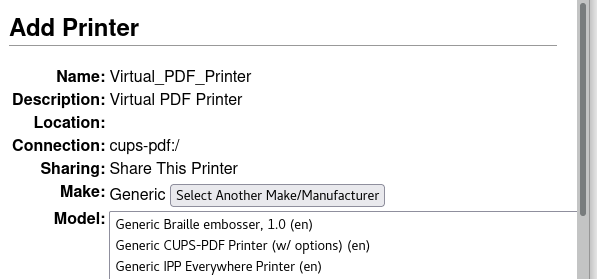
1. Select the “CUPS-PDF” local printer & press CONTINUE

Give the printer a name (e.g., Virtual\_PDF\_Printer), description and location.

1. Choose the sharing: ‘Share this printer’ option & press CONTINUE

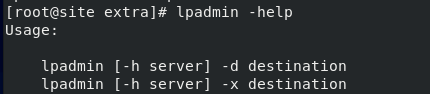
Choose the Make of printer: use generic → Generic CUPS-PDF Printer (w/options) (en).

1. Select “Add Printer”



In Linux, it also able to set printers up from the using command line:

lpadmin



**Task 2: Printing on Linux**

From browser, try printing current web page via File → Print, or press Ctrl-P

CUPS-PDF places the resultant file into folder /var/spool/cups-pdf/$USER by default.

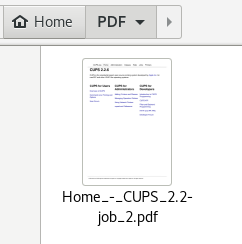
To see if the file printed, look in /var/spool/cups-pdf/root (as root). Use evince command to view PDF file.



Modify the folder by changing the out parameter in the

/etc/cups/cups-pdf.conf





**Printing from the command line**

Set default printing destination via the PRINTER (or LPDEST) environment variable.

export PRINTER=Virtual\_PDF\_Printer



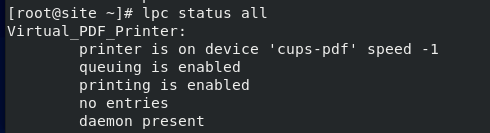
Else, use lpoptions command to set printer destination and other printing options. Do this in .profile to set the printer persistently

lpoptions –d Virtual\_PDF\_Printer –o PageSize=A4



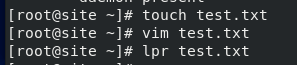
Check CUPS directly (If don’t know the printer name) via the:

lpc status all



To print in Linux, use the lp or lpr commands

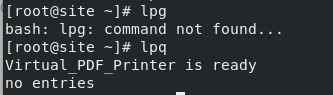
lpr test.txt





The result of the printed text in Linux.

Use lpq to view the status of a print queue or lprm to remove a print job.



**More tricks with CUPS**

CUPS other features :

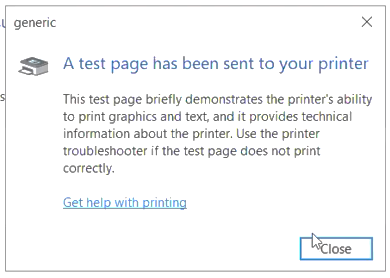
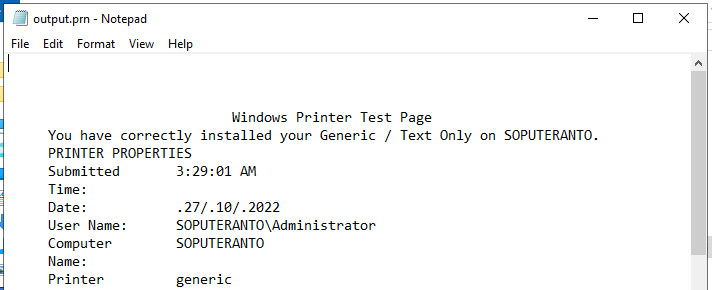
1. “Classes” (creating a pool of printers)
2. Scan network for other network printers and broadcast itself as a shared printer.
3. Communicate with Windows print servers.

**Lab 10b: Printing in Windows**

**Task 1: Create and Share a Printer on Windows**

On Windows Server, go Start → Windows Settings → Devices → Printers and scanners

* + - Add a new printer (automatically searching for a connected printer).
    - Choose “The printer that I want isn’t listed”
    - Choose “Add a local printer or network printer with manual settings”
    - Choose: Create a new port -> select “Local Port”
    - Enter c:\output.prn (saving the print file just for testing purposes) – (normally select the appropriate printer port - COM1, LPT1, USB1).
    - Install a printer driver: Choose the Generic manufacturer -> choose the “Generic/ Text Only”
    - Complete the wizard install steps.
    - Share this printer on the network (rename it to “generic”)
    - Print a test page and check the file (use notepad or the type command to view the contents)



The created printer on Windows.

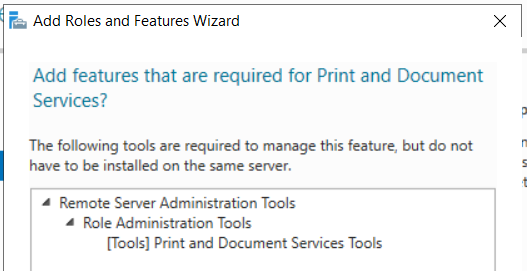
**Task 2: Set Up Windows as a Print Server**

Main objectives of adding printer server role to Windows Server are:

1. Centralise all shared printers onto a single server (including networked printing devices and secondary print servers).
2. Create a “pool” of printers.
3. Centrally manage security access to printers (via Active Directory or Windows security).
4. Deploy printer drivers for different windows clients (automatically download an appropriate printer driver).
5. Emulate Unix-style printing (for Unix, Linux, MacOS clients)
6. Centrally manage the printers from Windows management console

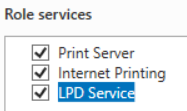
“Print and Document Services” role allows Windows Server to act as a printer server.

In Server Manager, choose Local Server → Roles and Features → Tasks → Add Roles and Features → Add “Print and Document Services” role.



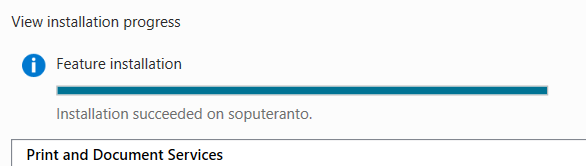
Install all 3 role services:

* + - Print Server
    - Internet Printing (enables IPP – creates a subdirectory on IIS called “/printers”)
    - LPD Service (emulates Unix printing on port 515)



\*\*Have IIS enabled in order to use Internet Printing.

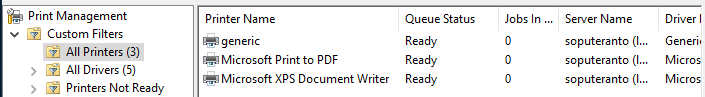
The wizard will automatically detect existing printers and begin to manage them.



**Task 3: Managing Printers**

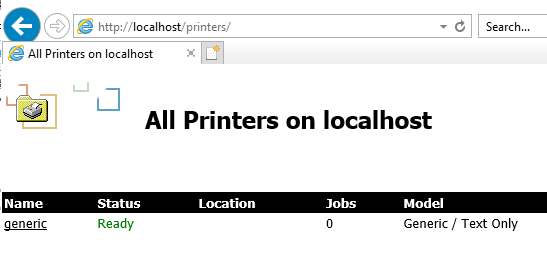
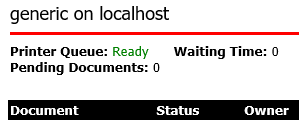
In Server Manager, use Tools menu → choose “Print Management”. Else, find it via Start → Windows Administrative Tools → Print Management.

View the printers (filter down to “All Printers” or use Print Servers folder to view printer and queues).



Use browser to check and explore the options by entering:

http://localhost/printers



Shown the printer checked through website.

**Task 4: Printing from Unix**

Use CUPS to direct printing to Windows Server.

There are three ways to set up CUPS to print to a Windows Server

* + - Use Samba subsystem in Linux to print to a Windows Printer server via SMB protocol
    - Use the LPD protocol
    - Use the Internet Printing Protocol

1. Logon to Linux server
2. Connect to CUPS admin panel via browser on Linux server

http://localhost:631

1. Use Administration Tab & the Add Printer wizard to add a printer.

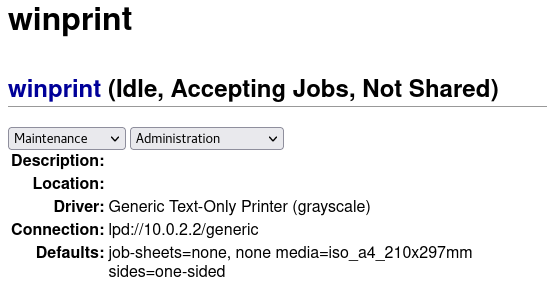
Choose “Other Network Printers” -> choose “LPD/LPR Host or Printer”

\*\*Do not choose “Internet Printing Protocol (ipp)” (windows does not listen on standard IPP port)

1. Choose the appropriate LPD URL for the Windows Server.

Should be: lpd://10.0.2.2/generic

1. Define the name (e.g.: winprint), description, location, sharing, make and type of printer (make this Generic / “Generic Text-Only Printer (en)”) and add the printer.
2. Check the Administration tab to view the completed printer definition.

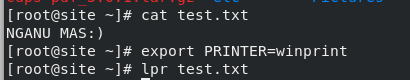


Added printer (was created from Windows) into Linux VM.

**Test Printing**

Check the printing to this server via

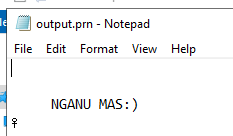
export PRINTER=winprint \



Use the JOBS tab in the CUPS interface to view the completed (or active) job.



Check that the print actually occurred, by viewing the output file of “generic” printer (eg: type c:\output.prn) on the Windows Server.



Viewed the printed output that is correct.

**Reflection**:

First week of the second term of the Journal, I learnt about how to set the network up statically on both client and server configuration. It is a clear new thing for me especially for the Linux and it felt quite complicated at the beginning. Most of my issue on this week is typing problem that often mistype that code. This leads to unsuccessful in running the code, especially in the ‘dig’ part that I often need to fix my command.

On the week 7, here I learnt more about the filesystem and all-around storage part. This week learning material is quite interesting as I could able to manage the disk including its permission by my own. The materials are not very hard, however, there is some content that need to change from the grub (root) file. Again, on this part I obtain a similar issue that I mistype on the grub file. This week issue is quite trouble some since a typo in grub file leading to a black screen after booting. Fortunately, I would be able to fix this part by recovering the grub file.

In the following week, the task is not plenty and is quite easy. The labs about NFS and samba is actually the upgrade and advance part from last week. There is no major problem, and it went well. It gives me new things to learn as especially about Samba that I never knew before.

Hereafter, week 9 is the most interesting part for me. This week I could learn a lot about configuring web server including Apache, virtual hosting, and IIS. The learning material is quite easy and fun as I could do more a lot thing in here. It really clarifies me about around the web server hosting and I really love it. The only issue occurs is my negligence in turning on the file extensions and wondering why the file type is not changing.

The last week 12 material is quite fast and easy. It learns about printing things that mostly is not complicated. It is a simple thing to be learned and I found no issue in learning this part. Everything went smoothly on this week.

**Reference List**

32520 Systems Administration, lecture 6, week 6: Domain Name System (DNS), lecture PowerPoint slides, viewed 26 September 2022, https://canvas.uts.edu.au/courses/24955/pages/lecture-6-materials?module\_item\_id=1072114

32520 System Administration, lecture 7, week 7: Filesystems and backups, lecture PowerPoint slides, Viewed 1 October 2022, https://canvas.uts.edu.au/courses/24955/pages/lecture-7-materials?module\_item\_id=1072116

Lansweeper 2020, *How to install IIS (Internet Information Services),* viewed 17 October 2022, https://www.lansweeper.com/knowledgebase/how-to-install-iis/

The Apache Software Foundation 2020, *VirtualHost Examples*, viewed 17 October 2022, <http://httpd.apache.org/docs/current/vhosts/examples.html>

Whitaker, J. 2020, Linux NFS Server: How to Set Up Server and Client, viewed 17 October 2022, https://cloud.netapp.com/blog/azure-anf-blg-linux-nfs-server-how-to-set-up-server-and-client#:~:text=What%20is%20Linux%20NFS%20Server,running%20the%20NFS%20server%20component.