

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

# 31338 Network Servers

Project –Network Design,  
Implementation, and  
Troubleshooting

Benjamin Lee 13248113



## Table of Contents

Task 1: Create partitions (10).....	2
Task 2: Set up static networking (5) .....	6
Task 3: Set up NFS service (5) .....	10
Task 4: Set up DNS server (10) .....	12
Task 5: Set up Web server (10) .....	12

## Task 1: Create partitions (10)

In this task we need to add a second hard disk on both virtual machines. We will create the following partitions as seen in the diagram below and mount them accordingly. The partitions will be **MBR** formatted.

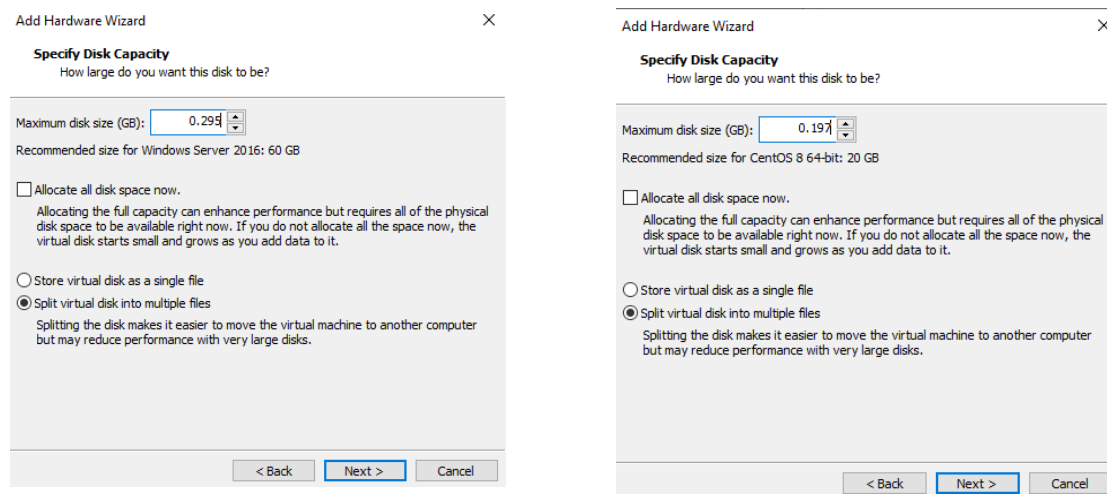
	Linux	Windows Server
<b>Disk</b>	<code>/dev/sdb</code>	<code>Disk 0</code>
<b>Partition 1</b>	Primary <b>swap</b> , 100MB	Primary partition 200MB, FAT32
<b>Partition 2</b>	Primary <b>EXT4</b> , 100MB	Primary partition 100MB, NTFS
<b>Mount partition 1 as:</b>	<code>swap</code>	<code>S:</code>
<b>Mount partition 2 as:</b>	<code>/www</code>	<code>H:</code>

For this task we may need to create a second hard disk for each VM.

For this we can select our Windows VM in VMware Fusion. To create a virtual hard disk we select it and Edit our VM settings. We should see an Add button.

1. After, select **Add → Hard Disk → SATA for windows VM → Create New Virtual Disk**
  - a. I tried NVMe but it creates a Disk 1 instead, so SATA seems to be a better solution
2. After select **Add → Hard Disk → SCSI for linux VM → Create New Virtual Disk**

**Remember:** When allocating disk capacity, VMware adds a few Megabytes when creating the virtual hard drive, so we minus a few MB as seen below.



### Windows:

In cmd as administrator, we can now use the **diskpart** command to partition and format our second hard disk.

```
C:\Users\Administrator>diskpart

Microsoft DiskPart version 10.0.17763.1

Copyright (C) Microsoft Corporation.
On computer: BENLEE

DISKPART> list disk

Disk ###  Status      Size      Free      Dyn  Gpt
-----  -
Disk 0    Online     301 MB    301 MB
Disk 1    Online     60 GB     0 B      *

```

We now use the following commands to partition and format second hard disk in order:

1. Diskpart
2. List disk (List disks), should have Disk 0 nearly 300MB, if it only shows Disk 1 then something is wrong. To fix this problem I used SATA instead
3. Select disk 0
4. Attributes disk clear readonly
5. Convert mbr
  - a. This command To convert disk to MBR format
6. Create partition primary size=200
7. Select partition 1
8. Format FS=FAT32
9. Assign letter=S
10. Create partition primary size=100
11. Select partition 2
12. Format FS=NTFS
13. Assign letter=H
14. Exit

```

Copyright (C) Microsoft Corporation.
On computer: BENLEE

DISKPART> list disk

   Disk ##  Status      Size   Free   Dyn  Gpt
   -----  -
   Disk 0    Online       302 MB  302 MB
   Disk 1    Online      60 GB   0 B   *

DISKPART> select disk 0
Disk 0 is now the selected disk.

DISKPART> attributes disk clear readonly
Disk attributes cleared successfully.

DISKPART> convert mbr
DiskPart successfully converted the selected disk to MBR format.

DISKPART> create partition primary size=200
DiskPart succeeded in creating the specified partition.

DISKPART> select partition 1
Partition 1 is now the selected partition.

DISKPART> format FS=FAT32
100 percent completed
DiskPart successfully formatted the volume.

DISKPART> assign letter=S
DiskPart successfully assigned the drive letter or mount point.

DISKPART> create partition primary size=100
DiskPart succeeded in creating the specified partition.

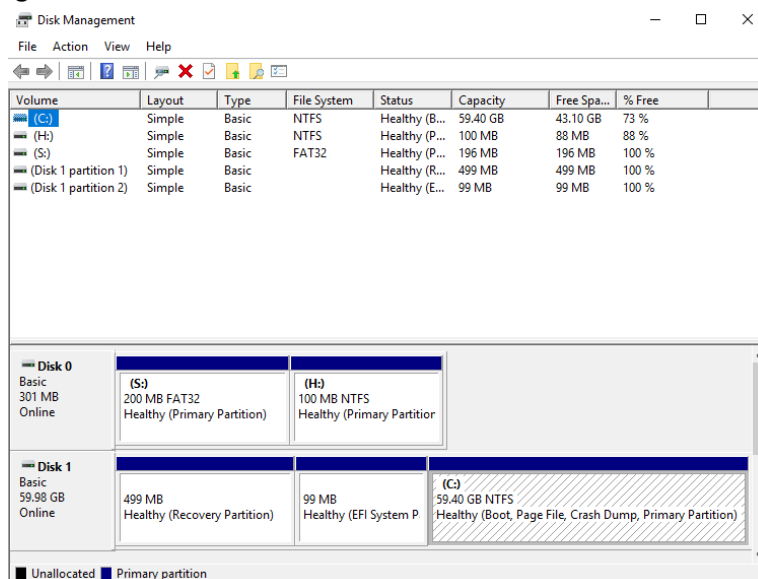
DISKPART> select partition 2
Partition 2 is now the selected partition.

DISKPART> format FS=NTFS
100 percent completed
DiskPart successfully formatted the volume.

DISKPART> assign letter=H
DiskPart successfully assigned the drive letter or mount point.

DISKPART> exit
  
```

Viola! We can now see our created partitions which should match with the red table on the last page via the 'Disk Management Tool'



**Linux:**

In linux, instead of using **diskpart** which is for windows, we use the **fdisk** command to create the partitions and mount them according to our red table on page 2. Use the follow commands as seen in the picture below:

```
[root@localhost ~]# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.32.1).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table.
Created a new DOS disklabel with disk identifier 0xfce40345.

Command (m for help): n
Partition type
   p   primary (0 primary, 0 extended, 4 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1):
First sector (2048-411647, default 2048):
Last sector, +sectors or +size{K,M,G,T,P} (2048-411647, default 411647): +100M

Created a new partition 1 of type 'Linux' and of size 100 MiB.

Command (m for help): n
Partition type
   p   primary (1 primary, 0 extended, 3 free)
   e   extended (container for logical partitions)
Select (default p): p
Partition number (2-4, default 2):
First sector (206848-411647, default 206848):
Last sector, +sectors or +size{K,M,G,T,P} (206848-411647, default 411647):

Created a new partition 2 of type 'Linux' and of size 100 MiB.

Command (m for help): w
The partition table has been altered.
Calling ioctl() to re-read partition table.
Syncing disks.
```

After this reboot with the **reboot** command.

Notes:

1. 'n' adds a new partition and 'p' prints partition table
2. We add +100M to make the first partition 100MiB, the second partition will default to 100 MiB after repeating same steps
3. 'w' will write changes in the table to the disk and exit

After the reboot, we now create our swap partition on partition 1 and make/format partition 2 as EXT4 as seen in the red table in page 2 via the following commands:

```
[root@localhost ~]# mkswap /dev/sdb1
Setting up swapspace version 1, size = 100 MiB (104853504 bytes)
no label, UUID=92238e48-3e61-4378-9db8-23c0b65fe7d2
[root@localhost ~]# swapon /dev/sdb1
[root@localhost ~]# swapon
NAME      TYPE      SIZE USED  PRIO
/dev/dm-1 partition 2G 268K  -2
/dev/sdb1 partition 100M  0B   -3
[root@localhost ~]# mkfs -t ext4 /dev/sdb2
mke2fs 1.45.4 (23-Sep-2019)
Creating filesystem with 102400 1k blocks and 25688 inodes
Filesystem UUID: c64ecc36-d017-44d9-a0a4-841ee53ff8aa
Superblock backups stored on blocks:
    8193, 24577, 40961, 57345, 73729

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done
```

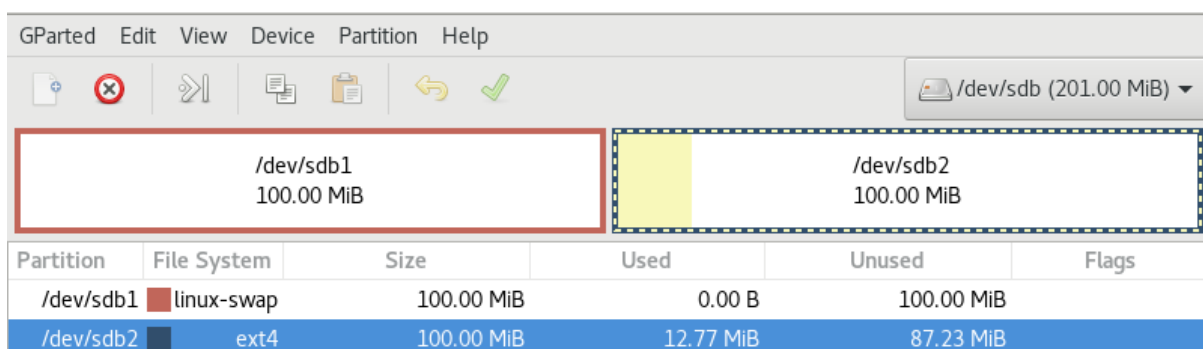
We now mount partition 2 **/dev/sdb2** to the **/www** directory and make mount point and swap of sdb2 permanent through **/etc/fstab**. We use the following commands:

```
[root@localhost ~]# mkdir /www
[root@localhost ~]# mount /dev/sdb2 /www
[root@localhost ~]# mount
```

We edit the lines in the **/etc/fstab** and reboot

- /dev/sdb1 swap swap defaults 0 0
- /dev/sdb2 /www ext4 defaults 0 0

We can now see our newly created partitions by installing **gparted** to see things easier via **yum install gparted**.



Partition	File System	Size	Used	Unused	Flags
/dev/sdb1	linux-swap	100.00 MiB	0.00 B	100.00 MiB	
/dev/sdb2	ext4	100.00 MiB	12.77 MiB	87.23 MiB	

## Task 2: Set up static networking (5)

This task focuses on configuring the static network via the table below

	Our Network	Linux Server	Windows Server
Network	1.2.3.0/24	.254	.250
Ethernet		ens37	Ethernet1
DNS Domain	feit.edu	feit.edu	feit.edu
Hostname		ben	lee
Gateway			

**Linux:**

First we should enable Network services via the **systemctl start NetworkManager** and **systemctl enable NetworkManager** commands.

```
root@localhost:~  
File Edit View Search Terminal Help  
[root@localhost ~]# systemctl start NetworkManager  
[root@localhost ~]# systemctl enable NetworkManager  
Created symlink /etc/systemd/system/multi-user.target.wants/NetworkManager.service → /usr/lib/systemd/system/NetworkManager.service.  
Created symlink /etc/systemd/system/dbus-org.freedesktop.nm-dispatcher.service → /usr/lib/systemd/system/NetworkManager-dispatcher.service.  
Created symlink /etc/systemd/system/network-online.target.wants/NetworkManager-wait-online.service → /usr/lib/systemd/system/NetworkManager-wait-online.service.
```

After this we should configure ens37 for a static IP address as seen for the table above. We need to create a new config file for ens37 so we use the **cp** command to copy the ens33 configuration for **ens37**. Commands are shown below:

```
[root@localhost ~]# cd /etc/sysconfig/network-scripts/  
[root@localhost network-scripts]# ls  
ifcfg-ens33  ifcfg-Wired_connection_1  
[root@localhost network-scripts]# cp ifcfg-ens33 ifcfg-ens37  
[root@localhost network-scripts]# vim ifcfg-ens37
```

Edit /etc/sysconfig/network-scripts/ifcfg-ens37 via vim and add:

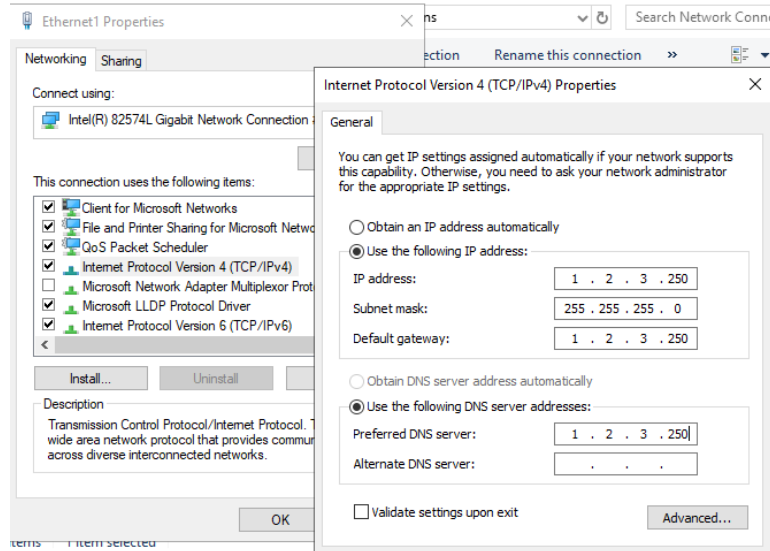
1. **BOOTPROTO=none**
2. **IPADDR=1.2.3.254**
3. **NETMASK=255.255.255.0**
4. **DEFROUTE=NO**
5. **ONBOOT=YES**
6. **NAME=ens37**
7. **DEVICE=ens37**
8. **Delete UUID**



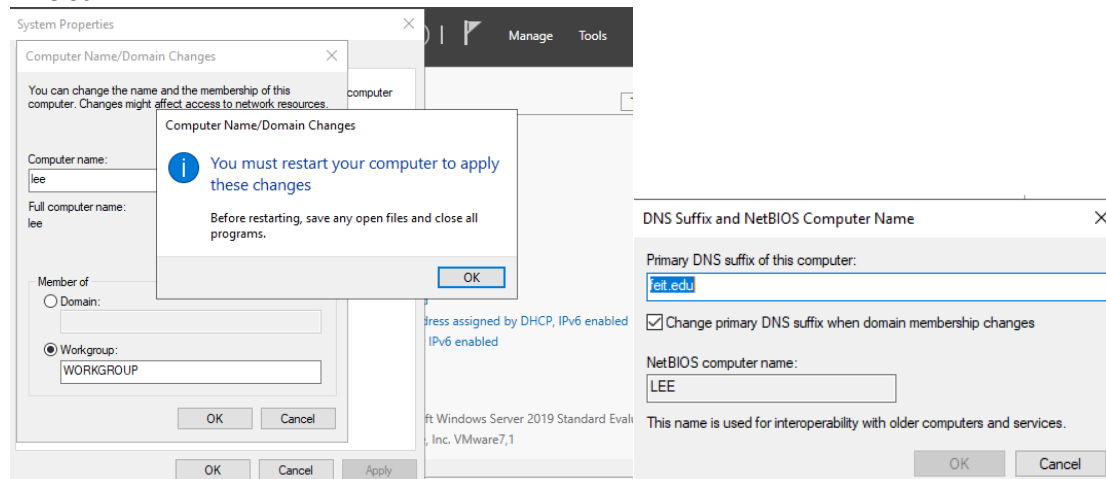


## Windows:

For Windows we should configure Eth1 with static IP of **1.2.3.250** and subnet mask of **255.255.255.0**. Subnet mask and preferred DNS Server should also be **1.2.3.250**. We do this via **Network and Internet settings → Change Adapter Options → Ethernet1 → Properties → IPv4** and edit following configuration

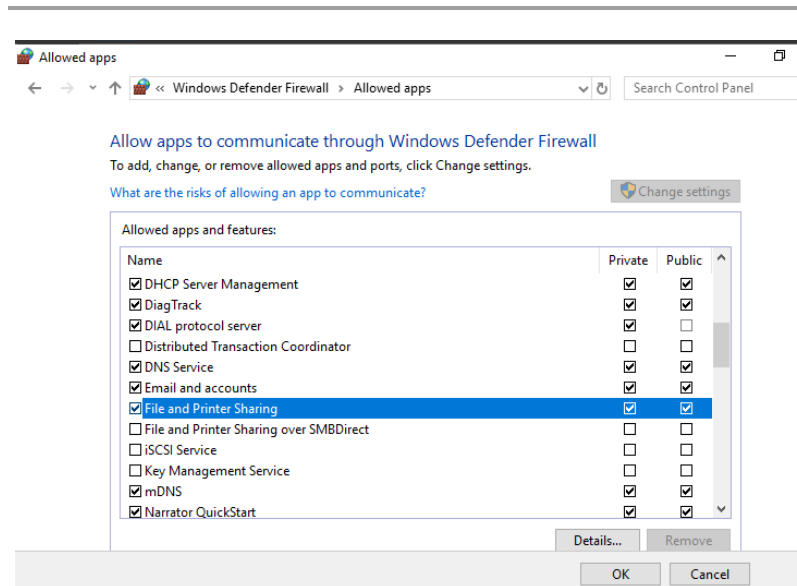


We can change hostname for Windows to be a surname, which is in my case 'lee' via **Server Manager → Computer Name → Change → Computer Name/Domain Changes** and change computer name to 'lee'. We can also change DNS Suffix to **feit.edu** by clicking **more** and changing DNS suffix.

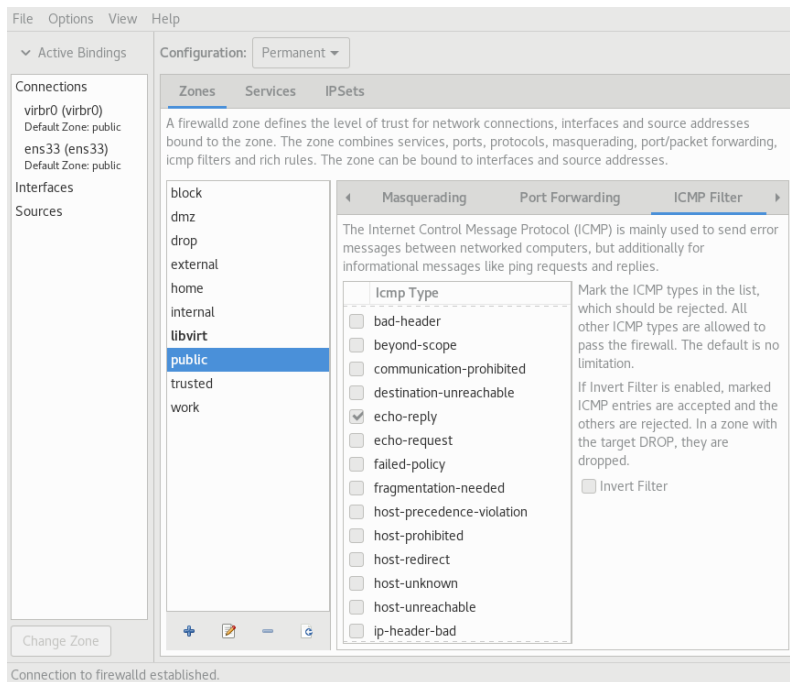


After this we can test if configurations are correct if we can ping both VM's.

In order to allow Linux to ping Windows the **"File and Printer Sharing"** option needs to be enabled in Windows Defender Firewall in **allowed apps**.



For linux, with the command **firewall-config**, we can allow echo-reply requests permanently via the ICMP filter in permanent.



After this, I could not get it working but I **fixed the problem** by using the following commands:

1. **[root@ben ~]# ifconfig ens37 1.2.3.254 netmask 255.255.255.0**
2. **[root@ben ~]# route add default gw 1.2.3.254**

After, I was able to ping and test each machine successfully showing that the configurations are correct.

### WINDOWS PINGING LINUX:

```
C:\Users\Administrator>ping 1.2.3.254

Pinging 1.2.3.254 with 32 bytes of data:
Reply from 1.2.3.254: bytes=32 time<1ms TTL=64
Reply from 1.2.3.254: bytes=32 time<1ms TTL=64
Reply from 1.2.3.254: bytes=32 time<1ms TTL=64
Reply from 1.2.3.254: bytes=32 time<1ms TTL=64

Ping statistics for 1.2.3.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

### LINUX PINGING WINDOWS:

```
[root@ben ~]# ping 1.2.3.250
PING 1.2.3.250 (1.2.3.250) 56(84) bytes of data.
64 bytes from 1.2.3.250: icmp_seq=1 ttl=128 time=0.851 ms
64 bytes from 1.2.3.250: icmp_seq=2 ttl=128 time=0.440 ms
64 bytes from 1.2.3.250: icmp_seq=3 ttl=128 time=0.496 ms
64 bytes from 1.2.3.250: icmp_seq=4 ttl=128 time=0.397 ms
^C
--- 1.2.3.250 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 78ms
```

### Task 3: Set up NFS service (5)

In task 3, we need to configure our linux server as an NFS server. To do this we can create a new directory called **/public** and give it read and write permissions. We use the commands below:

```
[root@ben ~]# mkdir /public
[root@ben ~]# chmod 777 /public
```

I then made sure **nfs-utils**, **rpcbind**, **nfs4-acl-tools** were installed via the **rpm** commands below:

```
[root@ben ~]# rpm -q nfs-utils
nfs-utils-2.3.3-31.el8.x86_64
[root@ben ~]# rpm -q rpcbind
rpcbind-1.2.5-7.el8.x86_64
[root@ben ~]# rpm -q nfs4-acl-tools
nfs4-acl-tools-0.3.5-3.el8.x86_64
```

To share the **/public** directory we edit the **/etc/exports** file via vim and add the line: **/public 1.2.3.0/24(rw)**

```
root@ben:~
File Edit View Search Terminal Help
/public 1.2.3.0/24(rw)
```

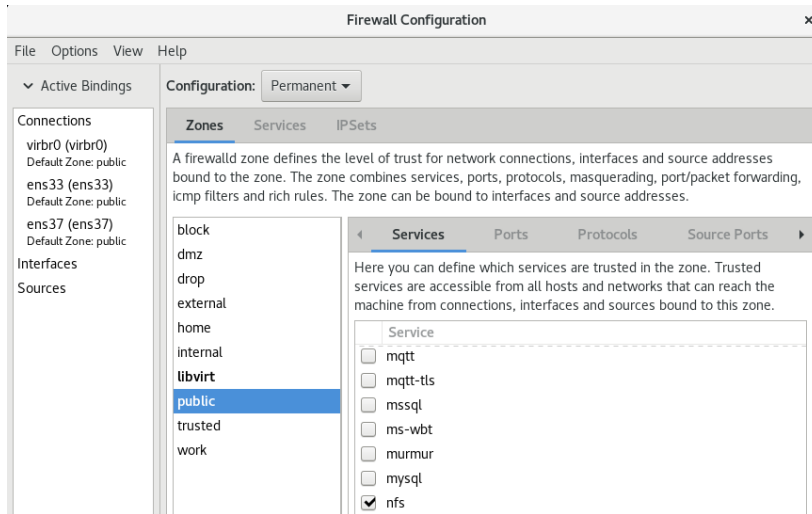
We must now start and enable the **rpcbind** and **nfs-server** service via the following commands:

```
[root@ben ~]# vim /etc/exports
[root@ben ~]# systemctl start rpcbind
[root@ben ~]# systemctl start nfs-server
[root@ben ~]# systemctl enable rpcbind
[root@ben ~]# systemctl enable nfs-server
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
/usr/lib/systemd/system/nfs-server.service.
```

We now run **exportfs -v** to update our NFS configuration and verify the directory is exported properly

```
[root@ben ~]# exportfs -v
/public 1.2.3.0/24(sync,wdelay,hide,no_subtree_check,sec=sys,rw,secure,root_squash,no_all_squash)
```

Via the **firewall-config** command we can allow NFS permanently as seen below:



After this, we can create **/mnt/public**, mounting NFS to it and make it permanent by editing **/etc/fstab** with the line: **1.2.3.254:/public /mnt/public nfs defaults 1 1**

```
[root@ben ~]# mkdir /mnt/public
[root@ben ~]# mount -t nfs4 1.2.3.254:/public /mnt/public
[root@ben ~]# vim /etc/fstab
```

We then reboot via the **reboot** command and mount via the command **mount**.

After this, we can test the configuration by creating a file. We can then use **ls -l** command to see the file we created in **/mnt/public**. This can be seen below

```
[root@ben ~]# touch /mnt/public/filetest1
[root@ben ~]# ls -l /mnt/public
total 0
-rw-r--r-- 1 root root 0 Nov  3 15:39 filetest1
```

Task 4: Set up DNS server (10)

N/A

Task 5: Set up Web server (10)

N/A