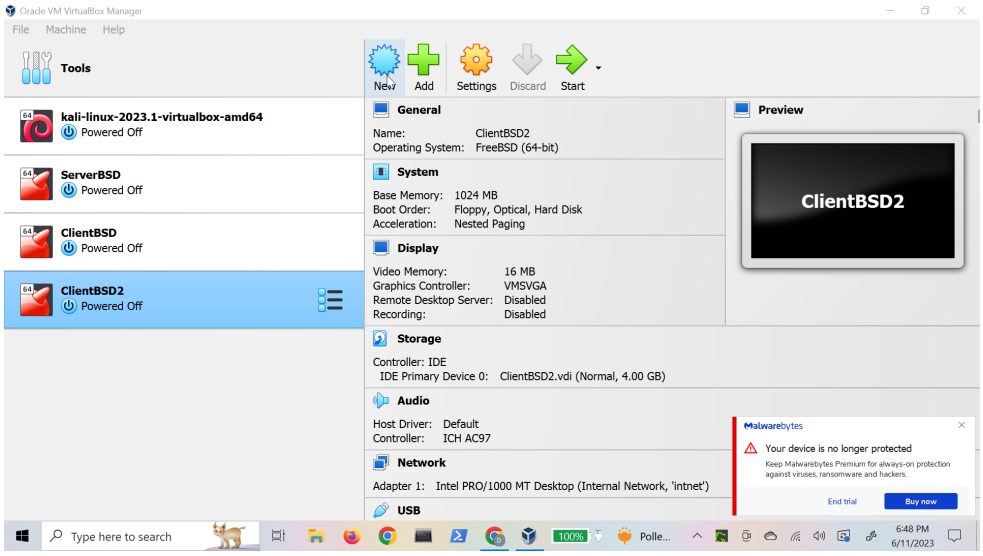
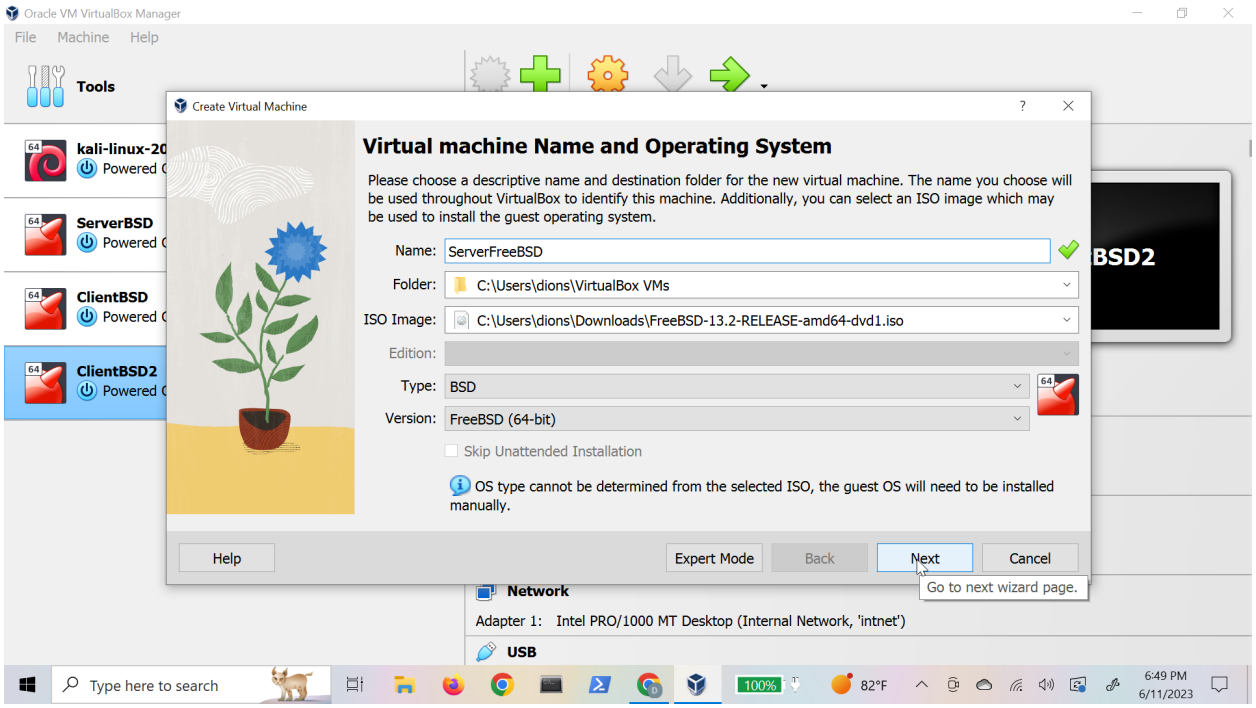
README

Preparation (Virtual Box Setup):

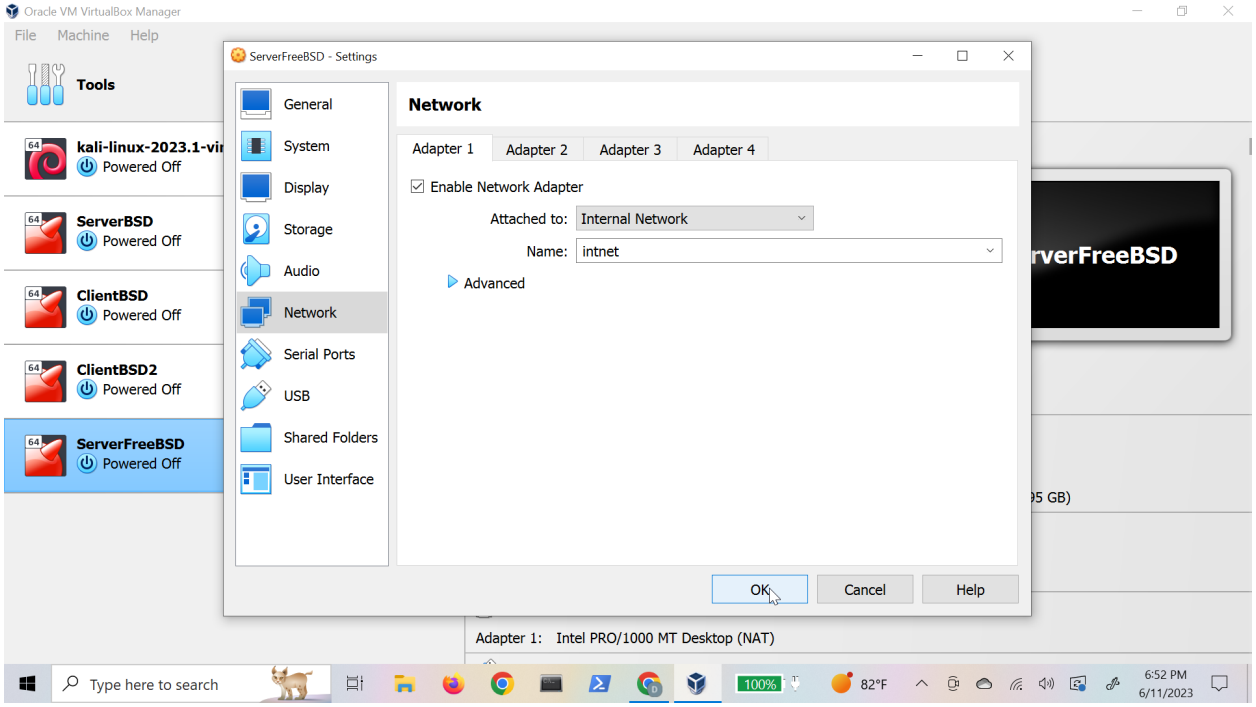


Setup the new vm

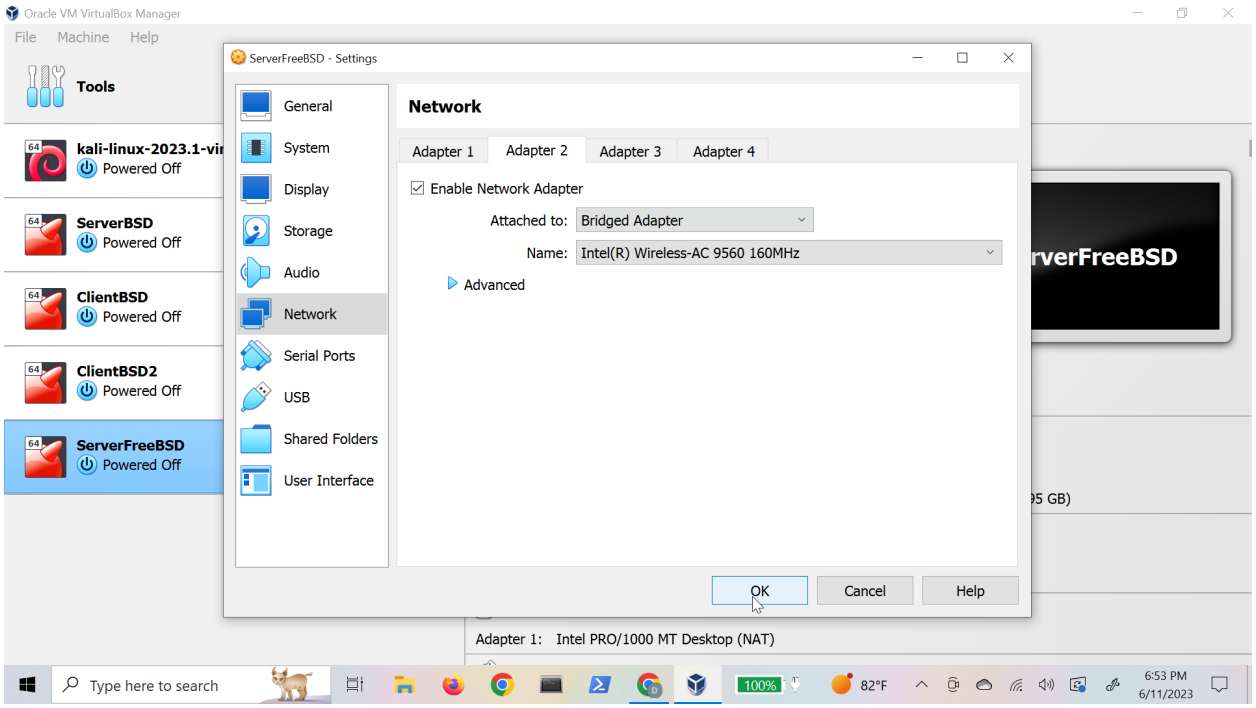


I am using a dvd file but a basic disc image is fine. Setup the RAM usage, CPU and storage as you like.

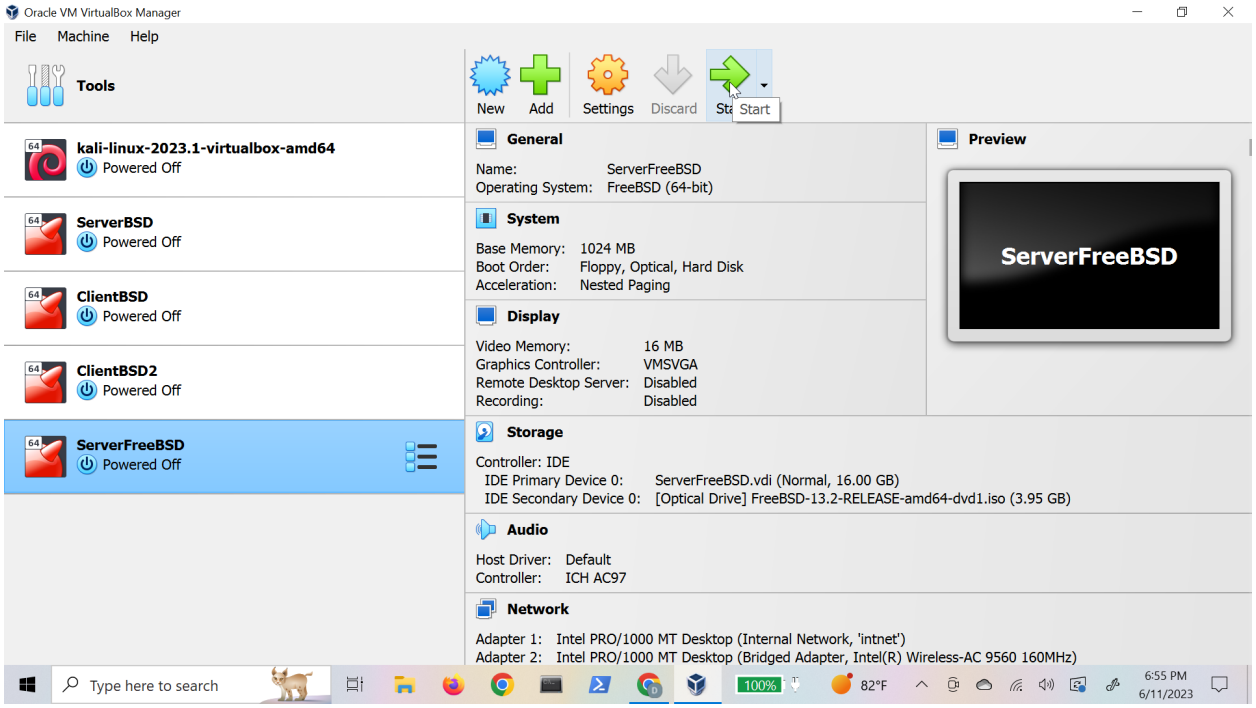
Setup the network adapters:



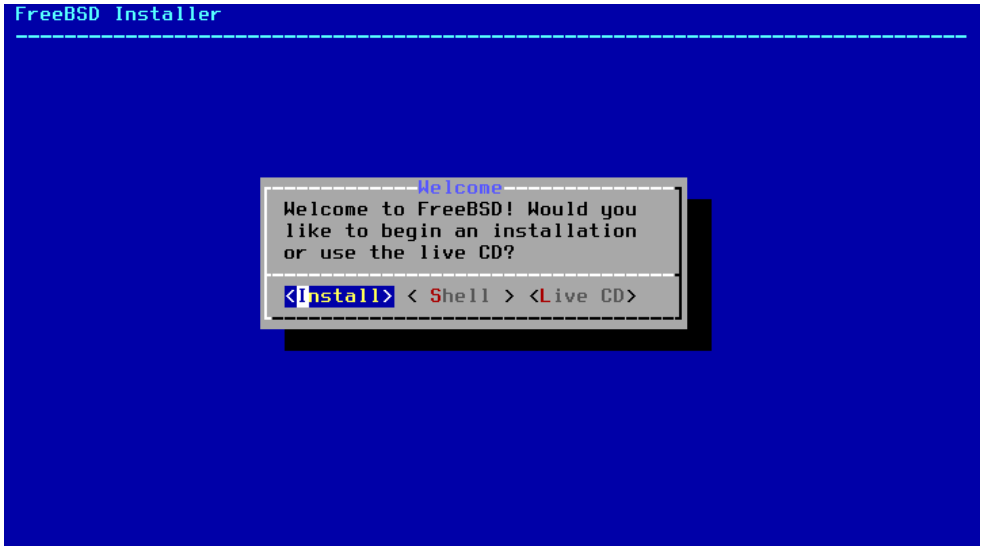
Go to the VM settings before you start and select the network and set the first adapter as an internal network.



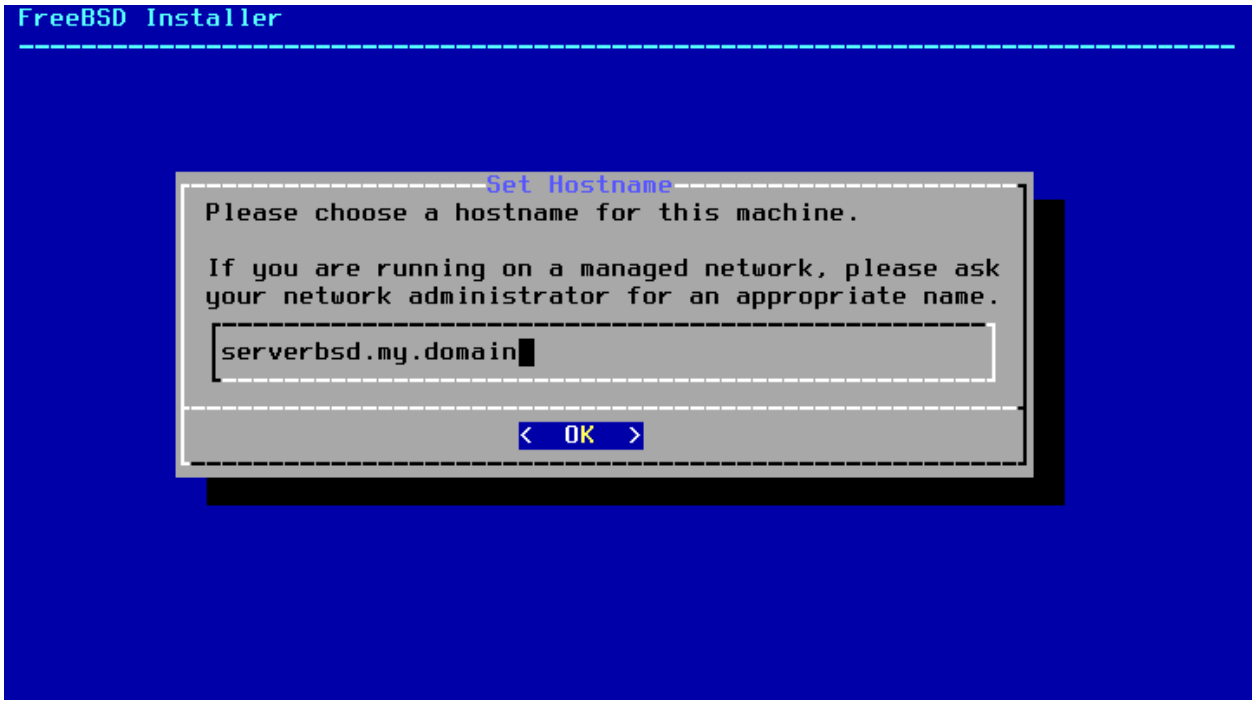
Set the second adapter as bridged for now. This is so you'll be able to ssh into the machine to add the setup script for the router. You can change it to NAT later if you wish and it will still work.



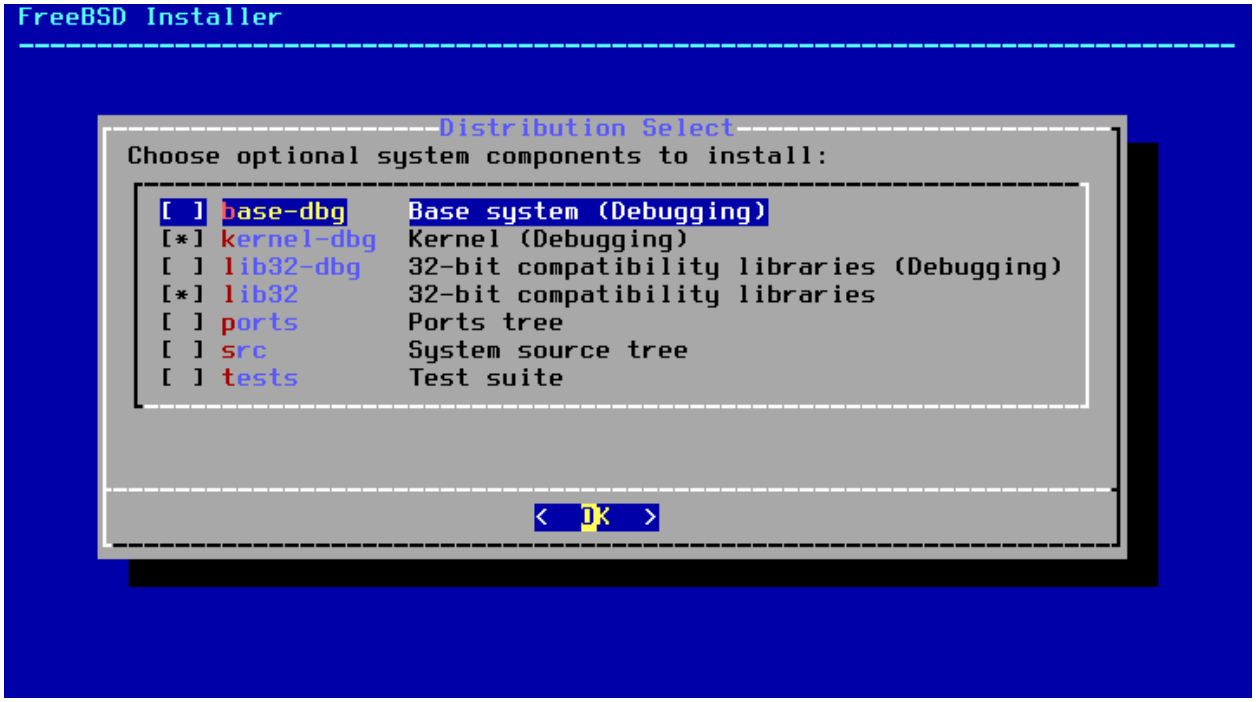
Start the VM and the installation process for FreeBSD



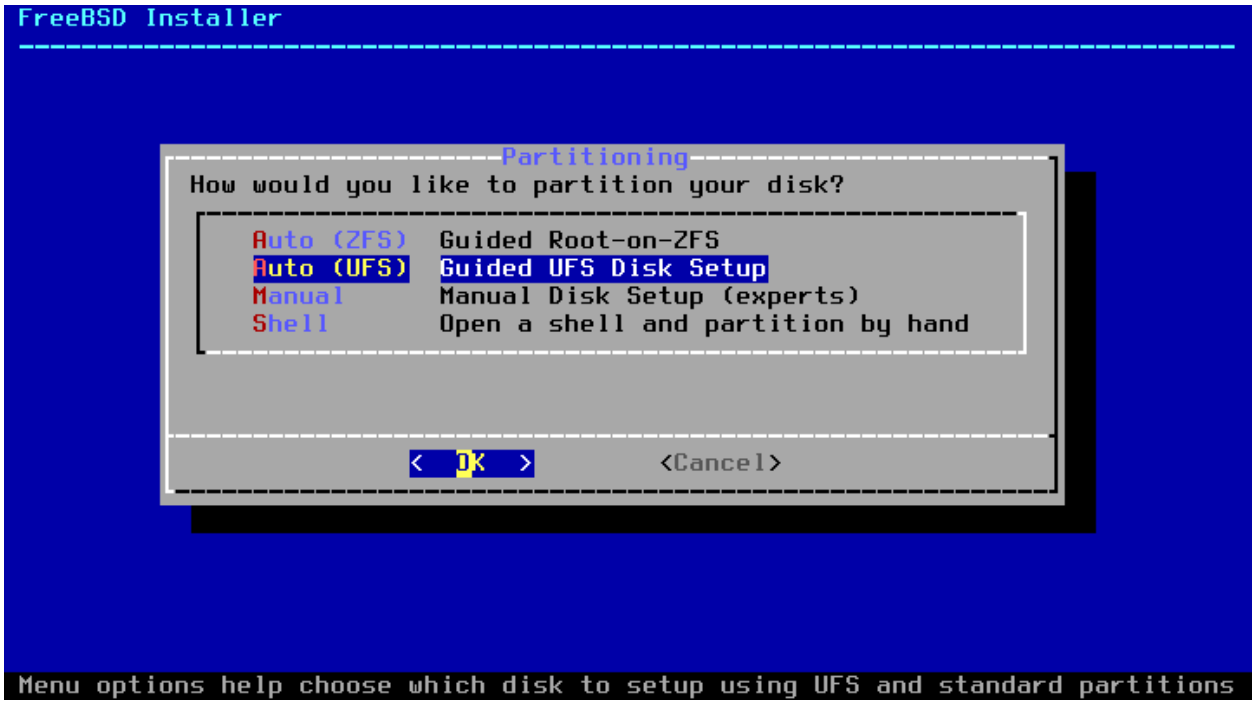
You should get a boot up screen then it should direct you to this screen. I go through the regular installer.



Set your host name to what you want.

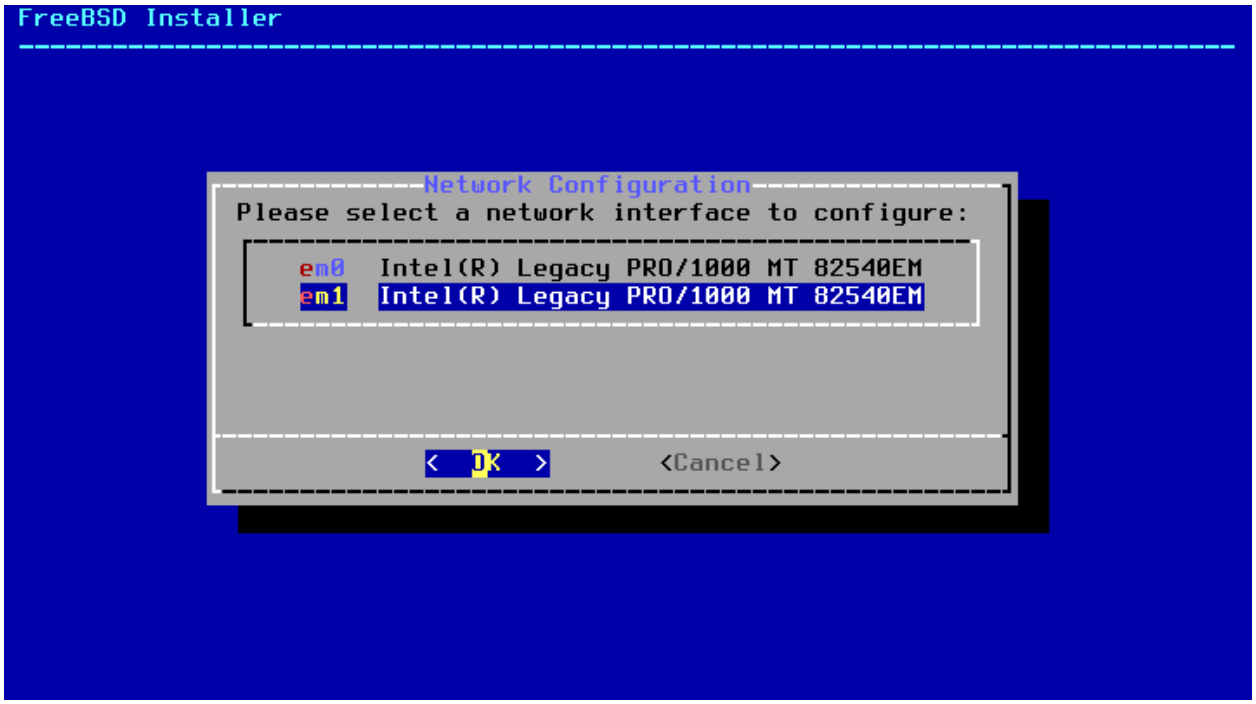


This can be left default so just press enter

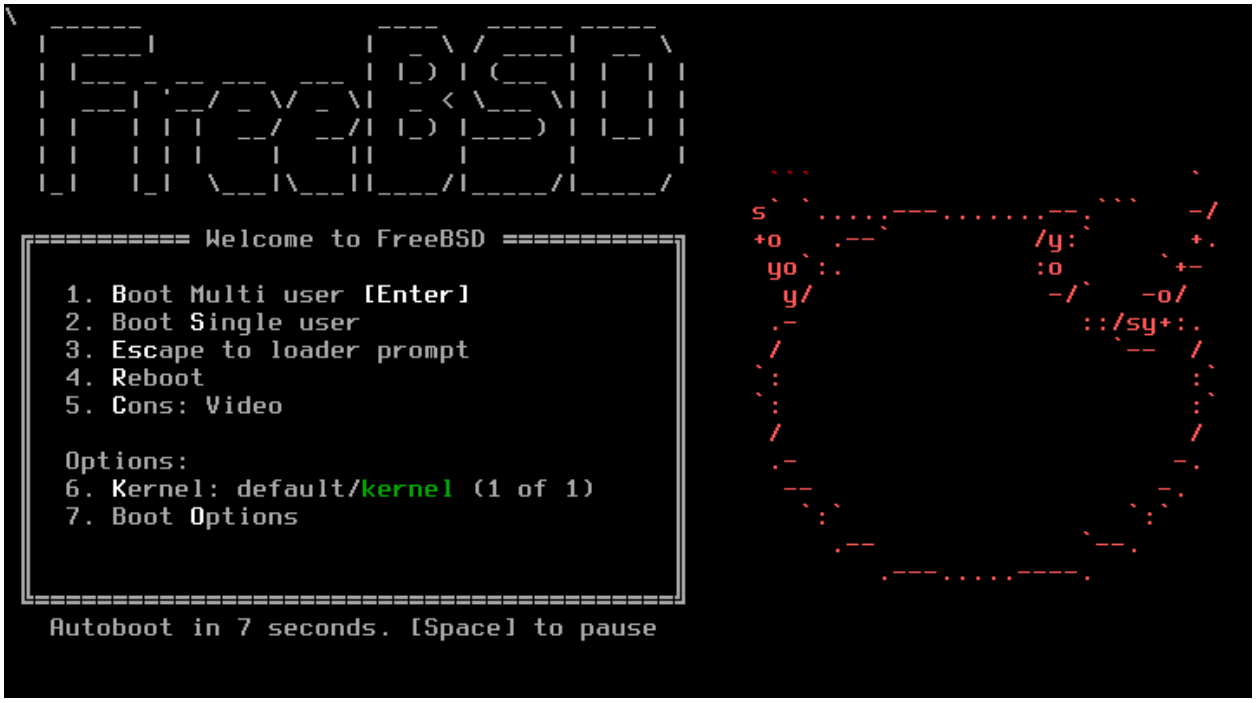


I use UFS for the disk setup and you should partition the entire disk

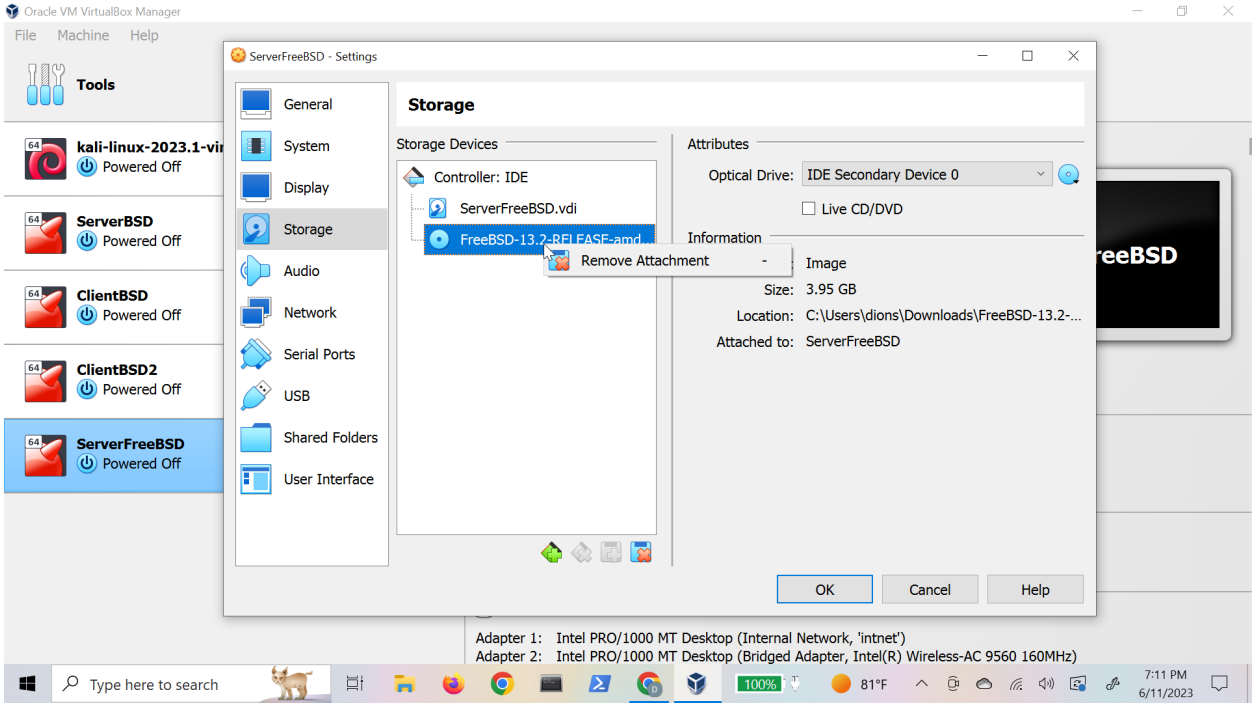
Then I use DOS partitions. The rest can be default and you can choose your own root password.



Select the em1 to configure since it's our WAN interface and choose yes to configure it for IPv4, ignore IPv6 configuration but you can if you want it's not necessary. The resolver configuration can be default along with the options. You can skip the add user option. (**I've had trouble getting it to actually add a user in the installation setup. It only worked for me when I rebooted the system after installation and ran the command adduser**) finally after pressing enter to everything be sure to reboot.

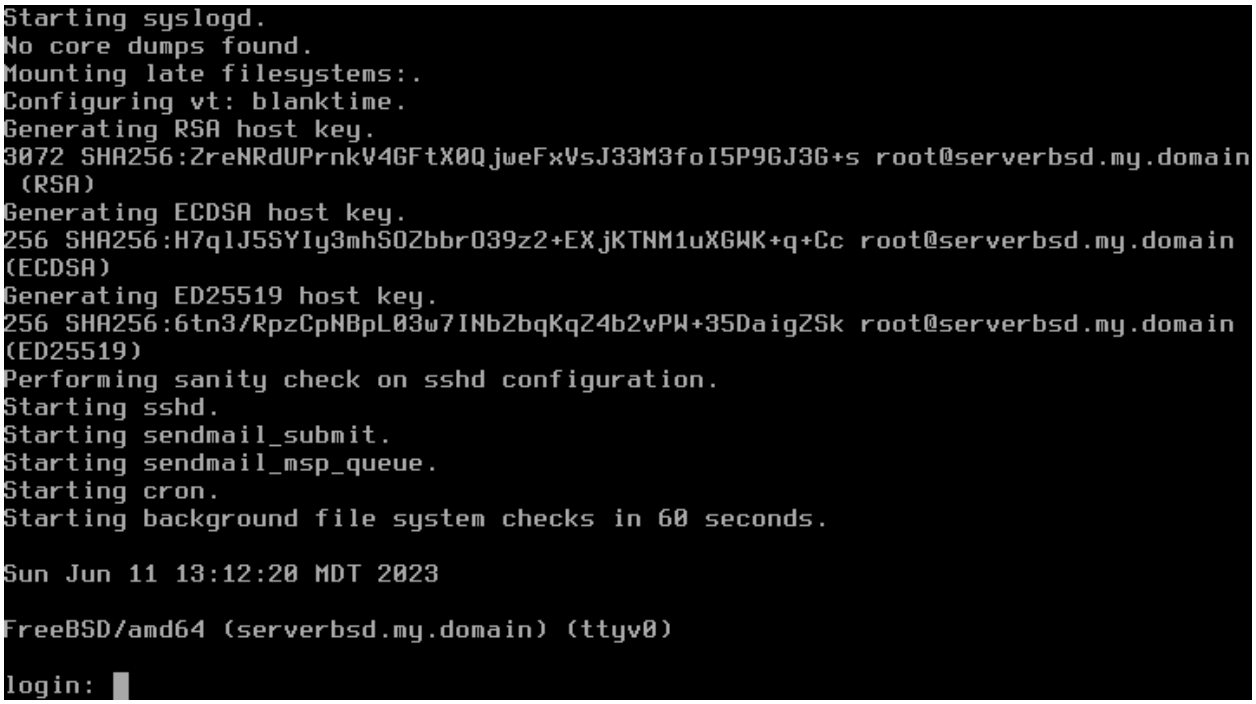


When you get back to this machine stop the vm

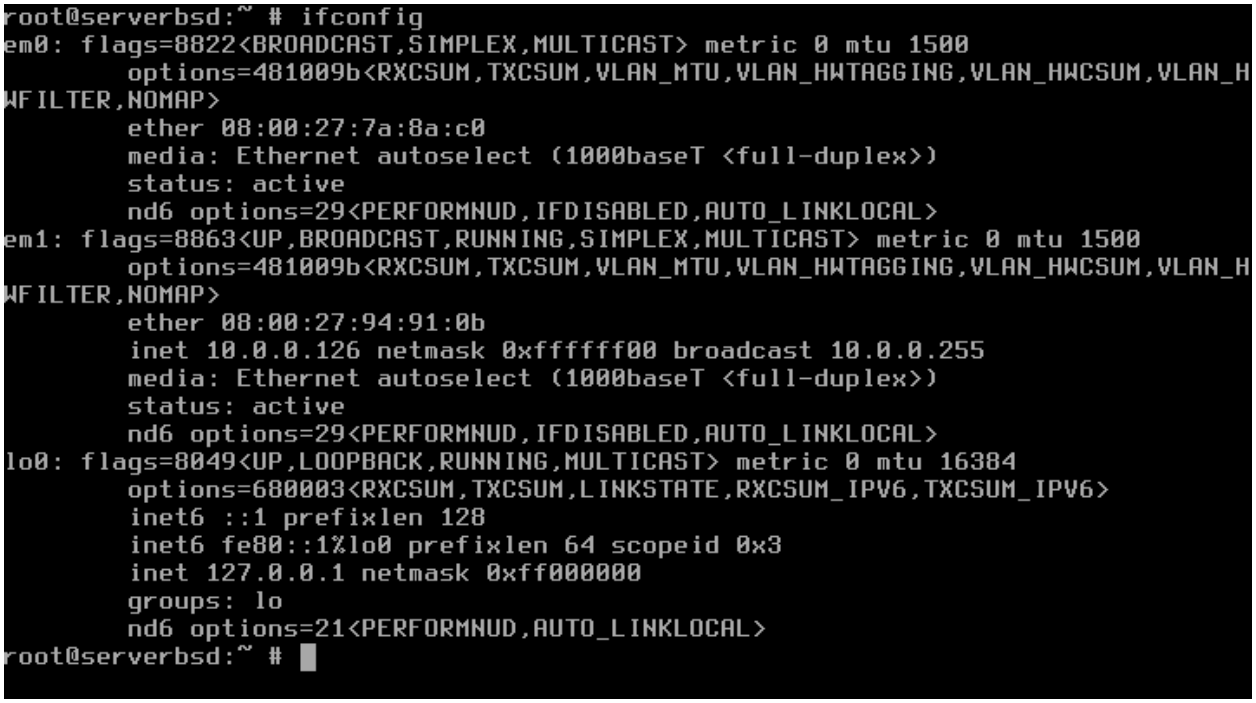


Remove the installation iso from the storage and then boot the vm.

VM Setup and SSH:



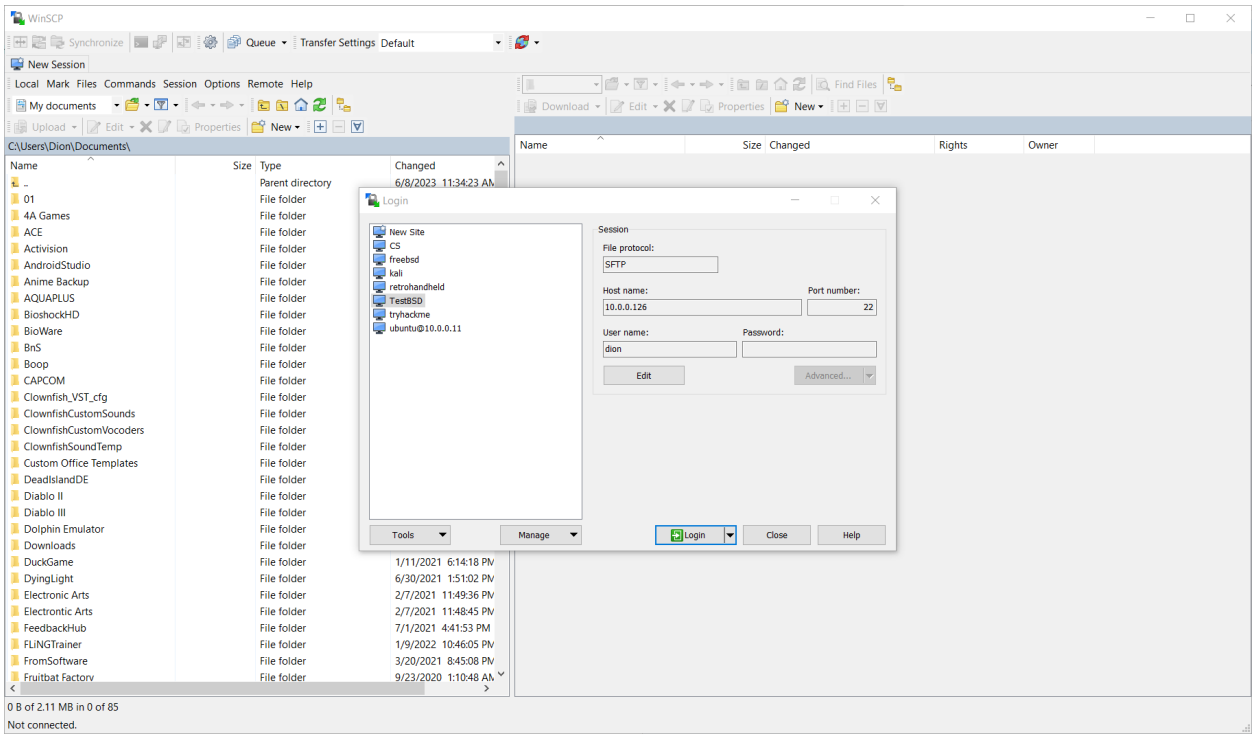
Login as root and the root password you set



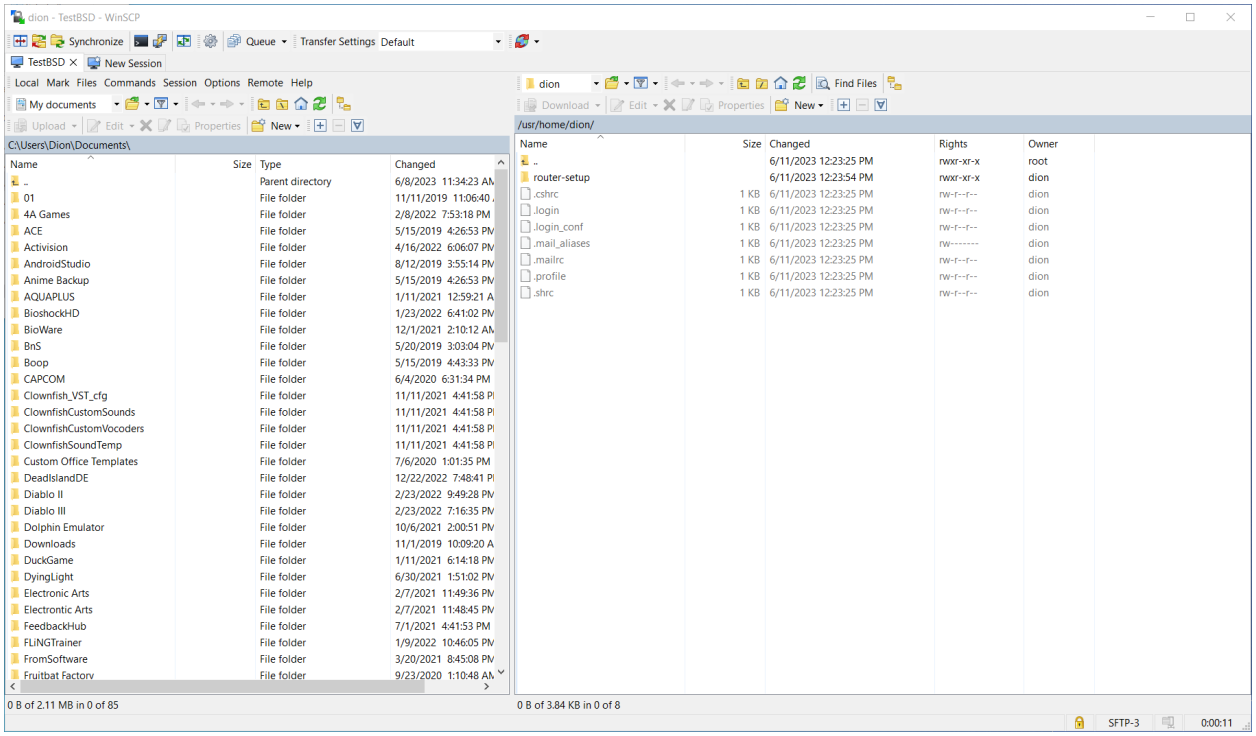
Run ifconfig to get the WAN interface ip so you can SSH into it. SSH should be enabled by default but if it's not. You just need to go into /etc/rc.conf and add sshd_enable=”YES” and restart the vm.

```
Shell (sh csh tcsh nologin) [sh]:
Home directory [/home/dion]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Passwords did not match!
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : dion
Password   : *****
Full Name  : dion
Uid        : 1001
Class      :
Groups     : dion wheel
Home       : /home/dion
Home Mode  :
Shell      : /bin/sh
Locked     : no
OK? (yes/no):
```

Next adduser and go through the setup for the user. (**You need a user so you can ssh into the machine.**)



I ssh into the vm using a file manager called WinSCP. I'm not sure what you might have but you'll need a SFTP client and FTP client to add the setup script into the vm. That's how I was able to get it working. You could try to get virtualbox drop and drop working but I never got it working plus I dont think it works with freebsd.



I would place the router-setup folding in your /home/[user] directory

Run Setup Script:

```
Uid             : 1001
Class           :
Groups          : dion wheel
Home            : /home/dion
Home Mode       :
Shell           : /bin/sh
Locked          : no
OK? (yes/no): Jun 11 13:23:06 serverbsd sshd[995]: error: PAM: Authentication er
ror for illegal user dion from 10.0.0.55

OK? (yes/no): yes
adduser: INFO: Successfully added (dion) to the user database.
Add another user? (yes/no): no
Goodbye!
root@serverbsd:~ # Jun 11 13:23:40 serverbsd syslogd: last message repeated 5 ti
mes
Jun 11 13:23:40 serverbsd sshd[995]: error: maximum authentication attempts exce
eded for invalid user dion from 10.0.0.55 port 54591 ssh2 [preauth]
ls
.cshrc          .k5login        .login          .profile        .shrc
root@serverbsd:~ # cd /home/dion/router-setup/
root@serverbsd:/home/dion/router-setup # ls
dhcpd.conf              pf.conf                 unbound.conf
packet_installer.sh     rc.conf
root@serverbsd:/home/dion/router-setup #
```

Make sure your still root so you can edit the network settings. Then run *cd /home/[user]/router-setup*.

```
Groups      : dion wheel
Home        : /home/dion
Home Mode   :
Shell       : /bin/sh
Locked      : no
OK? (yes/no): Jun 11 13:23:06 serverbsd sshd[995]: error: PAM: Authentication er
ror for illegal user dion from 10.0.0.55

OK? (yes/no): yes
adduser: INFO: Successfully added (dion) to the user database.
Add another user? (yes/no): no
Goodbye!
root@serverbsd:~ # Jun 11 13:23:40 serverbsd syslogd: last message repeated 5 ti
mes
Jun 11 13:23:40 serverbsd sshd[995]: error: maximum authentication attempts exce
eded for invalid user dion from 10.0.0.55 port 54591 ssh2 [preauth]
ls
.cshrc          .k5login        .login          .profile        .shrc
root@serverbsd:~ # cd /home/dion/router-setup/
root@serverbsd:/home/dion/router-setup # ls
dhcpd.conf              pf.conf                 unbound.conf
packet_installer.sh     rc.conf
root@serverbsd:/home/dion/router-setup # pkg
The package management tool is not yet installed on your system.
Do you want to fetch and install it now? [y/N]:
```

Run *pkg* to install the pkg management tool

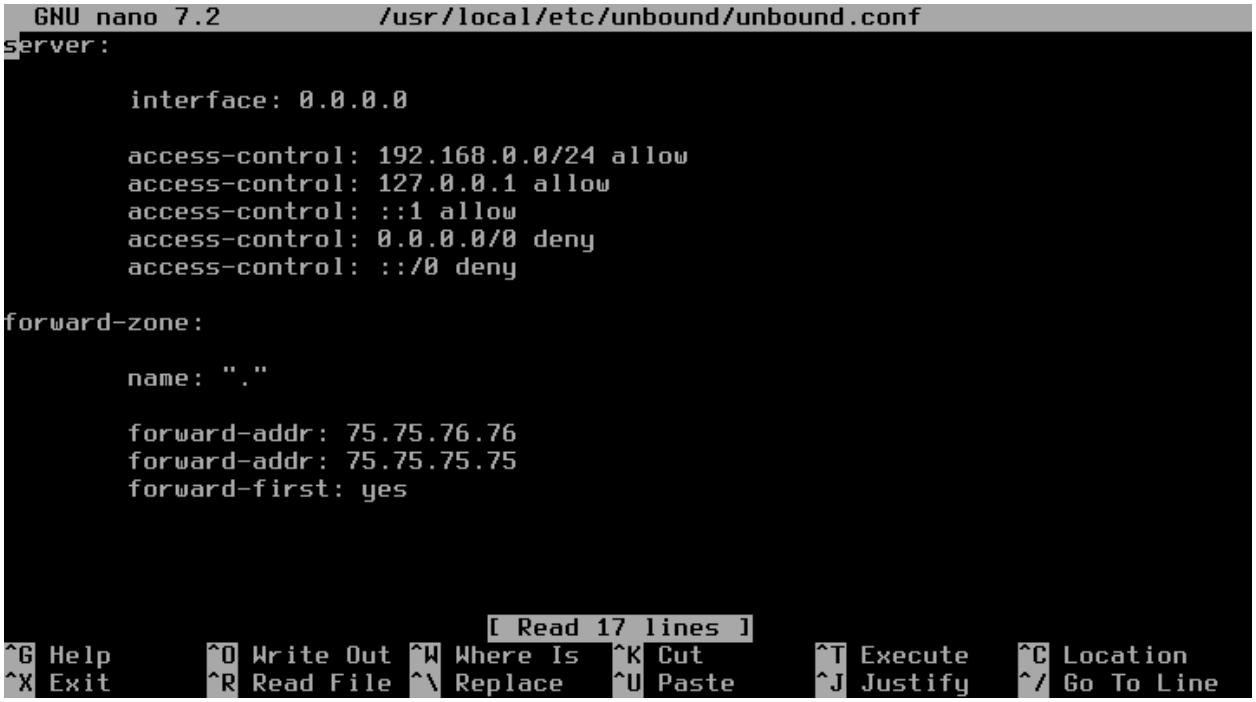
```
root@serverbsd:/home/dion/router-setup # pkg install bash
Updating FreeBSD repository catalogue...
Fetching meta.conf: 100%    163 B   0.2kB/s    00:01
Fetching packagesite.pkg: 100%    7 MiB   6.9MB/s    00:01
Processing entries: 100%
FreeBSD repository update completed. 32886 packages processed.
All repositories are up to date.
Updating database digests format: 100%
The following 4 package(s) will be affected (of 0 checked):

New packages to be INSTALLED:
        bash: 5.2.15
        gettext-runtime: 0.21.1
        indexinfo: 0.3.1
        readline: 8.2.1

Number of packages to be installed: 4

The process will require 12 MiB more space.
2 MiB to be downloaded.

Proceed with this action? [y/N]:
```

Then run *pkg install bash* to install bash so we can run the shell script.

Finally run *bash ./packet_installer.sh* to run the setup script. This should automatically set up the FreeBSD VM to work as a router. Replacing all necessary files with my custom ones. The only problem would be the DNS servers. I'll go over changing the nameservers specified to the ones you use. I use 75.75.75.75 and 75.75.76.76.

The files that need to be changed are /etc/resolv.conf, and /usr/local/etc/unbound/unbound.conf

Change the */etc/resolv.conf* and add your local namerservers in the form of *nameserver *.*.*.**



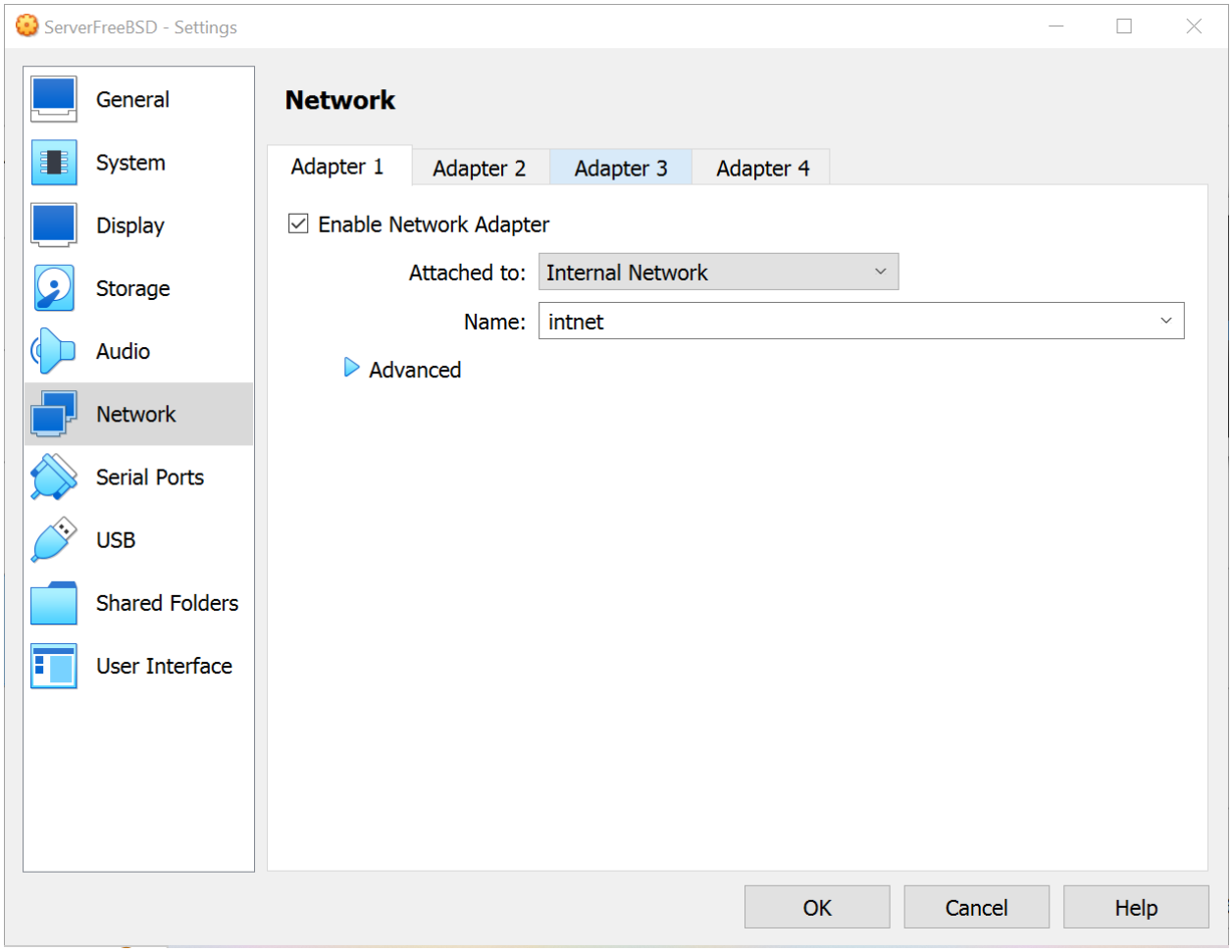
The unbound.conf is the next configuration file to change. If your using different nameserver then you need to change the forwarding server on *forward-addr* to the ones your using

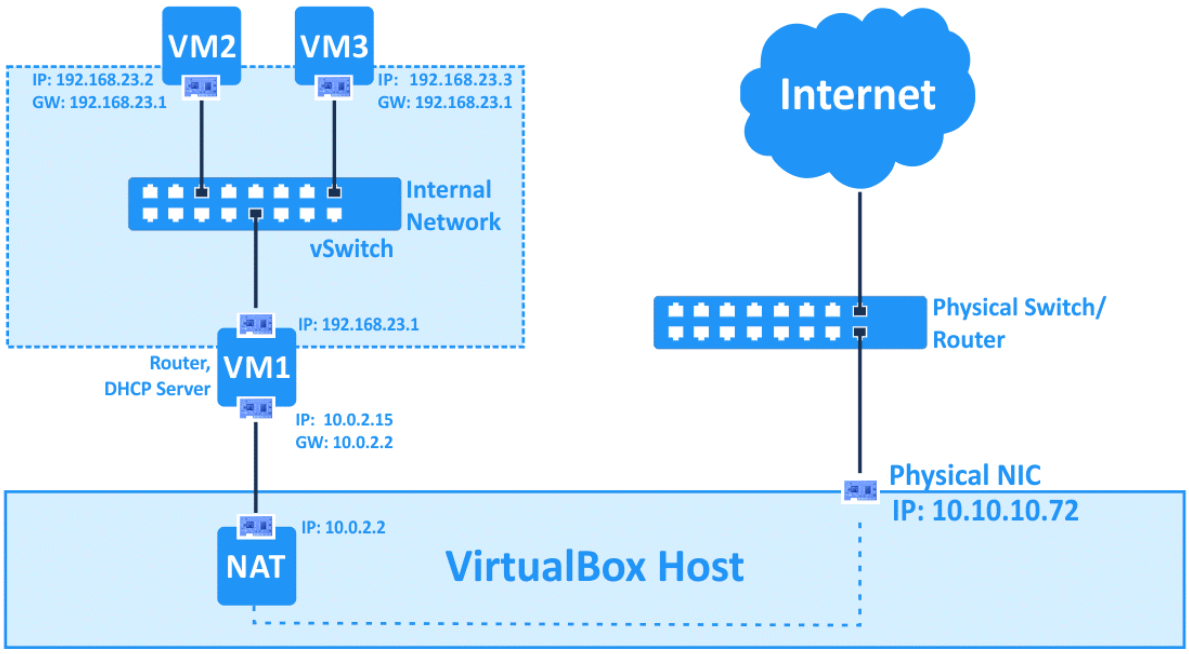
Lastly reboot the VM and afterwards everything should be working.

Features:

Switching:

Switching is internal to Virtualbox we essentially use the Virtualbox vswitch in the internal network to emulate this.





(Source: https://www.nakivo.com/blog/virtualbox-network-setting-guide/)

Firewall, NAT Layer, and Traffic Mirroring:

This is my pf.conf file that specifies all my rules for my firewall

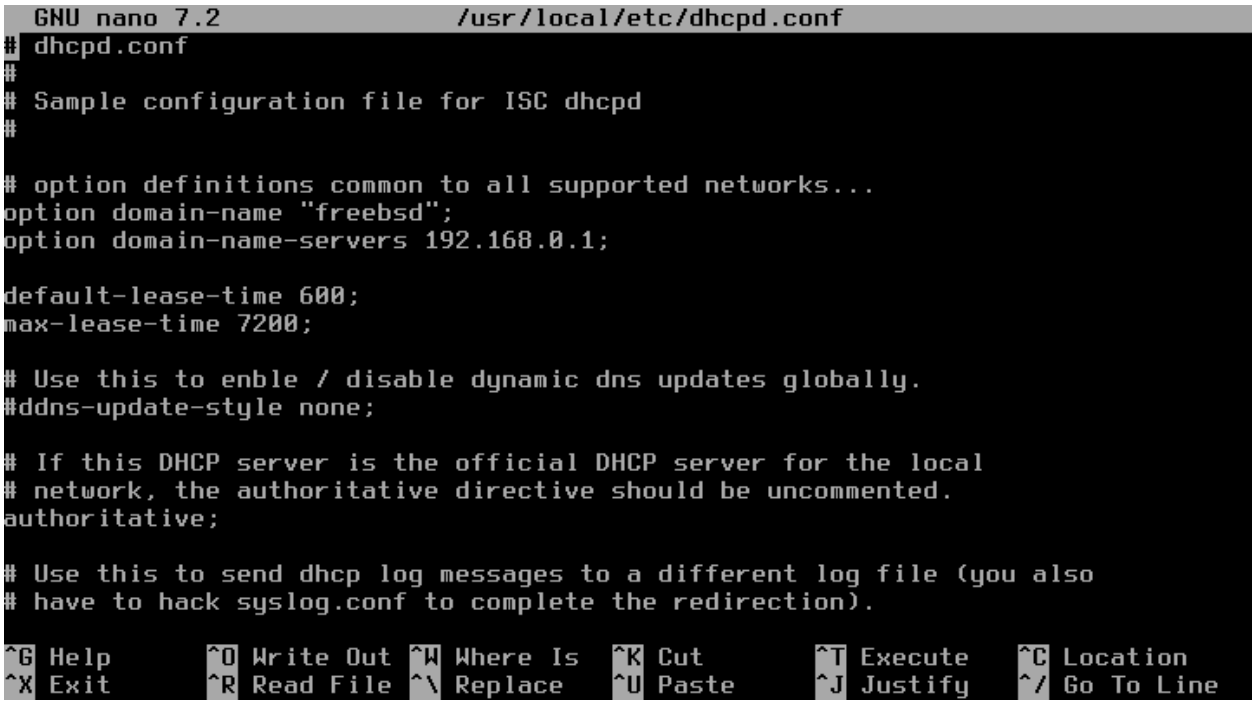
```
  GNU nano 7.2              /home/dion/router-setup/pf.conf
lan="em0"
wan="em1"
#mirror_ip="machine_ip"

set skip on lo0
set block-policy drop

nat on $wan from $lan:network to any -> ($wan)

block drop all

pass in on $lan from $lan:network to any keep state
pass in proto tcp to $wan port ssh flags S/SA keep state
pass out on $wan from $lan:network to any keep state
pass out on $wan from $wan:network to any keep state
#pass in on $wan dup-to ($lan $mirror_ip) to any keep state
#pass out on $wan dup-to ($lan $mirror_ip) to any keep state




                        [ Read 17 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute   ^C Location
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line
```

First I label my interfaces as a LAN variable and WAN variable. I skip the loopback since it doesn't need to be filtered and set the block policy to drop the packets. Then I translate all ip addresses from the lan network to the wan interface to a public ip when going out the WAN.

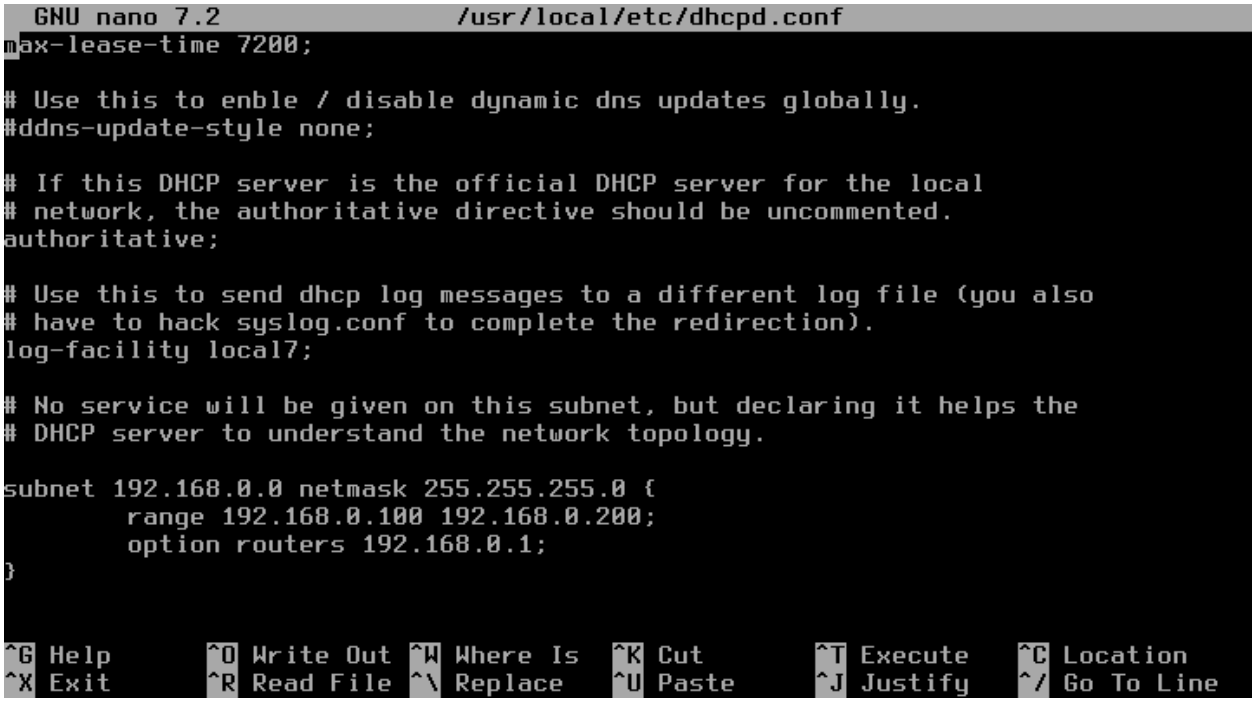
Then I allow all traffic coming in from the lan network to the lan and all SSH traffic coming in for SSH connection. Then I allow all traffic going out the WAN from the lan network and all traffic going out the WAN from the wan network. All these rules allow us to use SSH, and most protocols. To turn on the traffic mirroring you need to uncomment out the *mirror_ip* variable and set the ip address of the machine you want to duplicate/mirror the traffic to. Then uncomment the pass in and pass out rules that i have commented to enable it. Run *pfctl -f /etc/pf.conf* to reload the rules.

DHCP Server:

Next is the DHCP server settings



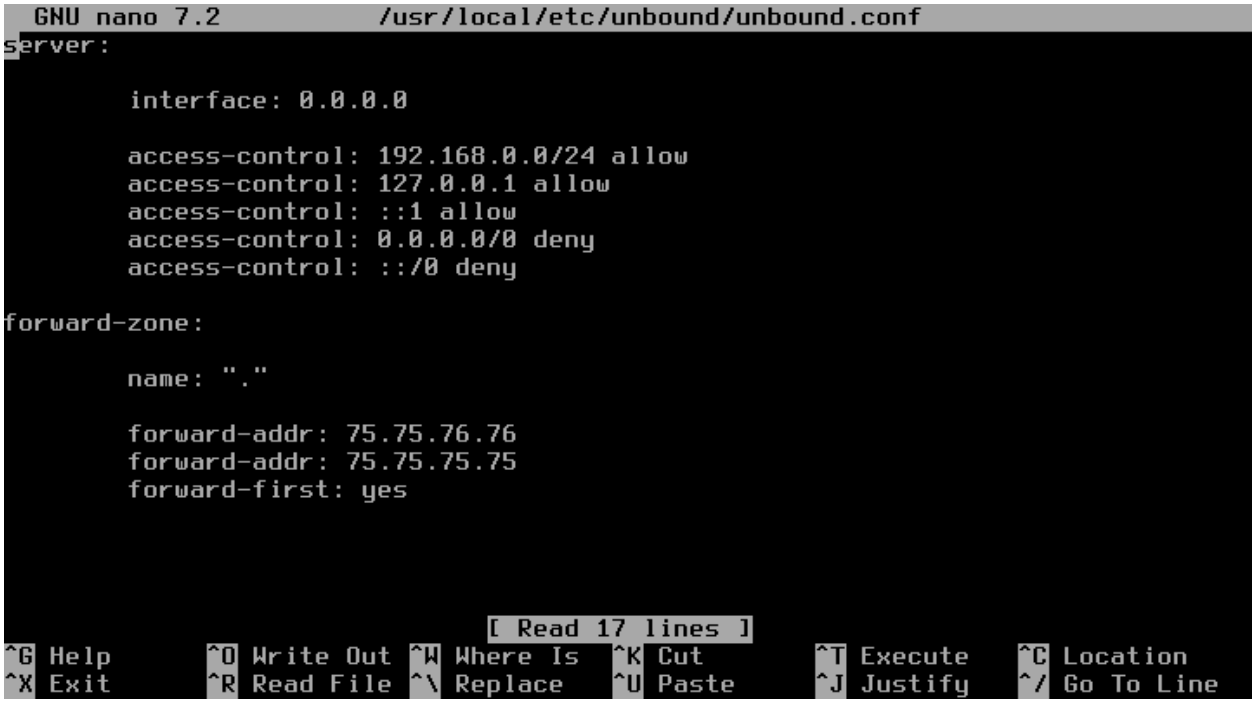
I have a dhcpd.conf file for my dhcp server. So the important setting is the name server option where I specify the nameserver that the associated nodes will use to query domain names. In this case because I have a dns server setup I'm using my lan interface as the DNS server. I uncommented the authoritative; so that the dhcp server will act as the official server of the network.



Then I set the subnet for the ip I will assign to the node. These are the IPs I can assign to the nodes requesting an IP from the server. I set the range from 192.168.0.100 - 192.168.0.200 with the router set to the lan interface IP.

DNS Unbound Server:

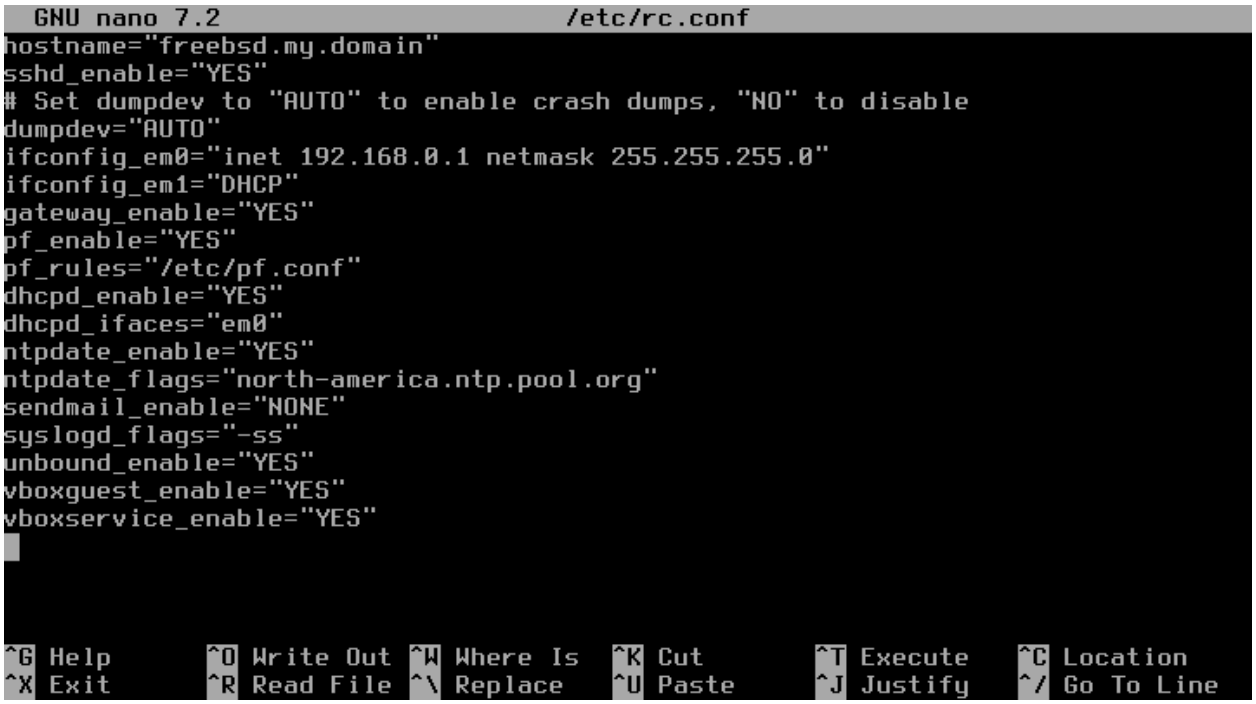
The next file is the for the DNS server configuration



In this case I'm using a package called unbound to make the server act as a DNS server. I allow unbound to listen to all interfaces through the
Interface: 0.0.0.0
You can specify the interface to listen on by just setting the IP address of the interface.
Then we set the access control or what network of computers or subnets can ask the server for information. We set the subnet to our DHCP subnet of our LAN. The next access control is for the loopback of the server and we want to allow access. Then the next one
access-control : 0.0.0.0/0 deny
We want to deny all other IPv4 networks and all other IPv6. The forward zone specifies the DNS servers we forward the request to if the DNS is not known. We allow all domain names. If you have a different DNS server or want to use a different one this is where you change the DNS forward server. Then finally we forward first so that if it doesn't know the answer it will forward first before it looks at its cache for that answer.

rc.conf:

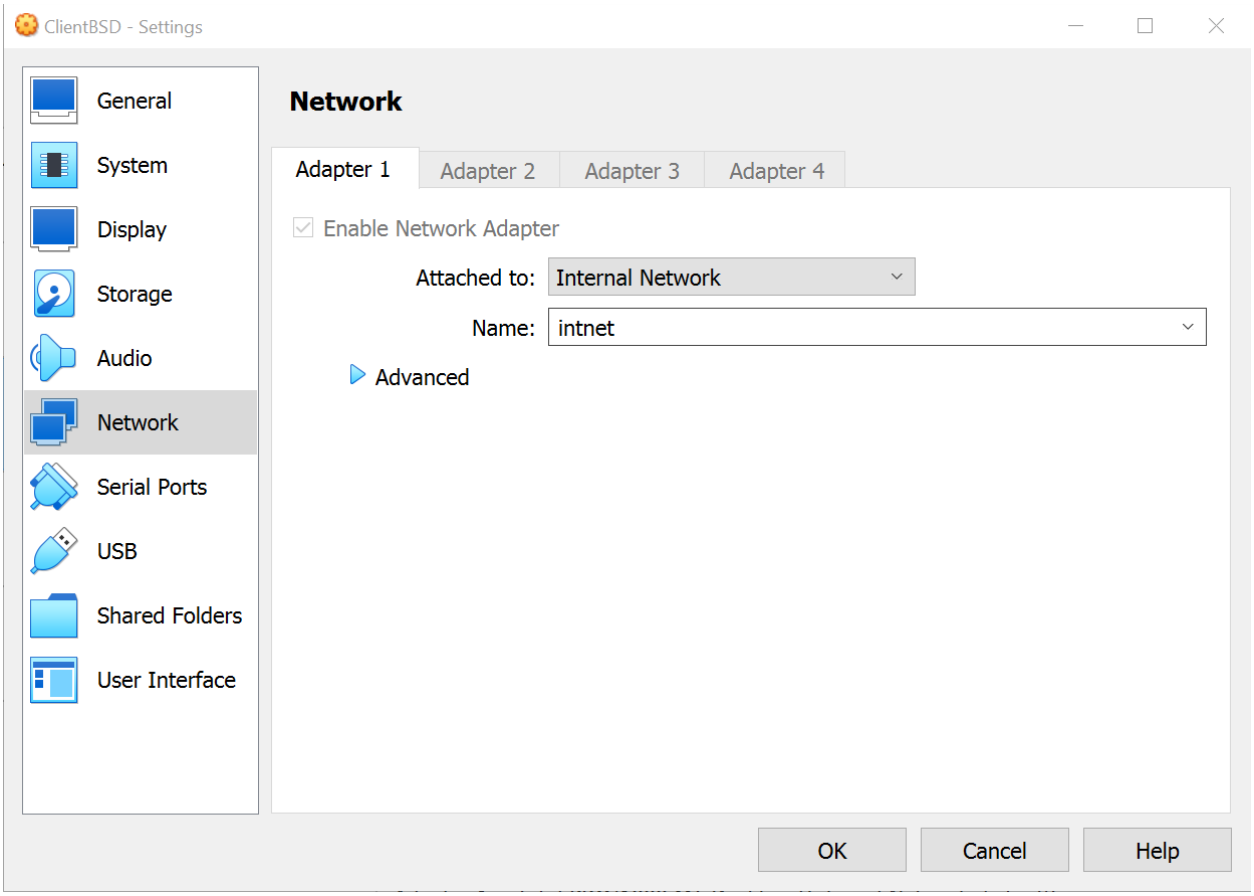
Finally to enable all these services and functionalities we have to change the rc.conf file.



We enable all the services and set up the LAN interface and WAN interface IPs along with enabling SSH.

Setting Up Nodes:

One this to note is that when setting up the nodes they need to be using the internal network adapter



So they are on the same switch. Then in the rc.conf the interface needs to be set to dhcp so that they are assigned an ip from our dhcp server.