

Name : Tarun Kumar Shori  
PRN : 230940127052

## Warewulf v4 installation:-

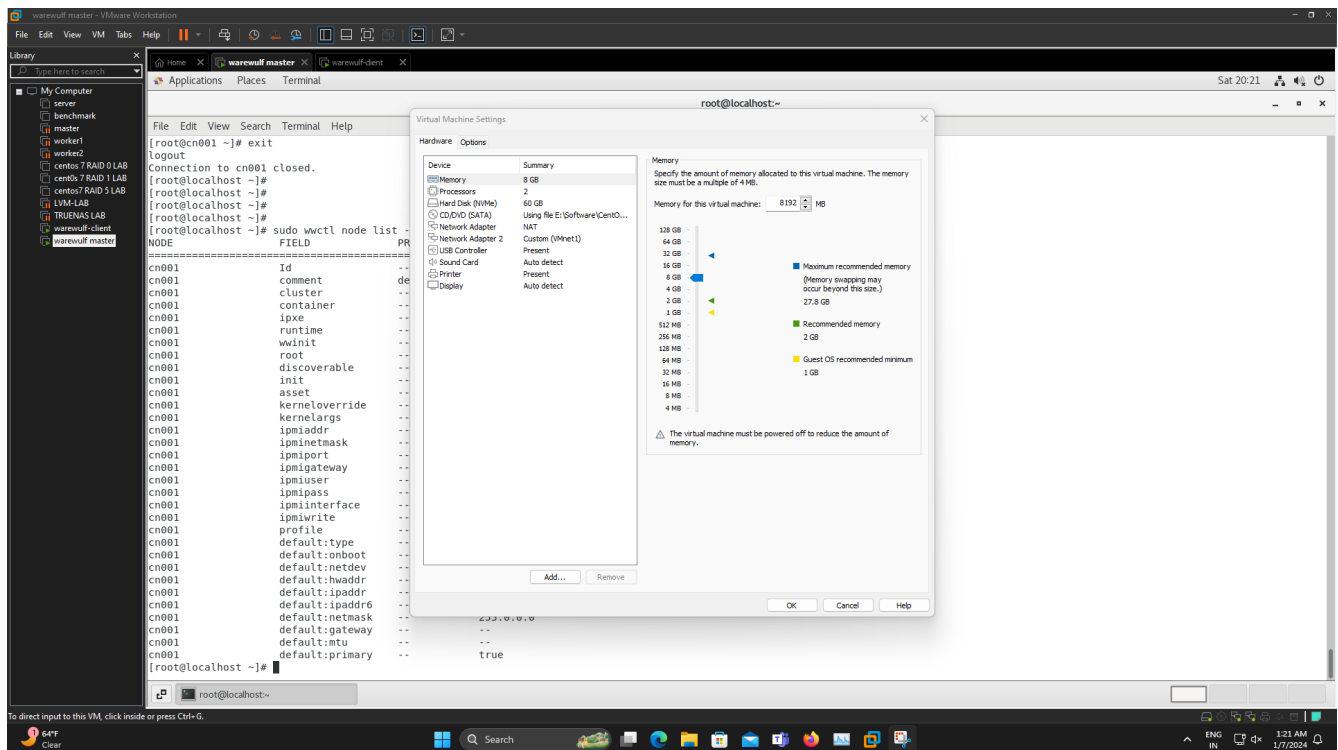
### prerequisites:

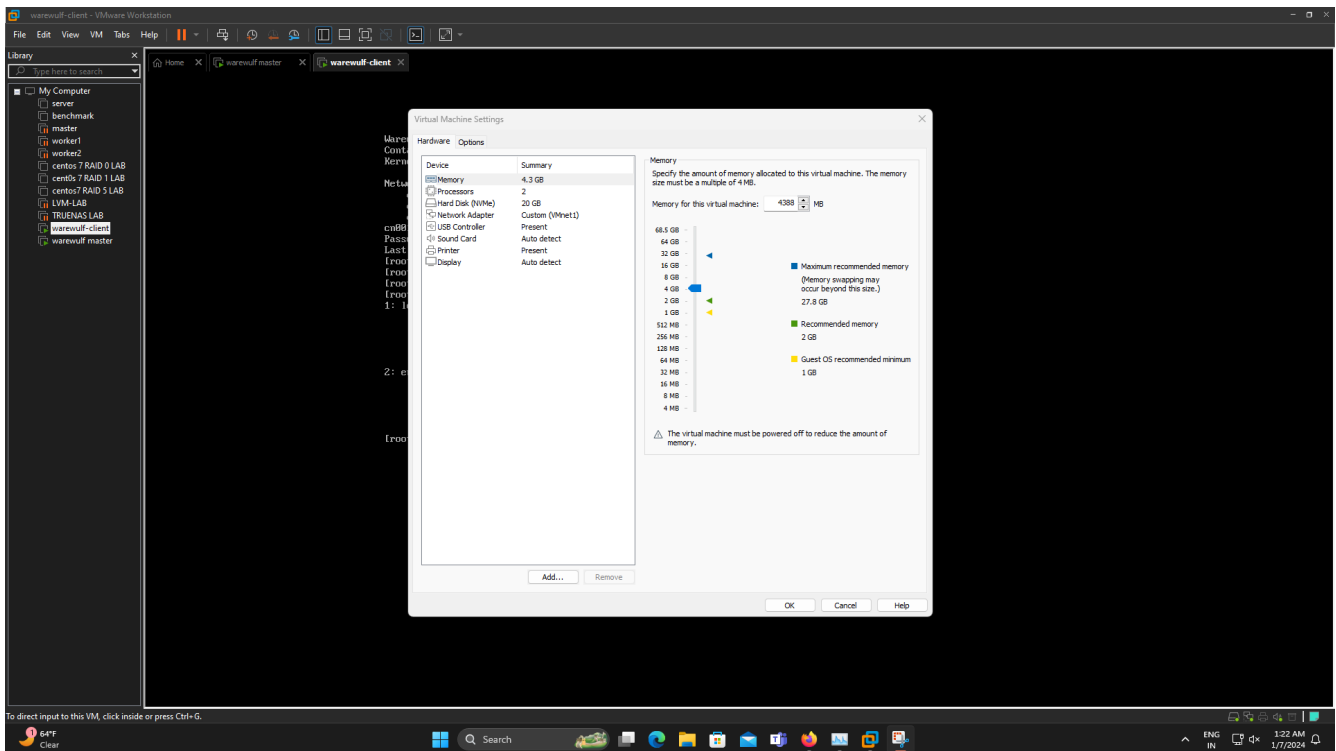
for master :-

OS : Centos-7  
RAM : 8 GB  
HDD : 60 GB  
core : 2

for compute node :-

OS : Rocky-8 Container  
RAM : 4GB  
HDD : 20 GB  
core : 2





# after installing the OS  
 -> Disable Selinux  
 -> Disable Firewall

## Configuration

The physical configuration I used to illustrate the installation of Warewulf was very simple . just one nodes a master node and one compute node .

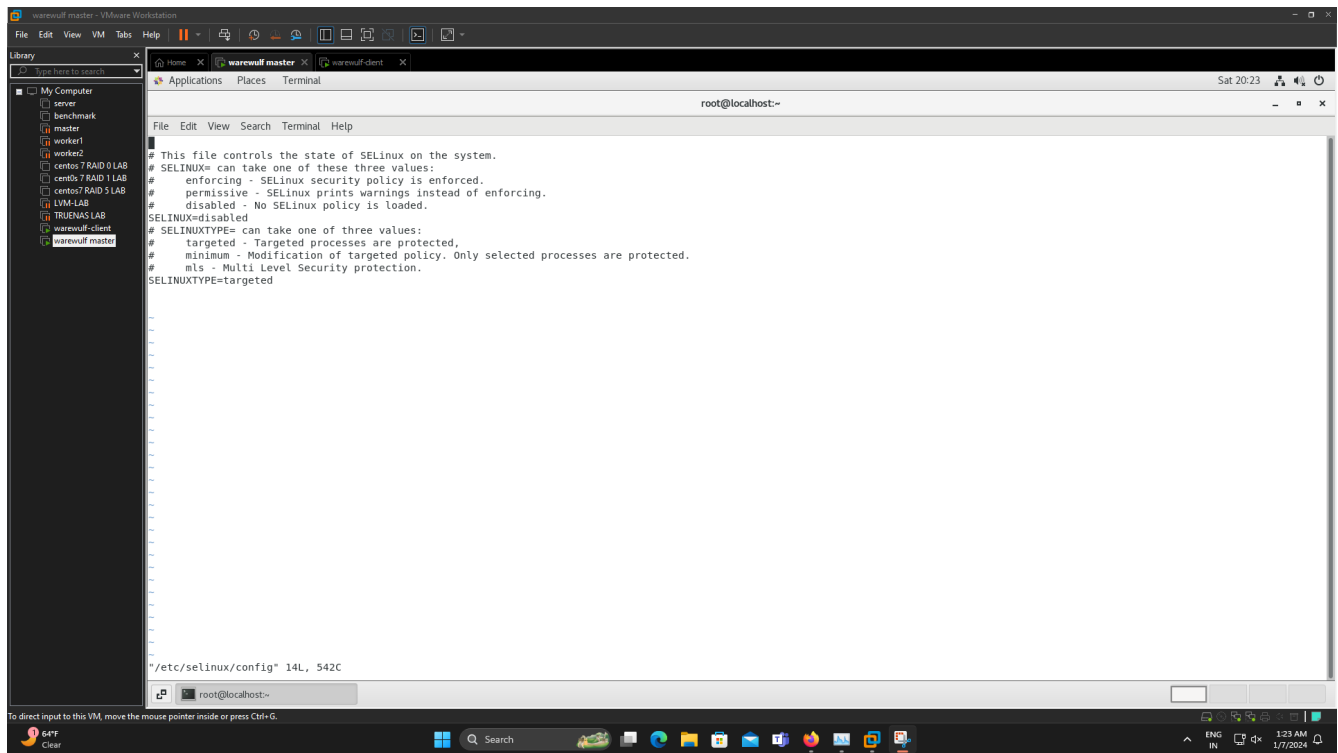
The master node has two network adapter, with one on the NAT connected to the local LAN using the Dynamic Host Configuration Protocol (DHCP). This network adapter is my home network (192.168.144.212). The second network adapter is connected to a private network (192.168.10.1,

subnet mask 255.255.255.0), which follows the network addressing discussed in the [Warewulf documentation](#). I will use the head node as a network filesystem (NFS) server.

## Master Node Installation

I began with a freshly installed Centos -7 distribution. I accepted the defaults to a *Server with a GUI* installation. I did make some changes to the OS. so I installed the Gnome desktop, but this change shouldn't affect anything. Once I installed the OS, I disabled the firewall on master node:

```
$ sudo systemctl stop firewalld  
$ sudo systemctl disable firewalld
```



Additionally, I disabled SELinux for the purposes of this article, but it's possible to keep it functioning while using Warewulf. [Disabling SELinux](#) is the subject of a number of articles online. It requires a reboot to complete.

Don't be afraid to add packages to your head node installation. They are unlikely to interfere with Warewulf. However, Warewulf can take care of installing and configuring three packages it uses for provisioning nodes:

- *dhcp*
- *tftp*
- *nfs-utils*

One more thing, when you create a user such as *administrator* or some other account that will be the primary Warewulf administrator, be sure to allow that user to be an administrator on the system so they can use *sudo* to run commands.

The next step is to [install Warewulf](#). The documentation covers several different methods, from cloning the Git repo to downloading the source. In general, the master node needs access to the outside world .

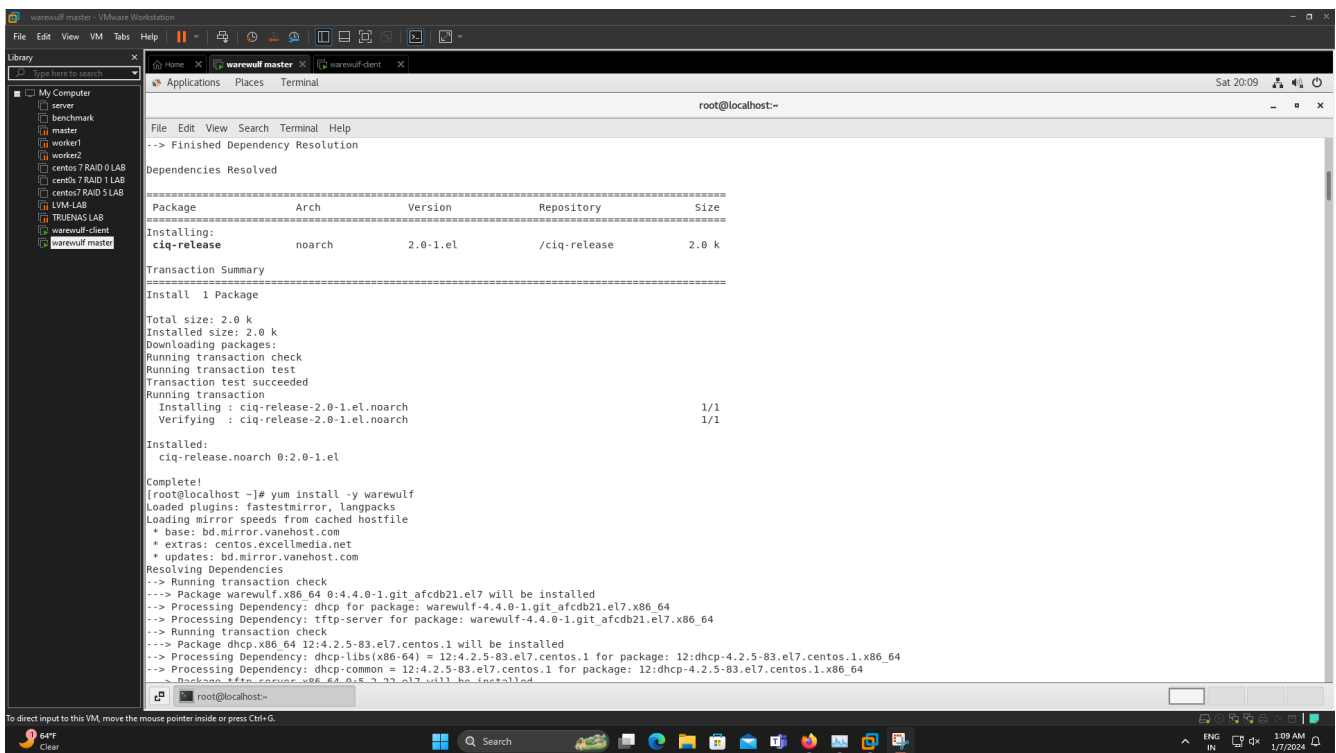
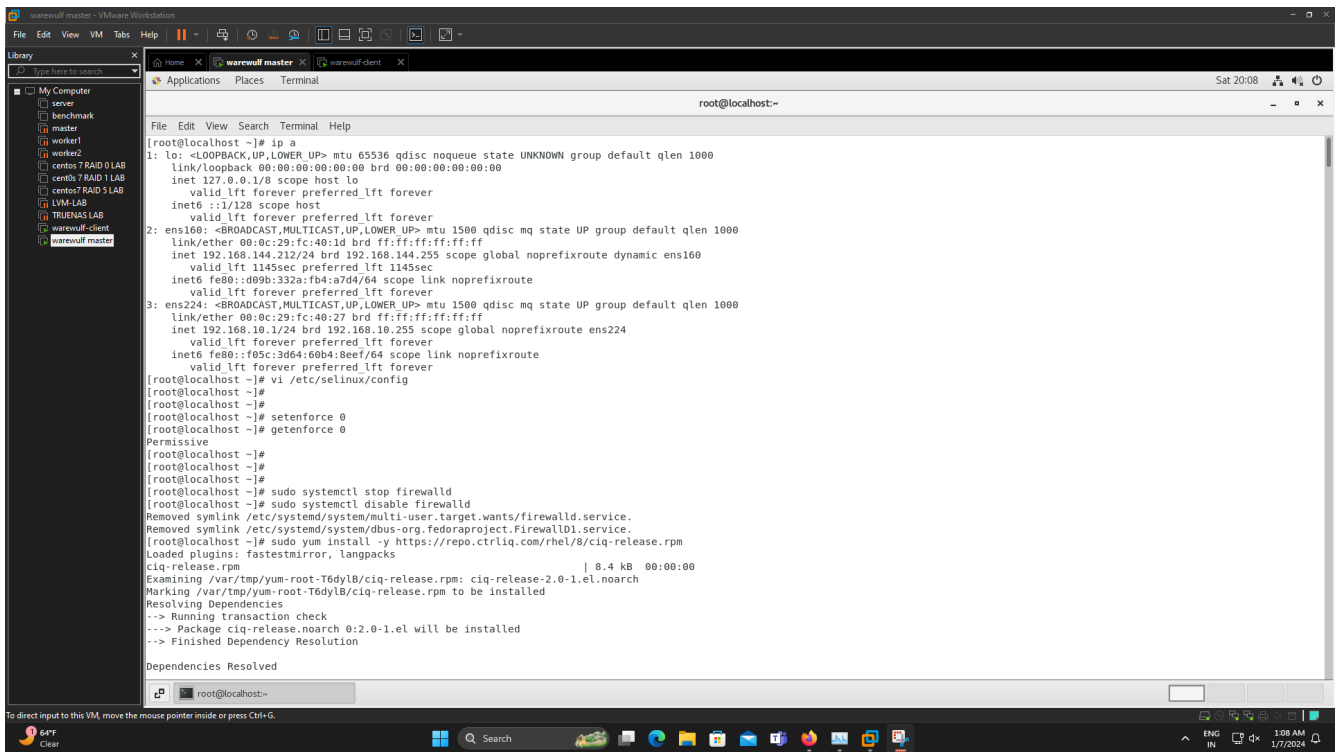
I chose to use binary RPMs because they are a little easier for me to update Warewulf. shows the results of the documentation guidance.

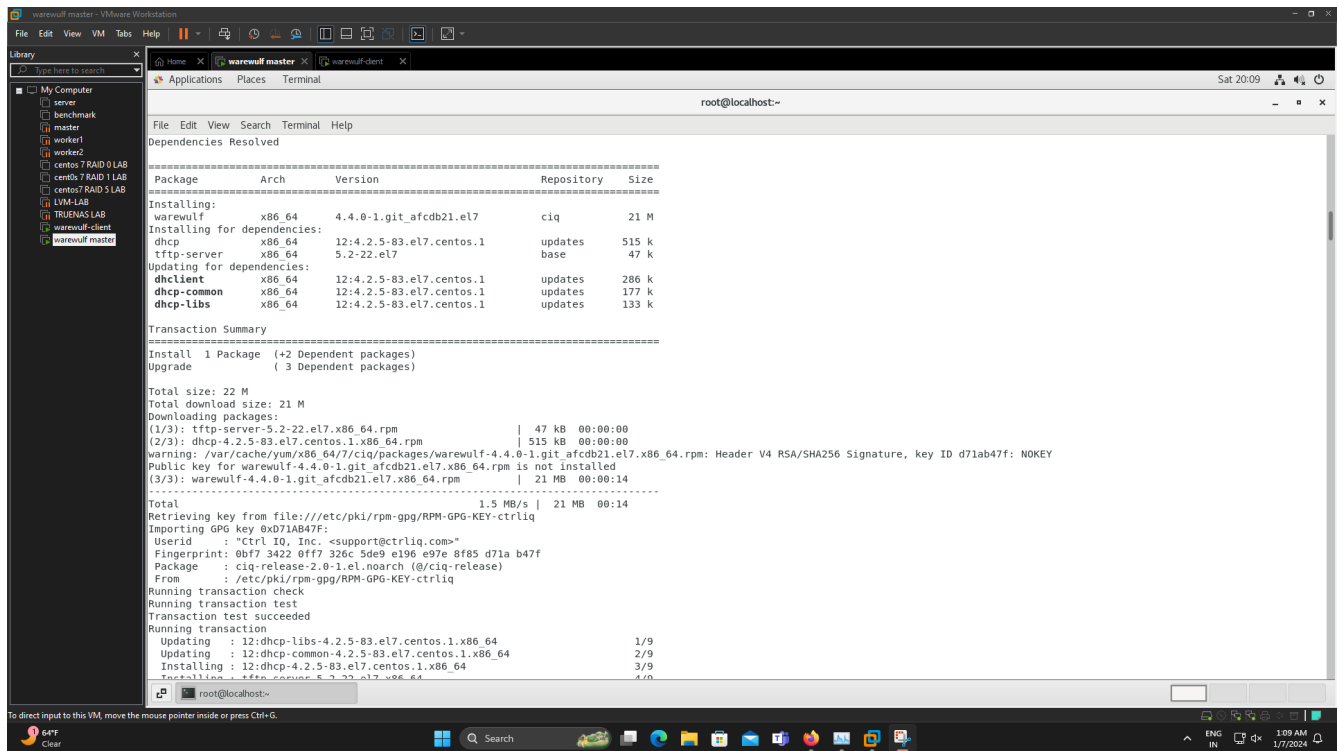
## **Warewulf Installation**

```
$ sudo yum install -y https://repo.ctrlq.com/rhel/8/ciq-release.rpm
$ yum install -y warewulf
```

**Once installed, you can configure Warewulf 4 on the master node (also called the host node).**

## **Configuring Warewulf on the master Node**



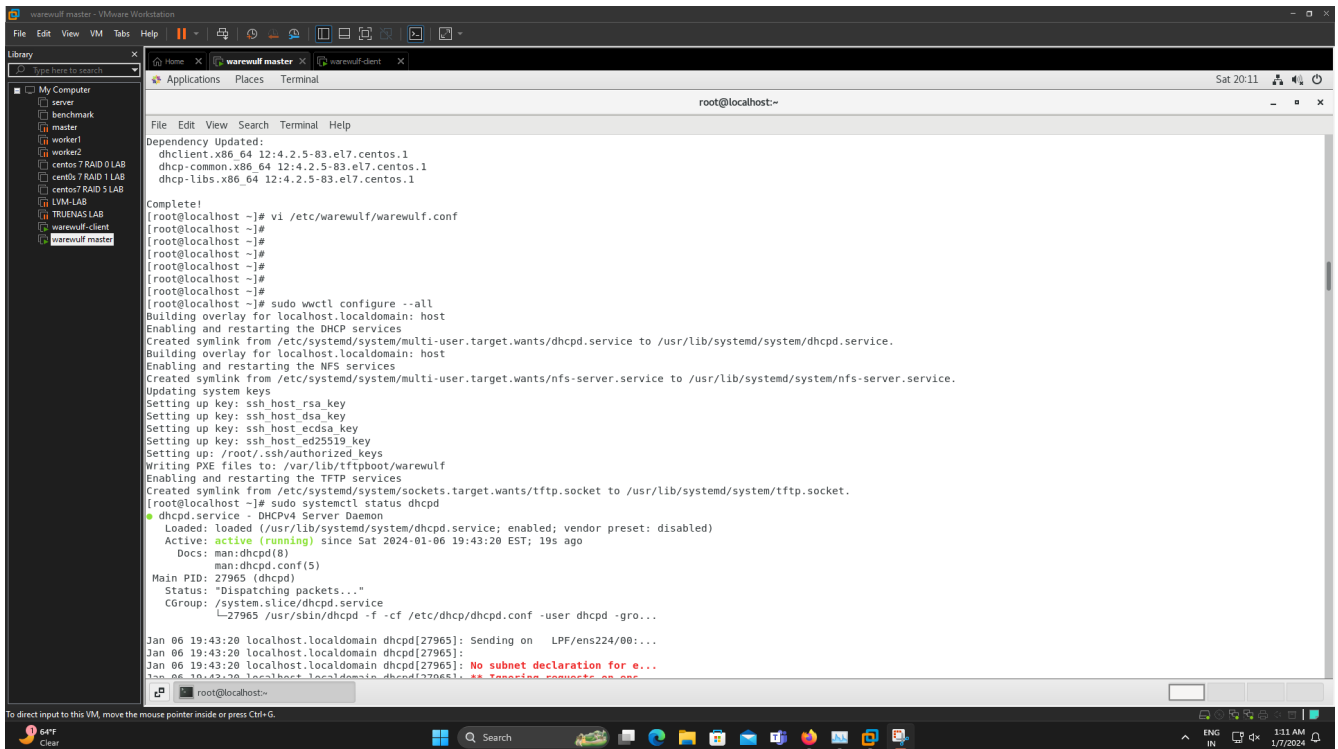
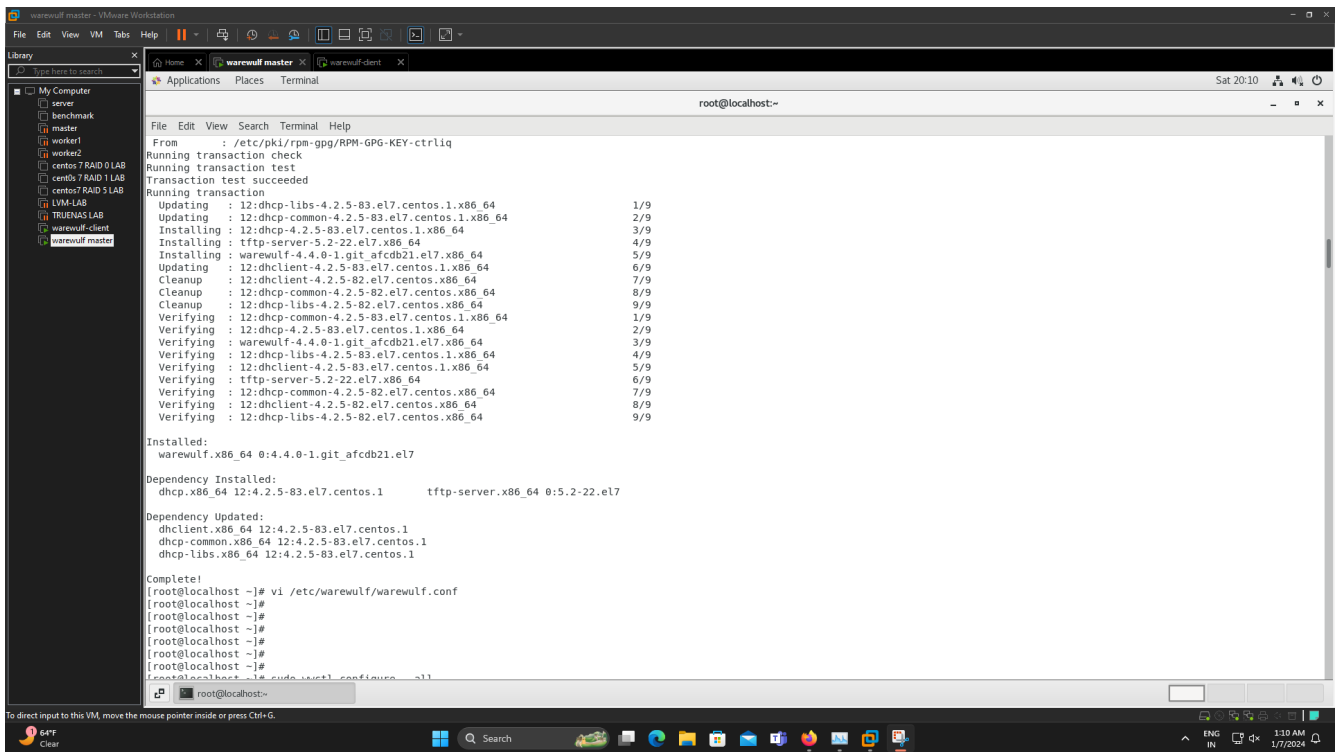


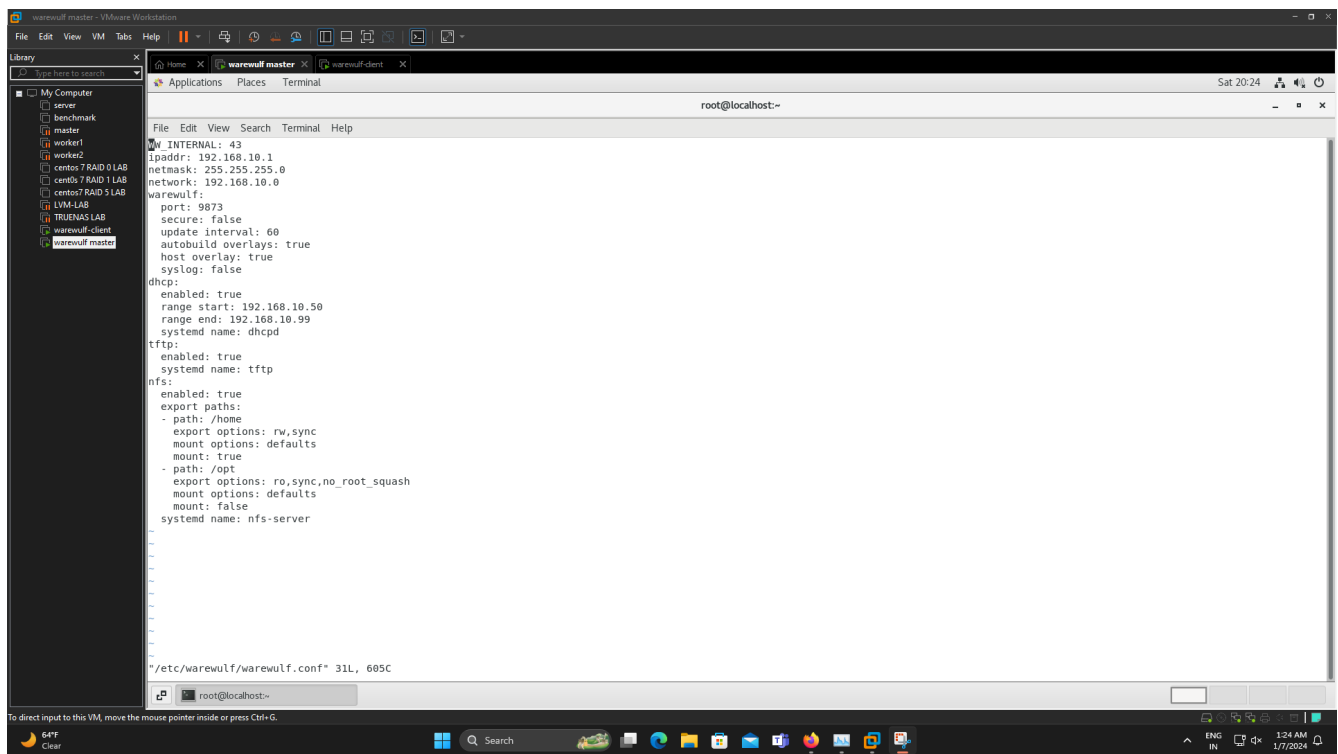
```
File Edit View Search Terminal Help
Dependencies Resolved
=====
Package Arch Version Repository Size
=====
Installing:
warewulf x86_64 4.4.0-1.git_afcdb21.el7 ciq 21 M
Installing for dependencies:
dhcp x86_64 12:4.2.5-83.el7.centos.1 updates 515 k
tftp-server x86_64 5.2-22.el7 base 47 k
Updating for dependencies:
dhclient x86_64 12:4.2.5-83.el7.centos.1 updates 286 k
dhcp-common x86_64 12:4.2.5-83.el7.centos.1 updates 177 k
dhcp-libs x86_64 12:4.2.5-83.el7.centos.1 updates 133 k
=====
Transaction Summary
=====
Install 1 Package (+2 Dependent packages)
Upgrade ( 3 Dependent packages)

Total size: 22 M
Total download size: 21 M
Downloading packages:
(1/3): tftp-server-5.2-22.el7.x86_64.rpm | 47 kB 00:00:00
(2/3): dhcp-4.2.5-83.el7.centos.1.x86_64.rpm | 515 kB 00:00:00
warning: /var/cache/yum/x86_64/7/ciq/packages/warewulf-4.4.0-1.git_afcdb21.el7.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID d71ab47f: NOKEY
Public key for warewulf-4.4.0-1.git_afcdb21.el7.x86_64.rpm is not installed
(3/3): warewulf-4.4.0-1.git_afcdb21.el7.x86_64.rpm | 21 MB 00:00:14
-----
Total
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-ctrlq
Importing GPG key 0xD71AB47F:
Userid : "Ctrl IO, Inc. <support@ctrlq.com>"
Fingerprint: 0b17 3422 0ff7 326c 5d69 e196 e97e 8f85 d71a b47f
Package : ciq-release-2.0-1.el.noarch (@/ciq-release)
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-ctrlq
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : 12:dhcp-libs-4.2.5-83.el7.centos.1.x86_64 1/9
  Updating : 12:dhcp-common-4.2.5-83.el7.centos.1.x86_64 2/9
  Installing : 12:dhcp-4.2.5-83.el7.centos.1.x86_64 3/9
  Installing : tftp-server-5.2-22.el7.x86_64 4/9
```

The configuration files for Warewulf are in `/etc/warewulf`. The primary command you will use to interact with Warewulf is `wwctl`. The first file you should examine is `/etc/warewulf/warewulf.conf`. My version of this file is shown in Listing 2.

## warewulf.conf





The default *warewulf.conf* can be different because it will be based on the IP addresses of the primary NIC in the master node. You will likely have to edit the file, so I'll walk you through that.

-> change ip address : 192.168.10.1

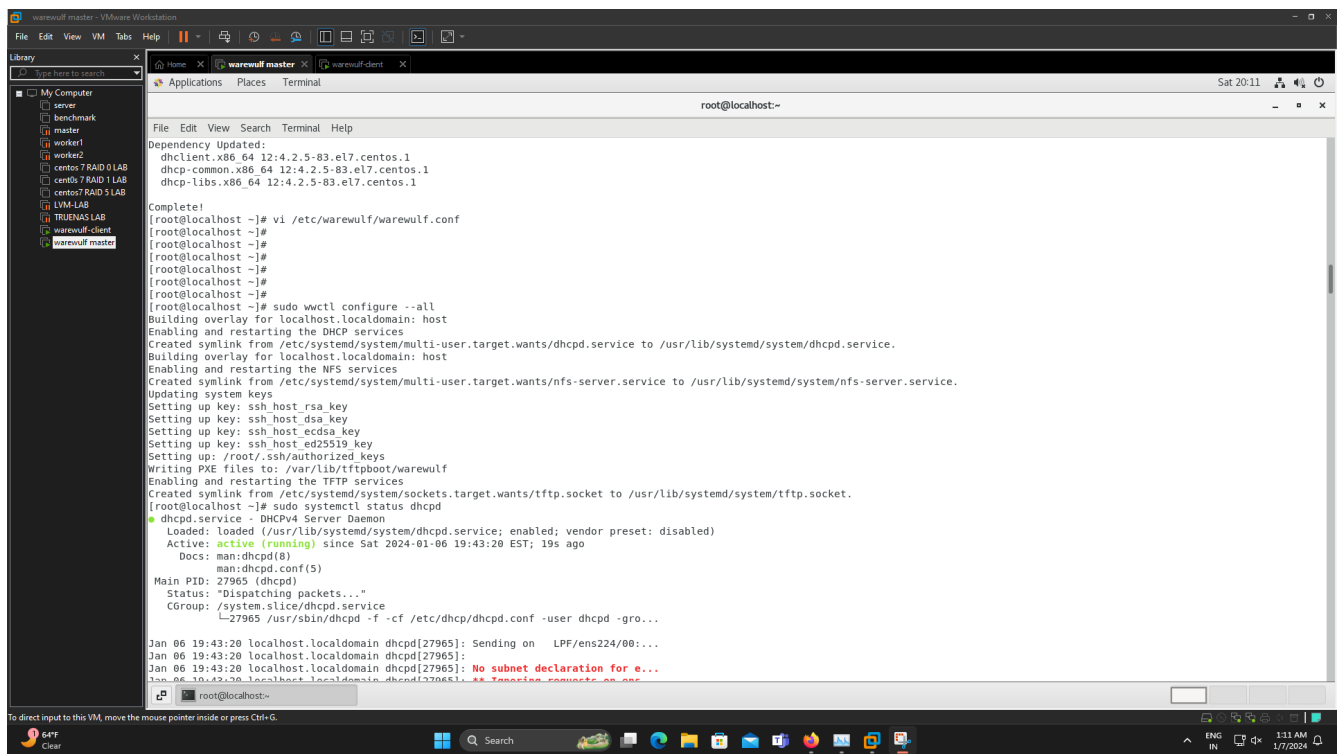
network : 192.168.10.0

dhcp range : 192.168.10.50 – 192.168.10.99

## Initializing Warewulf

At this point, Warewulf has been installed and configured. Now you can initialize it and have it start the needed system services. The general command with some sample output is below.





```
File Edit View Search Terminal Help
root@localhost:~

Dependency Updated:
dhclient.x86_64 12:4.2.5-83.el7.centos.1
dhcp-common.x86_64 12:4.2.5-83.el7.centos.1
dhcp-libraries.x86_64 12:4.2.5-83.el7.centos.1

Complete!
[root@localhost ~]# vi /etc/warewulf/warewulf.conf
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wctl configure --all
Building overlay for localhost.localdomain: host
Enabling and restarting the DHCP services
Created symlink from /etc/systemd/system/multi-user.target.wants/dhcpd.service to /usr/lib/systemd/system/dhcpd.service.
Building overlay for localhost.localdomain: host
Enabling and restarting the NFS services
Created symlink from /etc/systemd/system/multi-user.target.wants/nfs-server.service to /usr/lib/systemd/system/nfs-server.service.
Updating system keys
Setting up key: ssh_host_rsa_key
Setting up key: ssh_host_dsa_key
Setting up key: ssh_host_ecdsa_key
Setting up key: ssh_host_ed25519_key
Setting up: /root/.ssh/authorized_keys
Writing PXE files to: /var/lib/tftpboot/warewulf
Enabling and restarting the TFTP services
Created symlink from /etc/systemd/system/sockets.target.wants/tftp.socket to /usr/lib/systemd/system/tftp.socket.
[root@localhost ~]# sudo systemctl status dhcpd
● dhcpd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2024-01-06 19:43:20 EST; 19s ago
     Docs: man:dhcpd.conf(8)
           man:dhcpd.conf(5)
   Main PID: 27965 (dhcpd)
   Status: "Dispatching packets..."
   CGroup: /system.slice/dhcpd.service
           └─27965 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -gro...

Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: Sending on LPF/ens224/00:...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]:
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: No subnet declaration for e...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: ** Warning: requests on net...
```

## Checking the Services

System services don't always come up the way I intend, so I always, always check them. I think this is very important, particularly after a reboot of the master node. I'll start with DHCP .The first lines of data with date stamps can be ignored. the DHCP system service is up and functioning.

### Checking DHCP Service

Checking TFTP Service

Checking NFS Service

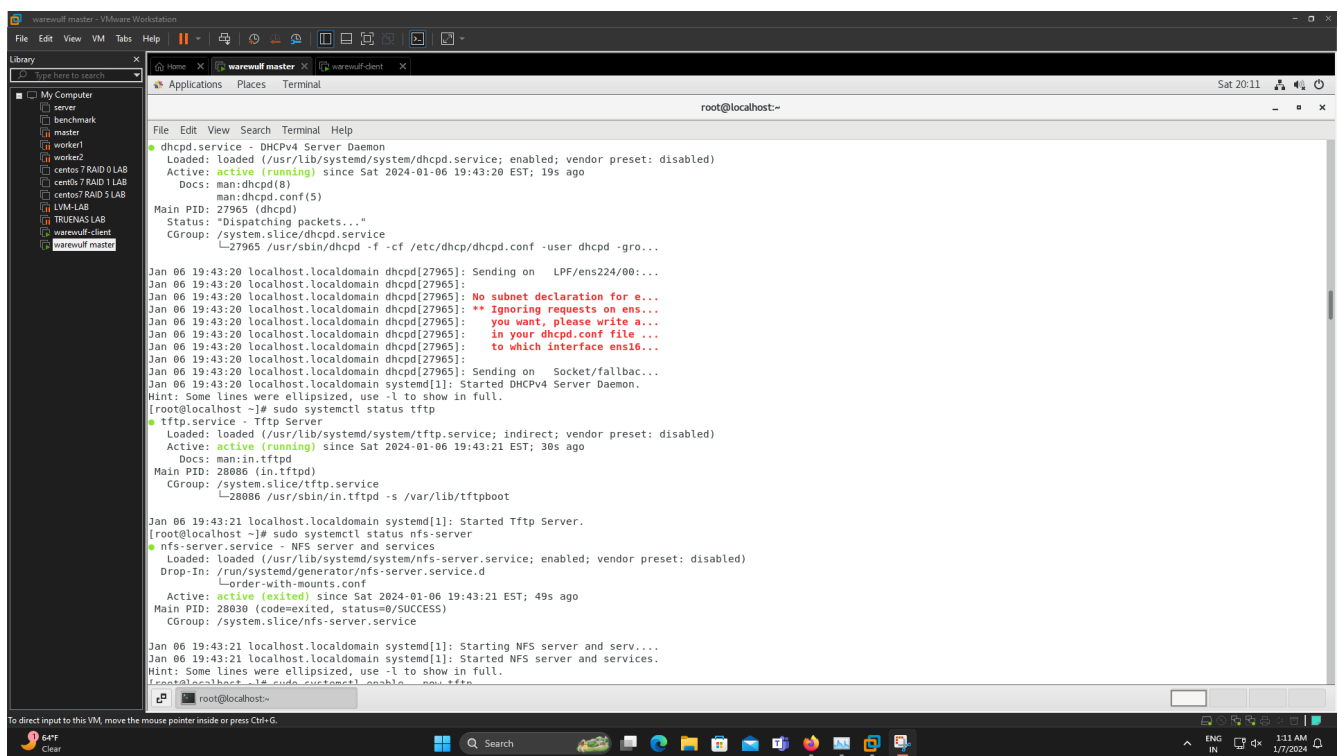
You can also verify with the *sudo exportfs* command. By default, you should see two filesystems exported: */home* and */opt*.

If the services are not running, the commands

```
$ sudo systemctl enable --now tftp
```

```
$ sudo systemctl start tftp
```

enable and start the services. If you don't use the `--now` option, they start on the next boot. Strictly speaking, the second command shouldn't be necessary, but I like to use it to be sure. (It won't hurt to use it.)



```
File Edit View Search Terminal Help
● dhcpd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Sat 2024-01-06 19:43:20 EST; 19s ago
     Docs: man:dhcpd(8)
           man:dhcpd.conf(5)
   Main PID: 27965 (dhcpd)
   Status: "Dispatching packets..."
   CGroup: /system.slice/dhcpd.service
           └─27965 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpd -gro...

Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: Sending on  LPF/ens224/00:...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: No subnet declaration for e...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: ** Ignoring requests on ens...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: you want, please write a...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: in your dhcpd.conf file ...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: to which interface ens16...
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]:
Jan 06 19:43:20 localhost.localdomain dhcpd[27965]: Sending on  Socket/fallbac...
Jan 06 19:43:20 localhost.localdomain systemd[1]: Started DHCPv4 Server Daemon.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# sudo systemctl status tftp
● tftp.service - Tftp Server
   Loaded: loaded (/usr/lib/systemd/system/tftp.service; indirect; vendor preset: disabled)
   Active: active (running) since Sat 2024-01-06 19:43:21 EST; 30s ago
     Docs: man:in.tftpd
   Main PID: 28086 (in.tftpd)
   CGroup: /system.slice/tftp.service
           └─28086 /usr/sbin/in.tftpd -s /var/lib/tftpboot

Jan 06 19:43:21 localhost.localdomain systemd[1]: Started Tftp Server.
[root@localhost ~]# sudo systemctl status nfs-server
● nfs-server.service - NFS server and services
   Loaded: loaded (/usr/lib/systemd/system/nfs-server.service; enabled; vendor preset: disabled)
   Drop-In: /run/systemd/generator/nfs-server.service.d
           └─order-with-mounts.conf
   Active: active (exited) since Sat 2024-01-06 19:43:21 EST; 49s ago
   Main PID: 28030 (code=exited, status=0/SUCCESS)
   CGroup: /system.slice/nfs-server.service

Jan 06 19:43:21 localhost.localdomain systemd[1]: Starting NFS server and serv...
Jan 06 19:43:21 localhost.localdomain systemd[1]: Started NFS server and services.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# sudo systemctl enable --now tftp
```

All of the dependent services have been started and are running at this point, so it's time to start the Warewulf daemon itself .

### Starting the Warewulf Daemon

```
$ sudo systemctl enable --now warewulfd  
$ sudo wwctl server status
```

As part of your system checks, you should also check *warewulfd*. Although I've never had any problems with it, it is always good to check.

At this point, all services should be up and running. If you like, look at the Warewulf logs, which should be pretty empty because the cluster is newly created:

```
$ sudo more /var/log/warewulfd.log
```

## Container Management for Compute Nodes

If you have used Warewulf before, you are familiar with VNFS. As the name suggests, it's a virtual filesystem that is basically the OS image for the compute nodes. In Warewulf 3 and before, it was created in a *chroot* directory, which in effect is a container but is really a directory on the head node system that serves as a root directory. Warewulf 4 has moved to using the term "container" more explicitly and using the container apparatus (*container*) for creating and using compute node OS images.

If you pull and try to boot a container such as *docker://rockylinux*, it will not boot properly; rather, it will boot into single-user mode and get stuck. It is better to look for containers in [Docker Hub](#) and even ask on the Warewulf mailing list or Slack channel about bootable containers.

I chose to pull a Rocky 8 container from the *warewulf* repo on Docker Hub . A quick note in case it is subtle: the head node needs access to the outside world to pull the container unless you have added it to a local repository.

### Pull Container on Docker Hub

```
warewulf-master: VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
centos 7 RAID 1 LAB
centos 7 RAID 5 LAB
LVM-LAB
TRUENAS-LAB
warewulf-client
warewulf-master
Applications Places Terminal
root@localhost:~
File Edit View Search Terminal Help
L-order-with-mounts.conf
Active: active (exited) since Sat 2024-01-06 19:43:21 EST; 49s ago
Main PID: 28030 (code=exited, status=0/SUCCESS)
CGroup: /system.slice/nfs-server.service

Jan 06 19:43:21 localhost.localdomain systemd[1]: Starting NFS server and serv....
Jan 06 19:43:21 localhost.localdomain systemd[1]: Started NFS server and services.
Hint: Some lines were ellipsized, use -l to show in full.
[root@localhost ~]# sudo systemctl enable --now tftp
[root@localhost ~]# sudo systemctl start tftp
[root@localhost ~]# sudo systemctl enable --now warewulf
Created symlink from /etc/systemd/system/multi-user.target.wants/warewulf.service to /usr/lib/systemd/system/warewulf.service.
[root@localhost ~]# sudo wwctl server status
SERV : Warewulf server is running at PID: 20217
[root@localhost ~]# sudo more /var/log/warewulf.log
[Sat Jan 6 19:44:43 EST 2024] SERV : Starting HTTPD REST service on port 9873
[root@localhost ~]# sudo wwctl container import docker://warewulf/rocky rocky-8
Copying blob 5461c86c2e54 done
Copying blob 2ef2dfdac739 done
Copying blob e4aa6f711855 done
Copying blob b8fd1f017bbd done
Copying blob 9231c4f09ac0 done
Copying blob 6c1511b659e1 done
Copying blob 154247c0c8a1 done
Copying config 91226788d9 done
Writing manifest to image destination
Storing signatures
2024/01/06 19:47:07 info unpack layer: sha256:5461c86c2e54b0c326aaf624b5a7719ff4b3a5362f1c8c1f30fea5a1d
2024/01/06 19:47:13 info unpack layer: sha256:2ef2dfdac739d58e532b819c29487535bf6cbfd6f850a82ec6fe911975c3dc08
2024/01/06 19:47:27 info unpack layer: sha256:e4aa6f71185580712bf6110e08622228c7d84e4bad13ce7aff9d0a132108f88e
2024/01/06 19:47:27 info unpack layer: sha256:b8fd1f017bbd97acbb6c77bb59bf92c8d7c1048dd166a26d9f04c5538c83b7d3
2024/01/06 19:47:28 info unpack layer: sha256:9231c4f09ac0e58207e33b7046f2706fdd7f7a60315ad2718396acdf154be14
2024/01/06 19:47:28 info unpack layer: sha256:6c1511b659e1f09b3e322a9db3fcd135dc2acbbf69b4cc0524be63c36511922
2024/01/06 19:47:28 info unpack layer: sha256:154247c0c8a17ac52d9a5ffeb7a1291b40a8a8342410ca9f17de24d2335abb7
Updating the container's /etc/resolv.conf
WARN : user: systemd-coredump:999:997 not present on host
WARN : user: systemd-resolve:193:193 not present on host
ERROR : error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
ERROR: error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# sudo wwctl container import --force docker://warewulf/rocky my-rocky-container
Copying blob 5461c86c2e54-skiped, already exists
```

\$ sudo wwctl container list

The image shows a Kali Linux virtual machine interface. At the top, there's a menu bar with 'File', 'Edit', 'View', 'VM', 'Tools', and 'Help'. Below the menu is a toolbar with various icons. The main window is titled 'Library' and contains a sidebar with a tree view showing the file system structure: 'My Computer' (server, benchmark, master, woker1, woker2, centos7 RAID 0 LAB, centos7 RAID 1 LAB, LINUX LAB, TRUENAS LAB, warewolf-client, warewolf-master). The main pane shows a terminal window with the command prompt 'root@localhost:~'. The terminal output shows the installation of a container named 'rocky' using Docker and Wcctl. The output includes the following commands and their results: 

```
File Edit View Search Terminal Help

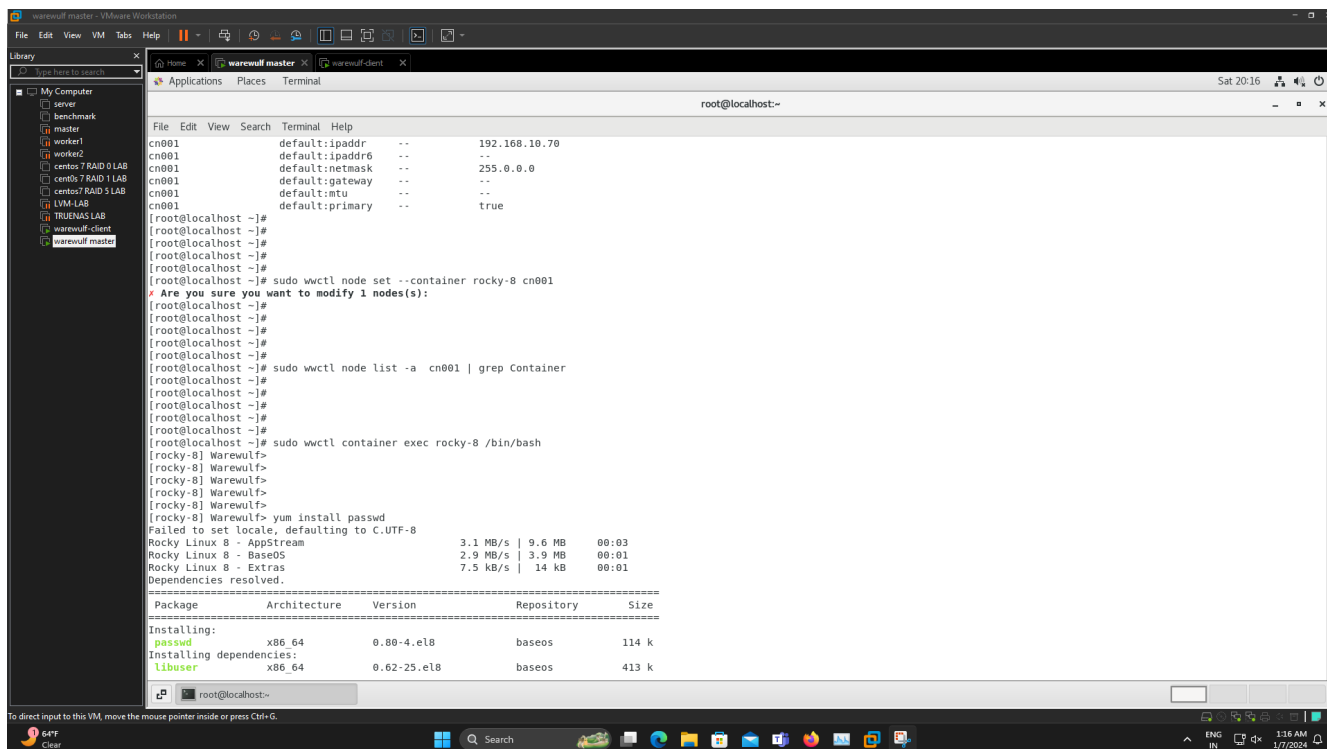
root@localhost:~

2024/01/06 19:47:27 info unpack Layer: sha256:b8fd1f017b9d97acb86c77b659bf92cd87c1048d166a26d9f04c5538c83b7d3
2024/01/06 19:47:27 info unpack Layer: sha256:9231c14f09ac0e8207e337b46b2706fde77a60315ad2718306acdf540e14
2024/01/06 19:47:28 info unpack Layer: sha256:6c1511b659e1fb9b3e3223a9db3fcd135dc2acbbf69b4ccb524be63c36511922
2024/01/06 19:47:28 info unpack Layer: sha256:154247c0c8a17ac52d9a5f7ebb7a1291b40a0834210ca9f17de24d2335abb7
Updating the container's /etc/resolv.conf
WARN : user: systemd-coredump:999:997 not present on host
WARN : user: systemd-resolve:193:193 not present on host
WARN : uid(999) collision for host: systemd-coredump and container: polkitd
ERROR : error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
ERROR: error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# sudo wcctl container import --force docker://warewolf/rocky my-rocky-container
Copying blob 5461c86c2e54 skipped: already exists
Copying blob 2ef2dfda739 skipped: already exists
Copying blob e4aa6f711855 skipped: already exists
Copying blob b8fd1f017b9d skipped: already exists
Copying blob 9231c14f09ac0 skipped: already exists
Copying blob 6c1511b659e1 skipped: already exists
Copying blob 154247c0c8a1 [-----] 0.0b / 0.0b
Copying config 91226788d9 done
Writing manifest to image destination
Storing signatures
2024/01/06 19:48:54 info unpack Layer: sha256:5461c86c2e548f89cd5d0c326aaf624b5a7f19ff4b3a5362f1c8c1f30fe5a1d
2024/01/06 19:48:56 info unpack Layer: sha256:2ef2dfda739d58e32b019c2948753bf6bdf6f050a82ec6f911975c3dc08
2024/01/06 19:49:01 info unpack Layer: sha256:4e4ab0711813108b8712b6f1108586222c8b619b6a132108f08
2024/01/06 19:49:01 info unpack Layer: sha256:b8fd1f017b9d97acb86c77b659bf92cd87c1048d166a26d9f04c5538c83b7d3
2024/01/06 19:49:02 info unpack Layer: sha256:9231c14f09ac0e8207e337b46b2706fde77a60315ad2718306acdf540e14
2024/01/06 19:49:02 info unpack Layer: sha256:6c1511b659e1fb9b3e3223a9db3fcd135dc2acbbf69b4ccb524be63c36511922
2024/01/06 19:49:02 info unpack Layer: sha256:154247c0c8a17ac52d9a5f7ebb7a1291b40a0834210ca9f17de24d2335abb7
Updating the container's /etc/resolv.conf
WARN : user: systemd-coredump:999:997 not present on host
WARN : user: systemd-resolve:193:193 not present on host
WARN : uid(999) collision for host: systemd-coredump and container: polkitd
ERROR : error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
ERROR: error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# wcctl container import docker://warewolf/rocky rocky-8
ERROR : VNFs Name exists, specify -force, --update, or choose a different name: rocky-8
ERROR: VNFs Name exists, specify -force, --update, or choose a different name: rocky-8
[root@localhost ~]# wcctl container syncuser --write rocky-8
WARN : user: systemd-coredump:999:997 not present on host
WARN : user: systemd-resolve:193:193 not present on host
```

## Modifying Containers on the master Node

The image shows a VMware Workstation interface with a virtual machine named 'warewolf-master' running. The terminal window is open, showing the following commands and output:  

```
add this user to /etc/passwd on host
ERROR: error: user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# wcctl container import docker://warewolf/rocky rocky-8
ERROR: VNFs Name exists, specify --force, --update, or choose a different name: rocky-8
ERROR: VNFs Name exists, specify --force, --update, or choose a different name: rocky-8
[root@localhost ~]# wcctl container syncuser --write rocky-8
WARN: user: systemd-coredump:999:997 not present on host
WARN: user: systemd-resolve:193:193 not present on host
WARN: uid(999) collision for host: systemd-coredump and container: polkitd
ERROR: Error in synchronize: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# sudo wcctl container list
ERROR: stat /var/lib/warewolf/container/my-rocky-container.img: no such file or directory
ERROR: stat /var/lib/warewolf/container/my-rocky-container.img: no such file or directory
ERROR: stat /var/lib/warewolf/container/my-rocky-container.img.gz: no such file or directory
ERROR: stat /var/lib/warewolf/container/rocky-8.img: no such file or directory
ERROR: stat /var/lib/warewolf/container/rocky-8.img: no such file or directory
ERROR: stat /var/lib/warewolf/container/rocky-8.img.gz: no such file or directory
CONTAINER NAME      NODES      KERNEL VERSION  CREATION TIME      MODIFICATION TIME      SIZE
my-rocky-container 0          4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:48 EST  31 Dec 69 19:00 EST  1.0 GiB
rocky-8            0          4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:45 EST  31 Dec 69 19:00 EST  1.0 GiB
[root@localhost ~]# sudo wcctl container exec rocky-8 /bin/bash
[rocky-8] Warewulf> cat /etc/rocky-release
Rocky Linux release 8.7 (Green Obsidian)
[rocky-8] Warewulf> exit
exit
Rebuilding container...
Created image for VNFs container rocky-8: /var/lib/warewolf/container/rocky-8.img
Compressed image for VNFs container rocky-8: /var/lib/warewolf/container/rocky-8.img.gz
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wcctl container list
ERROR: stat /var/lib/warewolf/container/my-rocky-container.img: no such file or directory
ERROR: stat /var/lib/warewolf/container/my-rocky-container.img: no such file or directory
ERROR: stat /var/lib/warewolf/container/my-rocky-container.img.gz: no such file or directory
CONTAINER NAME      NODES      KERNEL VERSION  CREATION TIME      MODIFICATION TIME      SIZE
my-rocky-container 0          4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:48 EST  31 Dec 69 19:00 EST  1.0 GiB
rocky-8            0          4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:45 EST  06 Jan 24 19:51 EST  2.5 GiB
[root@localhost ~]#
```



## Compute Node Kernel

Compute nodes need a Linux kernel to boot. Warewulf allows you to separate the kernel from the container, which can give you great flexibility; however, you have to check that the container can boot and function with a different kernel.

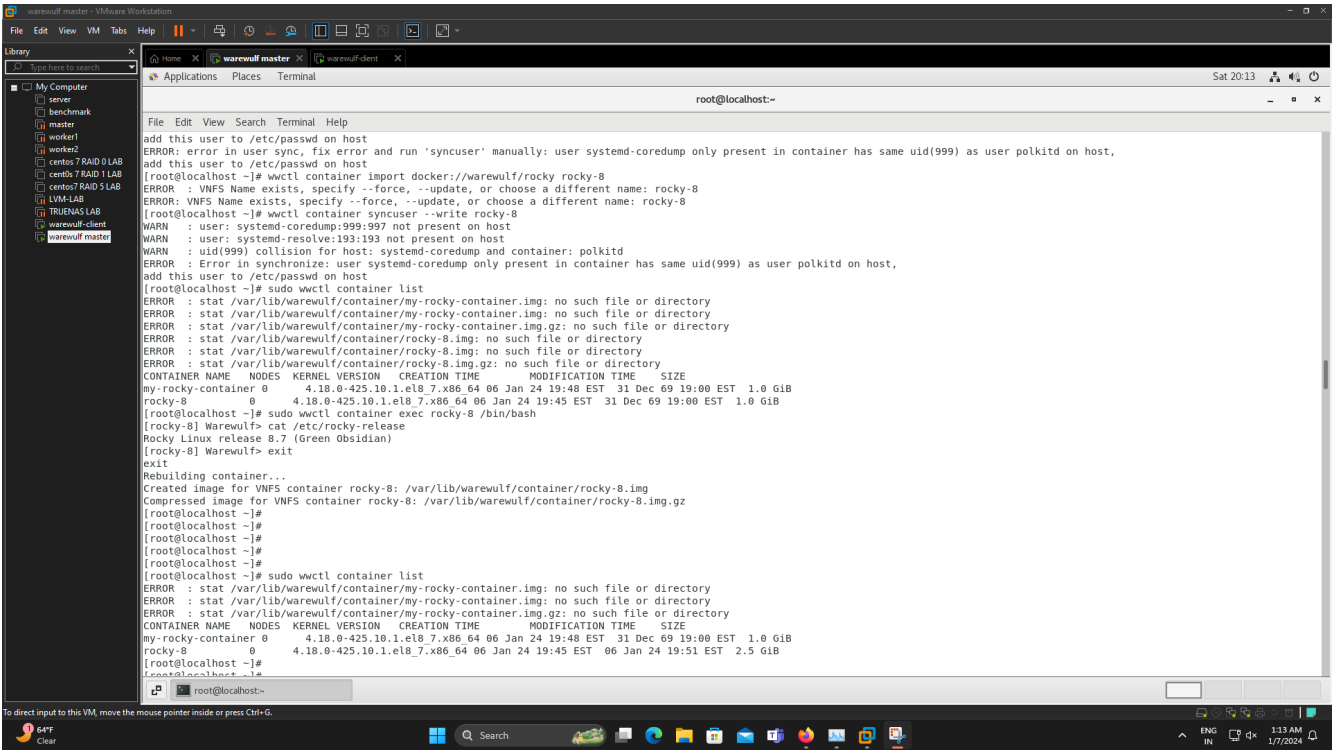
The simplest way is to install the desired kernel in the container or use the one that comes with the container. Warewulf will locate the kernel in an uploaded container by default and use it for any node that uses that container. Note that it can use the same container for nodes but have them boot with different kernels, but please test the combination before putting it in production.

Combining a container with a kernel is done with a *profile*. Personally, I like building containers with specific kernels rather than mixing and matching, but I can see the utility of being able to use separate kernels.

You can see the kernel used in the container with the command:

\$ sudo wwctl container list

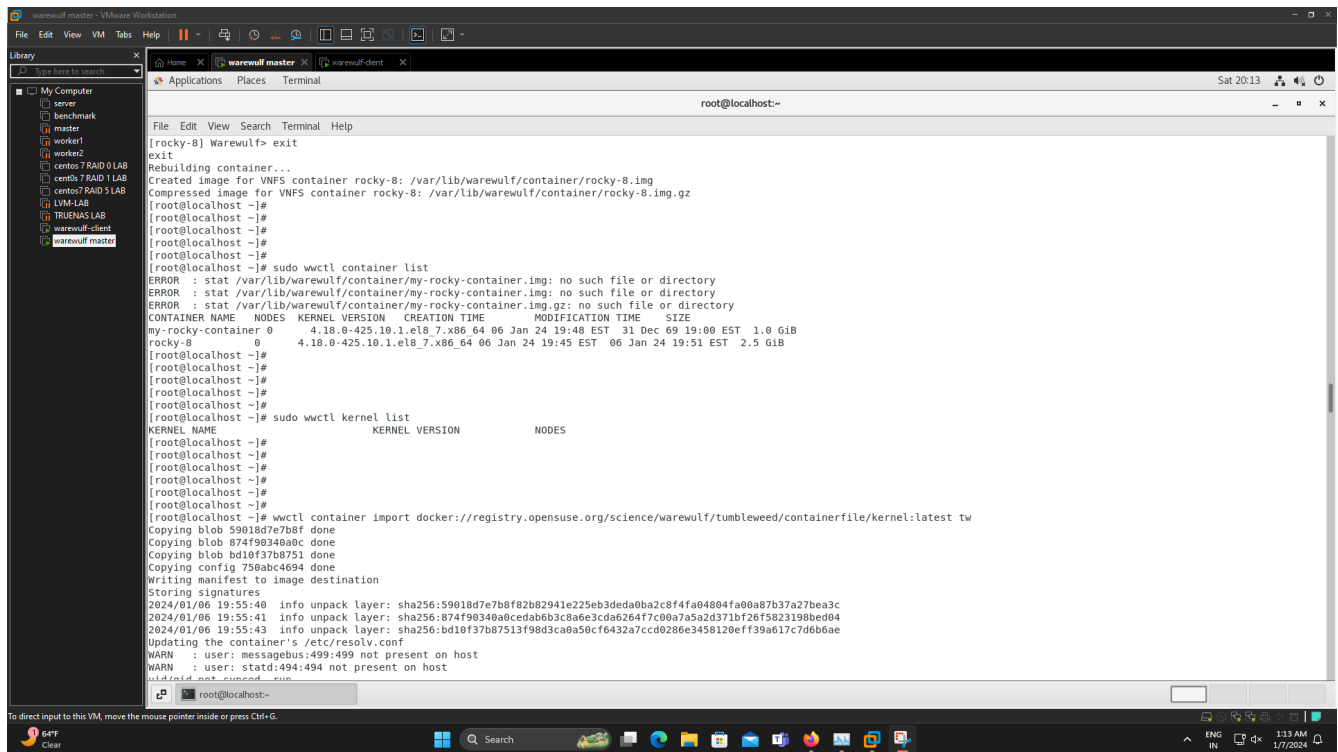
\$ sudo wwctl kernel list



```
warewulf-master - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
server
benchmark
master
worklet
worklet2
centos7 RAID 0 LAB
centos7 RAID 1 LAB
LVM LAB
TRUENAS LAB
warewulf-client
warewulf-master
warewulf-master
Applications Places Terminal
root@localhost:~
File Edit View Search Terminal Help
add this user to /etc/passwd on host
ERROR: error in user sync, fix error and run 'syncuser' manually: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# wwctl container import docker://warewulf/rocky rocky-8
ERROR: VNFs Name exists, specify --force, --update, or choose a different name: rocky-8
[root@localhost ~]# wwctl container syncuser --write rocky-8
WARN : user: systemd-coredump:999:997 not present on host
WARN : user: systemd-resolve:193:193 not present on host
WARN : uid(999) collision for host: systemd-coredump and container: polkitd
ERROR : Error in synchronize: user systemd-coredump only present in container has same uid(999) as user polkitd on host,
add this user to /etc/passwd on host
[root@localhost ~]# sudo wwctl container list
ERROR : stat /var/lib/warewulf/container/my-rocky-container.img: no such file or directory
ERROR : stat /var/lib/warewulf/container/my-rocky-container.img.gz: no such file or directory
ERROR : stat /var/lib/warewulf/container/my-rocky-container.img.gz: no such file or directory
ERROR : stat /var/lib/warewulf/container/rocky-8.img: no such file or directory
ERROR : stat /var/lib/warewulf/container/rocky-8.img.gz: no such file or directory
CONTAINER NAME  NODES  KERNEL VERSION  CREATION TIME  MODIFICATION TIME  SIZE
my-rocky-container 0      4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:48 EST  31 Dec 69 19:00 EST  1.0 GiB
rocky-8         0      4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:45 EST  31 Dec 69 19:00 EST  1.0 GiB
[root@localhost ~]# sudo wwctl container exec rocky-8 /bin/bash
[rocky-8] Warewulf> cat /etc/rocky-release
Rocky Linux release 8.7 (Green Obsidian)
[rocky-8] Warewulf> exit
exit
Rebuilding container...
Created image for VNFs container rocky-8: /var/lib/warewulf/container/rocky-8.img
Compressed image for VNFs container rocky-8: /var/lib/warewulf/container/rocky-8.img.gz
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wwctl container list
ERROR : stat /var/lib/warewulf/container/my-rocky-container.img: no such file or directory
ERROR : stat /var/lib/warewulf/container/my-rocky-container.img: no such file or directory
ERROR : stat /var/lib/warewulf/container/my-rocky-container.img.gz: no such file or directory
CONTAINER NAME  NODES  KERNEL VERSION  CREATION TIME  MODIFICATION TIME  SIZE
my-rocky-container 0      4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:48 EST  31 Dec 69 19:00 EST  1.0 GiB
rocky-8         0      4.18.0-425.10.1.el8_7.x86_64 06 Jan 24 19:45 EST  06 Jan 24 19:51 EST  2.5 GiB
[root@localhost ~]#
[root@localhost ~]#
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F Clear
Search
ENG IN
1:13 AM
1/7/2024
```







## Configuring Compute Nodes

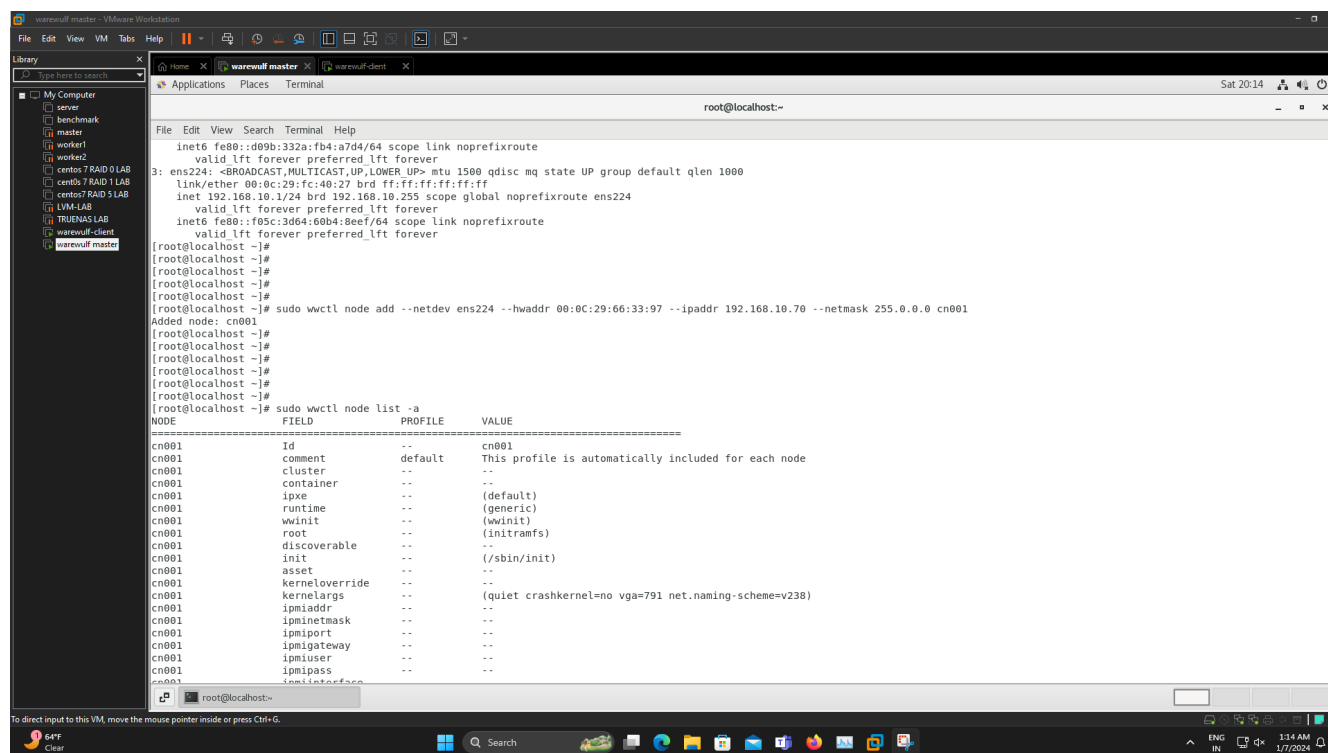
You're almost ready to start booting compute nodes, but you need to configure them in Warewulf before doing so. Recall that `/etc/warewulf/nodes.conf` is the database of compute nodes in a simple flat text YAML format. You can get the list of defined compute nodes with the command:

```
$ sudo wwctl node list -a
```

The command at this point should have no output, unless you or someone else with Warewulf access has created one. If you see that any node has been created that you don't expect, you might want to uninstall Warewulf and start over.

Adding a node is as simple as:

```
sudo wwctl node add --netdev ens224 --hwaddr 00:0C:29:66:33:97 --ipaddr 192.168.10.70 --netmask 255.255.255.0 cn001
```



```
File Edit View Search Terminal Help
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~# sudo wwctl node add --netdev ens224 --hwaddr 00:0C:29:66:33:97 --ipaddr 192.168.10.70 --netmask 255.0.0.0 cn001
Added node: cn001
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~# sudo wwctl node list -a
=====
NODE      FIELD      PROFILE  VALUE
-----
cn001     Id         --       cn001
cn001     comment   default  This profile is automatically included for each node
cn001     cluster   --       --
cn001     container --       --
cn001     ipxe      --       (default)
cn001     runtime   --       (generic)
cn001     winit     --       (winit)
cn001     root      --       (initramfs)
cn001     discoverable --       --
cn001     init      --       (/sbin/init)
cn001     asset     --       --
cn001     kerneloverride --       --
cn001     kernelargs --       (quiet crashkernel=no vga=791 net.naming-scheme=v238)
cn001     ipmiaddr  --       --
cn001     ipminetmask --       --
cn001     ipmiport  --       --
cn001     ipmigateway --       --
cn001     ipmiuser  --       --
cn001     ipmipass  --       --
cn001     ipmiinterface --       --
```

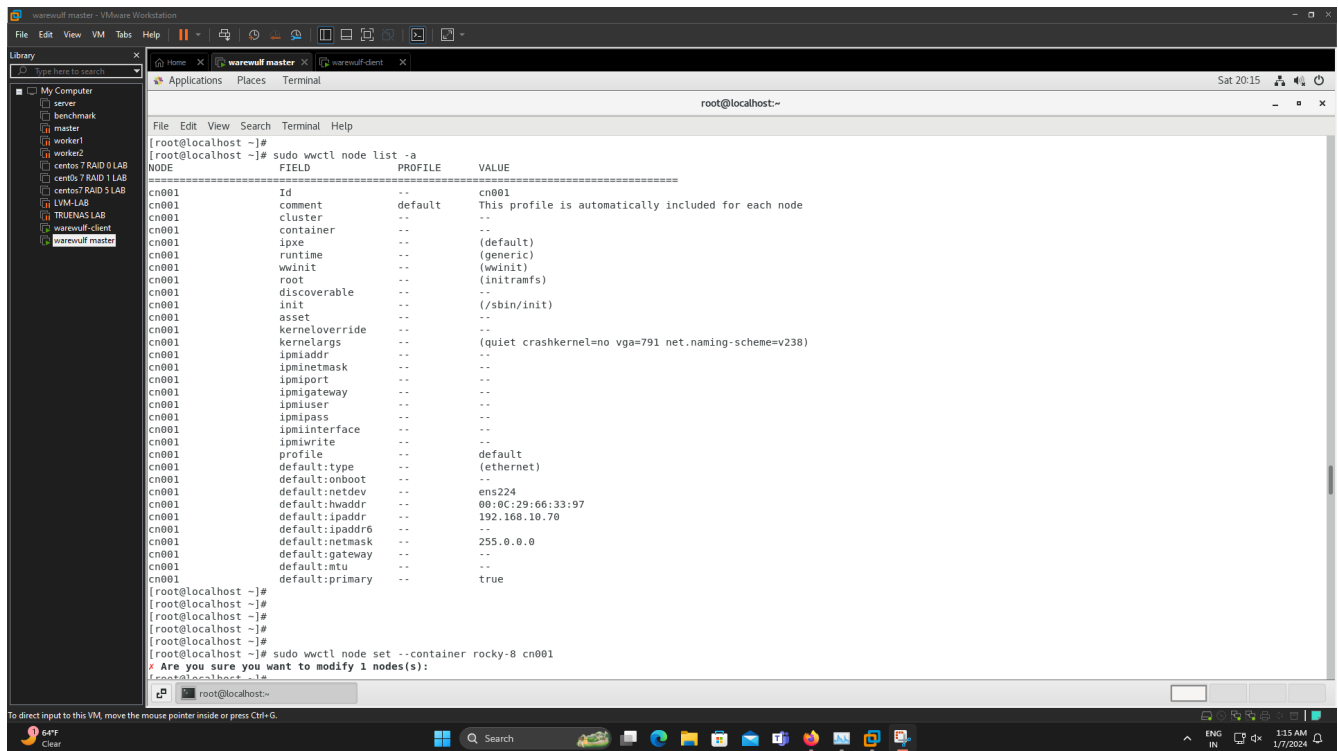
If you don't know the MAC address of the compute node, you can boot the nodes one by one and have Warewulf capture the MAC–node name association. When you define a node in Warewulf for which you don't know the MAC, you use the command:

```
sudo wwctl node add NODENAME.CLUSTER --ipaddr aaa.bbb.ccc.ddd --discoverable
```

ou should see the current node, the MAC address, and that the *discoverable* flag is *false*.

```
$ sudo wwctl node list -a
```

## List of Defined Nodes



```
root@localhost:~# sudo wwctl node list -a
NODE      FIELD      PROFILE  VALUE
=====
cn001     Id         --      cn001
cn001     comment    --      This profile is automatically included for each node
cn001     cluster    --      --
cn001     container  --      --
cn001     ipxe       --      (default)
cn001     runtime    --      (generic)
cn001     winit      --      (winit)
cn001     root       --      (initramfs)
cn001     discoverable --      --
cn001     init       --      (/sbin/init)
cn001     asset      --      --
cn001     kerneloverride --      --
cn001     kernelargs --      (quiet crashkernel=no vga=791 net.naming-scheme=v238)
cn001     ipmiaddr   --      --
cn001     ipminetmask --      --
cn001     ipmiport   --      --
cn001     ipmi gateway --      --
cn001     ipmiuser   --      --
cn001     ipmipass   --      --
cn001     ipmiinterface --      --
cn001     ipmiwrite  --      --
cn001     profile    --      default
cn001     default:type --      (ethernet)
cn001     default:onboot --      --
cn001     default:netdev --      ens224
cn001     default:hwaddr --      00:0C:29:66:33:97
cn001     default:ipaddr --      192.168.10.70
cn001     default:ipaddr6 --      --
cn001     default:netmask --      255.0.0.0
cn001     default:gateway --      --
cn001     default:mtu --      --
cn001     default:primary --      true

root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~#
root@localhost:~# sudo wwctl node set --container rocky-8 cn001
Are you sure you want to modify 1 nodes(s):
y
```

The values listed in parentheses are values that can be changed with *wwctl*.

At this point, I like to assign the container to the node with a simple command:

```
sudo wwctl node set --container rocky-8 n0001
```

```
warewulf-master - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
server
benchmark
master
workert1
workert2
centos7 RAID 0 LAB
centos7 RAID 1 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf-master

root@localhost~
File Edit View Search Terminal Help
cn001 default:ipaddr -- 192.168.10.70
cn001 default:ipaddr6 --
cn001 default:netmask -- 255.0.0.0
cn001 default:gateway --
cn001 default:mtu --
cn001 default:primary -- true
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wvctl node set --container rocky-8 cn001
Are you sure you want to modify 1 nodes(s):
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wvctl node list -a cn001 | grep Container
[root@localhost ~]#
[root@localhost ~]# sudo wvctl container exec rocky-8 /bin/bash
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf> yum install passwd
Failed to set locale, defaulting to C.UTF-8
Rocky Linux 8 - AppStream 3.1 MB/s | 9.6 MB 00:03
Rocky Linux 8 - BaseOS 2.9 MB/s | 3.9 MB 00:01
Rocky Linux 8 - Extras 7.5 kB/s | 14 kB 00:01
Dependencies resolved.
=====
Package Architecture Version Repository Size
=====
Installing:
passwd x86_64 0.80-4.el8 baseos 114 k
Installing dependencies:
libuser x86_64 0.62-25.el8 baseos 413 k
=====
root@localhost~
```

To add packages to the container, exec into the container:

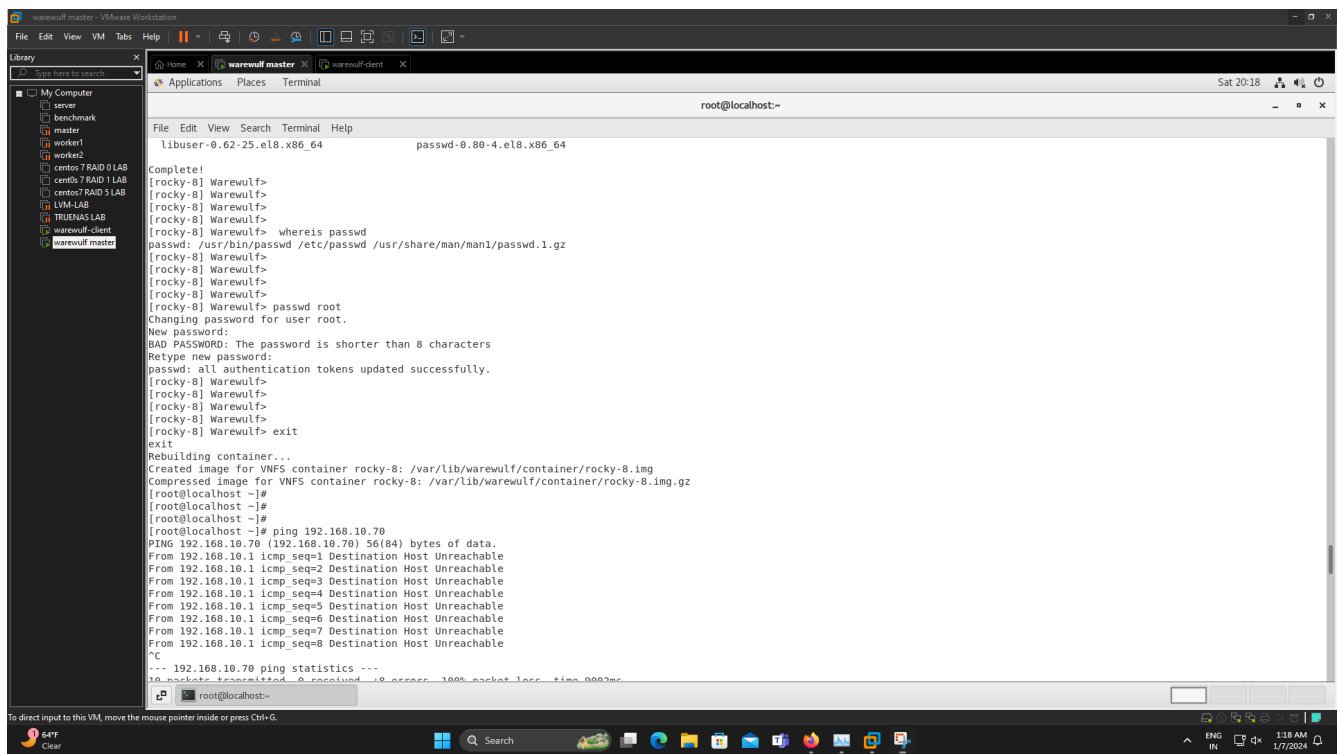
```
$ sudo wvctl container exec rocky-8 /bin/bash
```

Once “inside” the container, you can use *yum* to install *passwd*, the package for the password tool , then create a password for the root user . Again, notice that on exit, the container is rebuilt.

## Installing passwd

### Creating a Password Root





## Booting Nodes

At this point, you can boot compute nodes. For the first compute node, I would plug in a monitor and keyboard to watch the boot process and debug what is happening. After that, you can just turn on the power to the node.

You should see the node get the correct the IP address over DHCP; then you should see iPXE get copied over, and the node starts booting. The container is copied over the node and pivot boots into the container. Finally, you should see a prompt on the monitor plugged into the node indicating that the node is ready to be used.

Because only SSH is used for authentication, you won't be able to log in to the compute node as a user; however, the rootuser has a password, so you can log in as root with the monitor and keyboard. When you do this, poke around the compute node and make sure everything looks correct. For example,

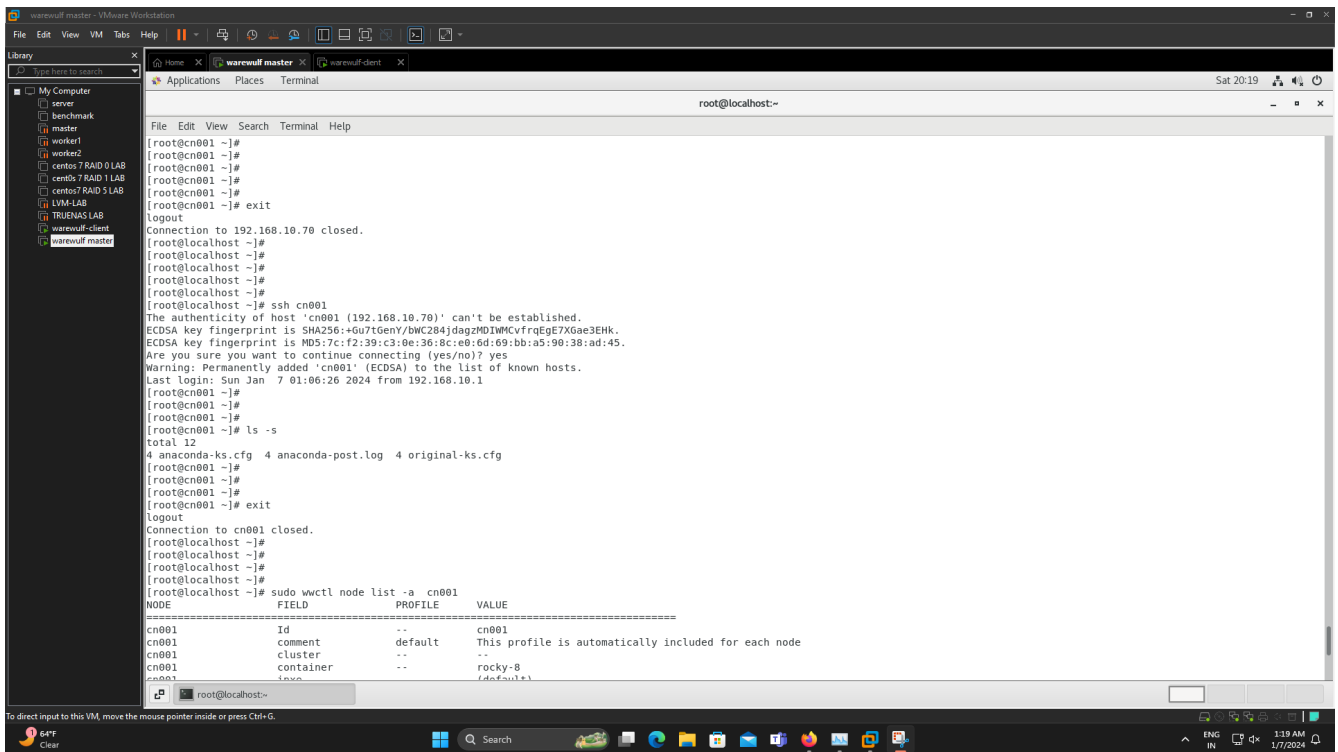
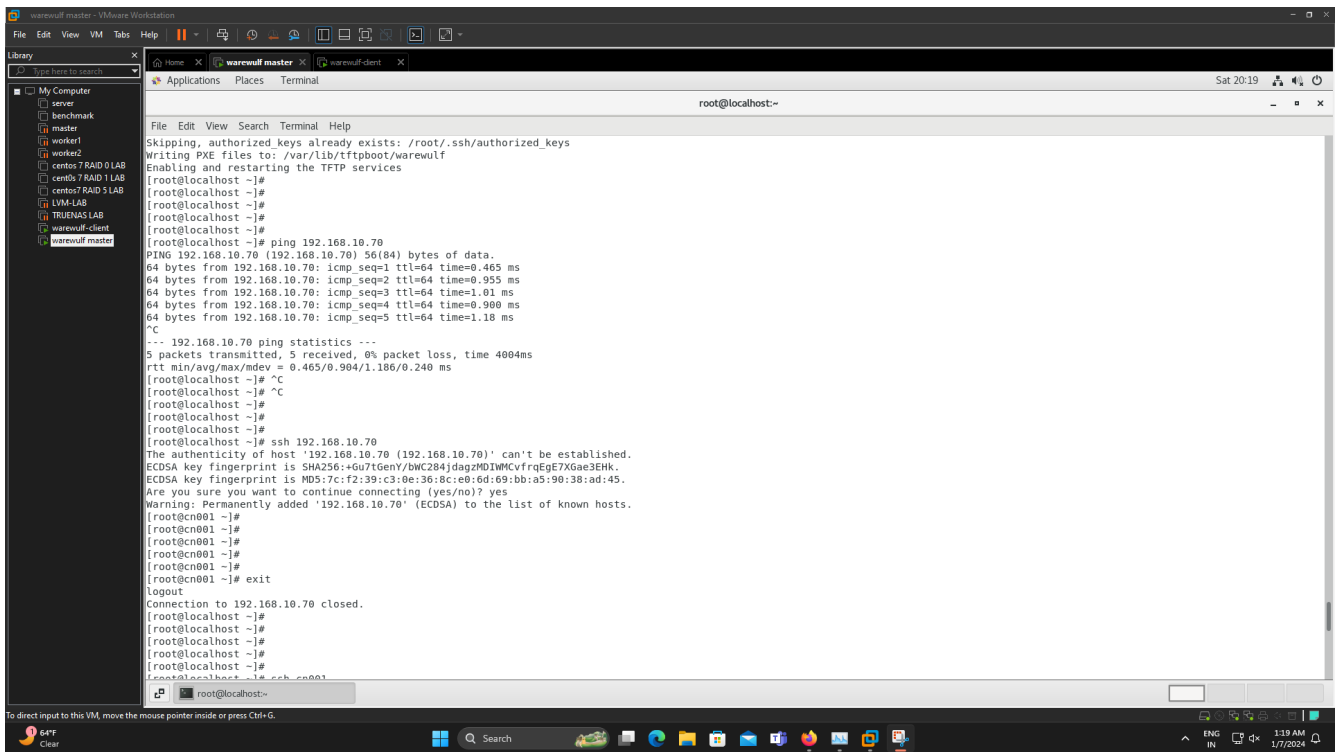
- check that the NFS exported files have mounted (use the *mount* command),
- look at the local IP address (*ip addr show*),
- ping the host node with the IP address (*ping 192.168.10.70*), and
- check the hostname of the node (*more /etc/hostname*).

These are just a few things you can do to check the compute node, but you can do whatever you like to satisfy that the compute node is up and working correctly.

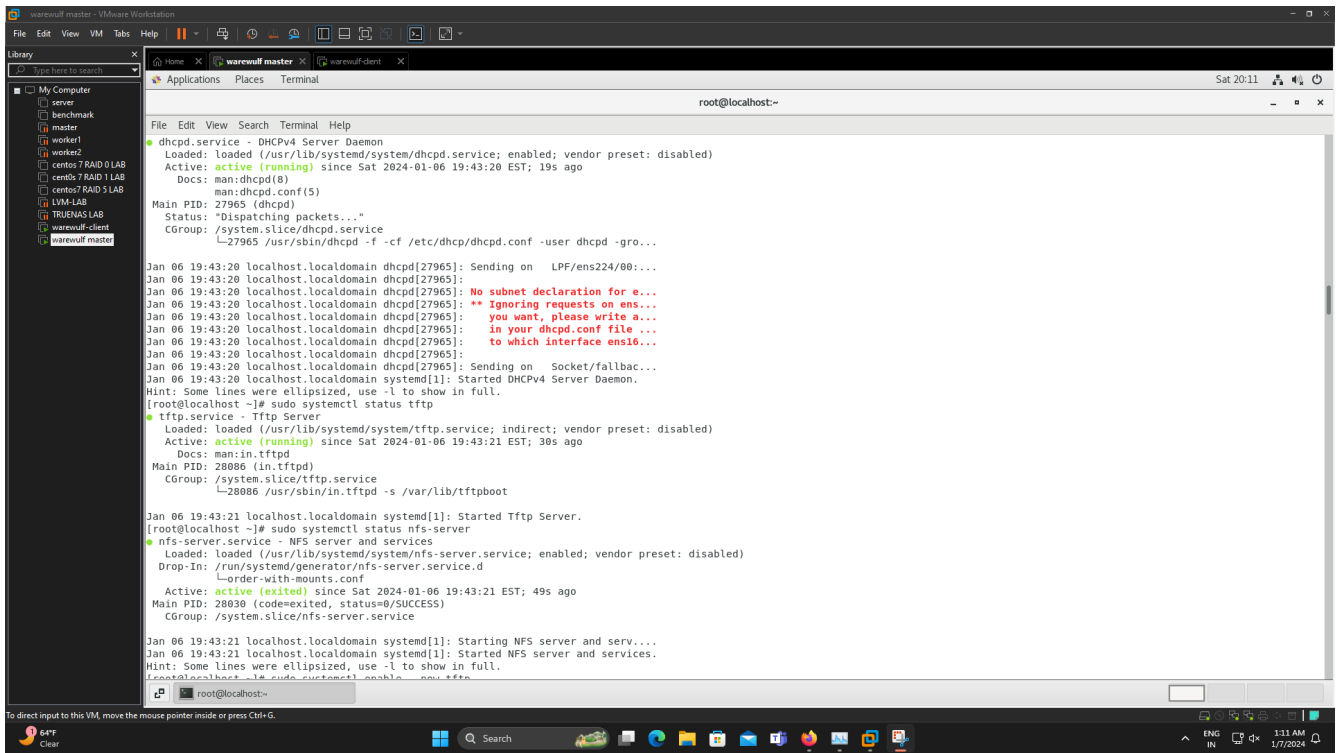
At this point, you might go back to the host node and check out the compute node. The first thing to do is ping the IP address of the compute node then ping the hostname of the compute node .

## Pinging the Compute IP Address

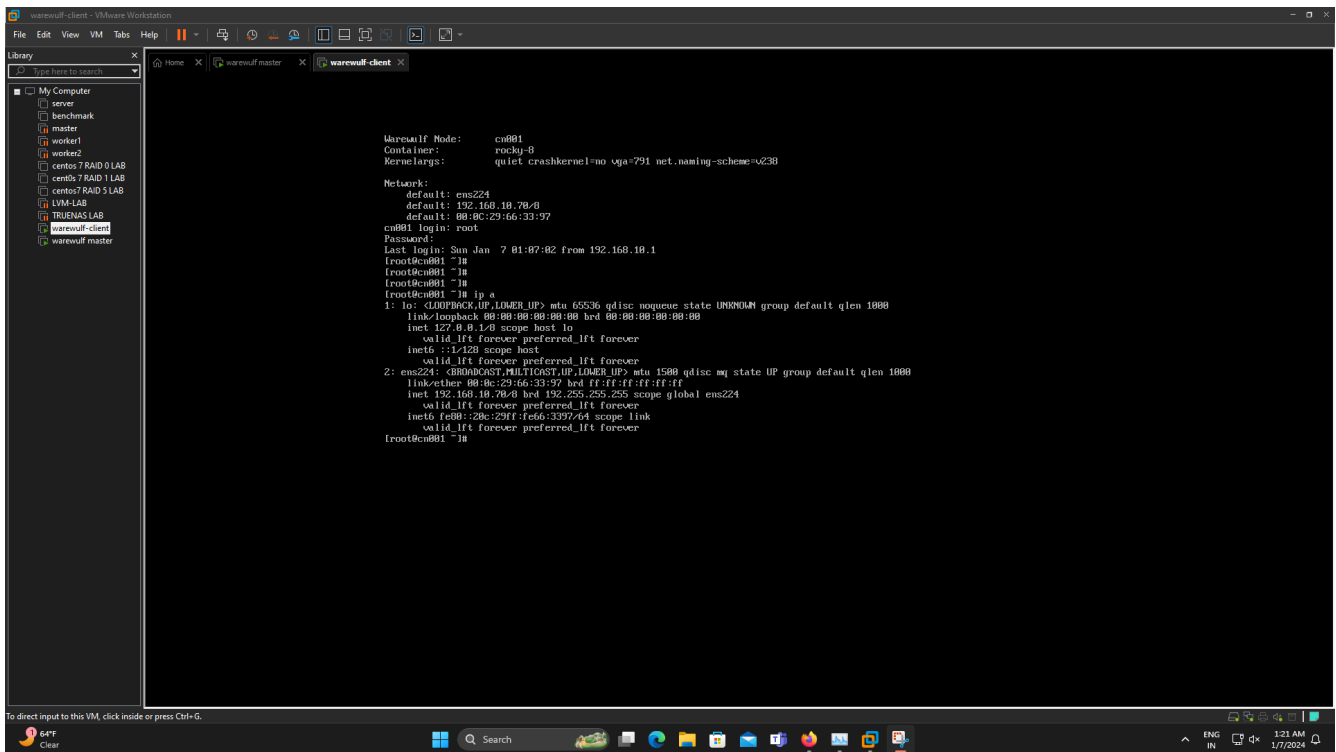
```
warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
  server
  benchmark
  master
  worker1
  worker2
  centos 7 RAID 0 LAB
  centos 7 RAID 1 LAB
  LVM-LAB
  TRUENAS LAB
  warewulf-client
  warewulf-master
warewulf master
Applications Places Terminal
root@localhost:~
File Edit View Search Terminal Help
From 192.168.10.1 icmp_seq=8 Destination Host Unreachable
^C
--- 192.168.10.70 ping statistics ---
10 packets transmitted, 0 received, +8 errors, 100% packet loss, time 9002ms
pipe 4
[root@localhost ~]# ^C
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ping 192.168.10.70
PING 192.168.10.70 (192.168.10.70) 56(84) bytes of data.
From 192.168.10.1 icmp_seq=1 Destination Host Unreachable
From 192.168.10.1 icmp_seq=2 Destination Host Unreachable
From 192.168.10.1 icmp_seq=3 Destination Host Unreachable
From 192.168.10.1 icmp_seq=4 Destination Host Unreachable
^C
--- 192.168.10.70 ping statistics ---
6 packets transmitted, 0 received, +4 errors, 100% packet loss, time 5000ms
pipe 4
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wvctl configure -a
Building overlay for localhost.localdomain: host
Enabling and restarting the DHCP services
Building overlay for localhost.localdomain: host
Enabling and restarting the NFS services
Updating system keys
Skipping, key already exists: ssh_host_rsa_key
Skipping, key already exists: ssh_host_dsa_key
Skipping, key already exists: ssh_host_ecdsa_key
Skipping, key already exists: ssh_host_ed25519_key
Skipping, authorized_keys already exists: /root/.ssh/authorized_keys
Writing PXE files to: /var/lib/tftpboot/warewulf
Enabling and restarting the TFTP services
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ping 192.168.10.70
root@localhost:~
```

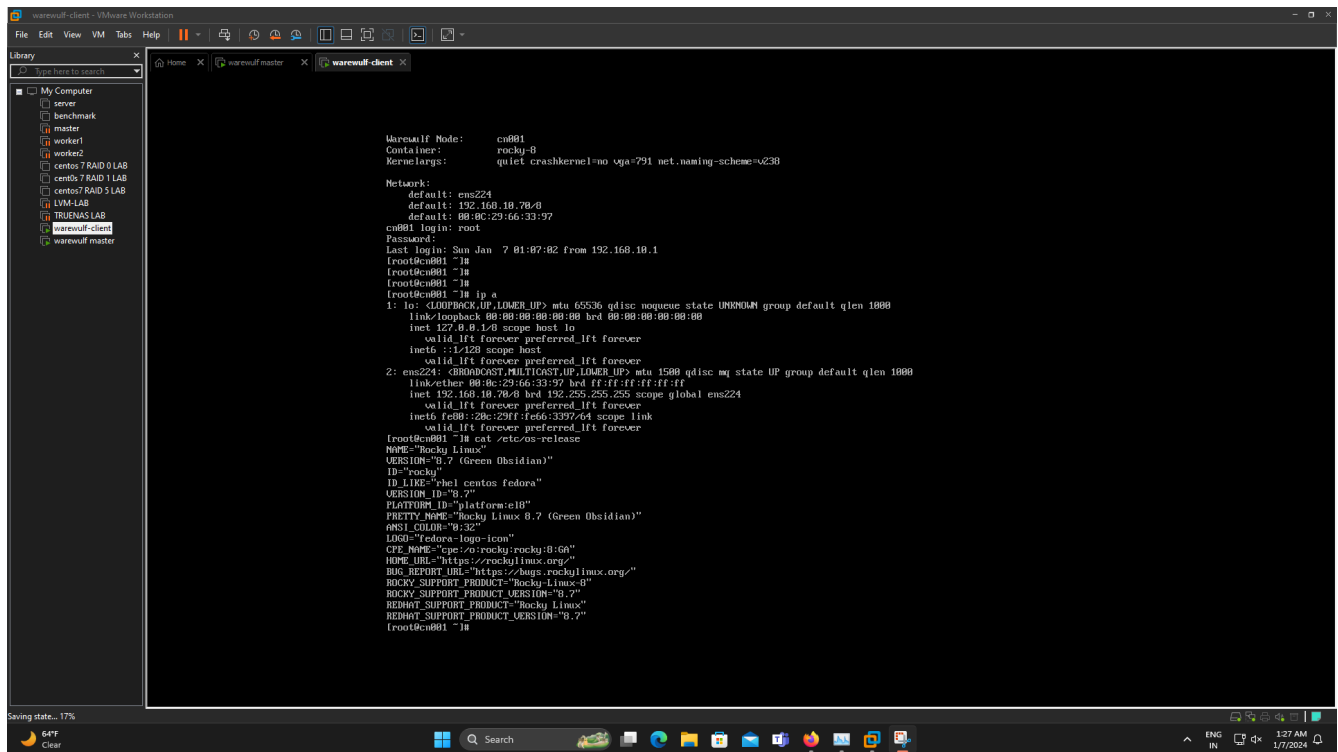






check compute node :





## Summary

Booting compute nodes is a huge step forward in building your cluster. Without this step, everything following is kind of meaningless. However, I am still a few steps away from having what I think is a modern cluster with compilers, MPI libraries, and a resource manager (aka a job scheduler). Moreover, my compute nodes have GPUs in them, and they are currently not configured.