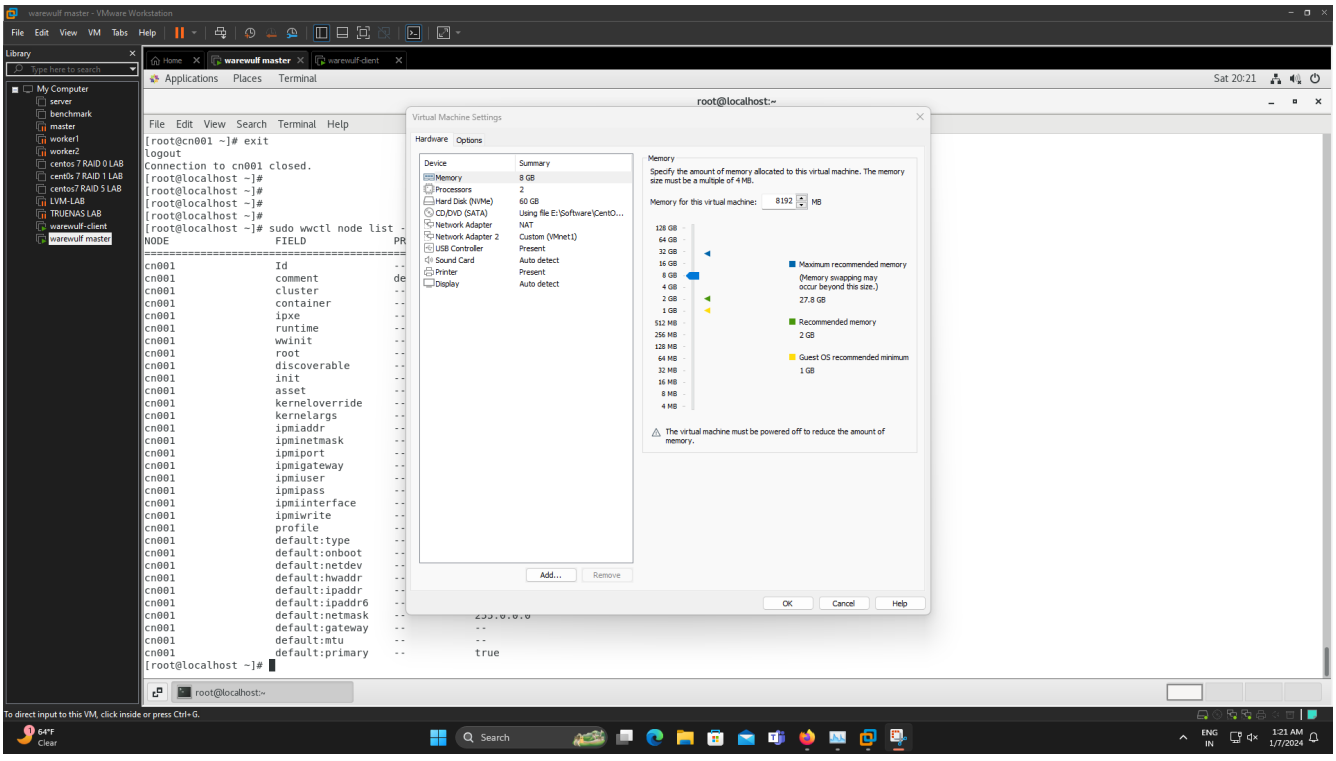
Name : Pranjali Patil
PRN : 230940127009

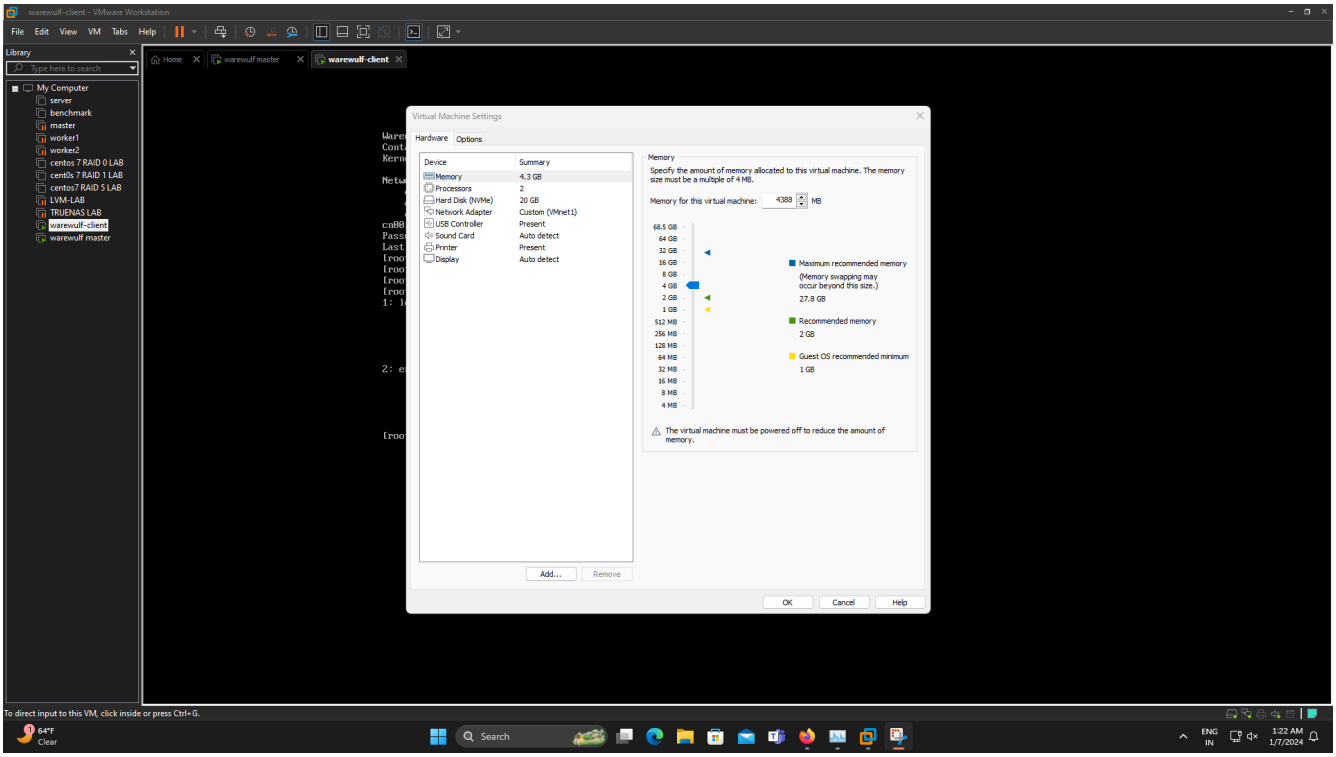
Warewulf v4 installation:-

prerequisites:

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

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





after installing the OS

-> Disable Selinux

-> Disable Firewalld

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 .

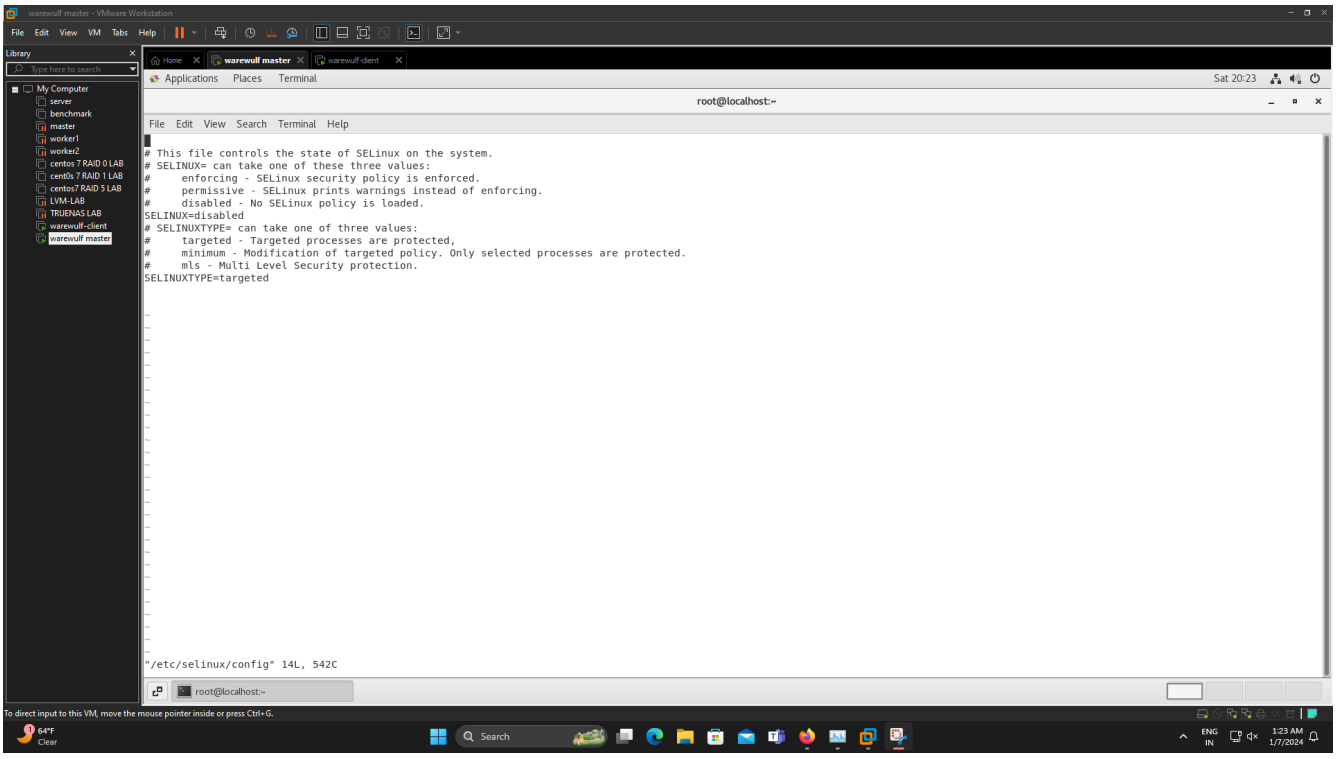
The master node has two network adapter, with one on the NAT connected to the local LAN using the Dynamic Host Configuration Protocol (DHCP). This network adapter is my home network (192.168.144.212). The second network adapter is connected to a private network (192.168.10.1,

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 .

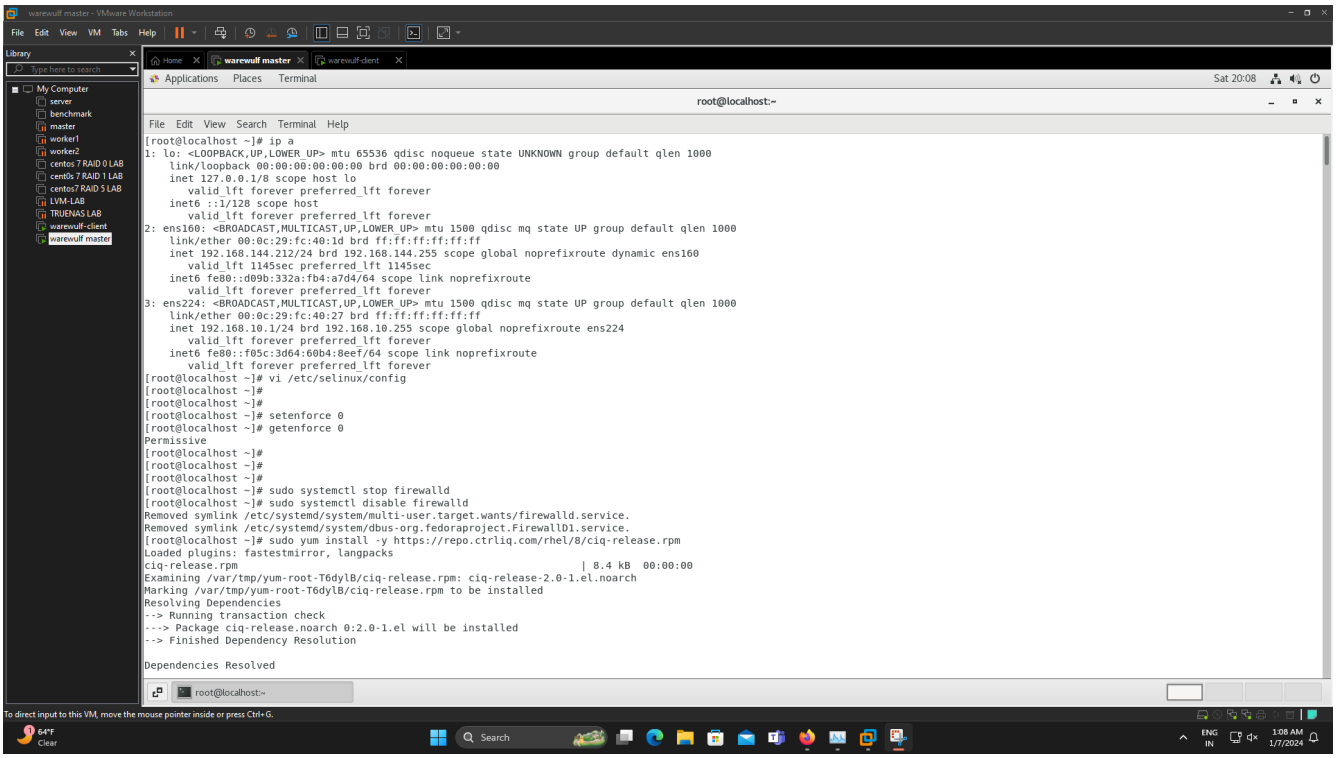
I chose to use binary RPMs because they are a little easier for me to update Warewulf. shows the results of the documentation guidance.

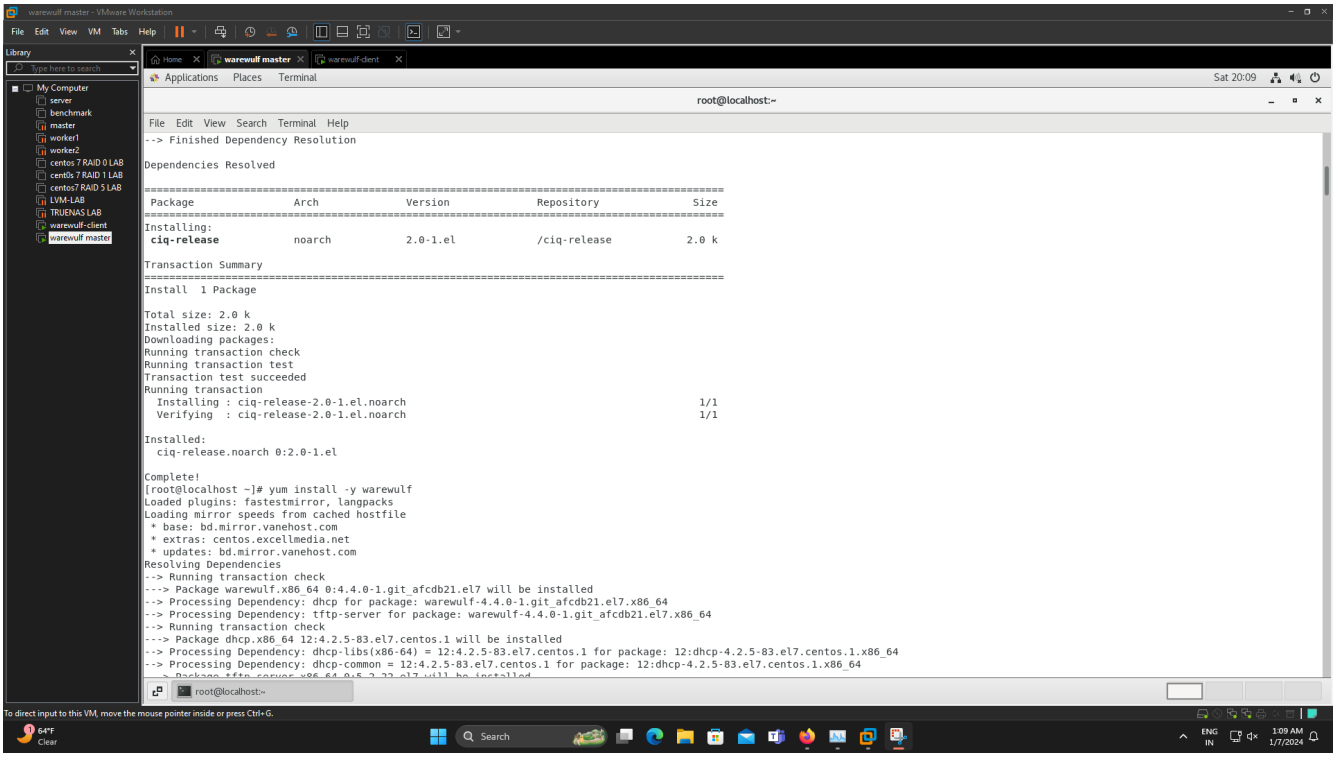
Warewulf Installation

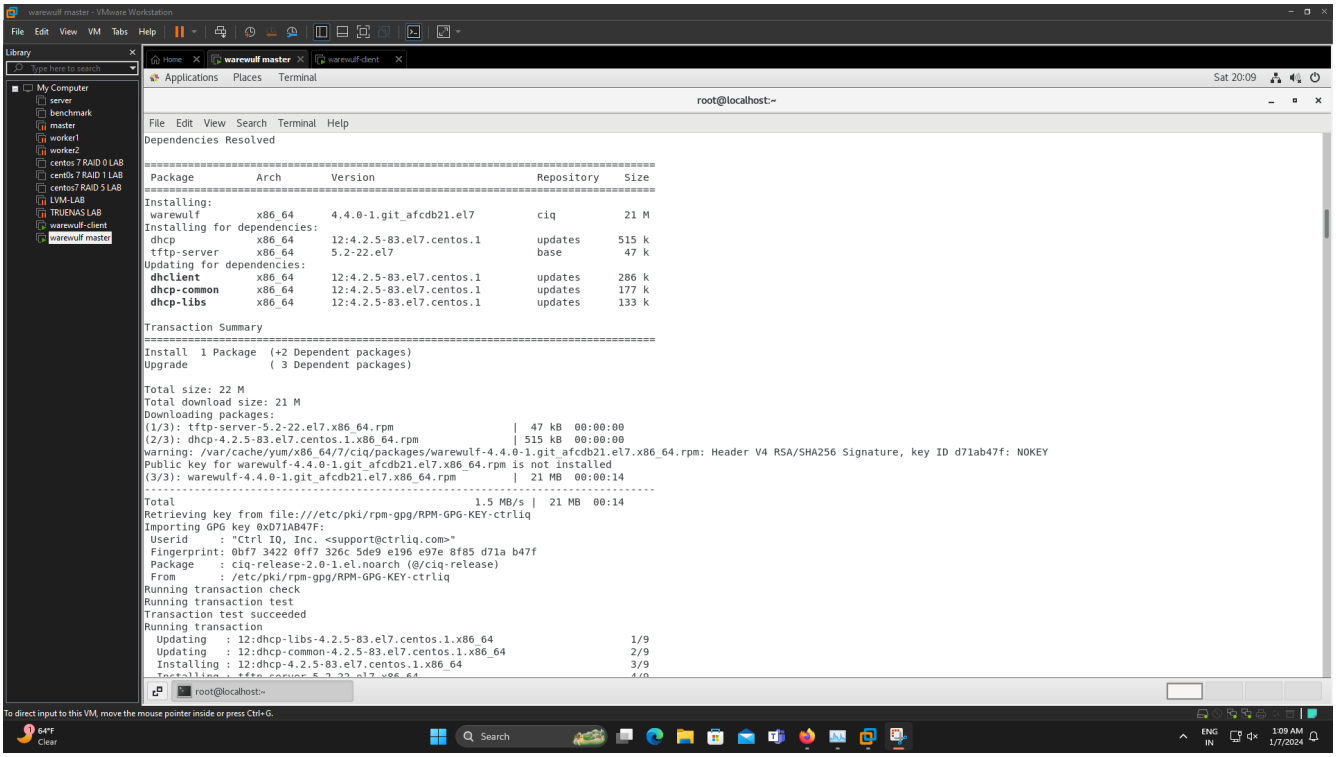
```
$ sudo yum install -y https://repo.ctrliq.com/rhel/8/ciq-release.rpm
$ yum install -y warewulf
```

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

Configuring Warewulf on the master Node

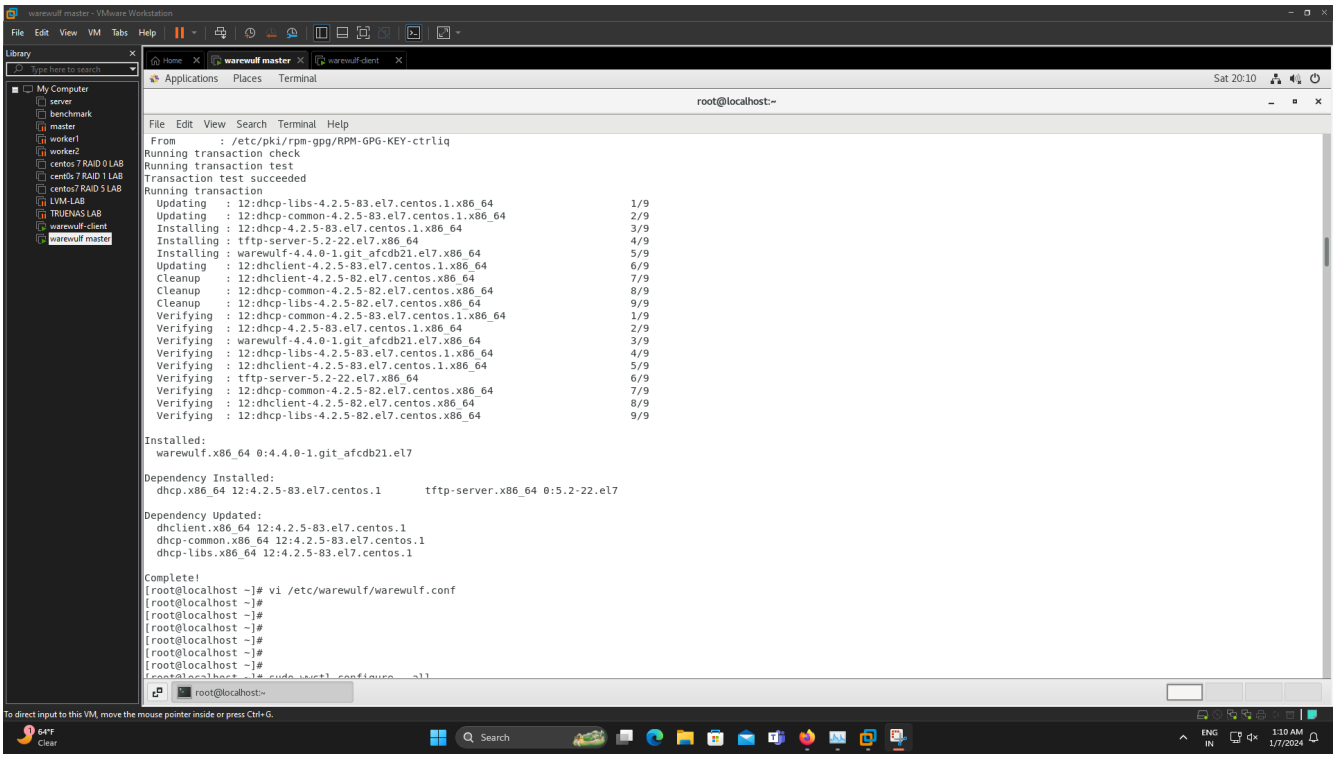

warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:08
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:0c:29:fc:40:1d brd ff:ff:ff:ff:ff:ff
inet 192.168.144.212/24 brd 192.168.144.255 scope global noprefixroute dynamic ens160
valid_lft 1145sec preferred_lft 1145sec
inet6 fe80::d09b:332a:fb4:a7d4/64 scope link noprefixroute
valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:0c:29:fc:40:27 brd ff:ff:ff:ff:ff:ff
inet 192.168.10.1/24 brd 192.168.10.255 scope global noprefixroute ens224
valid_lft forever preferred_lft forever
inet6 fe80::f05c:3d64:60b4:8eef/64 scope link noprefixroute
valid_lft forever preferred_lft forever
[root@localhost ~]# vi /etc/selinux/config
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# setenforce 0
[root@localhost ~]# getenforce 0
Permissive
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo systemctl stop firewalld
[root@localhost ~]# sudo systemctl disable firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@localhost ~]# sudo yum install -y https://repo.ctrliq.com/rhel/8/ciq-release.rpm
Loaded plugins: fastestmirror, langpacks
ciq-release.rpm | 8.4 kB 00:00:00
Examining /var/tmp/yum-root-T6dylB/ciq-release.rpm: ciq-release-2.0-1.el.noarch
Marking /var/tmp/yum-root-T6dylB/ciq-release.rpm to be installed
Resolving Dependencies
--> Running transaction check
---> Package ciq-release.noarch 0:2.0-1.el will be installed
--> Finished Dependency Resolution
Dependencies Resolved
root@localhost:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F Clear
Search
ENG IN
1:08 AM 1/7/2024

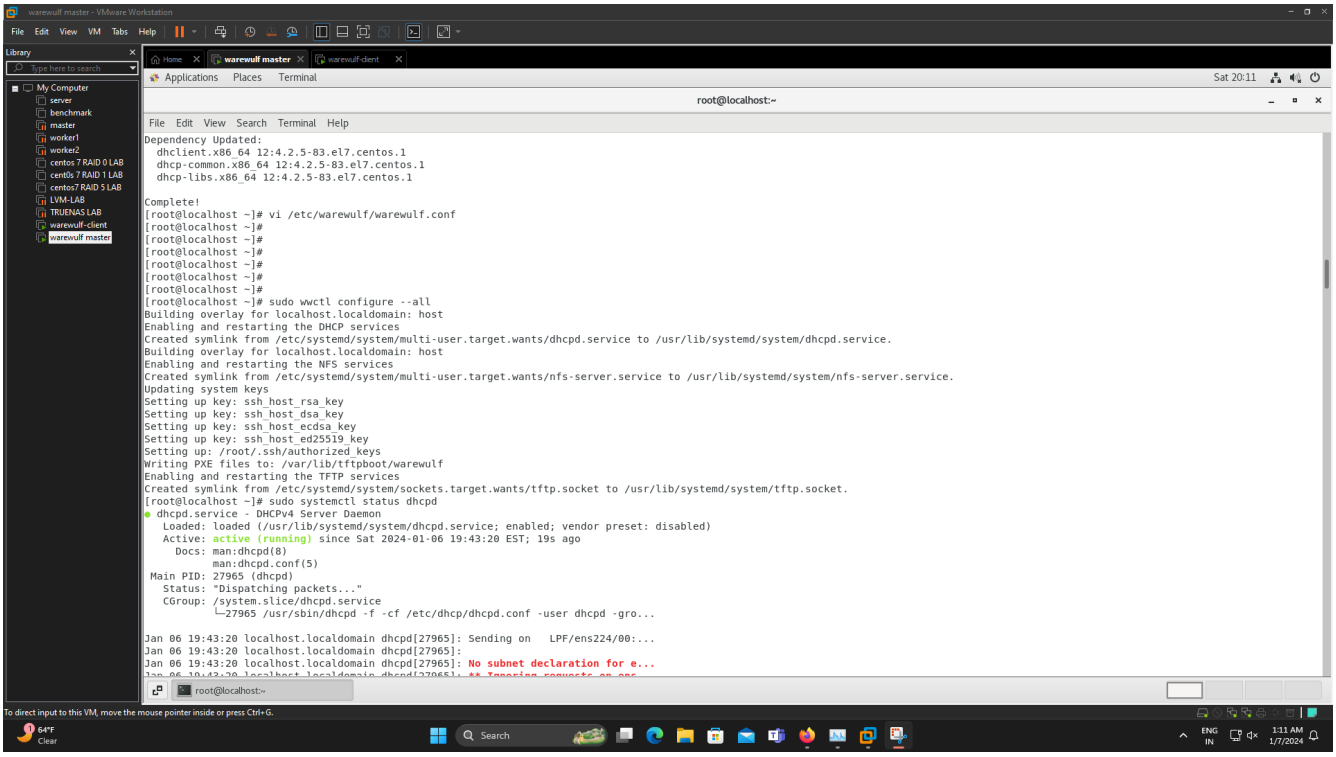

warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:09
root@localhost:~
File Edit View Search Terminal Help
--> Finished Dependency Resolution
Dependencies Resolved
Package Arch Version Repository Size
Installing:
ciq-release noarch 2.0-1.el /ciq-release 2.0 k
Transaction Summary
Install 1 Package
Total size: 2.0 k
Installed size: 2.0 k
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Installing : ciq-release-2.0-1.el.noarch 1/1
Verifying : ciq-release-2.0-1.el.noarch 1/1
Installed:
ciq-release.noarch 0:2.0-1.el
Complete!
[root@localhost ~]# yum install -y warewulf
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: bd.mirror.vanehost.com
* extras: centos.excellmedia.net
* updates: bd.mirror.vanehost.com
Resolving Dependencies
--> Running transaction check
---> Package warewulf.x86_64 0:4.4.0-1.git_afcdb21.el7 will be installed
--> Processing Dependency: dhcp for package: warewulf-4.4.0-1.git_afcdb21.el7.x86_64
--> Processing Dependency: tftp-server for package: warewulf-4.4.0-1.git_afcdb21.el7.x86_64
--> Running transaction check
---> Package dhcp.x86_64 12:4.2.5-83.el7.centos.1 will be installed
--> Processing Dependency: dhcp-libs(x86-64) = 12:4.2.5-83.el7.centos.1 for package: 12:dhcp-4.2.5-83.el7.centos.1.x86_64
--> Processing Dependency: dhcp-common = 12:4.2.5-83.el7.centos.1 for package: 12:dhcp-4.2.5-83.el7.centos.1.x86_64
root@localhost:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F Clear
Search
ENG IN
1:09 AM 1/7/2024

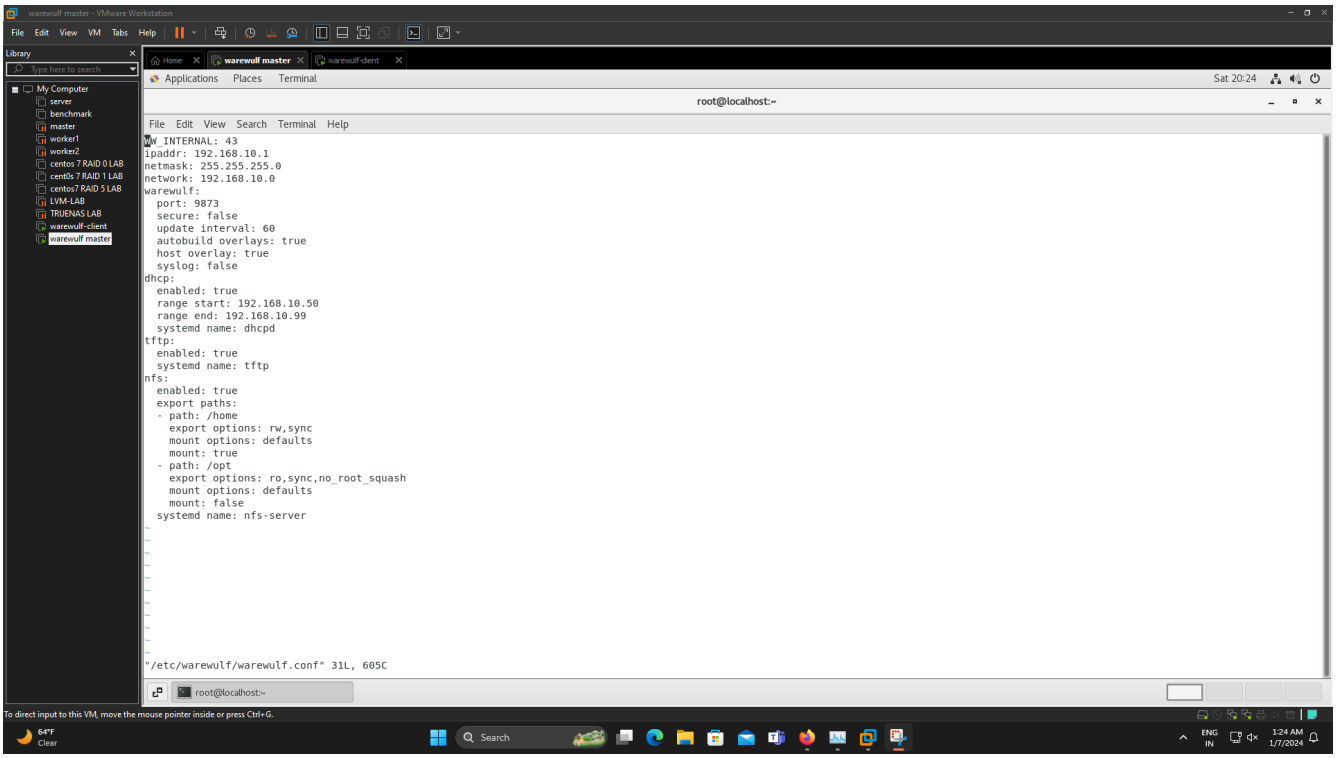


The configuration files for Warewulf are in */etc/warewulf*. The primary command you will use to interact with Warewulf is *wwctl*. The first file you should examine is */etc/warewulf/warewulf.conf*. My version of this file is shown in Listing 2.

warewulf.conf


warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:10
root@localhost:~
File Edit View Search Terminal Help
From : /etc/pki/rpm-gpg/RPM-GPG-KEY-ctrliq
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
Installing : warewulf-4.4.0-1.git_afcdb21.el7.x86_64 5/9
Updating : 12:dhclient-4.2.5-83.el7.centos.1.x86_64 6/9
Cleanup : 12:dhclient-4.2.5-82.el7.centos.x86_64 7/9
Cleanup : 12:dhcp-common-4.2.5-82.el7.centos.x86_64 8/9
Cleanup : 12:dhcp-libs-4.2.5-82.el7.centos.x86_64 9/9
Verifying : 12:dhcp-common-4.2.5-83.el7.centos.1.x86_64 1/9
Verifying : 12:dhcp-4.2.5-83.el7.centos.1.x86_64 2/9
Verifying : warewulf-4.4.0-1.git_afcdb21.el7.x86_64 3/9
Verifying : 12:dhcp-libs-4.2.5-83.el7.centos.1.x86_64 4/9
Verifying : 12:dhclient-4.2.5-83.el7.centos.1.x86_64 5/9
Verifying : tftp-server-5.2-22.el7.x86_64 6/9
Verifying : 12:dhcp-common-4.2.5-82.el7.centos.x86_64 7/9
Verifying : 12:dhclient-4.2.5-82.el7.centos.x86_64 8/9
Verifying : 12:dhcp-libs-4.2.5-82.el7.centos.x86_64 9/9
Installed:
warewulf.x86_64 0:4.4.0-1.git_afcdb21.el7
Dependency Installed:
dhcp.x86_64 12:4.2.5-83.el7.centos.1 tftp-server.x86_64 0:5.2-22.el7
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-libs.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:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F Clear
Search
ENG IN
1:10 AM 1/7/2024


warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:11
root@localhost:~
File Edit View Search Terminal Help
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-libs.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 wwctl 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(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...
root@localhost:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F Clear
Search
ENG IN
1:11 AM 1/7/2024



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.

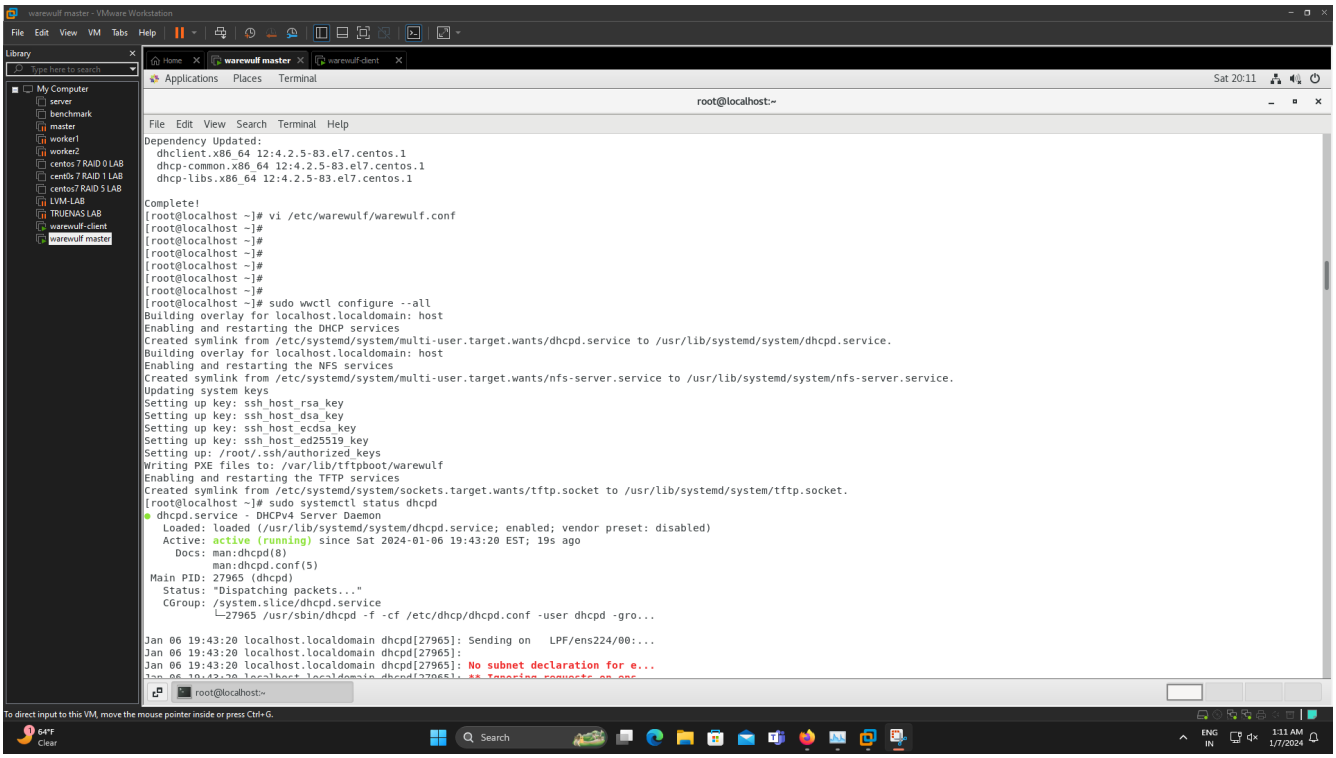
-> change ip address : 192.168.10.1

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.



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

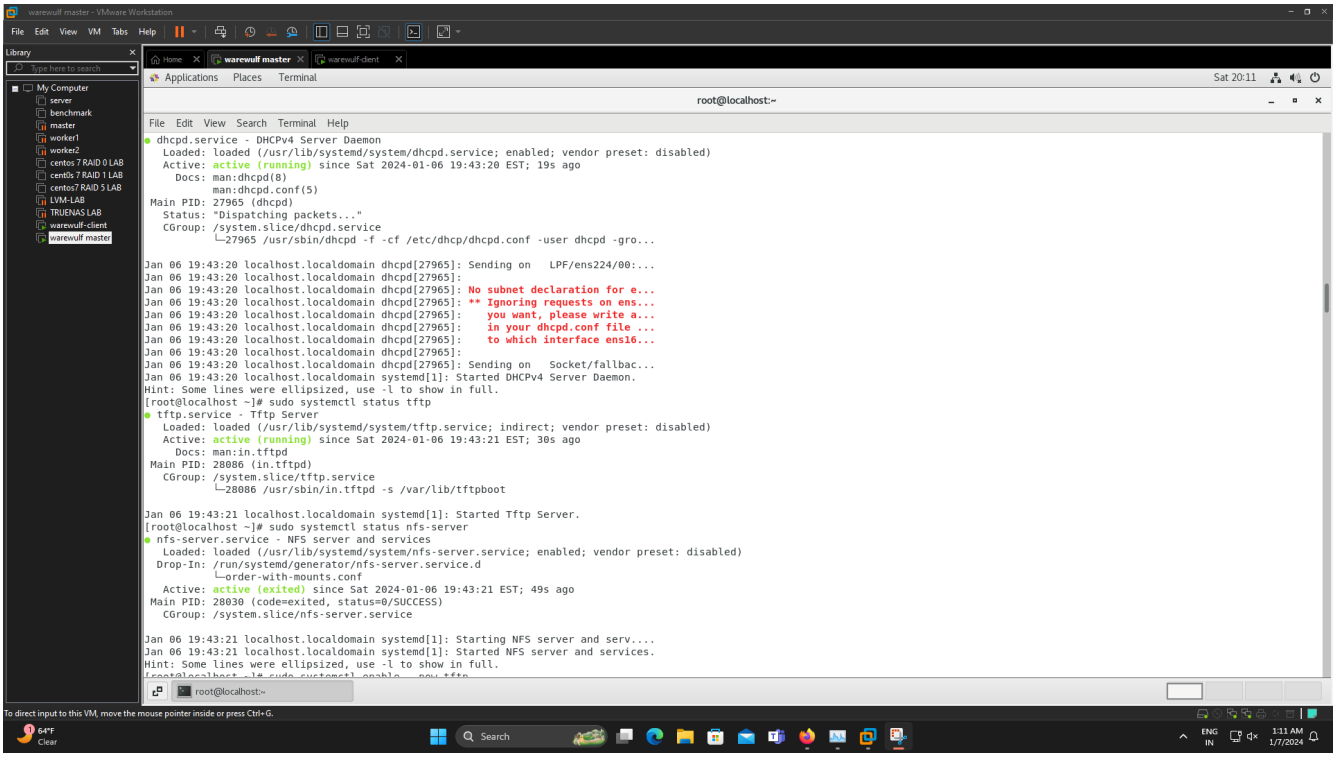
You can also verify with the *sudo exportfs* command. By default, you should see two filesystems exported: */home* and */opt*.

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.)



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
```

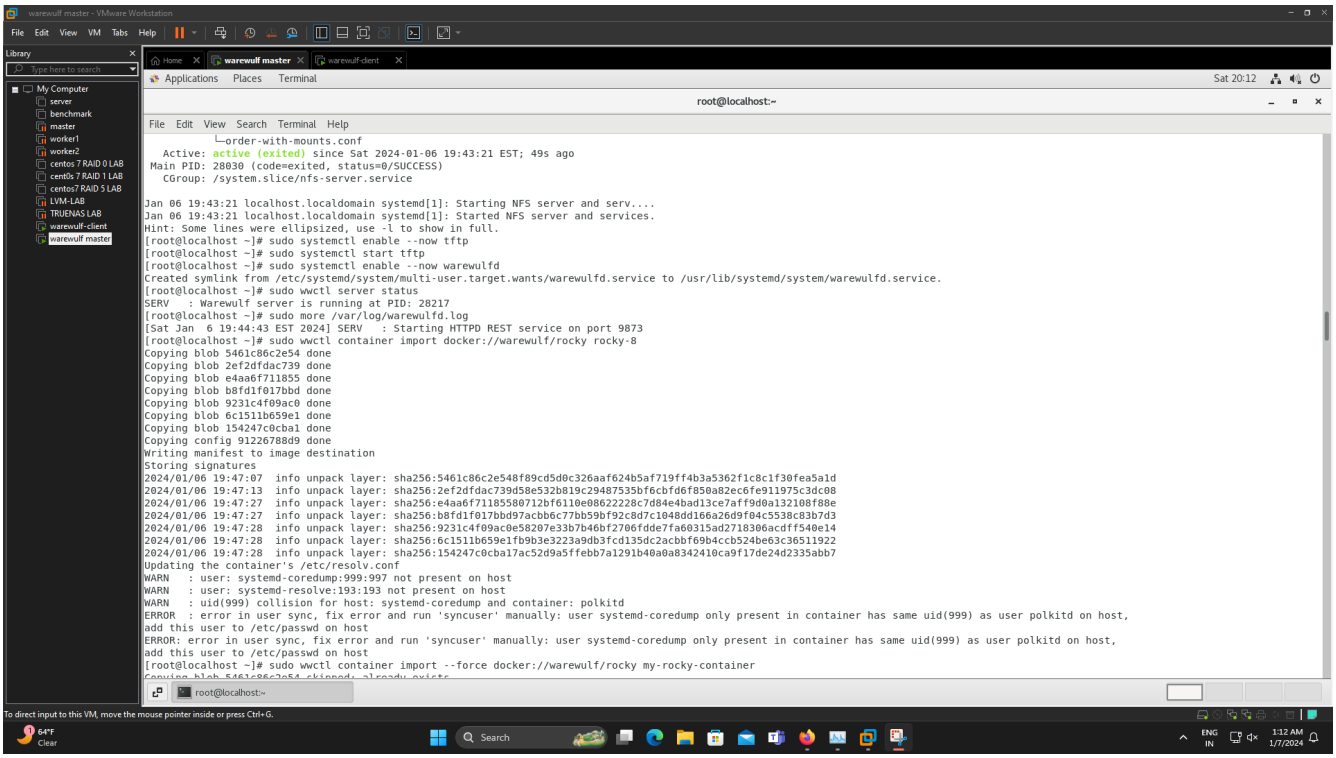
Container Management for Compute Nodes

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.

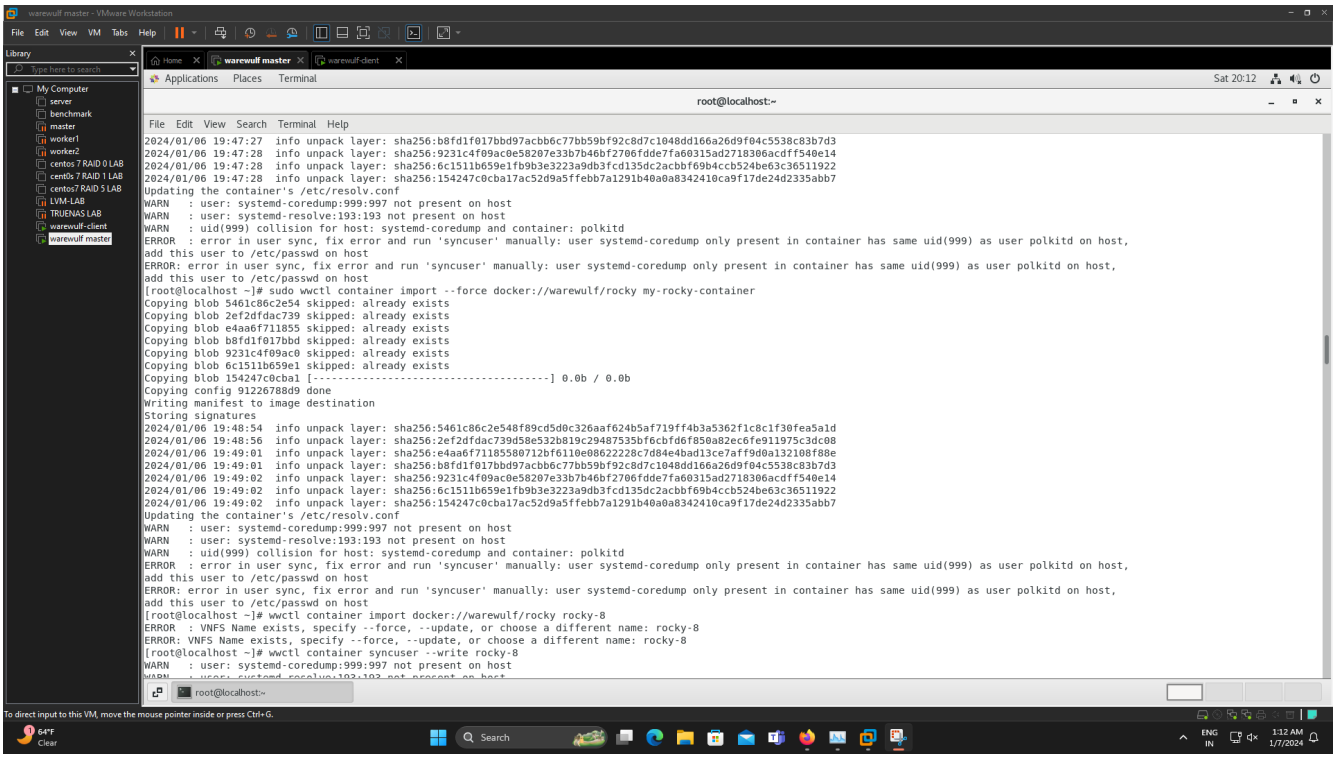
If you pull and try to boot a container such as *docker://rockylinux*, it will not boot properly; rather, it will boot into single-user mode and get stuck. It is better to look for containers in Docker Hub and even ask on the Warewulf mailing list or Slack channel about bootable containers.

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.

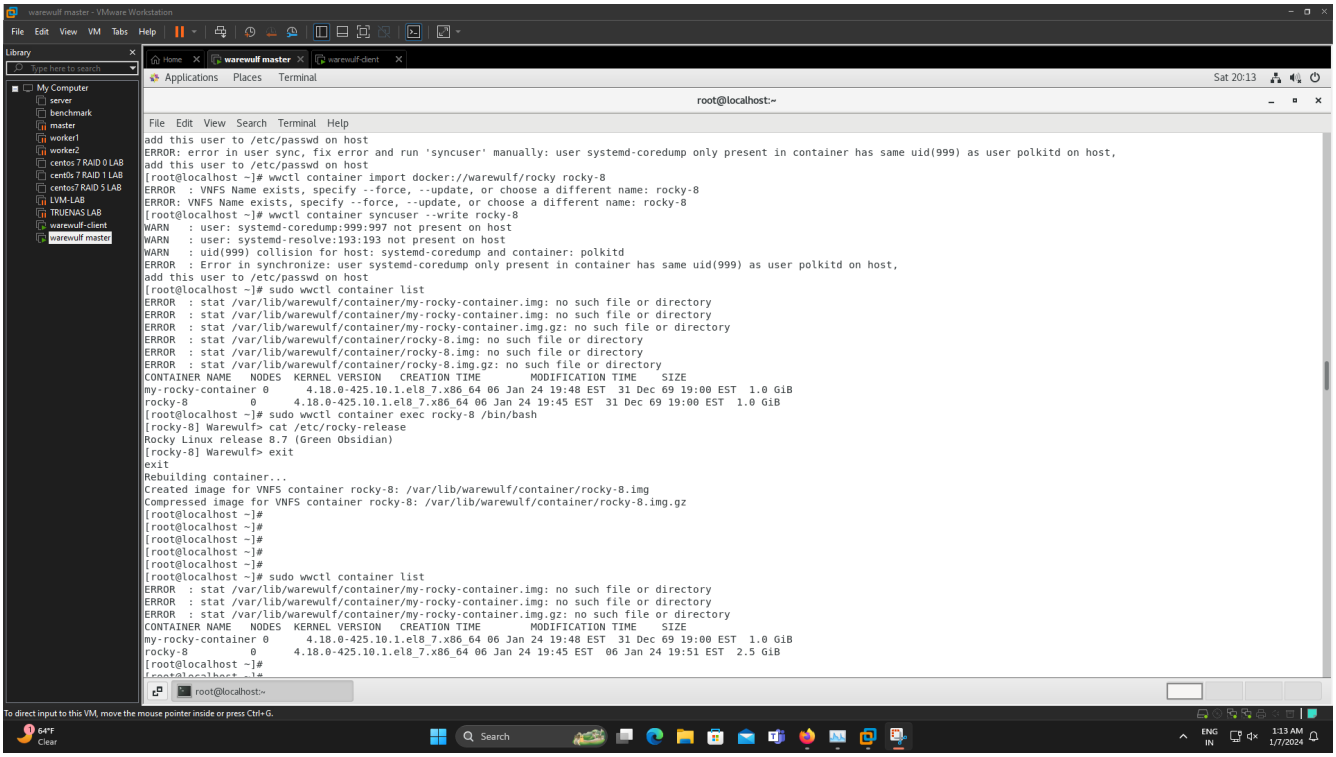
Pull Container on Docker Hub

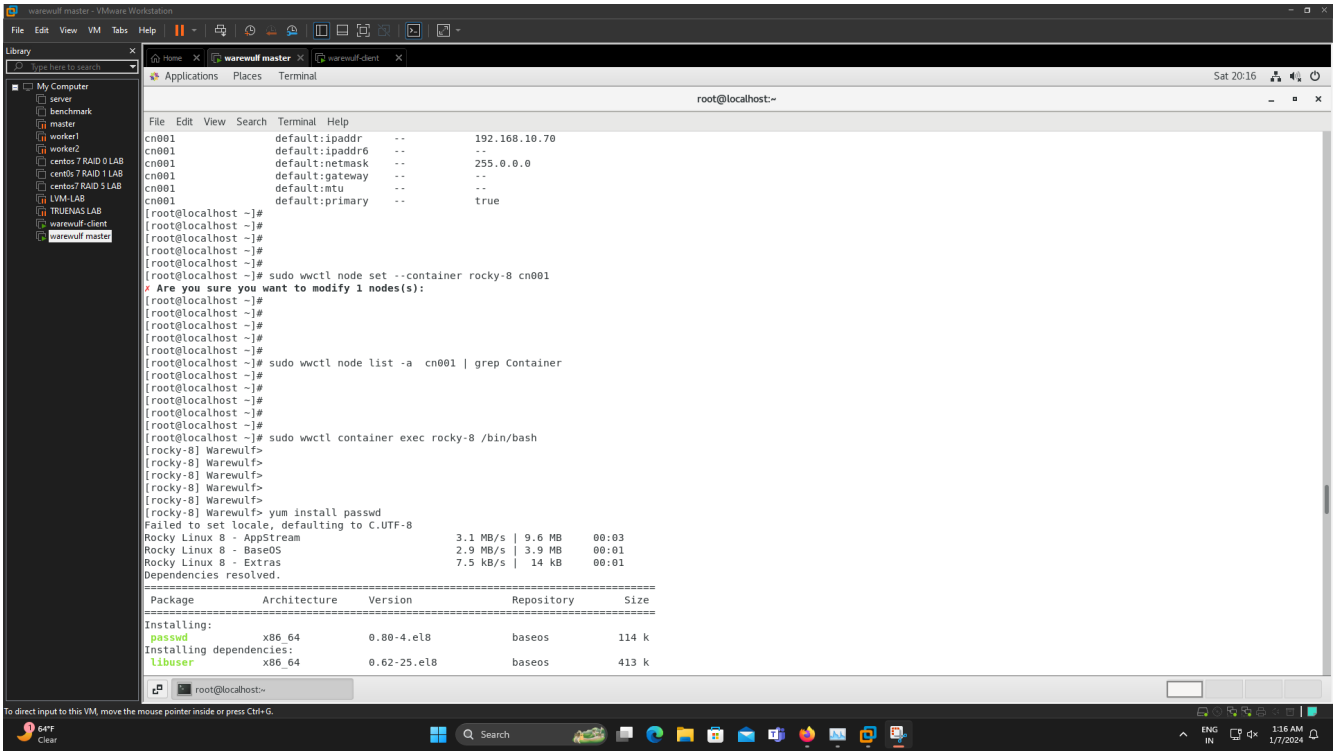


$ sudo wwctl container list



Modifying Containers on the master Node





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.

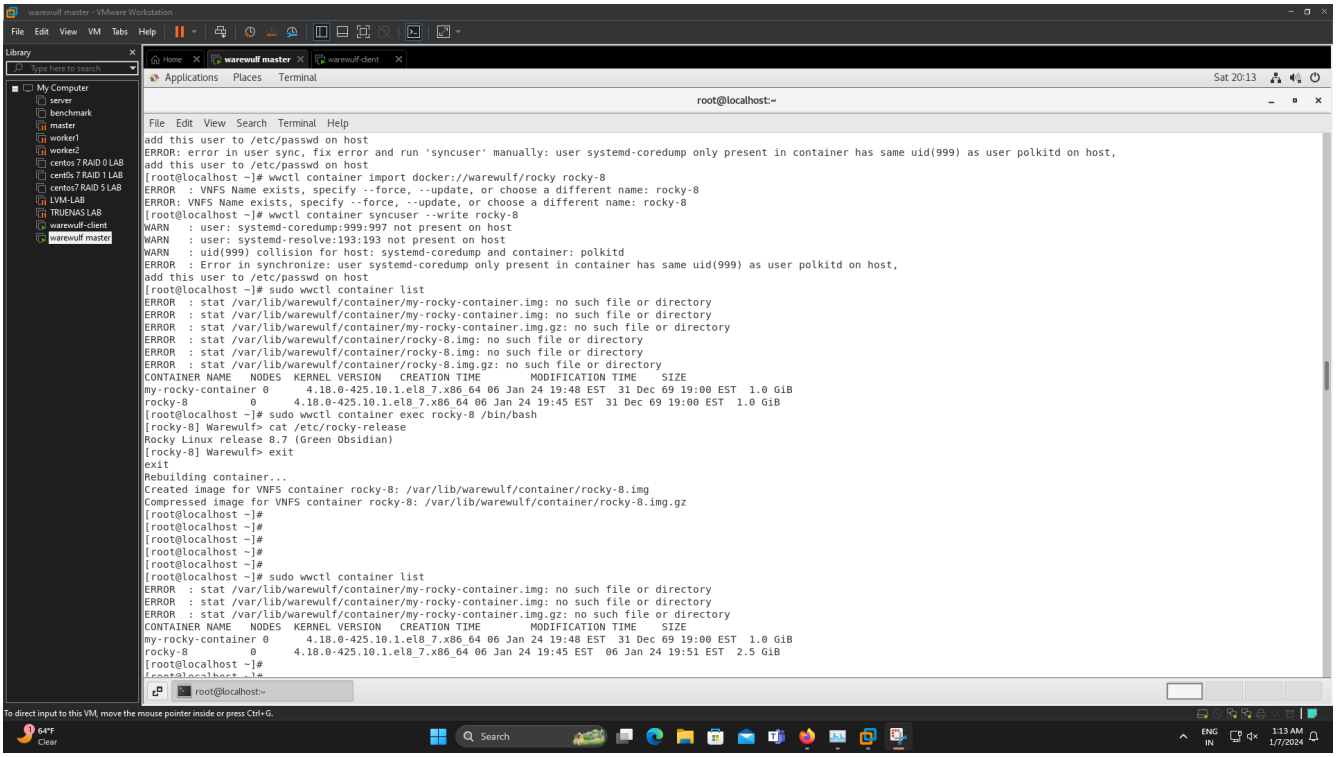
The simplest way is to install the desired kernel in the container or use the one that comes with the container. Warewulf will locate the kernel in an uploaded container by default and use it for any node that uses that container. Note that it can use the same container for nodes but have them boot with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

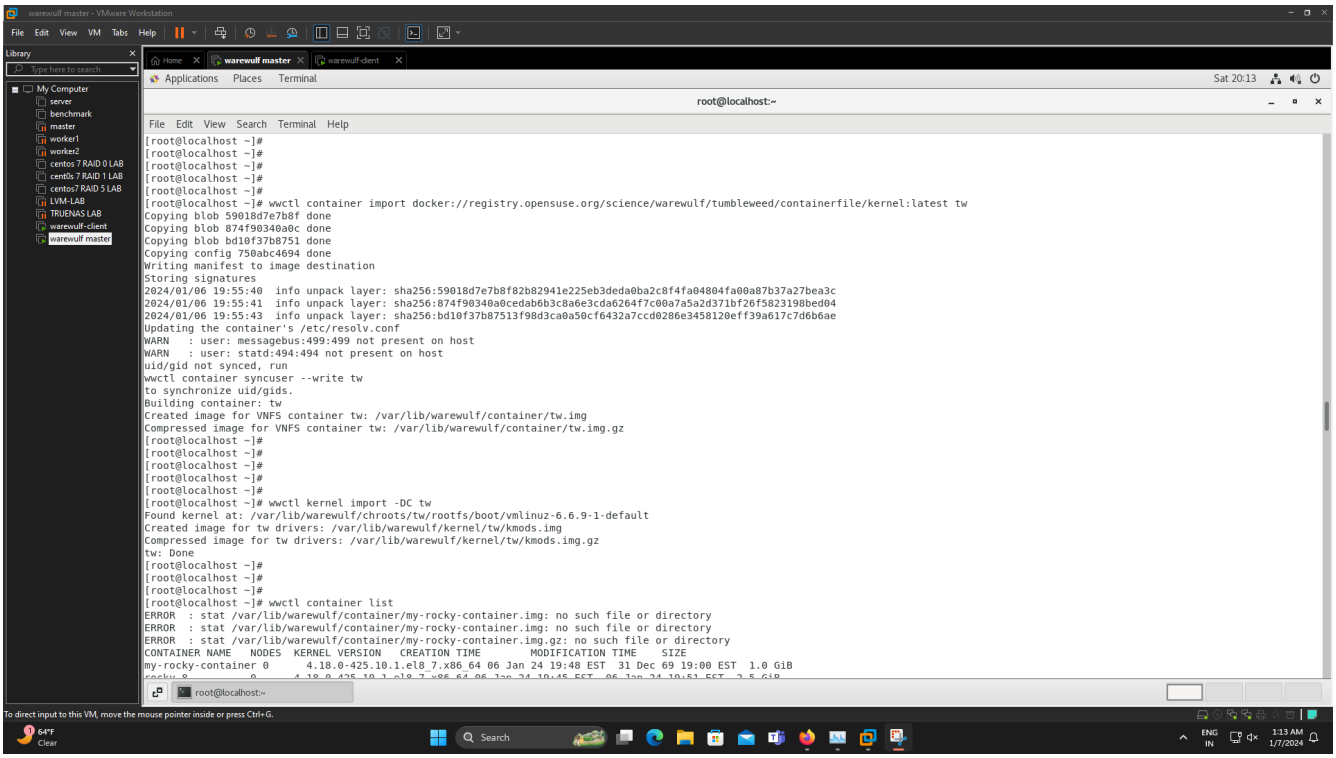
Combining a container with a kernel is done with a *profile*. Personally, I like building containers with specific kernels rather than mixing and matching, but I can see the utility of being able to use separate kernels.

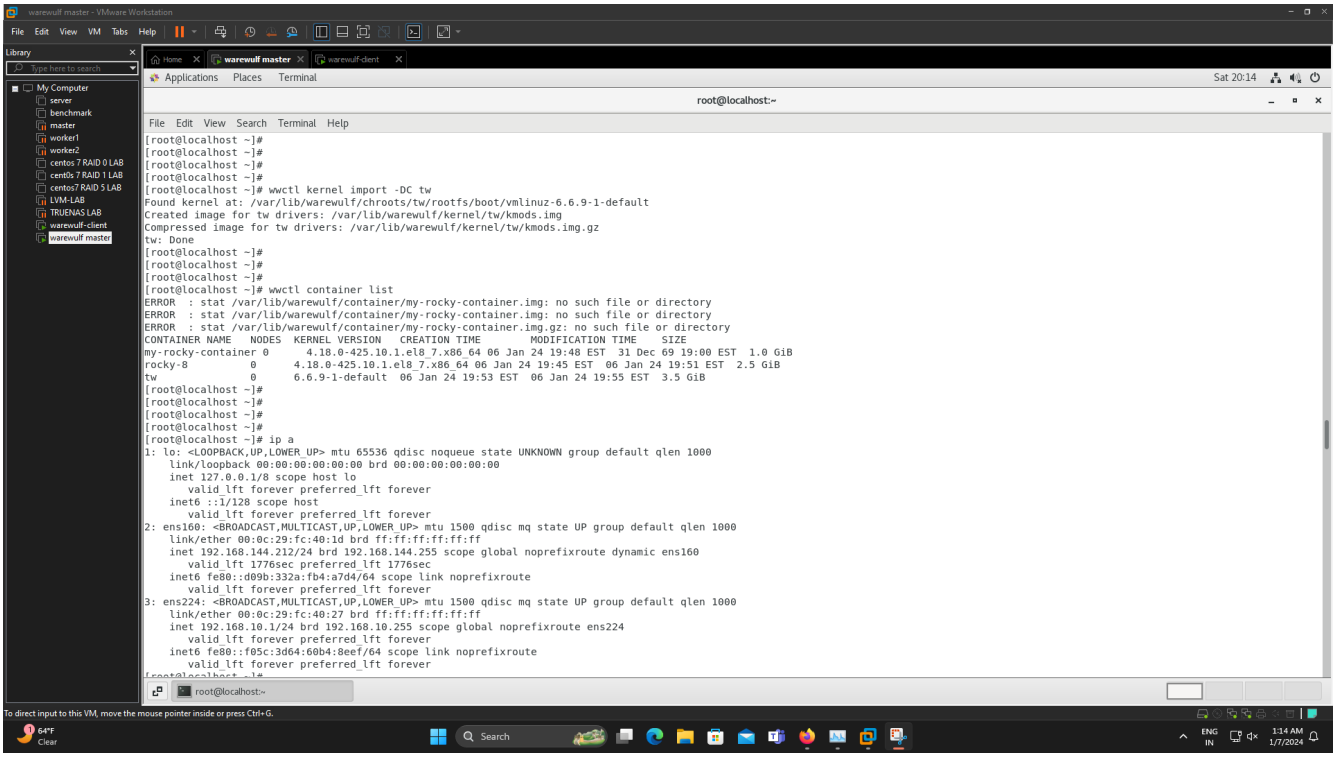
You can see the kernel used in the container with the command:

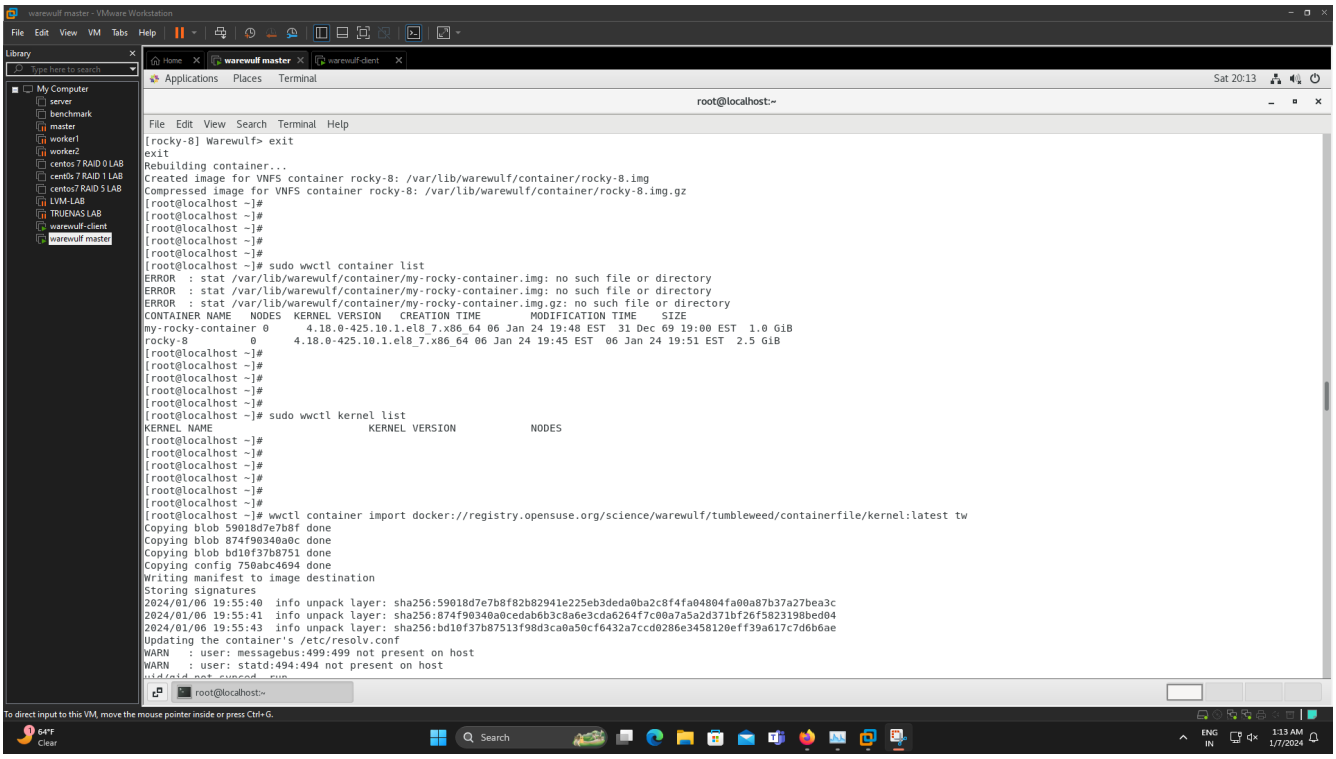
$ sudo wwctl container list

$ sudo wwctl kernel list




warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:13
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# wwctl container import docker://registry.opensuse.org/science/warewulf/tumbleweed/containerfile/kernel:latest tw
Copying blob 59018d7e7b8f done
Copying blob 874f90340a0c done
Copying blob bd10f37b8751 done
Copying config 750abc4694 done
Writing manifest to image destination
Storing signatures
2024/01/06 19:55:40 info unpack layer: sha256:59018d7e7b8f82b82941e225eb3deda0ba2c8f4fa04804fa00a87b37a27bea3c
2024/01/06 19:55:41 info unpack layer: sha256:874f90340a0cedab6b3c8a6e3cda6264f7c00a7a5a2d371bf26f5823198bed04
2024/01/06 19:55:43 info unpack layer: sha256:bd10f37b87513f98d3ca0a50cf6432a7ccd0286e3458120eff39a617c7d6b6ae
Updating the container's /etc/resolv.conf
WARN : user: messagebus:499:499 not present on host
WARN : user: statd:494:494 not present on host
uid/gid not synced, run
wwctl container syncuser --write tw
to synchronize uid/gids.
Building container: tw
Created image for VNFS container tw: /var/lib/warewulf/container/tw.img
Compressed image for VNFS container tw: /var/lib/warewulf/container/tw.img.gz
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# wwctl kernel import -DC tw
Found kernel at: /var/lib/warewulf/chroots/tw/rootfs/boot/vmlinuz-6.6.9-1-default
Created image for tw drivers: /var/lib/warewulf/kernel/tw/kmods.img
Compressed image for tw drivers: /var/lib/warewulf/kernel/tw/kmods.img.gz
tw: Done
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# 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
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


warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:14
root@localhost:~
File Edit View Search Terminal Help
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# wwctl kernel import -DC tw
Found kernel at: /var/lib/warewulf/chroots/tw/rootfs/boot/vmlinuz-6.6.9-1-default
Created image for tw drivers: /var/lib/warewulf/kernel/tw/kmods.img
Compressed image for tw drivers: /var/lib/warewulf/kernel/tw/kmods.img.gz
tw: Done
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# 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
tw 0 6.6.9-1-default 06 Jan 24 19:53 EST 06 Jan 24 19:55 EST 3.5 GiB
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
inet 127.0.0.1/8 scope host lo
valid_lft forever preferred_lft forever
inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:0c:29:fc:40:1d brd ff:ff:ff:ff:ff:ff
inet 192.168.144.212/24 brd 192.168.144.255 scope global noprefixroute dynamic ens160
valid_lft 1776sec preferred_lft 1776sec
inet6 fe80::d09b:332a:fb4:a7d4/64 scope link noprefixroute
valid_lft forever preferred_lft forever
3: ens224: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
link/ether 00:0c:29:fc:40:27 brd ff:ff:ff:ff:ff:ff
inet 192.168.10.1/24 brd 192.168.10.255 scope global noprefixroute ens224
valid_lft forever preferred_lft forever
inet6 fe80::f05c:3d64:60b4:8eef/64 scope link noprefixroute
valid_lft forever preferred_lft forever
root@localhost:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F Clear
Search
ENG IN
1:14 AM 1/7/2024



Configuring Compute Nodes

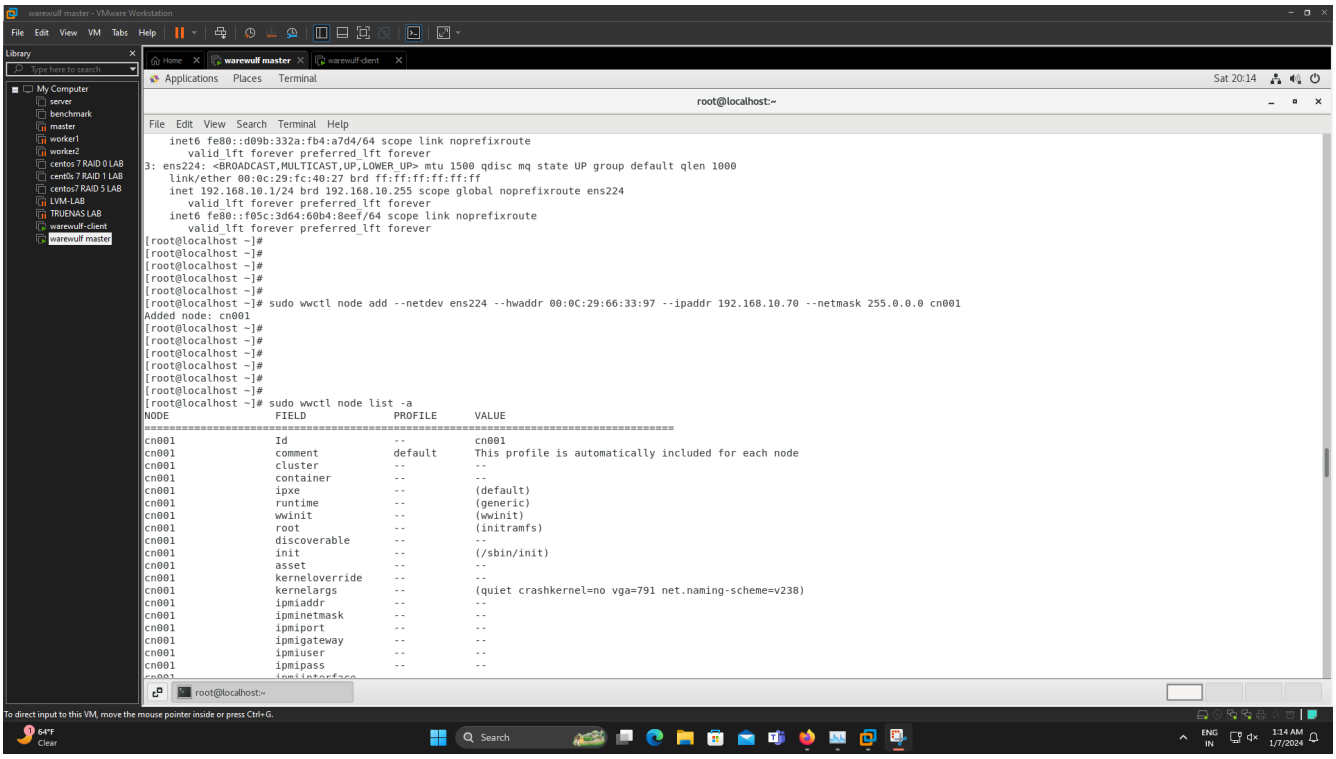
You're almost ready to start booting compute nodes, but you need to configure them in Warewulf before doing so. Recall that */etc/warewulf/nodes.conf* is the database of compute nodes in a simple flat text YAML format. You can get the list of defined compute nodes with the command:

```
$ 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



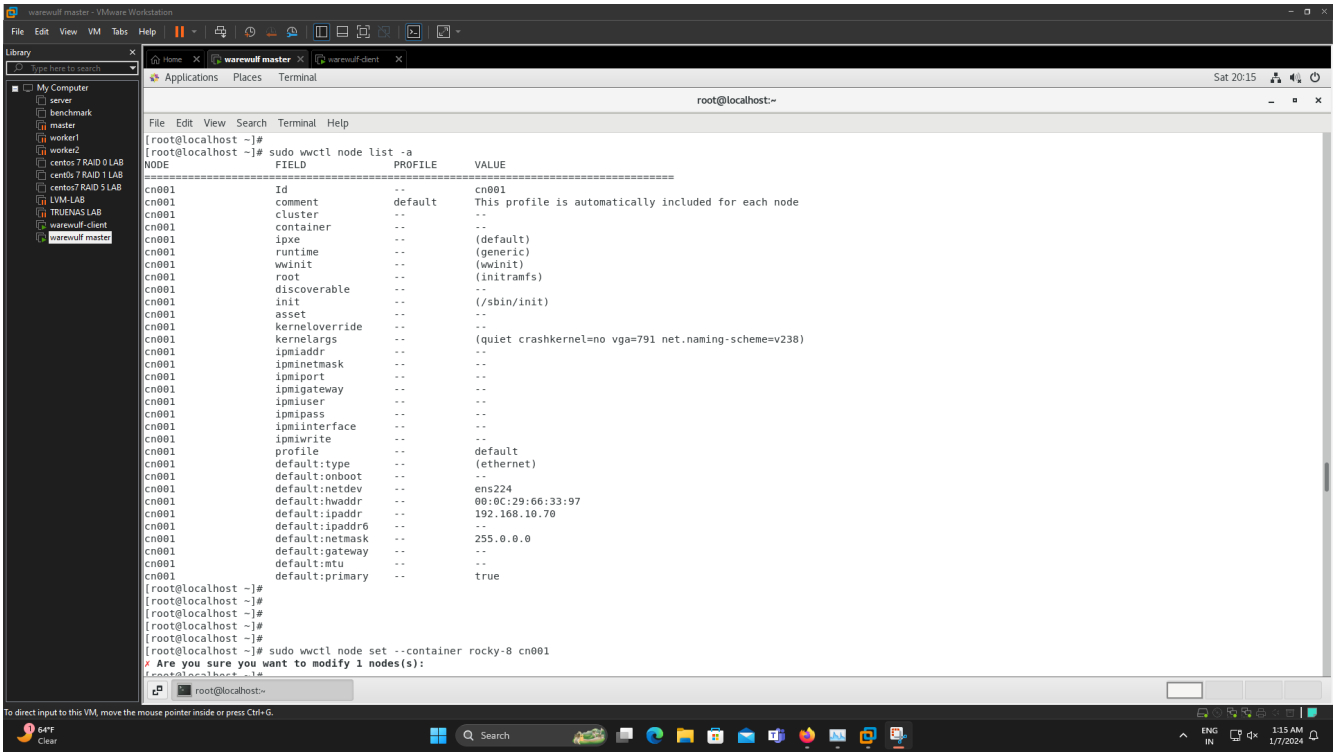
If you don't know the MAC address of the compute node, you can boot the nodes one by one and have Warewulf capture the MAC–node name association. When you define a node in Warewulf for which you don’t know the MAC, you use the command:

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

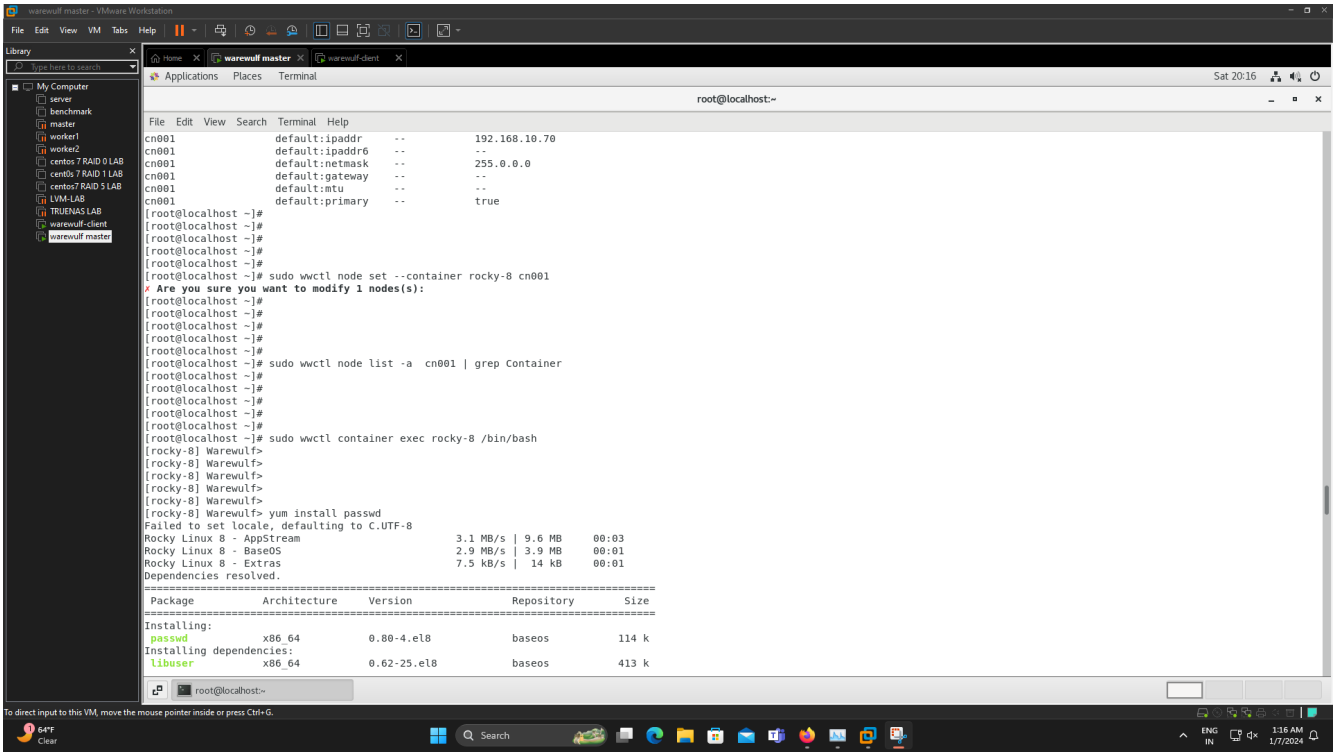
List of Defined Nodes



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



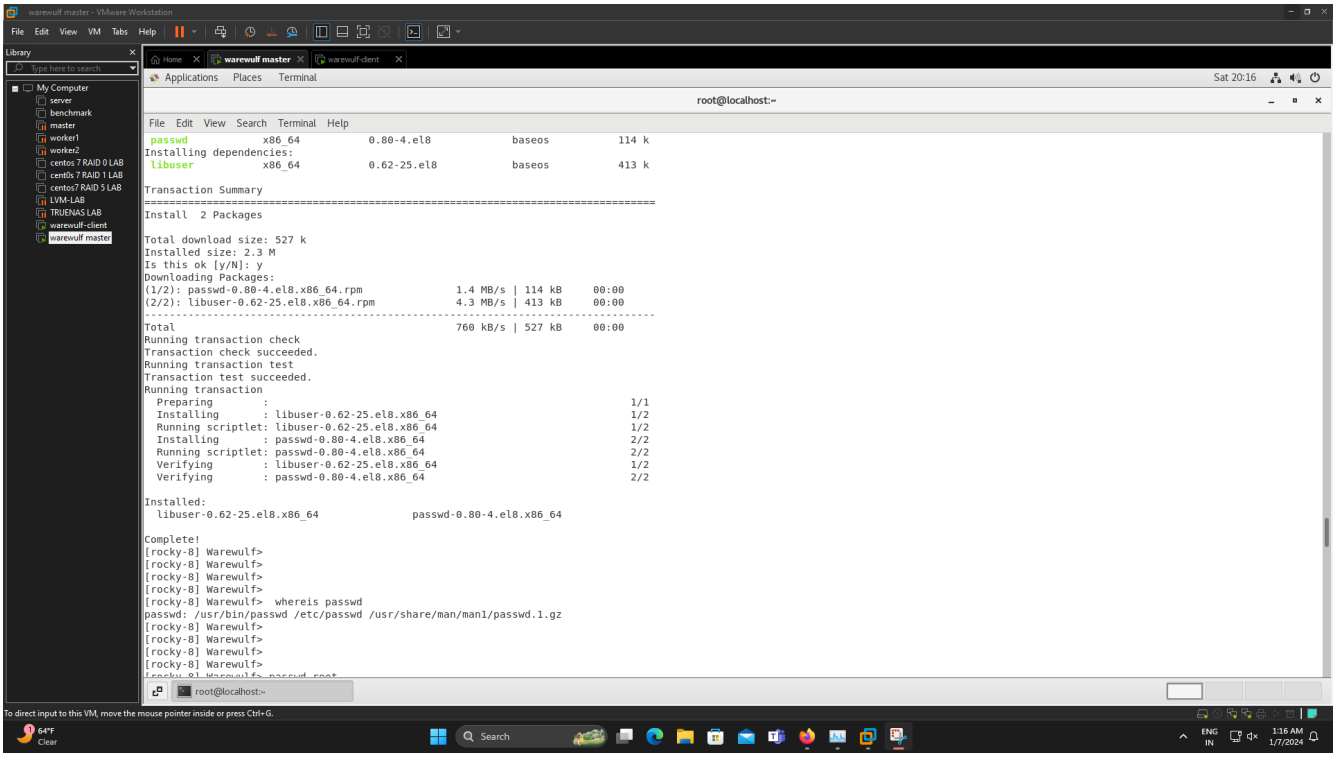
To add packages to the container, *exec* into the container:

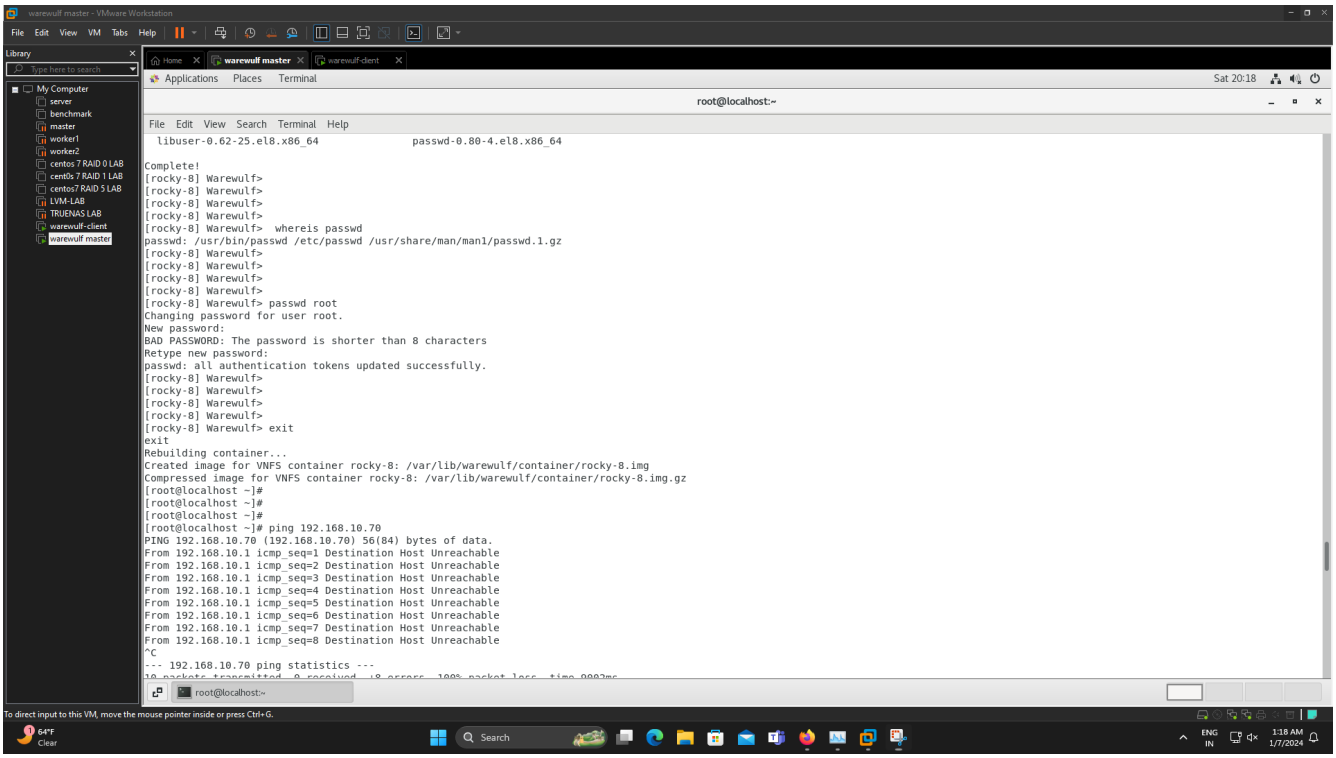
```
$ sudo wwctl container exec rocky-8 /bin/bash
```

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


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
cent0s 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:16
root@localhost:~
File Edit View Search Terminal Help
passwd x86_64 0.80-4.el8 baseos 114 k
Installing dependencies:
libuser x86_64 0.62-25.el8 baseos 413 k
Transaction Summary
Install 2 Packages
Total download size: 527 k
Installed size: 2.3 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): passwd-0.80-4.el8.x86_64.rpm 1.4 MB/s | 114 kB 00:00
(2/2): libuser-0.62-25.el8.x86_64.rpm 4.3 MB/s | 413 kB 00:00
Total 760 kB/s | 527 kB 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : libuser-0.62-25.el8.x86_64 1/2
Running scriptlet: libuser-0.62-25.el8.x86_64 1/2
Installing : passwd-0.80-4.el8.x86_64 2/2
Running scriptlet: passwd-0.80-4.el8.x86_64 2/2
Verifying : libuser-0.62-25.el8.x86_64 1/2
Verifying : passwd-0.80-4.el8.x86_64 2/2
Installed:
libuser-0.62-25.el8.x86_64 passwd-0.80-4.el8.x86_64
Complete!
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf> whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1.gz
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf>
[rocky-8] Warewulf>
root@localhost:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F
Clear
Search
ENG
IN
1:16 AM
1/7/2024



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.

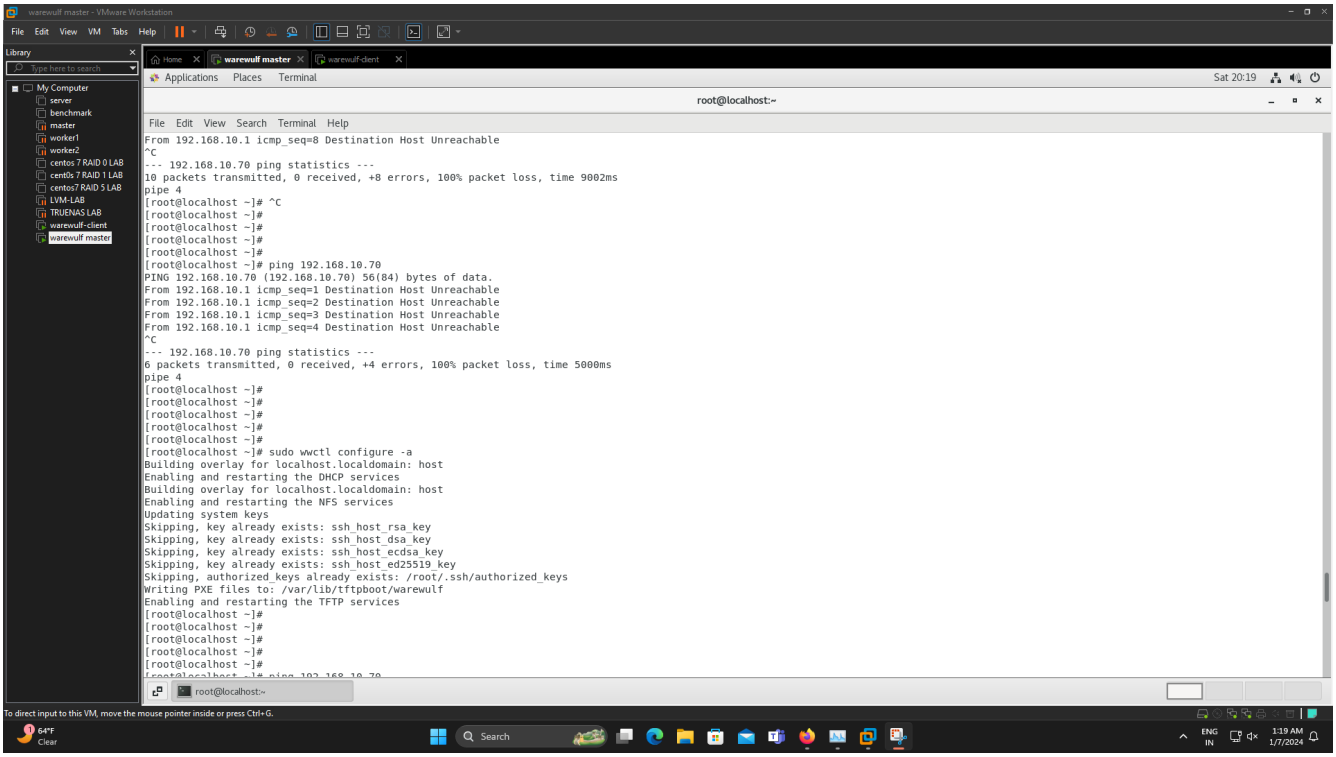
Because only SSH is used for authentication, you won't be able to log in to the compute node as a user; however, the rootuser has a password, so you can log in as root with the monitor and keyboard. When you do this, poke around the compute node and make sure everything looks correct. For example,

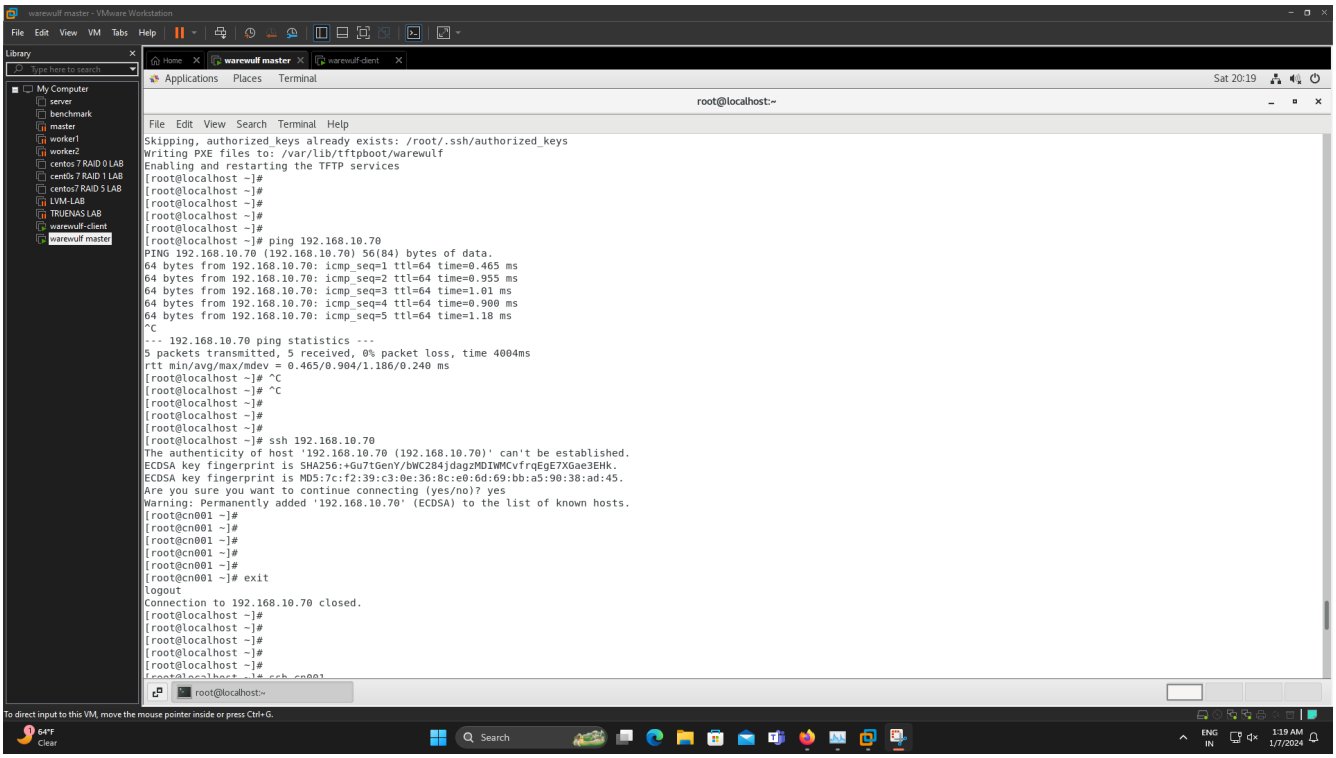
- check that the NFS exported files have mounted (use the *mount* command),
- look at the local IP address (*ip addr show*),
- ping the host node with the IP address (*ping 192.168.10.70*), and
- check the hostname of the node (*more /etc/hostname*).

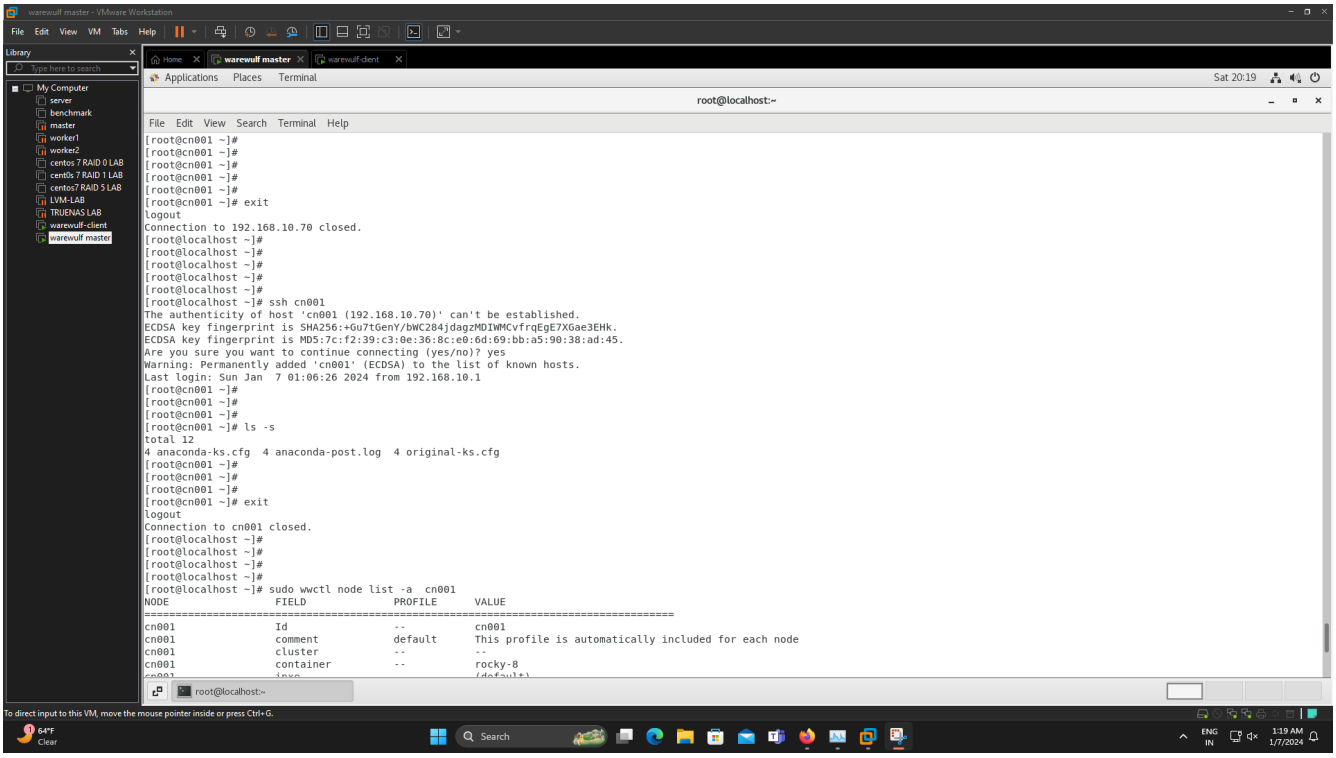
These are just a few things you can do to check the compute node, but you can do whatever you like to satisfy that the compute node is up and working correctly.

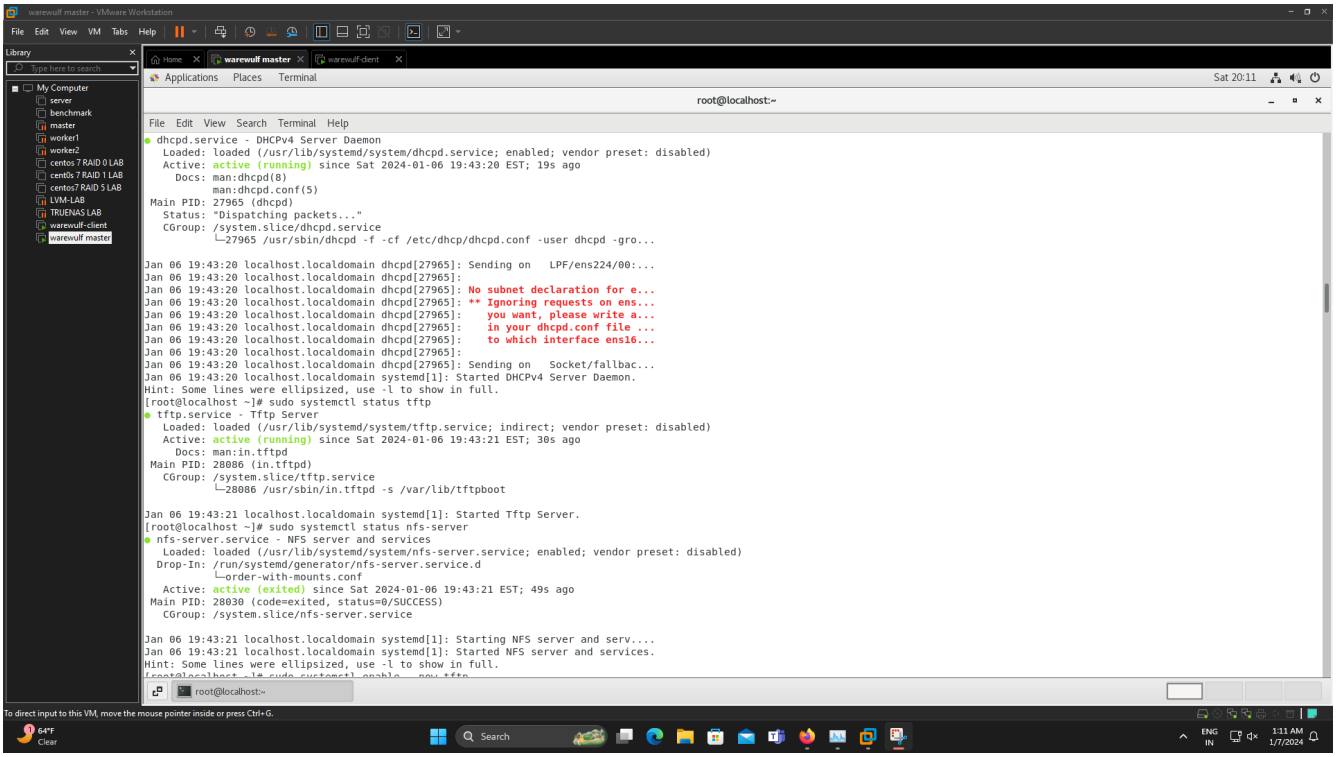
At this point, you might go back to the host node and check out the compute node. The first thing to do is ping the IP address of the compute node then ping the hostname of the compute node .

Pinging the Compute IP Address

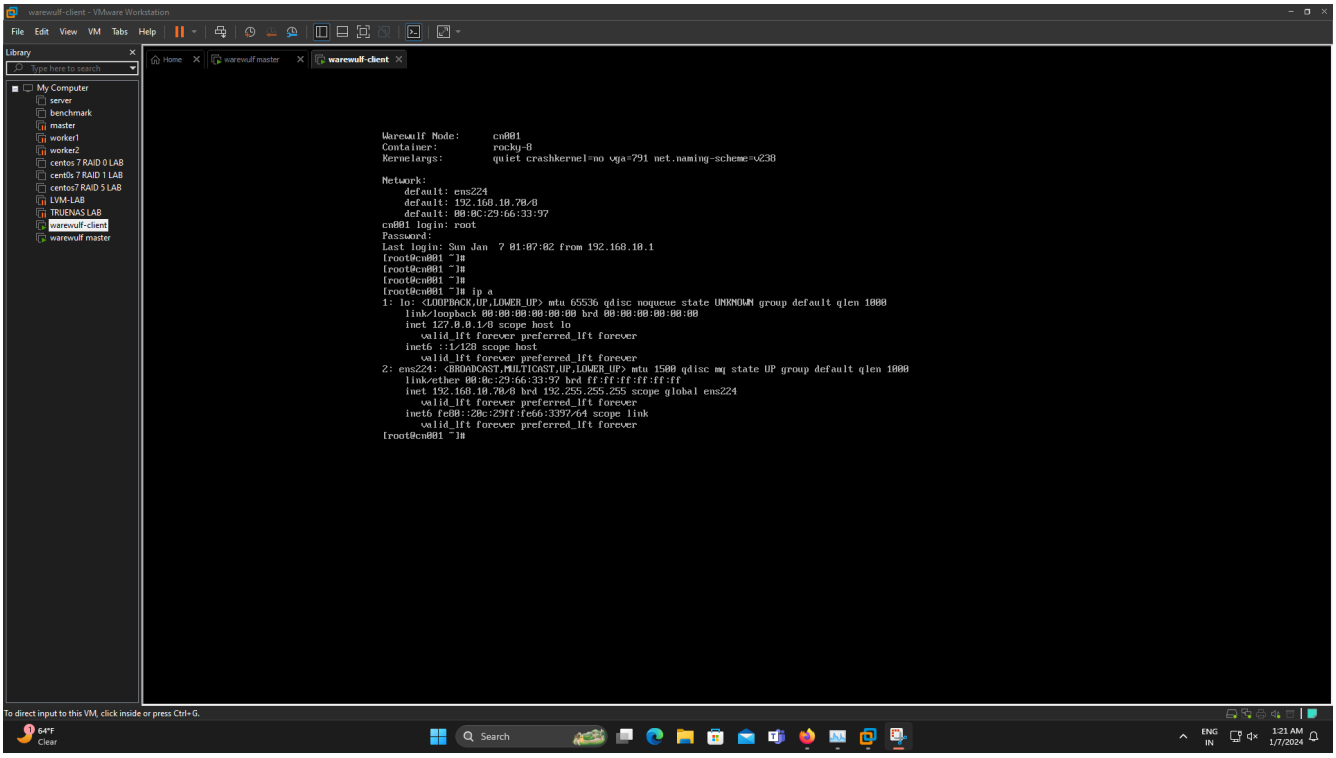


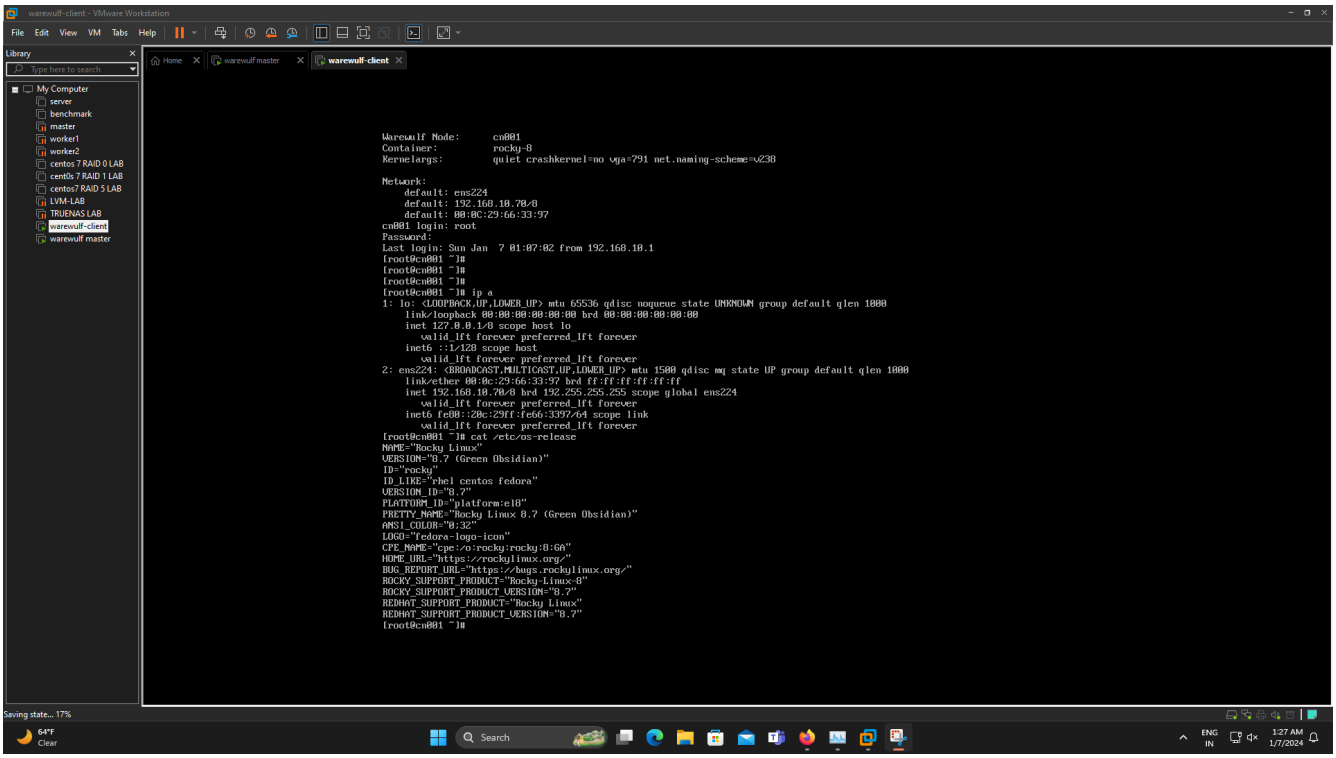

warewulf master - VMware Workstation
File Edit View VM Tabs Help
Library
Type here to search
My Computer
server
benchmark
master
worker1
worker2
centos 7 RAID 0 LAB
centos 7 RAID 1 LAB
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:19
root@localhost:~
File Edit View Search Terminal Help
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 ~]#
[root@localhost ~]#
[root@localhost ~]# ping 192.168.10.70
PING 192.168.10.70 (192.168.10.70) 56(84) bytes of data.
64 bytes from 192.168.10.70: icmp_seq=1 ttl=64 time=0.465 ms
64 bytes from 192.168.10.70: icmp_seq=2 ttl=64 time=0.955 ms
64 bytes from 192.168.10.70: icmp_seq=3 ttl=64 time=1.01 ms
64 bytes from 192.168.10.70: icmp_seq=4 ttl=64 time=0.900 ms
64 bytes from 192.168.10.70: icmp_seq=5 ttl=64 time=1.18 ms
^C
--- 192.168.10.70 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4004ms
rtt min/avg/max/mdev = 0.465/0.904/1.186/0.240 ms
[root@localhost ~]# ^C
[root@localhost ~]# ^C
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ssh 192.168.10.70
The authenticity of host '192.168.10.70 (192.168.10.70)' can't be established.
ECDSA key fingerprint is SHA256:+Gu7tGenY/bWC284jdagzMDIWMCvfrqEgE7XGae3EHk.
ECDSA key fingerprint is MD5:7c:f2:39:c3:0e:36:8c:e0:6d:69:bb:a5:90:38:ad:45.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.10.70' (ECDSA) to the list of known hosts.
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]# exit
logout
Connection to 192.168.10.70 closed.
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[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
1:19 AM
1/7/2024


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
centos7 RAID 5 LAB
LVM-LAB
TRUENAS LAB
warewulf-client
warewulf master
Home
warewulf master
warewulf-client
Applications Places Terminal
Sat 20:19
root@localhost:~
File Edit View Search Terminal Help
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]# exit
logout
Connection to 192.168.10.70 closed.
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# ssh cn001
The authenticity of host 'cn001 (192.168.10.70)' can't be established.
ECDSA key fingerprint is SHA256:+Gu7tGenY/bWC284jdagzMDIWMCvfrqEgE7XGae3EHk.
ECDSA key fingerprint is MD5:7c:f2:39:c3:0e:36:8c:e0:6d:69:bb:a5:90:38:ad:45.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'cn001' (ECDSA) to the list of known hosts.
Last login: Sun Jan 7 01:06:26 2024 from 192.168.10.1
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]# ls -s
total 12
4 anaconda-ks.cfg 4 anaconda-post.log 4 original-ks.cfg
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]#
[root@cn001 ~]# exit
logout
Connection to cn001 closed.
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]#
[root@localhost ~]# sudo wwctl node list -a cn001
NODE FIELD PROFILE VALUE
cn001 Id -- cn001
cn001 comment default This profile is automatically included for each node
cn001 cluster -- --
cn001 container -- rocky-8
root@localhost:~
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
64°F
Clear
Search
ENG
IN
1:19 AM
1/7/2024



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.