

*Project Report*  
*on*  
***Deploying HPC Cluster  
On Containers***



*Submitted  
in partial fulfillment  
for the award of the Degree of*

***Post Graduate Diploma in High Performance Computing  
System Administration***

***C-DAC ACTS (Pune)***

***Guided By:***  
Mr. Roshan Gami

***Submitted By:***  
Atharv Pise (240840127006)  
Kaustubh Veer (240840127012)  
Eshan (240840127009)  
Abhijit Kadam (240840127003)  
Hiteshwari Dukare (240840127008)

***Centre of Development of Advanced Computing  
(C-DAC), ACTS (Pune- 411008)***

---

## *ACKNOWLEDGEMENT*

---

This project “ **Deploying HPC Cluster on Containers** ” was a great learning experience for us and we are submitting this work to Advanced Computing Training School (CDAC ACTS).

We all are very glad to mention the name of **Mr. Roshan Gami** for his valuable guidance to work on this project. Overcome various obstacles and intricacies during the course of project work.

We are highly grateful to HPC tech team (ACTS training Centre, C-DAC), For his valuable guidance and support whenever necessary while doing this course Post Graduate Diploma in **High Performance Computing System Administration (PG- DHPCSA)** Through CDAC ACTS , Pune.

Our most heartfelt thank goes to **Ms. Pratiksha Gacche** (Course Coordinator, PG- DHPCSA) who gave all the required support and kind coordination to provide all the necessities like required hardware, internet facility and extra Lab hours to complete the project and throughout the course up to the last day here in C-DAC ACTS, Pune.

*From:*

**Atharv Pise (240840127006)**  
**Kaustubh Veer (240840127012)**  
**Eshan (240840127009)**  
**Abhijit Kadam (240840127003)**  
**Hiteshwari Dukare (240840127008)**

## ***ABSTRACT***

High-Performance Computing (HPC) is used to solve complex problems that need a lot of computing power. Traditional HPC clusters run on physical machines, which makes them hard to scale, manage, and maintain. Container technology provides a better way by making HPC clusters more flexible, portable, and easy to manage.

This project focuses on building an HPC cluster inside containers using Warewulf, SLURM, and Ganglia. Warewulf helps in setting up and managing cluster nodes. SLURM is used for job scheduling and resource allocation, making sure tasks run efficiently. Ganglia is added to monitor cluster performance and resource usage in real-time.

By using containers, this project makes it easier to deploy and manage HPC clusters. It reduces the setup time, improves flexibility, and removes hardware limitations. The system is tested with benchmark applications to check its performance and scalability. This work shows how container-based HPC clusters can be a better option for modern computing needs.

---

## ***TABLE OF CONTENTS***

---

S. No	Title	Page No.
	<b>Front Page</b>	<b>I</b>
	<b>Acknowledgement</b>	<b>II</b>
	<b>Abstract</b>	<b>III</b>
	<b>Table of Contents</b>	<b>IV</b>
<b>1</b>	<b>Introduction</b>	<b>5</b>
<b>2</b>	<b>Workflow</b>	<b>6</b>
<b>3</b>	<b>System Requirements</b>	<b>7</b>
<b>3.1</b>	Software	
<b>3.2</b>	Hardware	
<b>4</b>	<b>Setting up the Master</b>	<b>8 - 18</b>
<b>5</b>	<b>Installation of Warewulf</b>	<b>19 - 26</b>
<b>5.1</b>	Setting up the Node	
<b>5.2</b>	Provisioning of the Node	
<b>6</b>	<b>Installation of SLURM</b>	<b>27 - 36</b>
<b>7</b>	<b>Installation of Ganglia</b>	<b>37 - 41</b>
<b>8</b>	<b>Troubleshooting</b>	<b>42</b>
<b>9</b>	<b>References and Bibliography</b>	<b>44</b>
<b>10</b>	<b>Project Link</b>	<b>44</b>

---

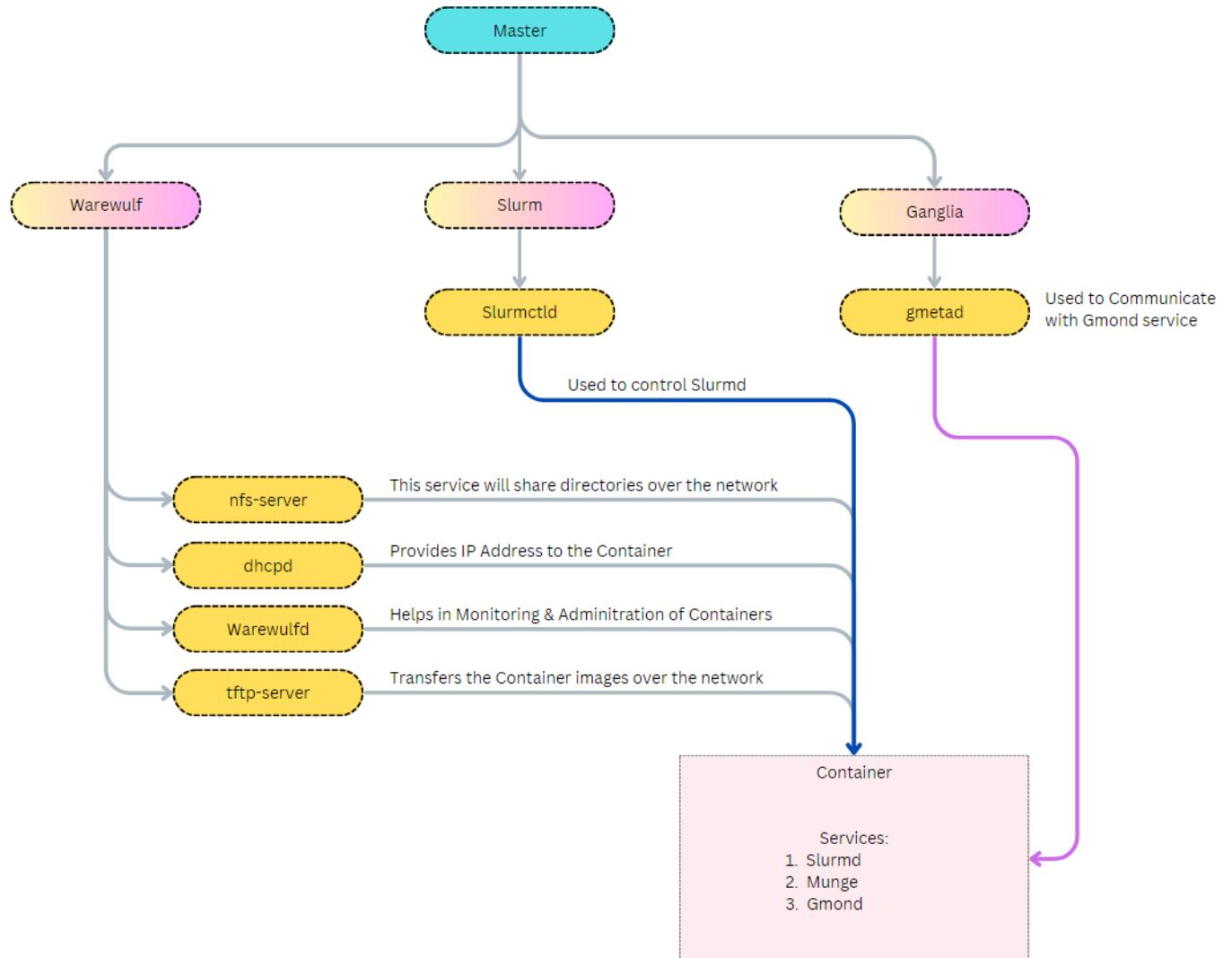
## *Introduction*

---

This project is about deployment of HPC on Containers using Warewulf.

For monitoring we have Ganglia. SLURM has been used to manage Resources and accounting. This HPC stacks uses Rocky 8.8 Linux as an Alternative to Centos 9.

## Workflow



---

## *System Requirements*

---

- RAM: 16 Gb
- HDD: 100 Gb
- Processors: 4 cores
- Network adapters
  - NAT
  - Host Only

---

## *Software Requirements*

---

- Rocky Linux (8.8)
- Warewulf
- SLURM
- Ganglia

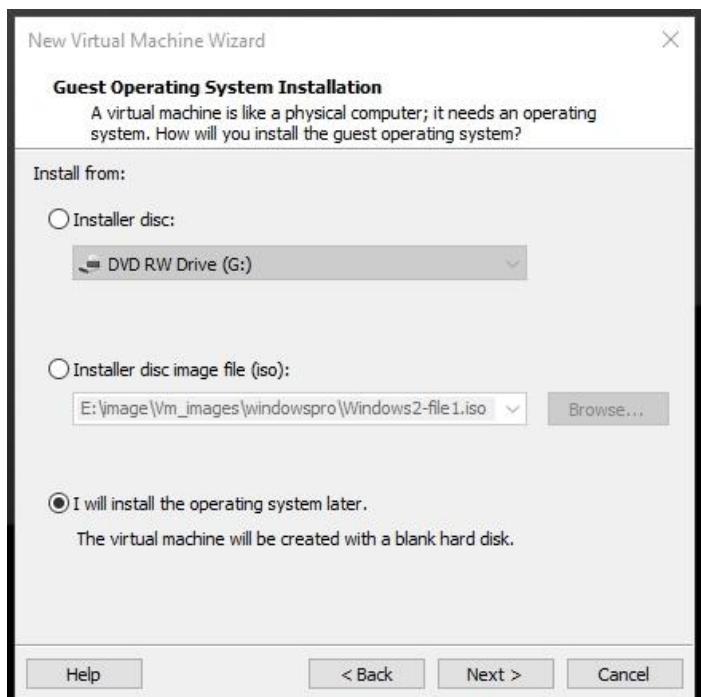
## *Setting up the master*

- create a Virtual machine for master node

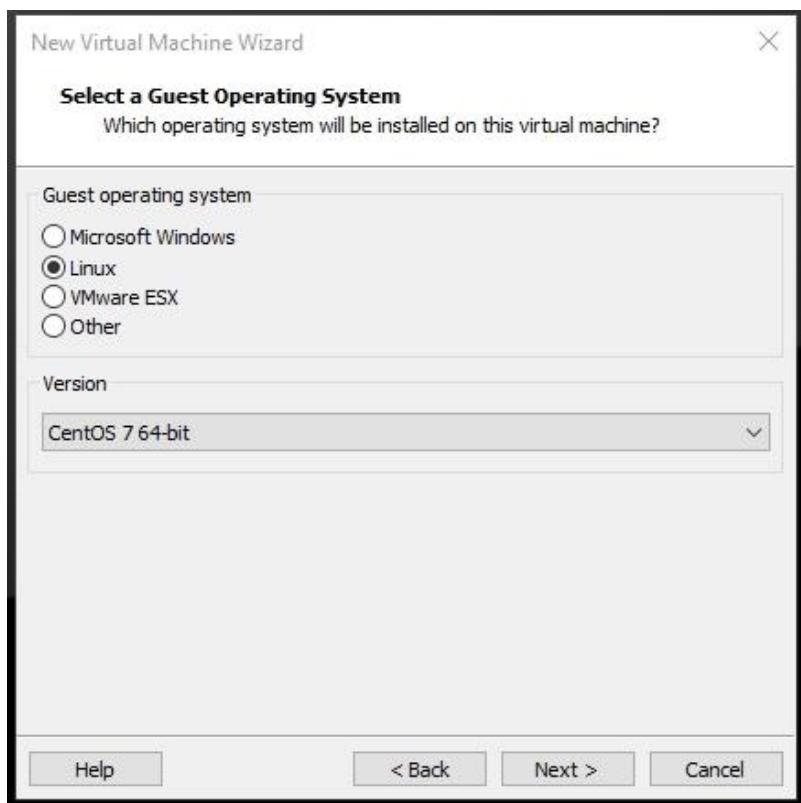
Step 1: Create new virtual machine in VMware Workstation



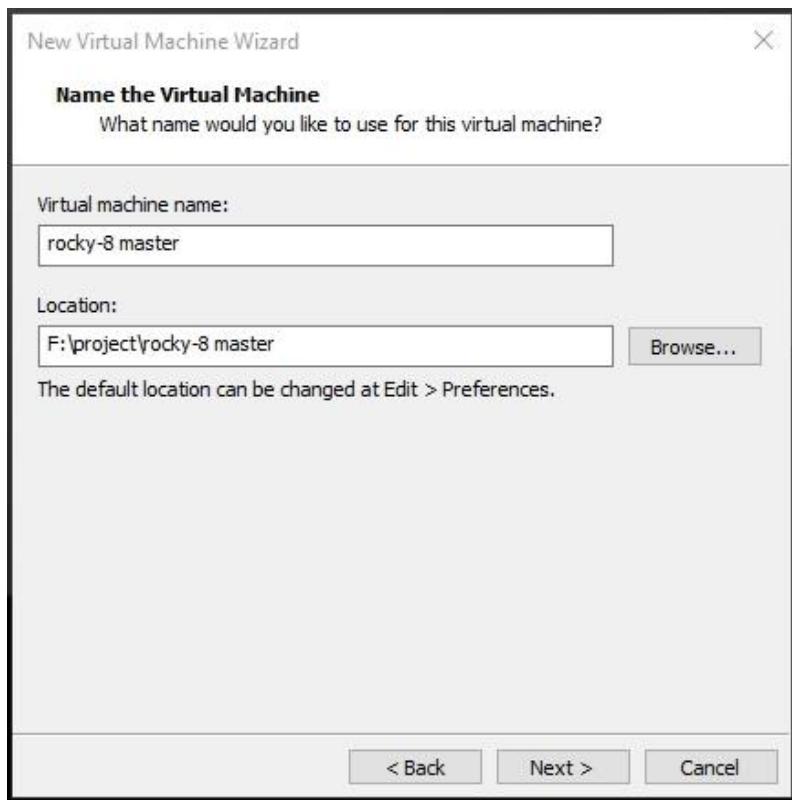
Step 2: Click next & choose `I will install Operating System Later`



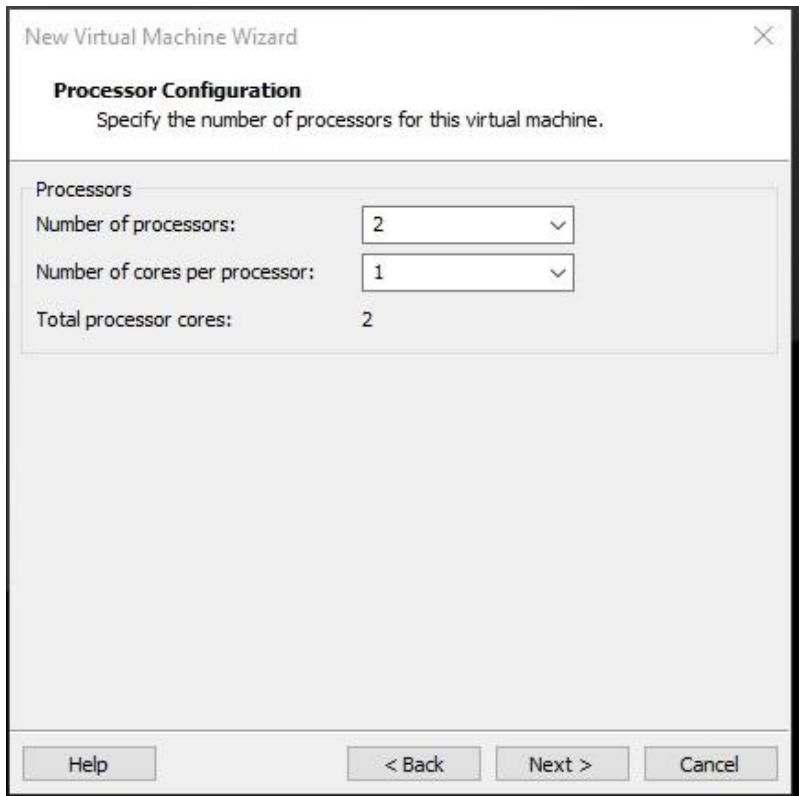
## Step 3: Select Guest Operating System Type



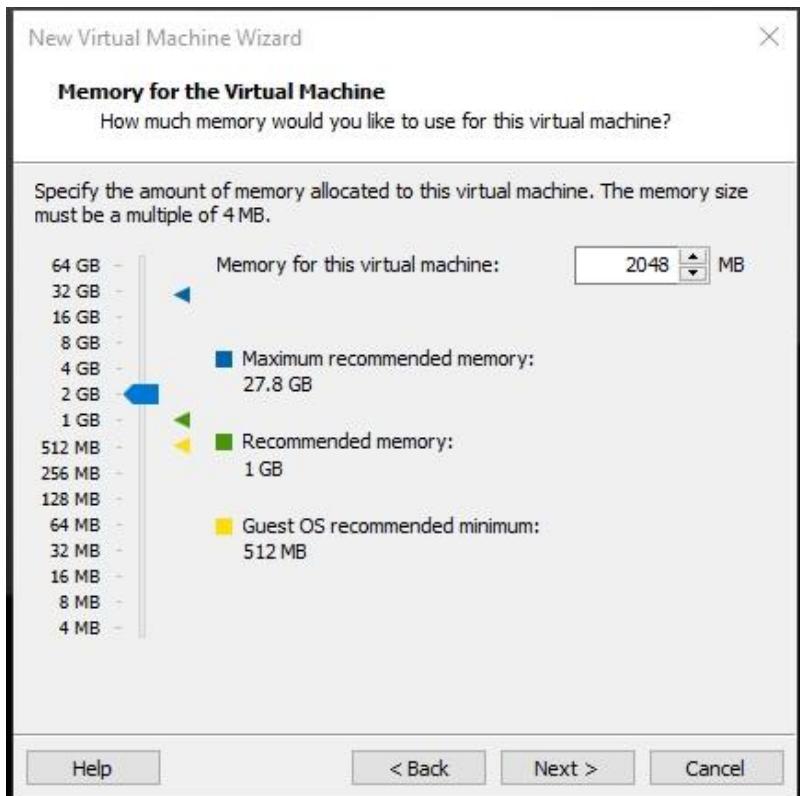
## Step 4: Give name to your New VM



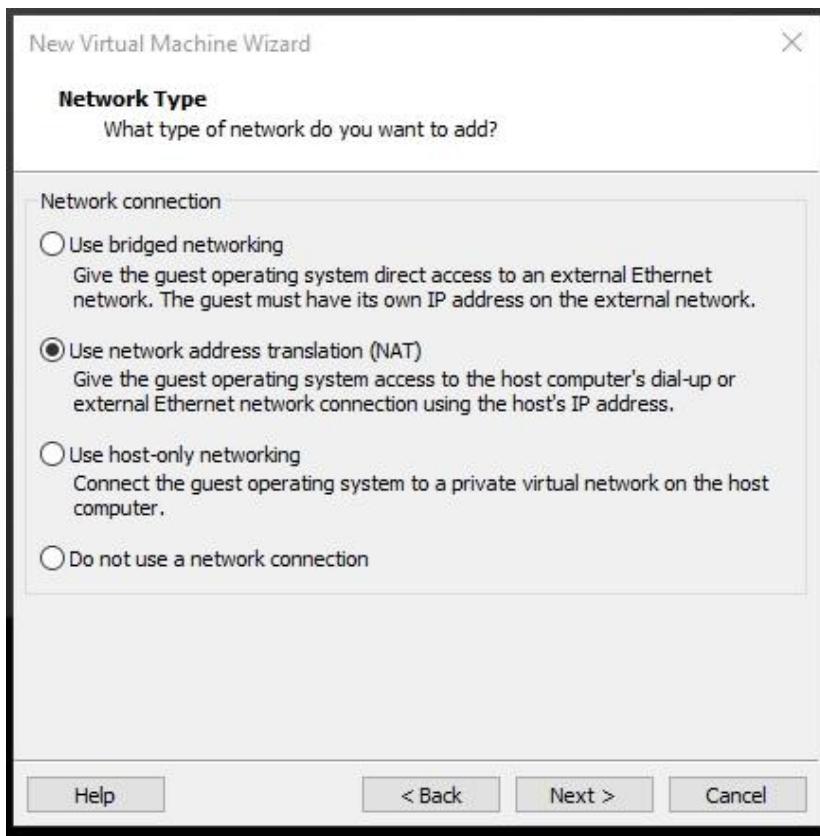
## Step 5: Select no of processors for master



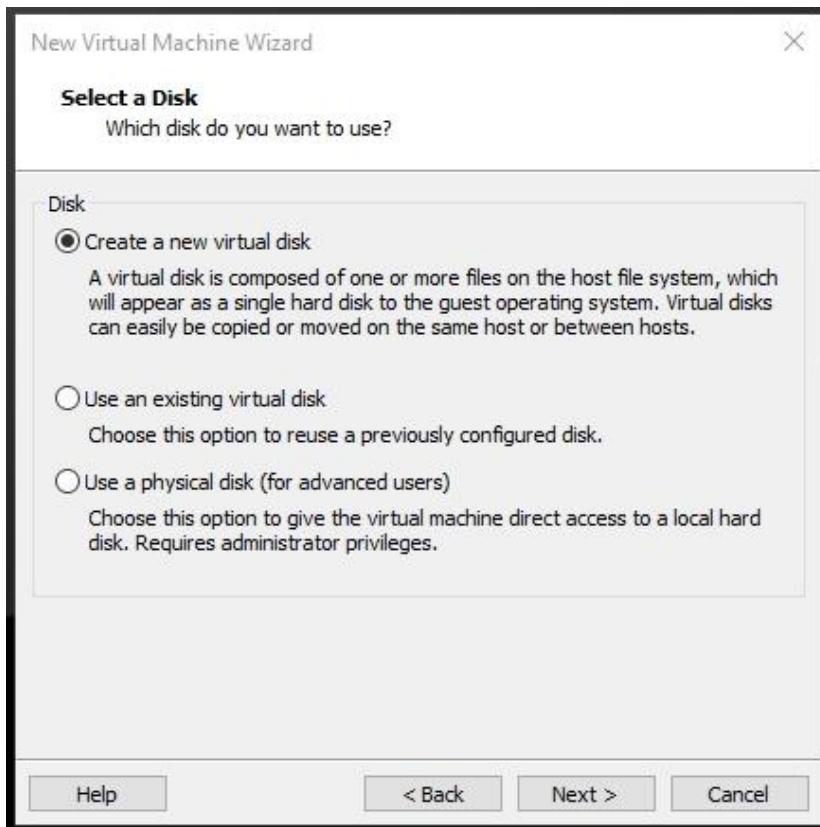
## Step 6: next step is to define ram for new master machine

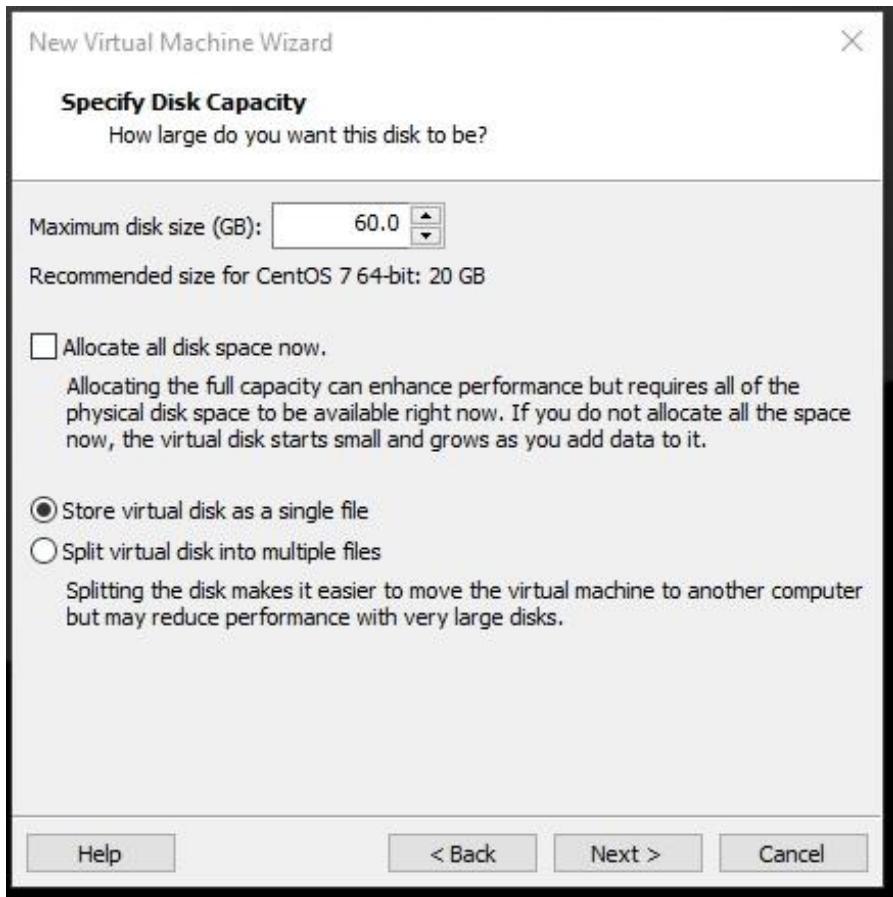


## Step 7: Next step is to select Network Type

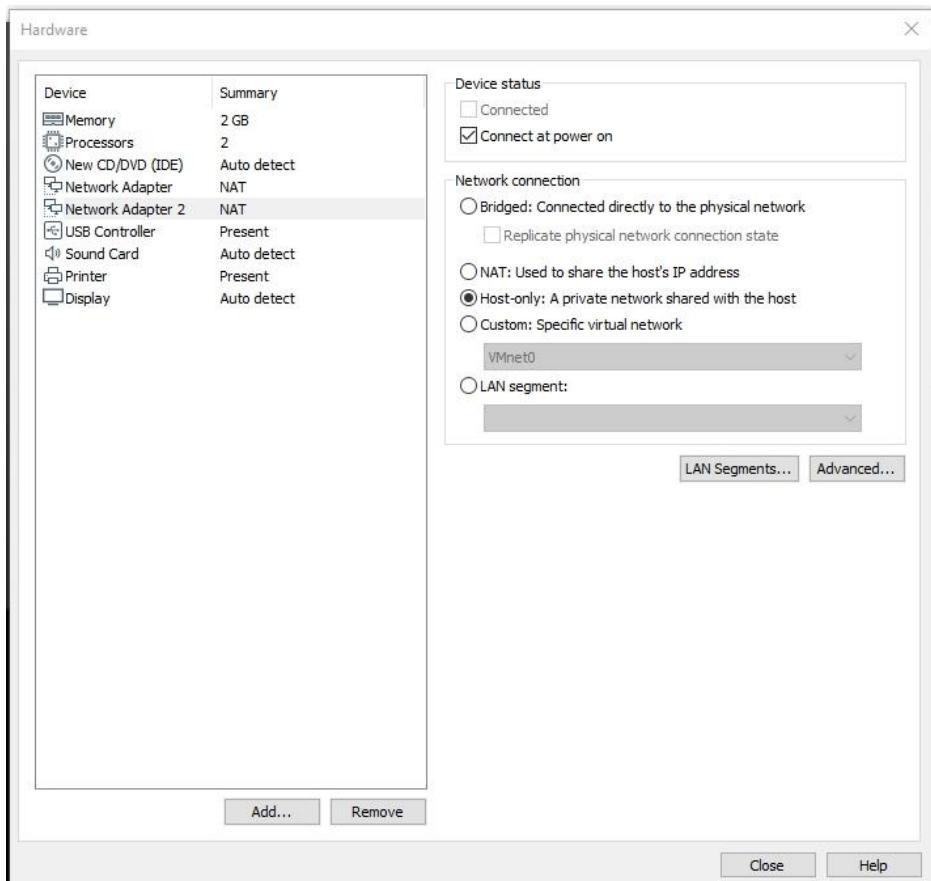


Step 8: Next step is to select disk & it's size

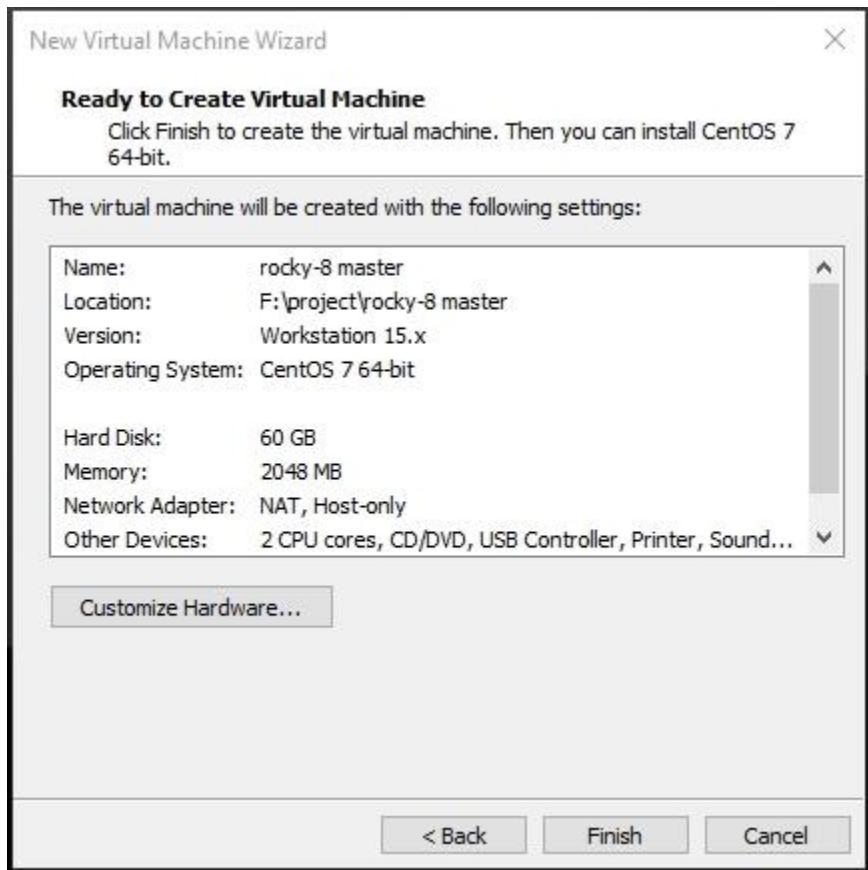




Step 9: Next Step is to add another network adapter & change its type to Host-only

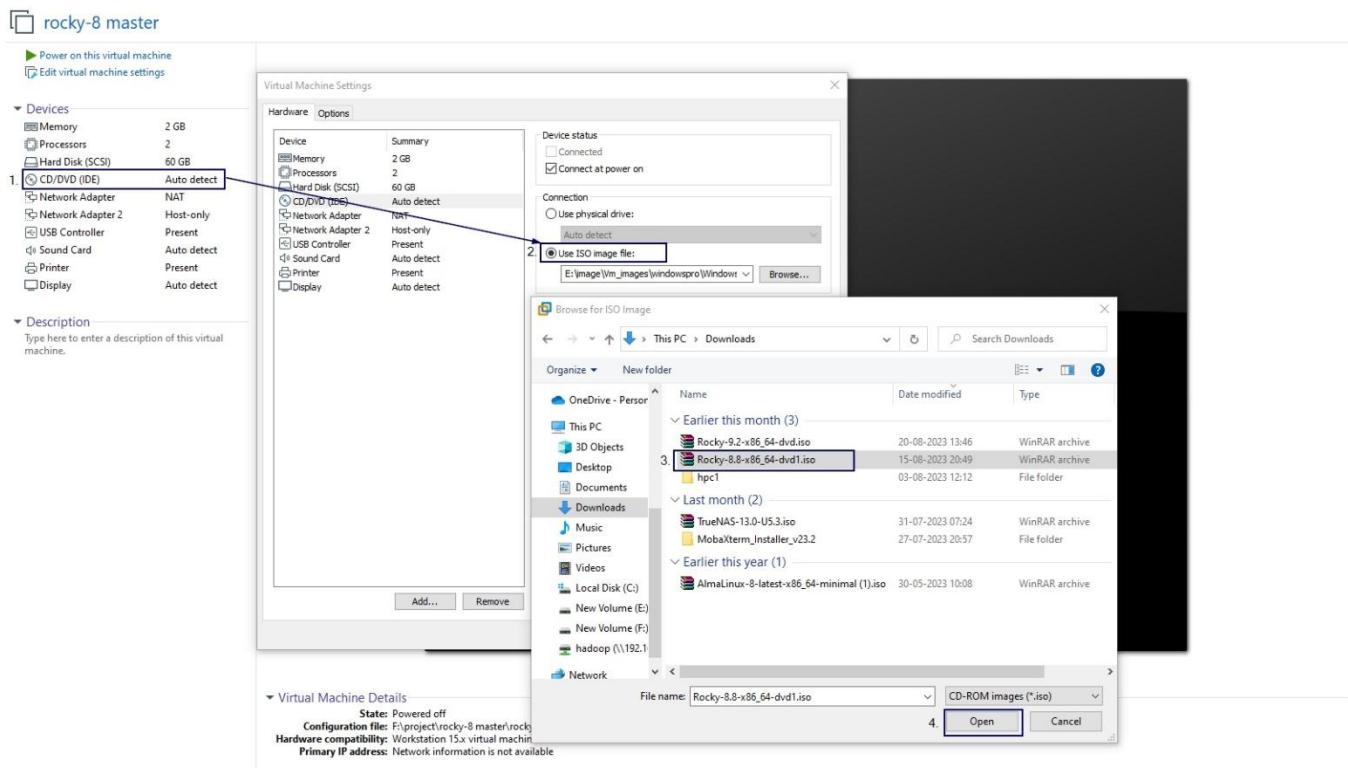


## Step 10: Final step is to confirm the configuration and click finish

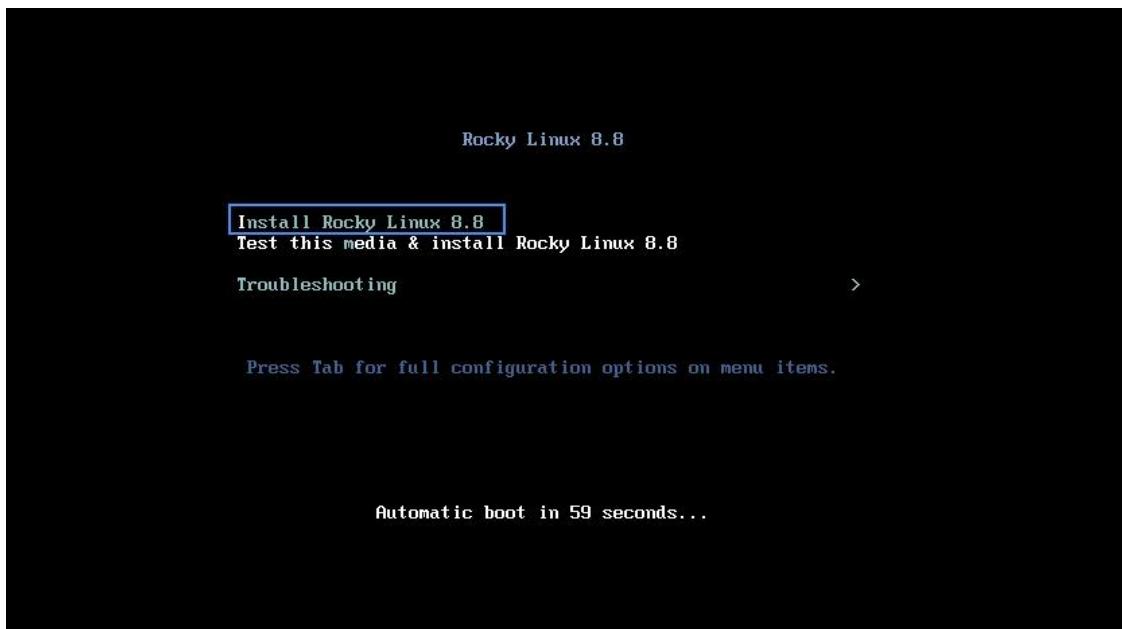


## Installing OS on new VM

Step 1: select boot device and choose rocky Linux 8.8 iso & click open. Then start the virtual machine



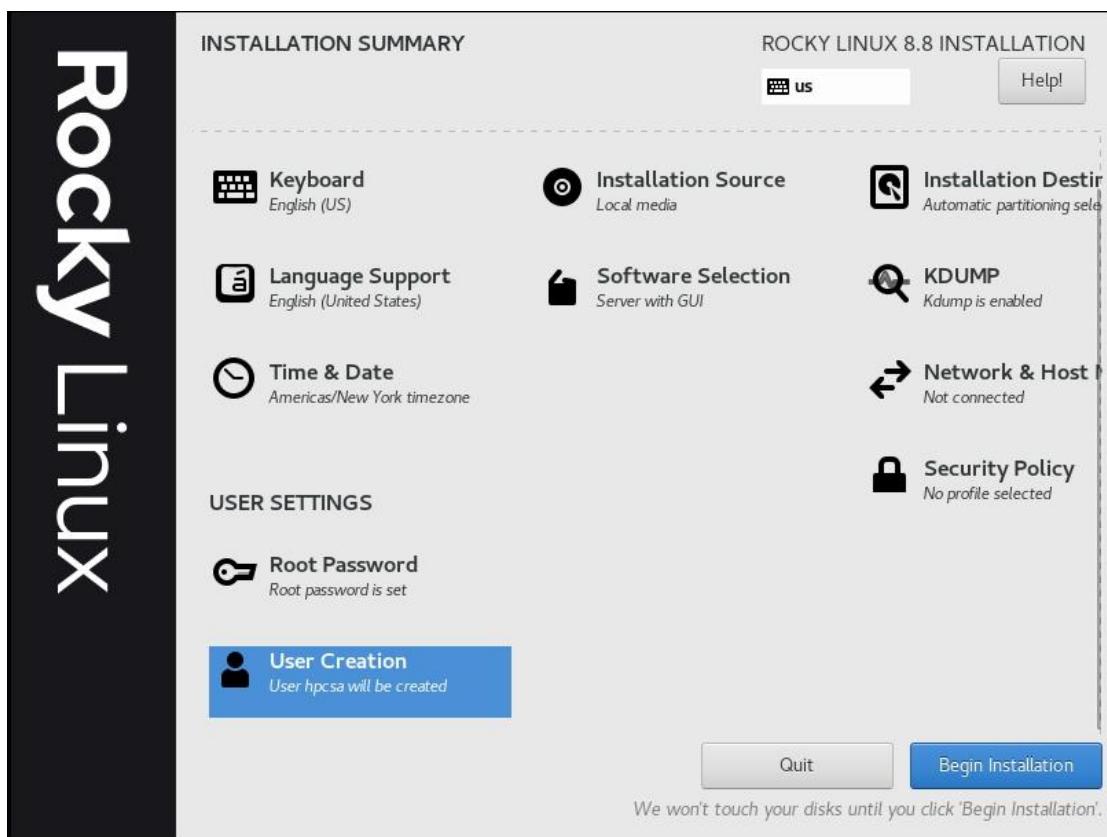
Step 2: After vm is started select the first option & press enter key



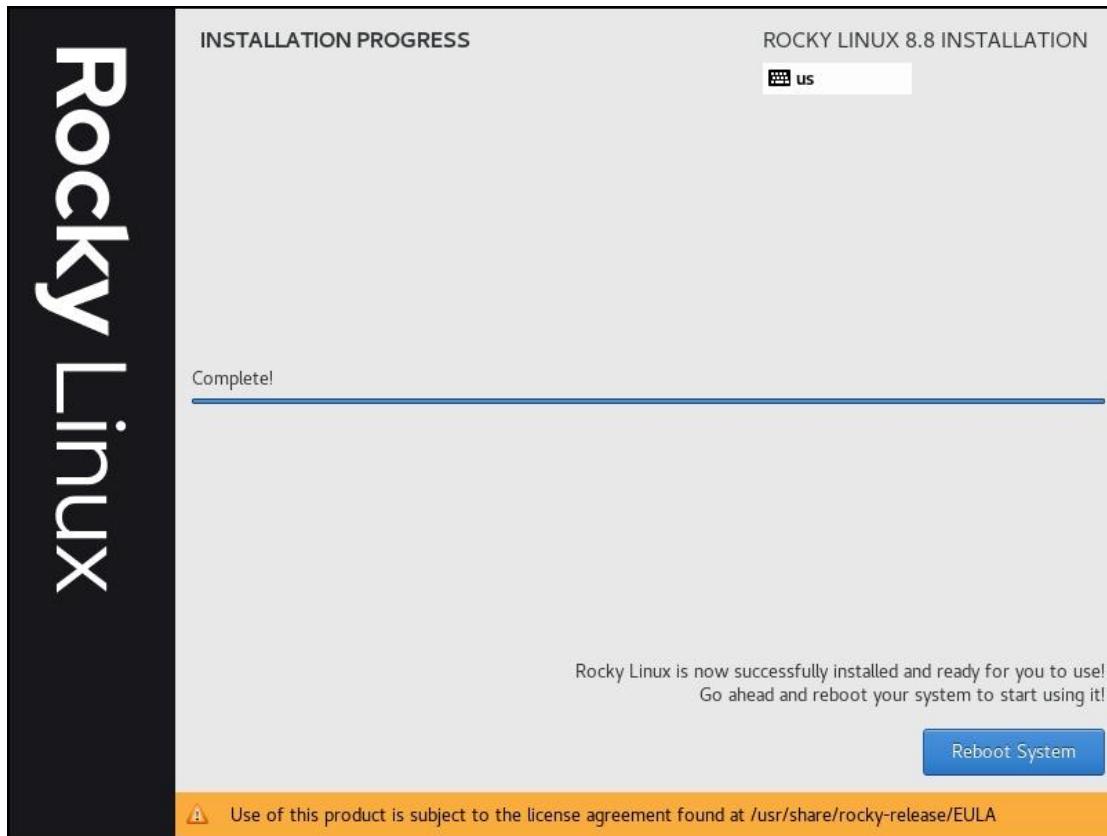
Step 3: Next step is to select language & press Continue button.



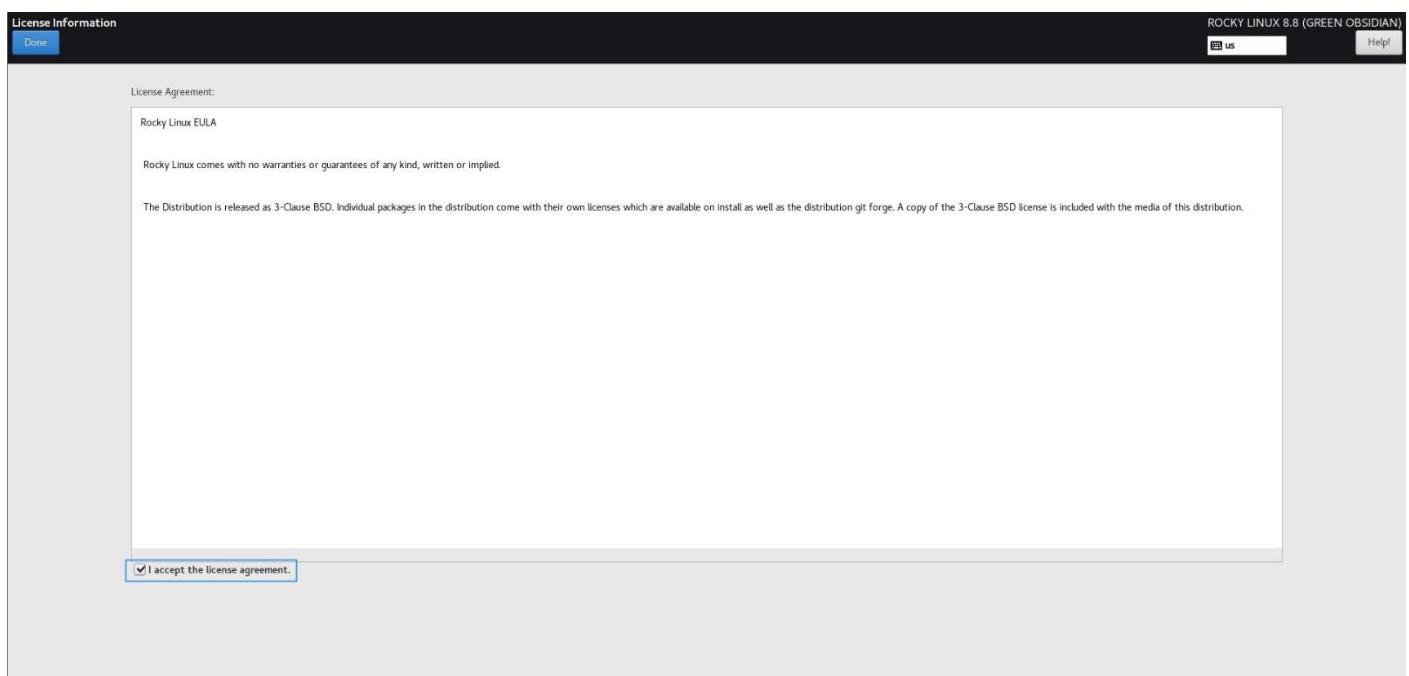
Step 4: setup the root password, Installation destination & click Begin Installation



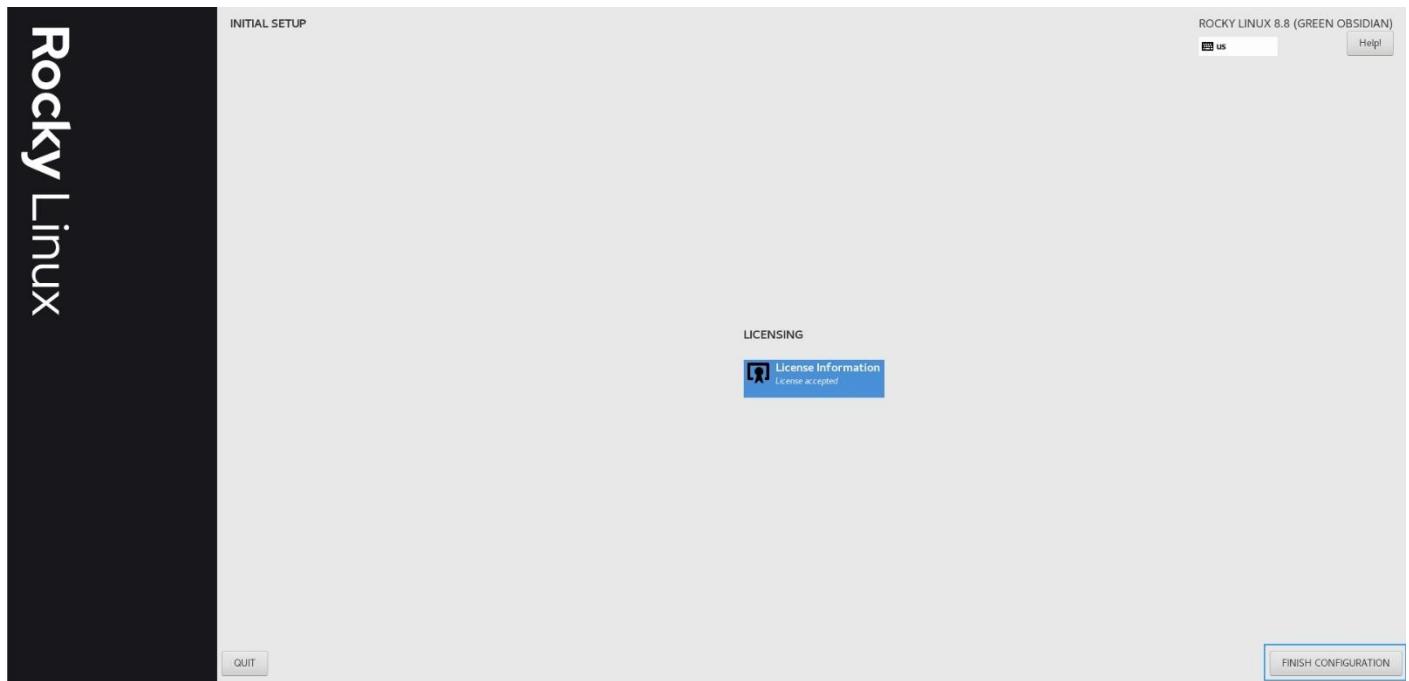
## Step 5: After system has been installed click Reboot System



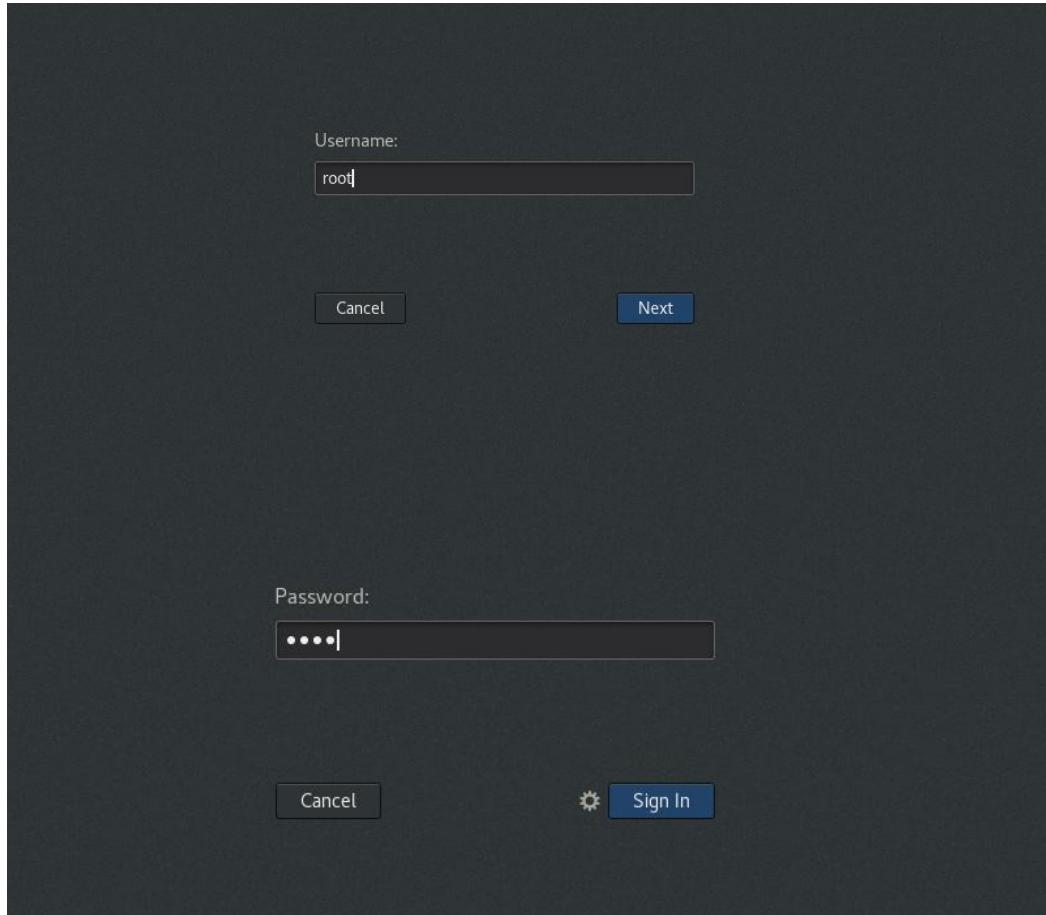
## Step 6: Next step is to accept the EULA agreement



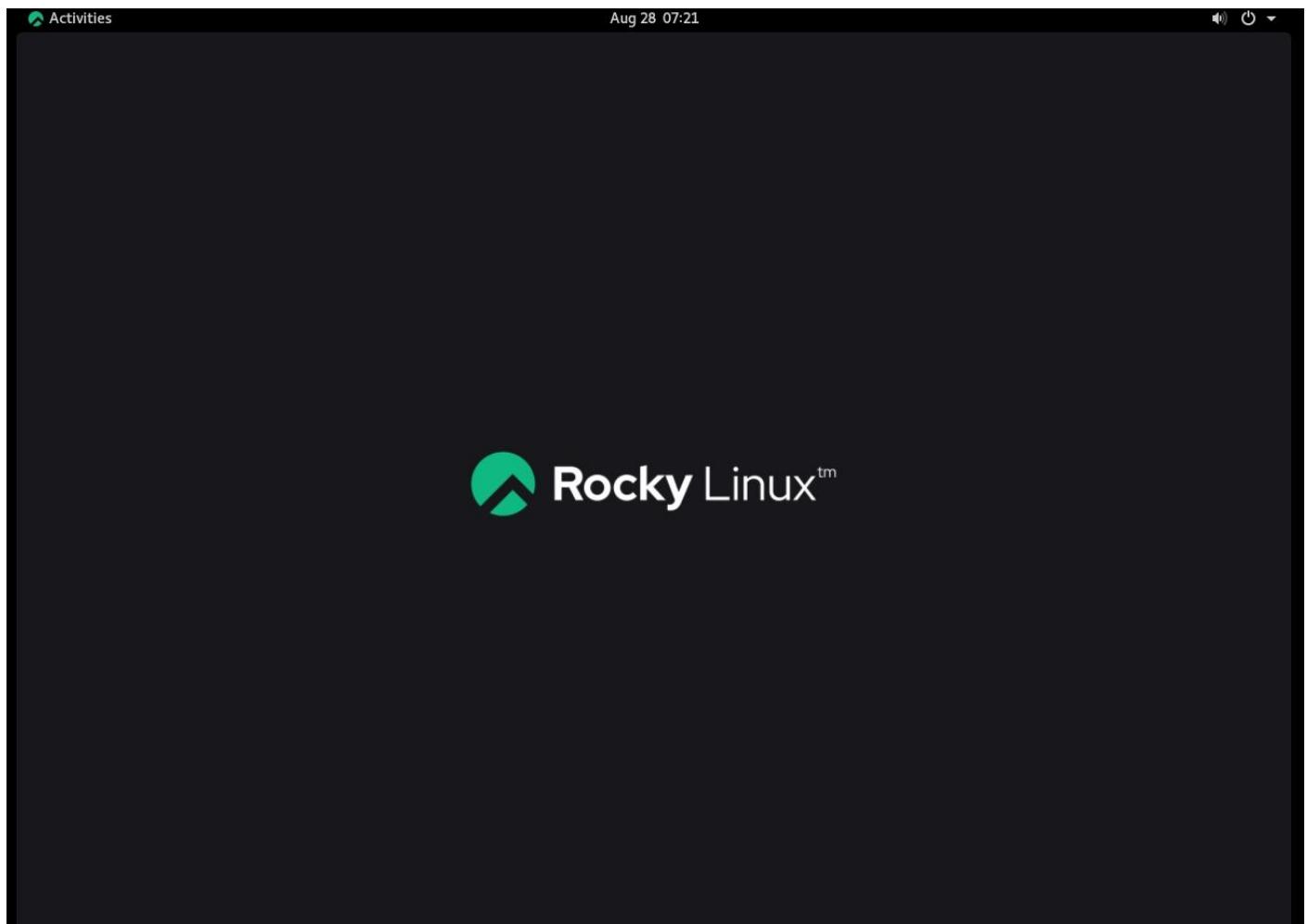
Step 7: Next step is to accept the agreement & click finish Configuration



Step 8: in Next step, Enter the username & password on the login screen

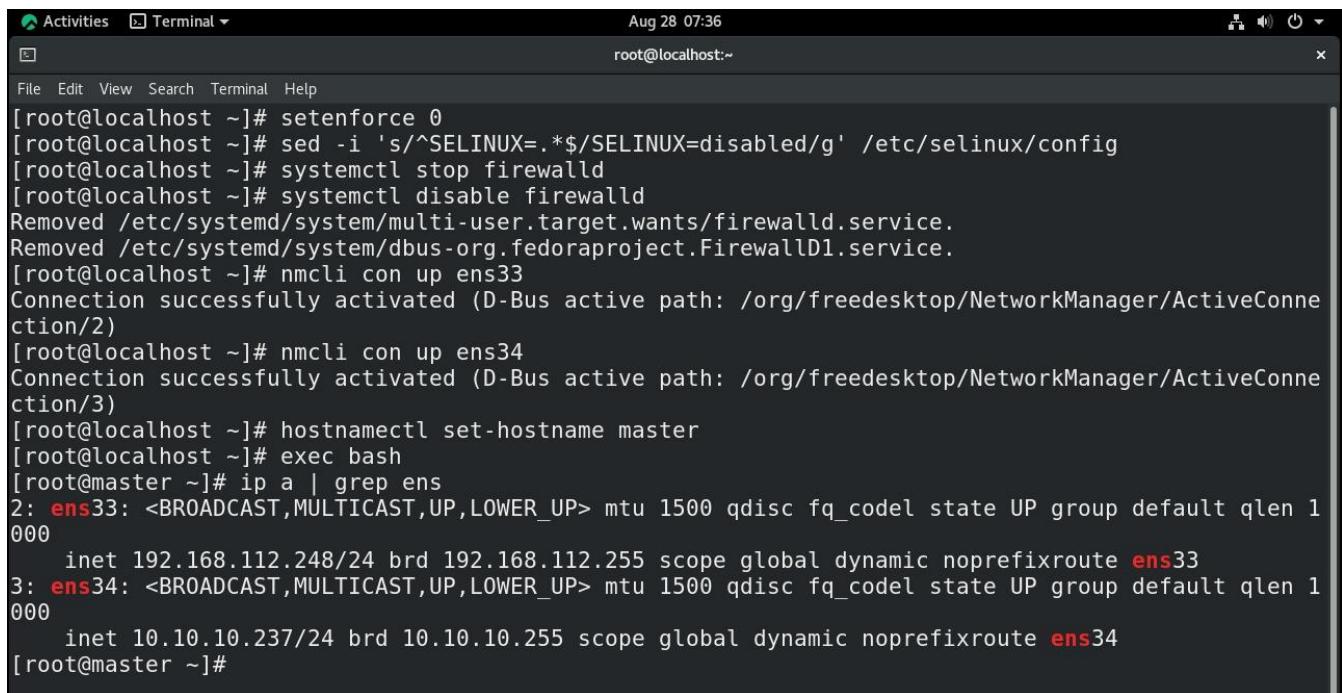


## Step 9: Finally, we land the Rock Linux desktop Environment



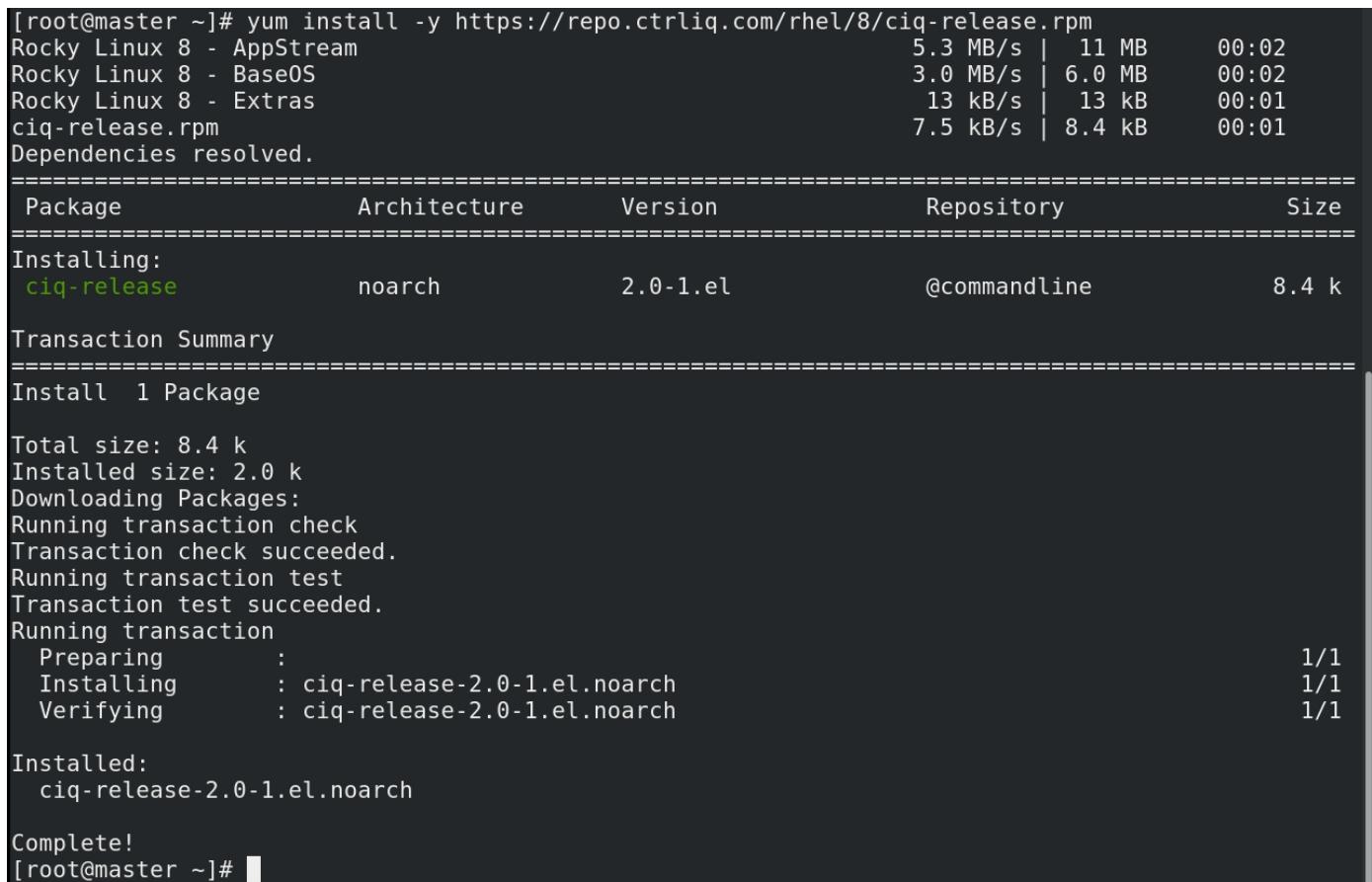
## Warewulf Installation

Step 1: First, we need to disable the SELINUX, firewall and change hostname



```
[root@localhost ~]# setenforce 0
[root@localhost ~]# sed -i 's/^SELINUX=.*/$SELINUX=disabled/g' /etc/selinux/config
[root@localhost ~]# systemctl stop firewalld
[root@localhost ~]# systemctl disable firewalld
Removed /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
[root@localhost ~]# nmcli con up ens33
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection(2))
[root@localhost ~]# nmcli con up ens34
Connection successfully activated (D-Bus active path: /org/freedesktop/NetworkManager/ActiveConnection(3))
[root@localhost ~]# hostnamectl set-hostname master
[root@localhost ~]# exec bash
[root@master ~]# ip a | grep ens
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 192.168.112.248/24 brd 192.168.112.255 scope global dynamic noprefixroute ens33
3: ens34: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    inet 10.10.10.237/24 brd 10.10.10.255 scope global dynamic noprefixroute ens34
[root@master ~]#
```

Step 2: Next step is to add warewulf repository



```
[root@master ~]# yum install -y https://repo.ctrlq.com/rhel/8/ciq-release.rpm
Rocky Linux 8 - AppStream
Rocky Linux 8 - BaseOS
Rocky Linux 8 - Extras
ciq-release.rpm
Dependencies resolved.
=====
Package           Architecture      Version       Repository      Size
=====
Installing:
  ciq-release     noarch          2.0-1.el      @commandline   8.4 k
Transaction Summary
=====
Install 1 Package

Total size: 8.4 k
Installed size: 2.0 k
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  Installing     : ciq-release-2.0-1.el.noarch
  Verifying      : ciq-release-2.0-1.el.noarch
Installed:
  ciq-release-2.0-1.el.noarch
Complete!
[root@master ~]#
```

## Step 3: Next step is to install warewulf

```
[root@master ~]# yum install -y warewulf
CIQ Stable
Dependencies resolved.
=====
Package           Architecture Version      Repository    Size
=====
Installing:
warewulf          x86_64      4.4.0-1.git_afcdb21.el8 ciq        21 M
Installing dependencies:
bind-export-libs   x86_64      32:9.11.36-8.el8_8.1 baseos      1.1 M
dhcp-common        noarch     12:4.3.6-49.el8   baseos      206 k
dhcp-libs          x86_64      12:4.3.6-49.el8   baseos      147 k
dhcp-server        x86_64      12:4.3.6-49.el8   baseos      529 k
tftp-server        x86_64      5.2-26.el8       appstream   49 k
Transaction Summary
=====
Install 6 Packages

Total download size: 23 M
Installed size: 71 M
Downloading Packages:
(1/6): tftp-server-5.2-26.el8.x86_64.rpm           893 kB/s | 49 kB  00:00
(2/6): dhcp-common-4.3.6-49.el8.noarch.rpm         720 kB/s | 206 kB 00:00
(3/6): bind-export-libs-9.11.36-8.el8_8.1.x86_64.rpm 3.9 MB/s | 1.1 MB 00:00
(4/6): dhcp-libs-4.3.6-49.el8.x86_64.rpm          584 kB/s | 147 kB 00:00
(5/6): dhcp-server-4.3.6-49.el8.x86_64.rpm         5.1 MB/s | 529 kB 00:00
```

## Step 4: Next step to edit config file

```
[root@master ~]# vim /etc/warewulf/warewulf.conf
WW INTERNAL: 43
ipaddr: 10.10.10.237
netmask: 255.255.255.0
network: 10.10.10.0
warewulf:
  port: 9873
  secure: false
  update interval: 60
  autobuild overlays: true
  host overlay: true
  syslog: false
dhcp:
  enabled: true
  range start: 10.10.10.240
  range end: 10.10.10.245
  systemd name: dhcpcd
tftp:
  enabled: true
  systemd name: tftp
nfs:
  enabled: true
  export paths:
    - path: /home
      export options: rw,sync
      mount options: defaults
      mount: true
    - path: /opt
      export options: ro,sync,no_root_squash
      mount options: defaults
      mount: false
  systemd name: nfs-server
~
```

## Step 5: Next step is to configure the warewulf

```
[root@master ~]# wwctl configure --all
Building overlay for master: host
Enabling and restarting the DHCP services
Created symlink /etc/systemd/system/multi-user.target.wants/dhcpd.service → /usr/lib/systemd/system/dhcpd.service.
Building overlay for master: host
Enabling and restarting the NFS services
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service → /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 /etc/systemd/system/sockets.target.wants/tftp.socket → /usr/lib/systemd/system/tftp.socket.
[root@master ~]#
```

## Step 6: Next we need to start the services required for provisioning

```
[root@master ~]# systemctl start dhcpd tftp nfs-server warewulfd
[root@master ~]# systemctl enable dhcpd tftp nfs-server warewulfd
Created symlink /etc/systemd/system/multi-user.target.wants/warewulfd.service → /usr/lib/systemd/system/warewulfd.service.
[root@master ~]# systemctl status dhcpd tftp nfs-server warewulfd
● dhcpd.service - DHCPv4 Server Daemon
   Loaded: loaded (/usr/lib/systemd/system/dhcpd.service; enabled; vendor preset: disabled)
   Active: active (running) since Mon 2023-08-28 07:43:23 EDT; 1min 0s ago
     Docs: man:dhcpd(8)
           man:dhcpd.conf(5)
 Main PID: 38907 (dhcpd)
   Status: "Dispatching packets..."
    Tasks: 1 (limit: 10802)
   Memory: 14.2M
      CGroup: /system.slice/dhcpd.service
              └─38907 /usr/sbin/dhcpd -f -cf /etc/dhcp/dhcpd.conf -user dhcpcd -group dhcpcd --no-pid

Aug 28 07:43:23 master systemd[1]: Started DHCPv4 Server Daemon.
Aug 28 07:43:23 master dhcpcd[38907]: 
Aug 28 07:43:23 master dhcpcd[38907]: No subnet declaration for ens33 (192.168.112.248).
Aug 28 07:43:23 master dhcpcd[38907]: ** Ignoring requests on ens33. If this is not what
Aug 28 07:43:23 master dhcpcd[38907]: you want, please write a subnet declaration
Aug 28 07:43:23 master dhcpcd[38907]: in your dhcpd.conf file for the network segment
Aug 28 07:43:23 master dhcpcd[38907]: to which interface ens33 is attached. **
Aug 28 07:43:23 master dhcpcd[38907]: Sending on   Socket/fallback/fallback-net
Aug 28 07:43:23 master dhcpcd[38907]: Server starting service.
```

```

● tftp.service - Tftp Server
  Loaded: loaded (/usr/lib/systemd/system/tftp.service; indirect; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 07:43:24 EDT; 59s ago
    Docs: man:in.tftpd(8)
 Main PID: 39015 (in.tftpd)
   Tasks: 1 (limit: 10802)
  Memory: 232.0K
    CGroup: /system.slice/tftp.service
            └─39015 /usr/sbin/in.tftpd -s /var/lib/tftpboot

Aug 28 07:43:24 master systemd[1]: Started Tftp 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 Mon 2023-08-28 07:43:23 EDT; 1min 0s ago
 Main PID: 38971 (code=exited, status=0/SUCCESS)
   Tasks: 0 (limit: 10802)
  Memory: 0B
    CGroup: /system.slice/nfs-server.service

Aug 28 07:43:23 master systemd[1]: Starting NFS server and services...
Aug 28 07:43:23 master systemd[1]: Started NFS server and services.

● warewulfd.service - Warewulf cluster management daemon
  Loaded: loaded (/usr/lib/systemd/system/warewulfd.service; enabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 07:44:11 EDT; 12s ago
    Docs: https://warewulf.hpcng.org/
 Main PID: 39081 (wwctl)
   Tasks: 6 (limit: 10802)
  Memory: 6.6M
    CGroup: /system.slice/warewulfd.service

```

Step 7: Next step is to import container from docker hub & set password for it.

```

[root@master ~]# wwctl container import docker://warewulf/rocky node1-container
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 154247c0cba1 done
Copying config 91226788d9 done
Writing manifest to image destination
Storing signatures
2023/08/28 07:47:46  info unpack layer: sha256:5461c86c2e548f89cd5d0c326aaaf624b5af719ff4b3a5362f1c8c1f30fea5a1d
2023/08/28 07:47:48  info unpack layer: sha256:2ef2dfdac739d58e532b819c29487535bf6cbfd6f850a82ec6fe911975c3dc08
2023/08/28 07:47:55  info unpack layer: sha256:e4aa6f71185580712bf6110e08622228c7d84e4bad13ce7aff9d0a132108f88e
2023/08/28 07:47:55  info unpack layer: sha256:b8fd1f017bbd97acbb6c77bb59bf92c8d7c1048dd166a26d9f04c5538c83b7d3
2023/08/28 07:47:56  info unpack layer: sha256:9231c4f09ac0e58207e33b7b46bf2706fdde7fa60315ad2718306acdff540e14
2023/08/28 07:47:56  info unpack layer: sha256:6c1511b659e1fb9b3e3223a9db3fc135dc2acbbf69b4ccb524be63c36511922
2023/08/28 07:47:56  info unpack layer: sha256:154247c0cba17ac52d9a5ffeb27a1291b40a0a8342410ca9f17de24d2335abb7
Updating the container's /etc/resolv.conf
uid/gid not synced, run
wwctl container syncuser --write node1-container
to synchronize uid/gids.
Building container: node1-container
Created image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img.gz

```

```
[root@master ~]# wwctl container shell node1-container
[node1-container] Warewulf> dnf install -y passwd
Failed to set locale, defaulting to C.UTF-8
Rocky Linux 8 - AppStream
Rocky Linux 8 - BaseOS
Rocky Linux 8 - Extras
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
Transaction Summary
=====
Install 2 Packages

Total download size: 527 k
Installed size: 2.3 M
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         4.5 MB/s | 413 kB   00:00
-----
Total                                         762 kB/s | 527 kB   00:00

[node1-container] Warewulf> passwd root
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[node1-container] Warewulf> exit
exit
Rebuilding container...
Created image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img.gz
[root@master ~]# wwctl overlay build
Building overlay for master: host
[root@master ~]# wwctl container syncuser --write node1-container
[root@master ~]# █
```

Step 8: Next step is to create a new node and test if it is booting or not.

**Warning :** RAM given to Node VM must be greater than the Container Size

1. click on create new VM
2. select “I will install the OS later”
3. Then choose OS type Linux
4. Choose No of processors
5. Give the RAM size for the Node
6. Then choose the Network mode(Host Only)
7. Then give the disk size. (I’ve given 0.001 Gb, in order to demonstrate that this process is stateless, doesn’t require HDD)

New Virtual Machine Wizard

### Guest Operating System Installation

A virtual machine is like a physical computer; it needs an operating system. How will you install the guest operating system?

Install from:

- Installer disc:  
DVD RW Drive (G:)
- Installer disc image file (iso):  
C:\Users\rajesh\Downloads\Rocky-8.8-x86\_64-dvd1.iso
- I will install the operating system later.  
The virtual machine will be created with a blank hard disk.

Help < Back Next > Cancel

New Virtual Machine Wizard

### Select a Guest Operating System

Which operating system will be installed on this virtual machine?

Guest operating system

- Microsoft Windows
- Linux
- VMware ESX
- Other

Version

CentOS 7 64-bit

Help < Back Next > Cancel

New Virtual Machine Wizard

### Processor Configuration

Specify the number of processors for this virtual machine.

Processors

Number of processors: 2

Number of cores per processor: 1

Total processor cores: 2

Help < Back Next > Cancel

New Virtual Machine Wizard

### Memory for the Virtual Machine

How much memory would you like to use for this virtual machine?

Specify the amount of memory allocated to this virtual machine. The memory size must be a multiple of 4 MB.

64 GB  
32 GB  
16 GB  
8 GB  
4 GB  
2 GB  
1 GB  
512 MB  
256 MB  
128 MB  
64 MB  
32 MB  
16 MB  
8 MB  
4 MB

Memory for this virtual machine: 8192 MB

Maximum recommended memory: 27.8 GB  
Recommended memory: 1 GB  
Guest OS recommended minimum: 512 MB

Help < Back Next > Cancel

New Virtual Machine Wizard

### Network Type

What type of network do you want to add?

Network connection

- Use bridged networking  
Give the guest operating system direct access to an external Ethernet network. The guest must have its own IP address on the external network.
- Use network address translation (NAT)  
Give the guest operating system access to the host computer's dial-up or external Ethernet network connection using the host's IP address.
- Use host-only networking  
Connect the guest operating system to a private virtual network on the host computer.
- Do not use a network connection

Help < Back Next > Cancel

New Virtual Machine Wizard

### Specify Disk Capacity

How large do you want this disk to be?

Maximum disk size (GB): .001

Recommended size for CentOS 7 64-bit: 20 GB

Allocate all disk space now.  
Allocating the full capacity can enhance performance but requires all of the physical disk space to be available right now. If you do not allocate all the space now, the virtual disk starts small and grows as you add data to it.

Store virtual disk as a single file

Split virtual disk into multiple files  
Splitting the disk makes it easier to move the virtual machine to another computer but may reduce performance with very large disks.

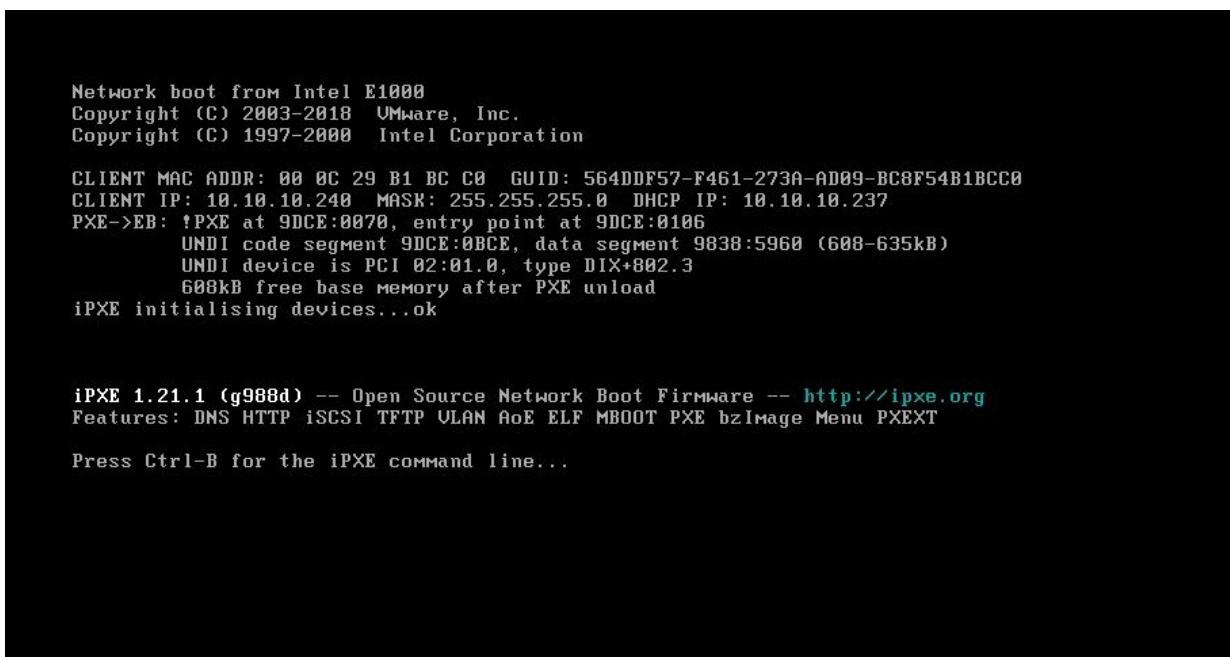
Help < Back Next > Cancel



## Step 9: add this node to warewulf

```
[root@master ~]# wwctl node add node1 --ipaddr 10.10.10.242 --discoverable ens34
Added node: node1
[root@master ~]# wwctl node set --container node1-container node1
? Are you sure you want to modify 1 nodes(s)? [y/N] y
[root@master ~]#
```

## Step 10: Start the node



```
00-0c-29-b1-bc-c0 : 2653 bytes [script]
=====
Warewulf v4 now booting: node1 (00:0c:29:b1:bc:c0)

Container: node1-container
Kernel: node1-container (container default)
KernelArgs: quiet crashkernel=no vga=791 net.naming-scheme=v238

Warewulf Controller: 10.10.10.237
Downloading Kernel Image:
http://10.10.10.237:9873/provision/00%3A0c%3A29%3Ab1%3Abc%3Ac0... ok

Use legacy initrd mode with compressed images
Downloading Container Image:
http://10.10.10.237:9873/provision/00%3A0c%3A29%3Ab1%3Abc%3Ac0... ok
Downloading System Overlay:
http://10.10.10.237:9873/provision/00%3A0c%3A29%3Ab1%3Abc%3Ac0... ok
Downloading Runtime Overlay:
http://10.10.10.237:9873/provision/00%3A0c%3A29%3Ab1%3Abc%3Ac0... ok
Booting in 3s...
boot kernel initrd=container initrd=system initrd=runtime wwid=00:0c:29:b1:bc:c0
quiet crashkernel=no vga=791 net.naming-scheme=v238
-
```

```
Warewulf Node:      node1
Container:        node1-container
Kernelargs:       quiet crashkernel=no vga=791 net.naming-scheme=v238

Network:
  default: eth0
  default: 10.10.10.242/24
  default: 00:0c:29:b1:bc:c0
node1 login: root
Password:
[root@node1 ~]# _
```

---

## *Installation of MUNGE*

---

### Step 1: Installing epel-release package

```
[root@master ~]# dnf install epel-release -y
Last metadata expiration check: 1:53:09 ago on Mon 28 Aug 2023 07:38:23 AM EDT.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
epel-release     noarch        8-18.el8   extras        24 k
=====
Transaction Summary
=====
Install 1 Package
```

### Step 2: enabling powertools & installing Munge packages

```
[root@master ~]# dnf config-manager --set-enabled powertools
[root@master ~]# dnf install munge* -y
Rocky Linux 8 - PowerTools
Extra Packages for Enterprise Linux 8 - x86_64
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
munge            x86_64        0.5.13-2.el8    appstream    121 k
munge-devel      x86_64        0.5.13-2.el8    powertools  27 k
munge-libs       x86_64        0.5.13-2.el8    appstream    29 k
=====
Transaction Summary
=====
Install 3 Packages
Total download size: 177 k
Installed size: 376 k
Downloading Packages:
(1/3): munge-libs-0.5.13-2.el8.x86_64.rpm          62 kB/s | 29 kB   00:00
(2/3): munge-devel-0.5.13-2.el8.x86_64.rpm         53 kB/s | 27 kB   00:00
(3/3): munge-0.5.13-2.el8.x86_64.rpm             156 kB/s | 121 kB  00:00
```

### Step 3: installing random number generator tools, this will help in generating the munge key

```
[root@master ~]# dnf install rng-tools.x86_64 langpacks-en glibc-all-langpacks -y
Last metadata expiration check: 0:05:24 ago on Mon 28 Aug 2023 09:35:31 AM EDT.
Package langpacks-en-1.0-12.el8.noarch is already installed.
Package glibc-all-langpacks-2.28-225.el8.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
rng-tools        x86_64        6.15-3.el8   baseos        78 k
=====
Transaction Summary
=====
Install 1 Package
```

## Step 4: Starting the rngd service

```
[root@master ~]# systemctl start rngd
[root@master ~]# systemctl enable rngd
[root@master ~]# systemctl status rngd
● rngd.service - Hardware RNG Entropy Gatherer Daemon
   Loaded: loaded (/usr/lib/systemd/system/rngd.service; enabled; vendor preset: enabled)
     Active: active (running) since Mon 2023-08-28 09:43:33 EDT; 1s ago
       Main PID: 42839 (rngd)
          Tasks: 2 (limit: 10802)
        Memory: 1.5M
      CGroup: /system.slice/rngd.service
             └─42839 /usr/sbin/rngd -f --fill-watermark=0 -x pkcs11 -x nist -x qrypt -D daemon:daemon

Aug 28 09:43:33 master systemd[1]: Started Hardware RNG Entropy Gatherer Daemon.
Aug 28 09:43:33 master rngd[42839]: Disabling 7: PKCS11 Entropy generator (pkcs11)
Aug 28 09:43:33 master rngd[42839]: Disabling 5: NIST Network Entropy Beacon (nist)
Aug 28 09:43:33 master rngd[42839]: Disabling 9: Qrypt quantum entropy beacon (qrypt)
Aug 28 09:43:33 master rngd[42839]: Initializing available sources
Aug 28 09:43:33 master rngd[42839]: [hwrng ]: Initialization Failed
Aug 28 09:43:33 master rngd[42839]: [drand]: Enabling RDOSEED rng support
Aug 28 09:43:33 master rngd[42839]: [drand]: Initialized
Aug 28 09:43:33 master rngd[42839]: [jitter]: JITTER timeout set to 5 sec
Aug 28 09:43:33 master rngd[42839]: [jitter]: Initializing AES buffer
[root@master ~]#
```

## Step 5: create the munge key

```
[root@master ~]# /usr/sbin/create-munge-key -r
Please type on the keyboard, echo move your mouse,
utilize the disks. This gives the random number generator
a better chance to gain enough entropy.
Generating a pseudo-random key using /dev/random completed.
[root@master ~]#
```

## Step 6: copying this munge key to container

```
[root@master ~]# wwctl container shell node1-container
[node1-container] Warewulf> cd
[node1-container] Warewulf> ll
total 12
-rw----- 1 root root 2403 Dec 19 2022 anaconda-ks.cfg
-rw-r--r-- 1 root root 737 Dec 19 2022 anaconda-post.log
-rw----- 1 root root 2041 Dec 19 2022 original-ks.cfg
[node1-container] Warewulf> mkdir tmp
[node1-container] Warewulf> ll
total 12
-rw----- 1 root root 2403 Dec 19 2022 anaconda-ks.cfg
-rw-r--r-- 1 root root 737 Dec 19 2022 anaconda-post.log
-rw----- 1 root root 2041 Dec 19 2022 original-ks.cfg
drwxr-xr-x 2 root root 6 Aug 28 13:47 tmp
[node1-container] Warewulf> exit
Rebuilding container...
Created Image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed Image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img.gz
[root@master ~]# wwctl container exec --bind /root/tmp:/root/tmp node1-container /bin/bash
[node1-container] Warewulf> cd
[node1-container] Warewulf> ll -R tmp/
tmp:
total 0
drwxr-xr-x 2 root root 23 Aug 28 13:45 munge
tmp/munge:
total 4
-r----- 1 root root 1024 Aug 28 13:45 munge.key
[node1-container] Warewulf>
```

## Step 7: installing the munge on the container

```
[node1:container] Warewulf> dnf install epel-release -y
Last metadata expiration check: 1:58:23 ago on Mon 28 Aug 2023 11:52:15 AM UTC.
Dependencies resolved.
=====
Package           Architecture   Version      Repository  Size
=====
Installing:
  epel-release     noarch       8-18.el8    extras       24 k
Installing dependencies:
  dbus-glib        x86_64       0.110-2.el8  baseos      126 k
  python3-dateutil noarch       1:2.6.1-6.el8 baseos      250 k
  python3-dbus     x86_64       1.2.4-15.el8 baseos      133 k
  python3-dnf-plugins-core noarch   4.0.21-19.el8_8 baseos      260 k
  python3-six      noarch       1.11.0-8.el8  baseos      37 k
  python3-systemd x86_64       234-8.el8    baseos      80 k
Installing weak dependencies:
  dnf-plugins-core noarch       4.0.21-19.el8_8 baseos      74 k
Transaction Summary
=====
Install 8 Packages

Total download size: 985 k
Installed size: 2.6 M
Downloading Packages:
(1/8): python3-dateutil-2.6.1-6.el8.noarch.rpm          2.8 MB/s | 250 kB  00:00
(2/8): dnf-plugins-core-4.0.21-19.el8_8.noarch.rpm      836 kB/s |  74 kB  00:00
(3/8): dbus-glib-0.110-2.el8.x86_64.rpm                1.4 MB/s | 126 kB  00:00
(4/8): python3-six-1.11.0-8.el8.noarch.rpm              1.8 MB/s |  37 kB  00:00
(5/8): python3-dbus-1.2.4-15.el8.x86_64.rpm            4.1 MB/s | 133 kB  00:00
(6/8): python3-systemd-234-8.el8.x86_64.rpm            3.5 MB/s |  80 kB  00:00
(7/8): python3-dnf-plugins-core-4.0.21-19.el8_8.noarch.rpm 4.7 MB/s | 260 kB  00:00
(8/8): epel-release-8-18.el8.noarch.rpm                  117 kB/s |  24 kB  00:00
```

## Step 8: copying the munge key from shared folder to munge path & changing it's ownership. Finally, we have to rebuild the container

```
[node1:container] Warewulf> cp tmp/munge/munge.key /etc/munge/
[node1:container] Warewulf> chown munge:munge /etc/munge/munge.key
[node1:container] Warewulf> exit
exit
WARN  : /etc/passwd has been modified, maybe you want to run syncuser
WARN  : /etc/group has been modified, maybe you want to run syncuser
Rebuilding container...
Created image for VNFs container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed image for VNFs container node1-container: /var/lib/warewulf/container/node1-container.img.gz
[root@master ~]# wctl container exec --bind /root/tmp:/root/tmp node1-container /bin/bash
[node1:container] Warewulf> echo "systemctl start munge" >> /root/.bashrc
[node1:container] Warewulf> exit
exit
Rebuilding container...
Created image for VNFs container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed image for VNFs container node1-container: /var/lib/warewulf/container/node1-container.img.gz
[root@master ~]# wctl overlay build
Building overlay for master: host
Building system overlays for node1: [winit]
Created Image for overlay node1/[winit]: /var/lib/warewulf/overlays/node1/winit.img
Compressed image for overlay node1/[winit]: /var/lib/warewulf/overlays/node1/winit.img.gz
Building runtime overlays for node1: [generic]
Created image for overlay node1/[generic]: /var/lib/warewulf/overlays/node1/generic.img
Compressed image for overlay node1/[generic]: /var/lib/warewulf/overlays/node1/generic.img.gz
[root@master ~]# wctl container syncuser --write node1-container
[root@master ~]#
```

## Step 9: checking status of munge service on Compute Node & Master

```
Warewulf Node:      node1
Container:         node1-container
Kernelargs:        quiet crashkernel=no vga=791 net.naming-scheme=v238

Network:
  default: eth0
  default: 10.10.10.242/24
  default: 00:0c:29:b1:bc:c0
node1 login: root
Password:
[root@node1 ~]# systemctl status munge
● munge.service - MUNGE authentication service
  Loaded: loaded (/usr/lib/systemd/system/munge.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 15:04:34 UTC; 6s ago
    Docs: man:munged(8)
   Process: 831 ExecStart=/usr/sbin/munged (code=exited, status=0/SUCCESS)
 Main PID: 833 (munged)
   Tasks: 4 (limit: 47187)
  Memory: 964.0K
   CGroup: /system.slice/munge.service
           └─833 /usr/sbin/munged

Aug 28 15:04:34 node1 systemd[1]: Starting MUNGE authentication service...
Aug 28 15:04:34 node1 systemd[1]: Started MUNGE authentication service.
[root@node1 ~]#
```

```
[root@master ~]# chown munge:munge /etc/munge/munge.key
[root@master ~]# systemctl start munge
[root@master ~]# systemctl status munge
● munge.service - MUNGE authentication service
  Loaded: loaded (/usr/lib/systemd/system/munge.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 11:29:17 EDT; 5s ago
    Docs: man:munged(8)
   Process: 45660 ExecStart=/usr/sbin/munged (code=exited, status=0/SUCCESS)
 Main PID: 45662 (munged)
   Tasks: 4 (limit: 10802)
  Memory: 1.1M
   CGroup: /system.slice/munge.service
           └─45662 /usr/sbin/munged

Aug 28 11:29:17 master systemd[1]: Starting MUNGE authentication service...
Aug 28 11:29:17 master systemd[1]: Started MUNGE authentication service.
[root@master ~]#
```

---

## Installation of SLURM

---

Step 1: First, we have to download rpm-build & make packages

```
[root@master ~]# dnf install rpm-build make -y
Last metadata expiration check: 0:18:40 ago on Mon 28 Aug 2023 11:16:31 AM EDT.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
make              x86_64        1:4.2.1-11.el8   baseos       497 k
rpm-build         x86_64        4.14.3-26.el8   appstream   173 k
Installing dependencies:
dwz               x86_64        0.12-10.el8    appstream   108 k
efi-srpm-macros noarch        3-3.el8      appstream   21 k
elfutils          x86_64        0.188-3.el8   baseos      543 k
gc                x86_64        7.6.4-3.el8    appstream   108 k
gdb-headless     x86_64        8.2-19.el8    appstream   3.7 M
ghc-srpm-macros noarch        1.4.2-7.el8    appstream   8.3 k
go-srpm-macros  noarch        2-17.el8     appstream   12 k
guile             x86_64        5:2.0.14-7.el8 appstream   3.5 M
libatomic_ops     x86_64        7.6.2-3.el8    appstream   37 k
libbabeitrace    x86_64        1.5.4-4.el8   baseos      199 k
libipt             x86_64        1.6.1-8.el8    appstream   49 k
ocaml-srpm-macros noarch        5-4.el8      appstream   8.3 k
openblas-srpm-macros noarch        2-2.el8     appstream   6.9 k
patch              x86_64        2.7.6-11.el8  baseos      137 k
perl-srpm-macros noarch        1-25.el8     appstream   9.7 k
python-rpm-macros noarch        3-45.el8    appstream   15 k
python-srpm-macros noarch        3-45.el8    appstream   15 k
python3-rpm-macros noarch        3-45.el8    appstream   14 k
qt5-srpm-macros  noarch        5.15.3-1.el8  appstream   9.5 k
redhat-rpm-config noarch        131-1.el8   appstream   90 k
rust-srpm-macros noarch        5-2.el8     appstream   8.2 k
zstd               x86_64        1.4.4-1.el8   appstream   392 k
```

Step 2 : Next step is to download Slurmd packages

```
[root@master ~]# cd Downloads/
[root@master Downloads]# wget https://download.schedmd.com/slurm/slurm-20.11.9.tar.bz2
--2023-08-28 11:36:26-- https://download.schedmd.com/slurm/slurm-20.11.9.tar.bz2
Resolving download.schedmd.com (download.schedmd.com)... 71.19.154.210
Connecting to download.schedmd.com (download.schedmd.com)|71.19.154.210|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6629247 (6.3M) [application/x-bzip2]
Saving to: 'slurm-20.11.9.tar.bz2'

slurm-20.11.9.tar.bz2          100%[=====] 6.32M  1.44MB/s  in 4.4s
2023-08-28 11:36:31 (1.44 MB/s) - 'slurm-20.11.9.tar.bz2' saved [6629247/6629247]

[root@master Downloads]# ll
total 6476
-rw-r--r--. 1 root root 6629247 May  4  2022 slurm-20.11.9.tar.bz2
```

Step 3 : Next step is to download dependencies for Slurmd packages

```
[root@master Downloads]# dnf install pam-devel python3 readline-devel perl-ExtUtils-MakeMaker gcc mysql-devel -y
Last metadata expiration check: 0:21:08 ago on Mon 28 Aug 2023 11:16:31 AM EDT.
Package python3-3.6.8-38.module+el8.5.0+671+195e4563.x86_64 is already installed.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
gcc              x86_64        8.5.0-18.el8   appstream   23 M
mysql-devel      x86_64        8.0.32-1.module+el8.8.0+1283+4b88a3a8.0.1 appstream   160 k
pam-devel        x86_64        1.3.1-25.el8  baseos      210 k
perl-ExtUtils-MakeMaker noarch        1:7.34-1.el8 appstream   299 k
readline-devel   x86_64        7.0-10.el8   baseos      203 k
Installing dependencies:
annobin          x86_64        10.94-1.el8   appstream   964 k
cpp              x86_64        8.5.0-18.el8   appstream   18 M
gcc-plugin-annobin x86_64        8.5.0-18.el8   appstream   34 k
glibc-devel      x86_64        2.28-225.el8  baseos      82 k
glibc-headers    x86_64        2.28-225.el8  baseos      487 k
isl               x86_64        0.16.1-6.el8  appstream   834 k
kernel-headers   x86_64        4.18.0-477.21.1.el8_8 baseos      11 M
keyutils-libs-devel x86_64        1.5.10-9.el8   baseos      47 k
krb5-devel       x86_64        1.18.2-22.el8_7 baseos      559 k
libcom_err-devel x86_64        1.45.6-5.el8   baseos      38 k
libkadm5          x86_64        1.18.2-22.el8_7 baseos      186 k
libselinux-devel x86_64        2.9-8.el8    baseos      199 k
libsepolicy-devel x86_64        2.9-3.el8    baseos      86 k
libvirt-devel     x86_64        0.3.2-2.el8   baseos      17 k
libxcrypt-devel   x86_64        4.1.1-6.el8   baseos      24 k
mariadb-connector-c-config noarch        3.1.11-2.el8_3 appstream   14 k
mysql-common      x86_64        8.0.32-1.module+el8.8.0+1283+4b88a3a8.0.1 appstream   137 k
mysql-lib          x86_64        8.0.32-1.module+el8.8.0+1283+4b88a3a8.0.1 appstream   1.5 M
ncurses-c++-libs  x86_64        6.1-9.20180224.el8 baseos      57 k
ncurses-devel     x86_64        6.1-9.20180224.el8 baseos      527 k
openssl-devel     x86_64        1:1.1.1k-9.el8_7 baseos      2.3 M
pcre2-devel       x86_64        10.32-3.el8_6  baseos      604 k
```

## Step 4 : Next step is to build rpm from tar file

```
[root@master Downloads]# rpmbuild -ta slurm-20.11.9.tar.bz2
warning: Macro expanded in comment on line 22: %_prefix path           install path for commands, libraries, etc.
warning: Macro expanded in comment on line 170: %define _unpacked_files_terminate_build      0
Executing(%prep): /bin/sh -e /var/tmp/rpm-tmp.o1Xlkf
+ umask 022
+ cd /root/rpmbuild/BUILD
+ cd /root/rpmbuild/BUILD
+ rm -rf slurm-20.11.9
+ /usr/bin/bzip2 -dc /root/Downloads/slurm-20.11.9.tar.bz2
+ /usr/bin/tar -xvvoF -
drwxr-xr-x 1000/1000          0 2022-05-04 15:32 slurm-20.11.9/
-rw-r--r-- 1000/1000          8543 2022-05-04 15:32 slurm-20.11.9/LICENSE.OpenSSL
drwxr-xr-x 1000/1000          0 2022-05-04 15:32 slurm-20.11.9/auxdir/
-rw-r--r-- 1000/1000        306678 2022-05-04 15:32 slurm-20.11.9/auxdir/libtool.m4
-rw-r--r-- 1000/1000         5868 2022-05-04 15:32 slurm-20.11.9/auxdir/ax_gcc_builtin.m4
-rwxr-xr-x 1000/1000        15368 2022-05-04 15:32 slurm-20.11.9/auxdir/install-sh
-rw-r--r-- 1000/1000        32716 2022-05-04 15:32 slurm-20.11.9/auxdir/lmain.sh
-rw-r--r-- 1000/1000        2638 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_freeipmi.m4
-rw-r--r-- 1000/1000        1783 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_yaml.m4
-rw-r--r-- 1000/1000        2789 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_databases.m4
-rw-r--r-- 1000/1000        2018 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_http_parser.m4
-rwxr-xr-x 1000/1000        36136 2022-05-04 15:32 slurm-20.11.9/auxdir/config.sub
-rwxr-xr-x 1000/1000        23568 2022-05-04 15:32 slurm-20.11.9/auxdir/depcomp
-rw-r--r-- 1000/1000         902 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_c99.m4
-rw-r--r-- 1000/1000        16672 2022-05-04 15:32 slurm-20.11.9/auxdir/Makefile.in
-rw-r--r-- 1000/1000        1538 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_systemd.m4
-rw-r--r-- 1000/1000        14514 2022-05-04 15:32 slurm-20.11.9/auxdir/ltoptions.m4
-rw-r--r-- 1000/1000        8569 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_cray.m4
-rw-r--r-- 1000/1000        1449 2022-05-04 15:32 slurm-20.11.9/auxdir/x_ac_nvml.m4
```

## Step 5 : Next step is to create Slurm User to operate Slurm services

```
[root@master Downloads]# export SLURMUSER=900;
[root@master Downloads]# # add group
[root@master Downloads]# groupadd -g $SLURMUSER slurm;
[root@master Downloads]# # create user
[root@master Downloads]# useradd -m -c "SLURM workload manager" -d /var/lib/slurm -u $SLURMUSER -g slurm -s /bin/bash slurm;
[root@master Downloads]# cat /etc/passwd | grep slurm
slurm:x:900:900:SLURM workload manager:/var/lib/slurm:/bin/bash
[root@master Downloads]#
```

## Step 6 : Next step is to check rpmbuild packages after rpmbuild packages has been build

```
[root@master Downloads]# cd /root/rpmbuild/RPMS/x86_64/
[root@master x86_64]# ll
total 20344
-rw-r--r--. 1 root root 16291836 Aug 28 11:41 slurm-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root   21168 Aug 28 11:41 slurm-contribs-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root  84044 Aug 28 11:41 slurm-devel-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root  12312 Aug 28 11:41 slurm-example-configs-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 163316 Aug 28 11:41 slurm-libpmi-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 12990 Aug 28 11:41 slurm-openlava-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 169832 Aug 28 11:41 slurm-pam-slurm-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root  829072 Aug 28 11:41 slurm-perlapi-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 1432156 Aug 28 11:41 slurm-slurmctld-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root  793340 Aug 28 11:41 slurm-slurmd-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root  863268 Aug 28 11:41 slurm-slurmddbd-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 133760 Aug 28 11:41 slurm-torque-20.11.9-1.el8.x86_64.rpm
```

## Step 7 : Next step is to install rpmbuild packages

```
[root@master x86_64]# dnf --nogpgcheck localinstall * -y
Last metadata expiration check: 0:28:36 ago on Mon 28 Aug 2023 11:16:31 AM EDT.
Dependencies resolved.
=====
Package           Architecture   Version      Repository    Size
=====
Installing:
slurm            x86_64        20.11.9-1.el8  @commandline  16 M
slurm-contribs   x86_64        20.11.9-1.el8  @commandline  21 k
slurm-devel      x86_64        20.11.9-1.el8  @commandline  82 K
slurm-example-configs x86_64    20.11.9-1.el8  @commandline  12 K
slurm-libpml     x86_64        20.11.9-1.el8  @commandline  159 K
slurm-openlava    x86_64        20.11.9-1.el8  @commandline  13 K
slurm-pam_slurm  x86_64        20.11.9-1.el8  @commandline  165 K
slurm-perlapi    x86_64        20.11.9-1.el8  @commandline  810 K
slurm-slurmctld  x86_64        20.11.9-1.el8  @commandline  1.4 M
slurm-slurmd     x86_64        20.11.9-1.el8  @commandline  775 K
slurm-slurmdbd   x86_64        20.11.9-1.el8  @commandline  843 K
slurm-torque     x86_64        20.11.9-1.el8  @commandline  131 K

Transaction Summary
=====
Install 12 Packages

Total size: 20 M
Installed size: 86 M
Downloading Packages:
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
```

## Step 8 : Next step is to copy Slurm packages from Master to Container

```
[root@master x86_64]# ll
total 20344
-rw-r--r... 1 root root 16291836 Aug 28 11:41 slurm-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 21168 Aug 28 11:41 slurm-contribs-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 84044 Aug 28 11:41 slurm-devel-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 12312 Aug 28 11:41 slurm-example-configs-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 163316 Aug 28 11:41 slurm-libpml-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 12900 Aug 28 11:41 slurm-openlava-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 169832 Aug 28 11:41 slurm-pam_slurm-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 829872 Aug 28 11:41 slurm-perlapi-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 1432156 Aug 28 11:41 slurm-slurmctld-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 793340 Aug 28 11:41 slurm-slurmd-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 863260 Aug 28 11:41 slurm-slurmdbd-20.11.9-1.el8.x86_64.rpm
-rw-r--r... 1 root root 133760 Aug 28 11:41 slurm-torque-20.11.9-1.el8.x86_64.rpm
[root@master x86_64]# mkdir /root/tmp/slurm/
[root@master x86_64]# cp * /root/tmp/slurm/
[root@master x86_64]# cd /root/tmp/slurm/
[root@master slurm]# rm slurm-slurmctld-20.11.9-1.el8.x86_64.rpm slurm-slurmd-20.11.9-1.el8.x86_64.rpm
rm: remove regular file 'slurm-slurmctld-20.11.9-1.el8.x86_64.rpm'? yes
rm: remove regular file 'slurm-slurmd-20.11.9-1.el8.x86_64.rpm'? yes
[root@master slurm]#
```

## Step 9 : Next step is to make directory , change ownership and give permission to Slurm User

```
[root@master slurm]# mkdir /var/spool/slurm
[root@master slurm]# chown slurm:slurm /var/spool/slurm
[root@master slurm]# chmod 755 /var/spool/slurm
[root@master slurm]# mkdir /var/log/slurm
[root@master slurm]# chown slurm:slurm /var/log/slurm
[root@master slurm]# touch /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
[root@master slurm]# chown slurm:slurm /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
[root@master slurm]# cp /etc/slurm/slurm.conf.example /etc/slurm/slurm.conf
```

## Step 10 : Next step is to install Slurm on container

```
[root@master slurm]# wwctl container exec --bind /root/tmp:/root/tmp/ node1-container /bin/bash
System has not been booted with systemd as init system (PID 1). Can't operate.
Failed to connect to bus: Host is down
[node1-container] Warewulf> rpm -qa | grep slurm
[node1-container] Warewulf> cd /root/tmp/slurm/
[node1-container] Warewulf> ll
total 18100
-rw-r--r--. 1 root root 16291836 Aug 28 15:47 slurm-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 21168 Aug 28 15:47 slurm-contribs-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 84044 Aug 28 15:47 slurm-devel-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 12312 Aug 28 15:47 slurm-example-configs-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 163316 Aug 28 15:47 slurm-libpmi-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 12900 Aug 28 15:47 slurm-openlava-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 169032 Aug 28 15:47 slurm-pam_slurm-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 829072 Aug 28 15:47 slurm-perlapi-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 793340 Aug 28 15:47 slurm-slurmd-20.11.9-1.el8.x86_64.rpm
-rw-r--r--. 1 root root 133760 Aug 28 15:47 slurm-torque-20.11.9-1.el8.x86_64.rpm
[node1-container] Warewulf> dnf --nogpgcheck localinstall * -y
Last metadata expiration check: 5:17:11 ago on Mon 28 Aug 2023 01:51:33 PM UTC.
Dependencies resolved.
=====
 Package           Architecture   Version      Repository    Size
=====
Installing:
 slurm             x86_64        20.11.9-1.el8      @commandline  16 M
 slurm-contribs    x86_64        20.11.9-1.el8      @commandline  21 k
 slurm-devel       x86_64        20.11.9-1.el8      @commandline  82 k
 slurm-example-configs x86_64        20.11.9-1.el8      @commandline  12 k
 slurm-libpmi       x86_64        20.11.9-1.el8      @commandline  159 k
 slurm-openlava     x86_64        20.11.9-1.el8      @commandline  13 k
 slurm-pam_slurm   x86_64        20.11.9-1.el8      @commandline  165 k
 slurm-perlapi      x86_64        20.11.9-1.el8      @commandline  810 k
 slurm-slurmd       x86_64        20.11.9-1.el8      @commandline  775 k
 slurm-torque       x86_64        20.11.9-1.el8      @commandline  131 k
Installing dependencies:
=====

```

## Step 11 : Next step is to edit the config file

Line no 11 : give name to this slurm cluster

Line no 12 : give hostname of slurmctld machine

Line no 92 : comment this line

Line no 93 : add node1 information, which can be obtained using “slurmd -C”

Save & quit config file

```
[root@master ~]# vim /etc/slurm/slurm.conf
[root@master ~]#
 5 #
 6 #
 7 # slurm.conf file generated by configurator.html.
 8 #
 9 # See the slurm.conf man page for more information.
10 #
11 ClusterName=hpcsa
12 ControlMachine=master
13 #ControlAddr=
14 #BackupController=
15 #BackupAddr=
16 #
17 SlurmUser=slurm
18 #SlurmUser=root
19 SlurmctldPort=6817
20 SlurmPort=6818
21 AuthType=auth/munge
22 #JobCredentialPrivateKey=
23 #JobCredentialPublicCertificate=
24 StateSaveLocation=/var/spool/slurm/ctld
25 SlurmSpoolDir=/var/spool/slurm/d
26 SwitchType=switch/none
27 MpDefault=None
28 SlurmctldPidFile=/var/run/slurmctld.pid
29 SlurmPidFile=/var/run/slurmd.pid
30 ProctrackType=proctrack/pgid
31 #PluginDir=
32 #FirstJobId=
33 ReturnToService=0
34 #MaxJobCount=
35 #PlugStackConfig=
36 #PropagatePrioProcess=
37 #PropagateResourceLimits=
38 #PropagateResourceLimitsExcept=
39 #Prolog=
91 # COMPUTE_NODES
92 # NodeName=linux[1-32] Procs=1 State=UNKNOWN
93 NodeName=node1 CPUs=2 Boards=1 SocketsPerBoard=2 CoresPerSocket=1 ThreadsPerCore=1 RealMemory=7928
94
95 PartitionName=debug Nodes=ALL Default=YES MaxTime=INFINITE State=UP
-
:wq
```

```
[root@node1 ~]# slurmd -C
NodeName=node1 CPUs=2 Boards=1 SocketsPerBoard=2 CoresPerSocket=1 ThreadsPerCore=1 RealMemory=7928
UpTime=0-00:01:36
[root@node1 ~]#
```

Note: to get this slurm configuration from node, node has to be in a booted state. Otherwise, this command will show master machine configuration.

**Step 12:** Next step is to change ownership to slurm and restart the service

```
[root@master ~]# chown slurm:slurm /etc/slurm/slurm.conf
[root@master ~]# systemctl restart slurmcld
[root@master ~]# systemctl start slurmcld
[root@master ~]# systemctl status slurmcld
● slurmcld.service - Slurm controller daemon
   Loaded: loaded (/usr/lib/systemd/system/slurmcld.service; enabled; vendor preset: disabled)
     Active: active (running) since Mon 2023-08-28 15:38:24 EDT; 10s ago
       Main PID: 102705 (slurmcld)
          Tasks: 7
         Memory: 7.4M
            CPU: /system.slice/slurmcld.service
                  └─102705 /usr/sbin/slurmcld -D

Aug 28 15:38:24 master systemd[1]: Started Slurm controller daemon.
```

**Step 13 :** Next step is to create spool & log directories for Slurmd on Container

```
[node1-container] Warewulf> mkdir /var/spool/slurm
[node1-container] Warewulf> chown slurm:slurm /var/spool/slurm
[node1-container] Warewulf> chmod 755 /var/spool/slurm
[node1-container] Warewulf> mkdir /var/log/slurm
[node1-container] Warewulf> chown slurm:slurm /var/log/slurm
[node1-container] Warewulf> touch /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
[node1-container] Warewulf> chown slurm:slurm /var/log/slurm_jobacct.log /var/log/slurm_jobcomp.log
[node1-container] Warewulf> ll /etc/slurm/slurm.conf
ls: cannot access '/etc/slurm/slurm.conf': No such file or directory
[node1-container] Warewulf> cp /root/tmp/slurm/slurm.conf /etc/slurm/
[node1-container] Warewulf> chown slurm:slurm /etc/slurm/slurm.conf
[node1-container] Warewulf> exit
exit
Rebuilding container...
Created image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img.gz
[root@master slurm]# wwctl overlay build; wwctl container syncuser --write node1-container
Building overlay for master: host
Building system overlays for node1: [wwinit]
Created image for overlay node1/[wwinit]: /var/lib/warewulf/overlays/node1/wwinit.img
Compressed image for overlay node1/[wwinit]: /var/lib/warewulf/overlays/node1/wwinit.img.gz
Building runtime overlays for node1: [generic]
Created image for overlay node1/[generic]: /var/lib/warewulf/overlays/node1/generic.img
Compressed image for overlay node1/[generic]: /var/lib/warewulf/overlays/node1/generic.img.gz
[root@master slurm]#
```

**Step 14 :** Next step is to reconfigure the warewulf

```
[root@master ~]# wwctl configure --all
Building overlay for master: host
Enabling and restarting the DHCP services
Building overlay for master: 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
```

## Step 15 : Next step is to reboot the Container & check all services

```
28/08/2023 19:42:25 /home/mobaxterm ssh -l root 10.10.10.242
root@10.10.10.242's password:
X11 forwarding request failed on channel 0
[root@node1 ~]# systemctl status munge slurmd
● munge.service - MUNGE authentication service
  Loaded: loaded (/usr/lib/systemd/system/munge.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 19:43:08 UTC; 8s ago
    Docs: man:munged(8)
   Process: 830 ExecStart=/usr/sbin/munged (code=exited, status=0/SUCCESS)
   Main PID: 832 (munged)
     Tasks: 4 (limit: 46900)
    Memory: 1000.0K
       CGrou: /system.slice/munge.service
          └─832 /usr/sbin/munged

Aug 28 19:43:08 node1 systemd[1]: Starting MUNGE authentication service...
Aug 28 19:43:08 node1 systemd[1]: Started MUNGE authentication service.

● slurmd.service - Slurm node daemon
  Loaded: loaded (/usr/lib/systemd/system/slurmd.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 19:43:08 UTC; 8s ago
  Main PID: 838 (slurmd)
    Tasks: 1
   Memory: 1.0M
      CGrou: /system.slice/slurmd.service
         └─838 /usr/sbin/slurmd -D

Aug 28 19:43:08 node1 systemd[1]: Started Slurm node daemon.
[root@node1 ~]#
```

## Step 16 : Next step is to confirm node has been added or not

```
[root@master ~]# sinfo
PARTITION AVAIL  TIMELIMIT  NODES  STATE NODELIST
debug*        up    infinite      1  down node1
[root@master ~]#
```

Note : Node is visible to Slurm controller

---

## *Installation of Ganglia*

---

Step 1 : First , we have to download Ganglia packages

```
[root@master ~]# dnf install ganglia ganglia-gmetad ganglia-gmond ganglia-web -y
Last metadata expiration check: 1:38:35 ago on Mon 28 Aug 2023 02:30:28 PM EDT.
Dependencies resolved.
=====
Package           Architecture Version      Repository  Size
=====
Installing:
ganglia          x86_64      3.7.2-40.el8   epel        113 k
ganglia-gmetad   x86_64      3.7.2-40.el8   epel        67 k
ganglia-gmond    x86_64      3.7.2-40.el8   epel        96 k
ganglia-web      x86_64      3.7.5-40.el8   epel       669 k
Installing dependencies:
apr              x86_64      1.6.3-12.el8   appstream  128 k
apr-util         x86_64      1.6.1-6.el8_1   appstream  104 k
httpd            x86_64      2.4.37-56.module+el8.8.0+1284+07ef499e.6 appstream 1.4 M
httpd-filesystem noarch     2.4.37-56.module+el8.8.0+1284+07ef499e.6 appstream  42 k
httpd-tools       x86_64      2.4.37-56.module+el8.8.0+1284+07ef499e.6 appstream 109 k
libconfuse        x86_64      3.3-7.el8     epel        196 k
libmemcached-libs x86_64      1.0.18-17.el8   appstream  136 k
mod_http2         x86_64      1.15.7-8.module+el8.8.0+1284+07ef499e.3 appstream 154 k
nginx-filesystem noarch     1:1.14.1-9.module+el8.4.0+542+81547229 appstream  23 k
php               x86_64      7.2.24-1.module+el8.4.0+413+c9202dda appstream 1.5 M
php-cli           x86_64      7.2.24-1.module+el8.4.0+413+c9202dda appstream  3.1 M
php-common        x86_64      7.2.24-1.module+el8.4.0+413+c9202dda appstream 660 k
php-gd            x86_64      7.2.24-1.module+el8.4.0+413+c9202dda appstream  83 k
php-xml           x86_64      7.2.24-1.module+el8.4.0+413+c9202dda appstream 187 k
```

Step 2 : Next step is to edit the gmetad file

Line 44 : Change the Cluster name

```
[root@master ~]# vim /etc/ganglia/gmetad.conf
[root@master ~]#
42 # data_source "another source" 1.3.4.7:8655 1.3.4.8
43
44 data_source "hpcsa cluster" localhost
45
46 #
47 # Round-Robin Archives
48 # You can specify custom Round-Robin archives here (defaults are listed below)
49 #
```

Step 4 : Next step is to edit gmond file

Line no 30 : give the cluster name

Line no 31 : give the hostname of the master machine

Line no 50 : give the ip address of the master

Line no 57 : comment this line

Line no 59 : comment this line

```
[root@master ~]# vim /etc/ganglia/gmond.conf
[root@master ~]#
29 cluster {
30   name = "hpcsa cluster"
31   owner = "master"
32   latlong = "unspecified"
33   url = "unspecified"
34 }
50   mcast_join = 10.10.10.237
51   port = 8649
52   ttl = 1
53 }
54
55 /* You can specify as many udp_recv_channels as you like as well. */
56 udp_recv_channel {
57   # mcast_join = 239.2.11.71
58   port = 8649
59   # bind = 239.2.11.71
60   retry_bind = true
61   # Size of the UDP buffer. If you are handling lots of metrics you really
62   # should bump it up to e.g. 10MB or even higher.
63   # buffer = 10485760
64 }
```

## Step 7 : Next step is to start the services

```
[root@master ~]# systemctl start gmetad gmond httpd
[root@master ~]# systemctl enable gmetad gmond httpd
Created symlink /etc/systemd/system/multi-user.target.wants/gmetad.service → /usr/lib/systemd/system/gmetad.service.
Created symlink /etc/systemd/system/multi-user.target.wants/gmond.service → /usr/lib/systemd/system/gmond.service.
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@master ~]# systemctl status gmetad gmond httpd
● gmetad.service - Ganglia Meta Daemon
  Loaded: loaded (/usr/lib/systemd/system/gmetad.service; enabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 16:16:20 EDT; 10s ago
    Main PID: 106568 (gmetad)
      Tasks: 9 (limit: 10802)
     Memory: 13.1M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/gmetad.service
                 └─106568 /usr/sbin/gmetad -d 1

● gmond.service - Ganglia Meta Daemon
  Loaded: loaded (/usr/lib/systemd/system/gmond.service; enabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 16:16:20 EDT; 10s ago
    Main PID: 106580 (gmond)
      Tasks: 2 (limit: 10802)
     Memory: 1.6M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/gmond.service
                 └─106580 /usr/sbin/gmond

Aug 28 16:16:20 master systemd[1]: Starting Ganglia Meta Daemon...
Aug 28 16:16:20 master systemd[1]: Started Ganglia Meta Daemon.

● httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
  Drop-In: /usr/lib/systemd/system/httpd.service.d
            └─php-fpm.conf
  Active: active (running) since Mon 2023-08-28 16:16:20 EDT; 10s ago
    Docs: man:httpd.service(8)
  Main PID: 106579 (httpd)
    Status: "Running, listening on: port 80"
      Tasks: 213 (limit: 10802)
     Memory: 17.9M
        CPU: 0.000 CPU(s) since start
       CGroup: /system.slice/httpd.service
                 ├─106579 /usr/sbin/httpd -DFOREGROUND
```

## Step 8 : Next step is to install ganglia on container

```
[node1-container] Warewulf> dnf install ganglia ganglia-gmond -y
Last metadata expiration check: 6:28:28 ago on Mon 28 Aug 2023 01:51:33 PM UTC.
Dependencies resolved.
=====
Package           Architecture      Version       Repository     Size
=====
Installing:
ganglia          x86_64          3.7.2-40.el8   epel          113 k
ganglia-gmond    x86_64          3.7.2-40.el8   epel          96 k
Installing dependencies:
apr              x86_64          1.6.3-12.el8   appstream    128 k
libconfuse        x86_64          3.3-7.el8     epel          196 k
Transaction Summary
=====
Install 4 Packages

Total download size: 534 k
Installed size: 1.9 M
Downloading Packages:
(1/4): apr-1.6.3-12.el8.x86_64.rpm                                2.1 MB/s | 128 kB   00:00
(2/4): ganglia-gmond-3.7.2-40.el8.x86_64.rpm                      280 kB/s | 96 kB   00:00
(3/4): ganglia-3.7.2-40.el8.x86_64.rpm                           163 kB/s | 113 kB   00:00
(4/4): libconfuse-3.3-7.el8.x86_64.rpm                            224 kB/s | 196 kB   00:00
=====
```

## Step 9 : Next step is to edit gmond file on cluster

Line no 30 : give the cluster name

Line no 31 : give the hostname of the master machine

Line no 50 : give the ip address of the master

Line no 57 : comment this line

Line no 59 : comment this line

```
29 cluster {
30   name = "hpcsa cluster"
31   owner = "master"
32   latlong = "unspecified"
33   url = "unspecified"
34 }
35
50   mcast_join = 10.10.10.237
51   port = 8649
52   ttl = 1
53 }
54
55 /* You can specify as many udp_recv_channels as you like as well. */
56 udp_recv_channel {
57   # mcast_join = 10.10.10.237
58   port = 8649
59   # bind = 239.2.11.71
60   retry_bind = true
61   # Size of the UDP buffer. If you are handling lots of metrics you really
62   # should bump it up to e.g. 10MB or even higher.
63   # buffer = 10485760
64 }
65
```

## Step 10 : Next step is to add the gmond service in bashrc file

```
[node1-container] Warewulf> echo "systemctl start gmond" >> /root/.bashrc
[node1-container] Warewulf> exit
exit
WARN  : /etc/passwd has been modified, maybe you want to run syncuser
WARN  : /etc/group has been modified, maybe you want to run syncuser
Rebuilding container...
Created image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img
Compressed image for VNFS container node1-container: /var/lib/warewulf/container/node1-container.img.gz
[root@master ~]# wectl overlay build; wectl container syncuser --write node1-container
Building overlay for master: host
Building system overlays for node1: [wwinit]
Created image for overlay node1/[wwinit]: /var/lib/warewulf/overlays/node1/wwinit.img
Compressed image for overlay node1/[wwinit]: /var/lib/warewulf/overlays/node1/wwinit.img.gz
Building runtime overlays for node1: [generic]
Created image for overlay node1/[generic]: /var/lib/warewulf/overlays/node1/generic.img
Compressed image for overlay node1/[generic]: /var/lib/warewulf/overlays/node1/generic.img.gz
[root@master ~]#
```

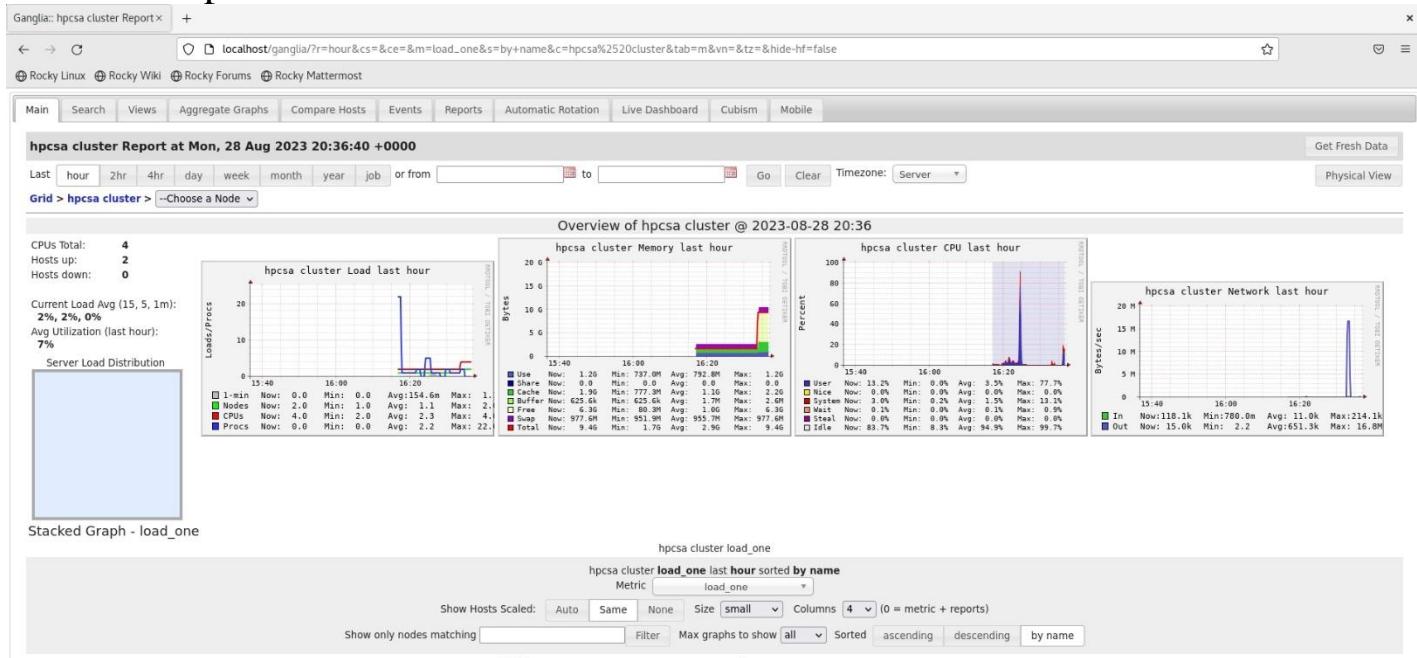
## Step 11 : Next step is to reboot the node & check the status of the gmond service

```
[root@node1 ~]# systemctl status gmond
● gmond.service - Ganglia Meta Daemon
  Loaded: loaded (/usr/lib/systemd/system/gmond.service; disabled; vendor preset: disabled)
  Active: active (running) since Mon 2023-08-28 20:33:34 UTC; 6s ago
    Process: 848 ExecStart=/usr/sbin/gmond (code=exited, status=0/SUCCESS)
   Main PID: 849 (gmond)
     Tasks: 2 (limit: 46841)
    Memory: 1.3M
      CGroup: /system.slice/gmond.service
             └─849 /usr/sbin/gmond

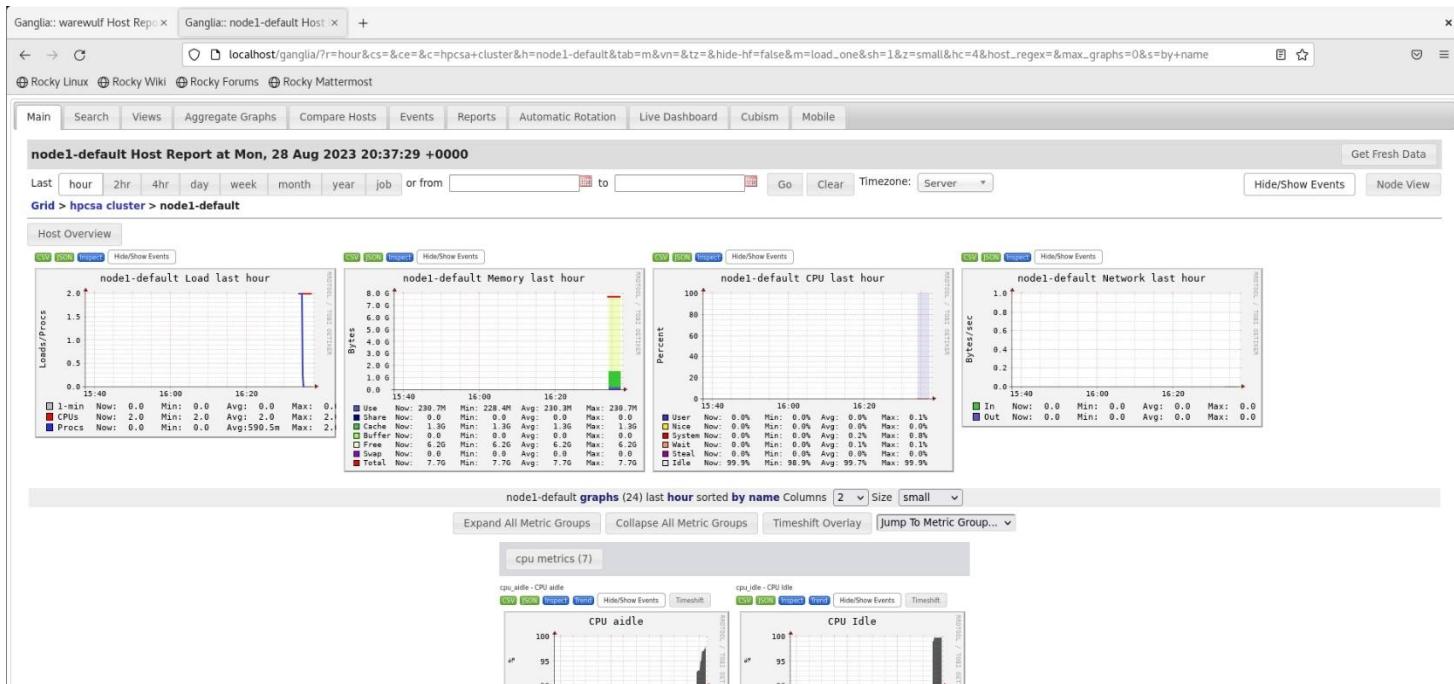
Aug 28 20:33:34 node1 systemd[1]: Starting Ganglia Meta Daemon...
Aug 28 20:33:34 node1 systemd[1]: Started Ganglia Meta Daemon.
[root@node1 ~]#
```

## Step 12 : Next step is to open browser on master & check the ganglia cluster output <https://localhost/ganglia>

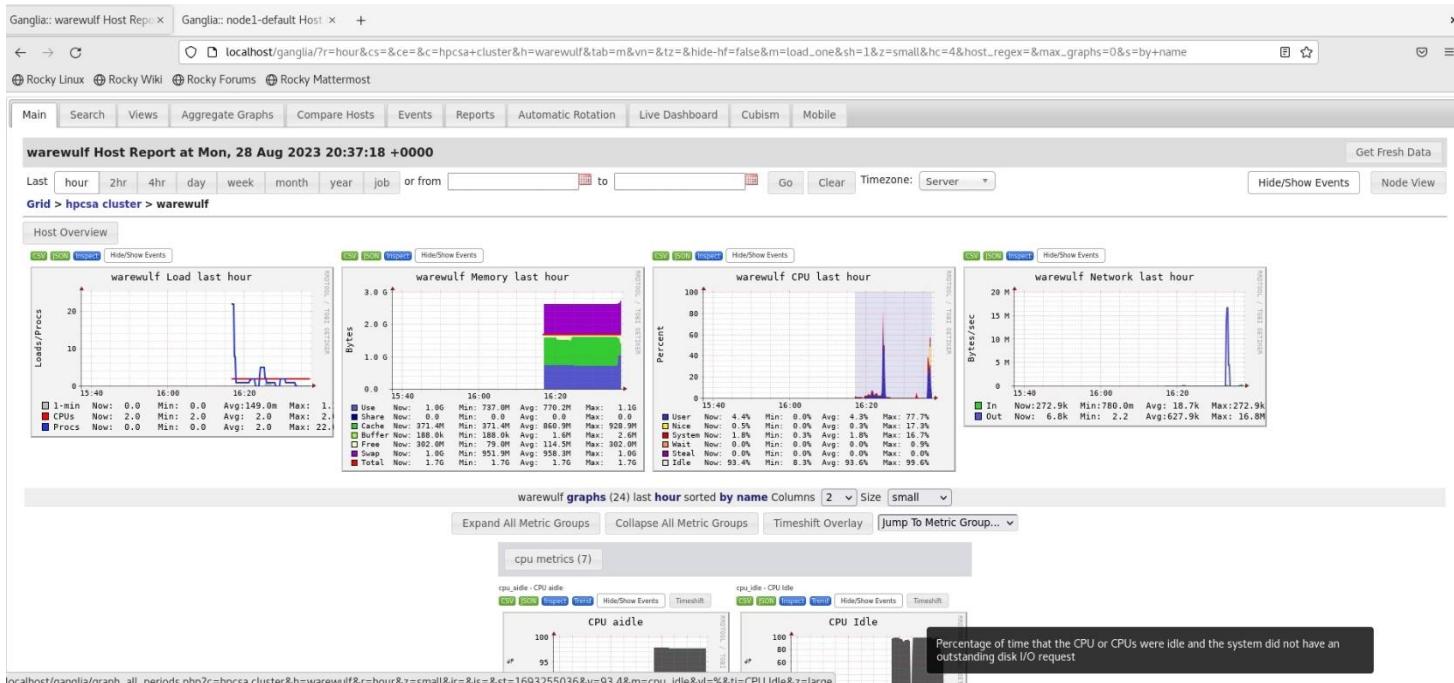
### Cluster Output:



## Node Result:



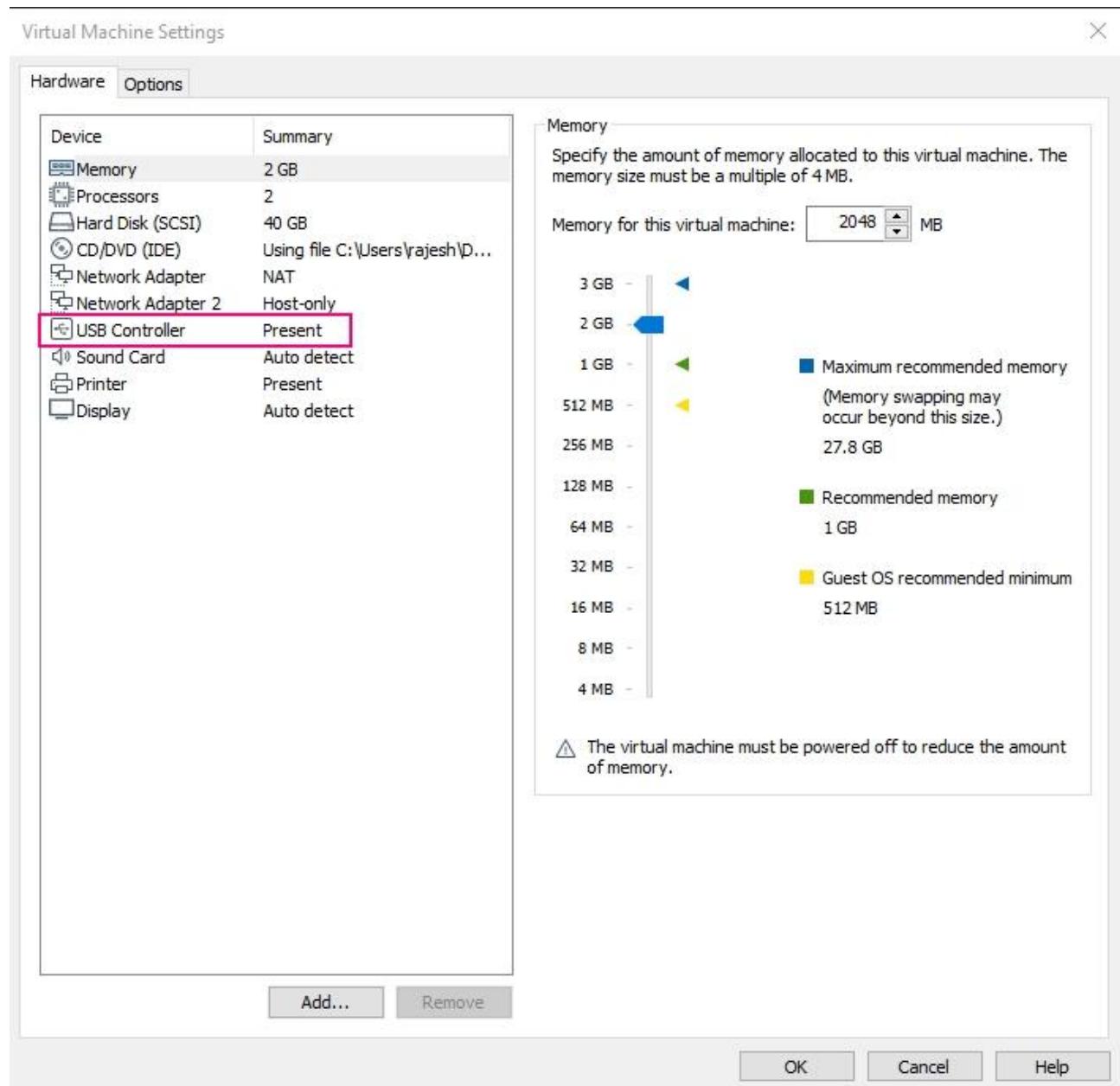
## Master result:



## Troubleshooting

Error 1: hub\_port\_status failed (err = 110)

Solution: Remove Usb Controller from Vm settings



## Error 2: Failed to set Locale, defaulting to C.UTF-8

Solution: install glibc-all-langpacks to resolve this issue

```
Issue:
[rocky-8] Warewulf> dnf config-manager --set-enabled powertools
Failed to set locale, defaulting to C.UTF-8
[rocky-8] Warewulf> dnf install langpacks-en glibc-all-langpacks -y
Failed to set locale, defaulting to C.UTF-8
Rocky Linux 8 - PowerTools
Extra Packages for Enterprise Linux 8 - x86_64
Last metadata expiration check: 0:00:01 ago on Wed Aug 23 09:38:21 2023.
Package langpacks-en-1.0-12.el8.noarch is already installed.
Dependencies resolved.
=====
Package           Architecture      Version       Repository   Size
=====
Installing:
glibc-all-langpacks    x86_64        2.28-225.el8  baseos      26 M
Upgrading:
glibc              x86_64        2.28-225.el8  baseos      2.2 M
glibc-common        x86_64        2.28-225.el8  baseos      1.0 M
glibc-minimal-langpack x86_64        2.28-225.el8  baseos      64 k
Installing weak dependencies:
glibc-gconv-extra   x86_64        2.28-225.el8  baseos      1.5 M
glibc-langpack-en   x86_64        2.28-225.el8  baseos      825 k
Transaction Summary
=====
Install 3 Packages
Upgrade 3 Packages

After Installing above Packages Issue Resolved:
[rocky-8] Warewulf> dnf config-manager --set-enabled powertools
[rocky-8] Warewulf>
```

---

## *References & Bibliography*

---

### 1. Warewulf Documentation

<https://warewulf.org/docs/development/>

### 2. Slurm Documentation

<https://slurm.schedmd.com/documentation.html>

---

## *Project Link*

---

Github: [https://github.com/AtharvNPise/HPCSA\\_Project](https://github.com/AtharvNPise/HPCSA_Project)