

## Warewulf CaseStudy

Name : Ankur jadhav  
PRN : 240340127039  
Name : Shrinidhi Matte  
PRN : 240340127006

Requirements :

Master host machine:

With centos 7 and two network interfaces one in NAT for internet connection and another one Host Only where ware Wulf services would run.

Client node Machine : no os , 8 gb ram , 60 gb hardisk, 1 interface hostonly

Steps to do :

1 : disable selinux and firewalld.

```
[root@osboxes ~]# systemctl status firewalld
● firewalld.service - firewalld - dynamic firewall daemon
  Loaded: loaded (/usr/lib/systemd/system/firewalld.service; disabled; vendor preset: enabled)
  Active: inactive (dead)
    Docs: man:firewalld(1)
[root@osboxes ~]# sestatus
SELinux status:                 disabled
[root@osboxes ~]#
```

2 : add an static ip address to hostonly interface using nmcli, ipconfig or ip

Eg : ip address add <ip address>/subnet dev <hostonly interface name> \* ip is added temporary

```
[root@osboxes ~]# ip r
default via 172.16.0.254 dev ens32 proto dhcp metric 100
172.16.0.0/24 dev ens32 proto kernel scope link src 172.16.0.134 metric 100
192.168.122.0/24 dev virbr0 proto kernel scope link src 192.168.122.1
192.168.253.0/24 dev ens34 proto kernel scope link src 192.168.253.130 metric 101
[root@osboxes ~]#
```

3 : make host entry in /etc/hosts

```
[root@osboxes ~]# echo 192.168.253.130 master >> /etc/hosts
[root@osboxes ~]# nl /etc/hosts
 1 127.0.0.1 localhost.localdomain localhost4.localdomain4
 2 ::1 localhost.localdomain localhost6.localdomain6
 3 192.168.253.130 master
[root@osboxes ~]#
```

4 : enable openhpc repository for local use

yum install [http://build.openhpc.community/OpenHPC:/1.3/CentOS\\_7/x86\\_64/ohpc-release-1.3-1.el7.x86\\_64.rpm](http://build.openhpc.community/OpenHPC:/1.3/CentOS_7/x86_64/ohpc-release-1.3-1.el7.x86_64.rpm)

```
[root@osboxes ~]# yum install http://build.openhpc.community/OpenHPC:/1.3/CentOS_7/x86_64/ohpc-release-1.3-1.el7.x86_64.rpm
[root@osboxes ~]# bash
[root@master_wv ~]# yum install http://build.openhpc.community/OpenHPC:/1.3/CentOS_7/x86_64/ohpc-release-1.3-1.el7.x86_64.rpm
Loaded plugins: fastestmirror, langpacks
ohpc-release-1.3-1.el7.x86_64.rpm | 4.4 kB 00:00:00
Examining /var/tmp/yum-root-XVWIi4/ohpc-release-1.3-1.el7.x86_64.rpm: ohpc-release-1.3-1.el7.x86_64
Marking /var/tmp/yum-root-XVWIi4/ohpc-release-1.3-1.el7.x86_64.rpm to be installed
Resolving Dependencies
--> Running transaction check
--> Package ohpc-release.x86_64 0:1.3-1.el7 will be installed
--> Finished Dependency Resolution
epel/x86_64/metalink | 9.2 kB 00:00:00
epel/x86_64 | 4.3 kB 00:00:00
epel/x86_64/group | 399 kB 00:00:00
epel/x86_64/updateinfo | 1.0 MB 00:00:01
epel/x86_64/primary_db | 8.7 MB 00:00:07
Dependencies Resolved
```

## 5: install base meta packages

yum -y install ohpc-base

yum -y install ohpc-warewulf

```
root@master_wv ~]# yum -y install ohpc-base
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: centos.excellmedia.net
* epel: epel.in.ssimmn.org
* extras: centos.excellmedia.net
* updates: centos.excellmedia.net
OpenHPC
OpenHPC-updates | 1.6 kB 00:00:00
(1/3): OpenHPC/group_gz | 1.2 kB 00:00:00
(2/3): OpenHPC/primary | 1.7 kB 00:00:00
(3/3): OpenHPC-updates/primary | 155 kB 00:00:01
| 341 kB 00:00:01
                                         821/821
OpenHPC
OpenHPC-updates | 1930/1930
Resolving Dependencies
--> Running transaction check
--> Package ohpc-base.x86_64 0:1.3.8-3.1.ohpc.1.3.8 will be installed
--> Processing Dependency: conman-ohpc for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: lmod-ohpc for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: examples-ohpc for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: screen for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: libunwind for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: ntp for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: libstdc++-devel for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: ipmitool for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: cmake-ohpc for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: binutils-devel for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: pdsh-ohpc for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: OpenIPMI for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: emacs-nox for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: losf-ohpc for package: ohpc-base-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Running transaction check
```

```
[root@master_wv ~]# yum -y install ohpc-warewulf
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: centos.excellmedia.net
 * epel: repo.extreme-ix.org
 * extras: centos.excellmedia.net
 * updates: centos.excellmedia.net
Resolving Dependencies
--> Running transaction check
--> Package ohpc-warewulf.x86_64 0:1.3.8-3.1.ohpc.1.3.8 will be installed
--> Processing Dependency: warewulf-provision-initramfs-x86_64-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-common-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-provision-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-vnfs-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-cluster-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-provision-server-ipxe-x86_64-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-ipmi-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Processing Dependency: warewulf-provision-server-ohpc for package: ohpc-warewulf-1.3.8-3.1.ohpc.1.3.8.x86_64
--> Running transaction check
--> Package warewulf-cluster-ohpc.x86_64 0:3.8.1-10.5.ohpc.1.3.6 will be installed
--> Package warewulf-common-ohpc.x86_64 0:3.8.1-14.2.ohpc.1.3.6 will be installed
--> Processing Dependency: mariadb-server for package: warewulf-common-ohpc-3.8.1-14.2.ohpc.1.3.6.x86_64
--> Processing Dependency: perl-Term-ReadLine-Gnu for package: warewulf-common-ohpc-3.8.1-14.2.ohpc.1.3.6.x86_64
--> Processing Dependency: perl-DBD-MySQL for package: warewulf-common-ohpc-3.8.1-14.2.ohpc.1.3.6.x86_64
--> Processing Dependency: perl(DBD::Pg) for package: warewulf-common-ohpc-3.8.1-14.2.ohpc.1.3.6.x86_64
--> Package warewulf-ipmi-ohpc.x86_64 0:3.8.1-12.3.ohpc.1.3.6 will be installed
--> Package warewulf-provision-initramfs-x86_64-ohpc.noarch 0:3.8.1-56.1.ohpc.1.3.9 will be installed
--> Package warewulf-provision-ohpc.x86_64 0:3.8.1-56.1.ohpc.1.3.9 will be installed
--> Package warewulf-provision-server-ipxe-x86_64-ohpc.noarch 0:3.8.1-56.1.ohpc.1.3.9 will be installed
```

## 6 : configure warewulf :

```
perl -pi -e "s/device = eth1/device = <hostonly interface name>/"  
/etc/warewulf/provision.conf
```

```
[root@master_wv ~]# perl -pi -e "s/device = eth1/device = ens34/" /etc/warewulf/provision.conf
[root@master_wv ~]#
[root@master_wv ~]# vim /etc/warewulf/provision.conf
[root@master_wv ~]# cat /etc/warewulf/provision.conf

# What is the default network device that the master will use to
# communicate with the nodes?
network device = ens34

# Which DHCP server implementation should be used?
dhcp server = isc

# What is the TFTP root directory that should be used to store the
# network boot images? By default Warewulf will try and find the
# proper directory. Just add this if it can't locate it.
#tftpboot = /var/lib/tftpboot

# Automatically generate and manage a dynamic_host virtual file
# object in the datastore? This is useful for provisioning this
# out to nodes so they always have a current /etc/hosts file.
generate dynamic_hosts = yes

# Should we manage and overwrite the local hostfile file on this
# system? This will cause all node entries to be added
# automatically to /etc/hosts.
update hostfile = yes

# If no cluster/domain is set on a node, should we add 'localdomain'
# as the default domain
use localdomain = yes

# The default kernel arguments to pass to the nodes boot kernel
default kargs = "net.ifnames=0 biosdevname=0 quiet"
```

## 7: enable tftp services, enable einternal interface and restart and enable services

```
perl -pi -e "s/device = eth1/device = <hostonly interface name>/"  
/etc/warewulf/provision.conf
```

```
perl -pi -e "s/^s+enable\s+= yes/ disable = no/" /etc/xinetd.d/tftp  
  
ifconfig <hostonly interface name> <hostonly ip > netmask <hostonly subnet> up
```

```
[root@master_wv ~]# perl -pi -e "s/^s+enable\s+= yes/ disable = no/" /etc/xinetd.d/tftp  
[root@master_wv ~]# ifconfig ens34 192.168.253.130 netmask 255.255.255.0 up  
[root@master_wv ~]# systemctl restart xinetd  
[root@master_wv ~]# systemctl enable mariadb.service  
[root@master_wv ~]# systemctl restart mariadb  
[root@master_wv ~]# systemctl enable httpd.service  
[root@master_wv ~]# systemctl restart httpd  
[root@master_wv ~]# systemctl enable dhcpd.service  
Created symlink from /etc/systemd/system/multi-user.target.wants/dhcpd.service to /usr/lib/systemd/system/dhcpd.service.  
[root@master_wv ~]#
```

## 8: defining chroot location and build initial chroot image

```
export CHROOT=/opt/ohpc/admin/images/centos7.7
```

```
wwmkchroot centos-7 $CHROOT
```

```
[root@master_wv ~]# export CHROOT=/opt/ohpc/admin/images/centos7.7  
[root@master_wv ~]# wwmkchroot centos-7 $CHROOT  
Configuring SSH for cluster access  
Loaded plugins: fastestmirror, langpacks  
Determining fastest mirrors  
os-base  
(1/2): os-base/x86_64/group_gz | 3.6 kB 00:00:00  
(2/2): os-base/x86_64/primary_db | 153 kB 00:00:00  
| 6.1 MB 00:00:02  
Resolving Dependencies  
--> Running transaction check  
--> Package basesystem.noarch 0:10.0-7.el7.centos will be installed  
--> Package bash.x86_64 0:4.2.46-34.el7 will be installed  
--> Processing Dependency: rtld(GNU_HASH) for package: bash-4.2.46-34.el7.x86_64  
--> Processing Dependency: libdl.so.2(GLIBC_2.2.5)(64bit) for package: bash-4.2.46-34.el7.x86_64  
--> Processing Dependency: libc.so.6(GLIBC_2.15)(64bit) for package: bash-4.2.46-34.el7.x86_64  
--> Processing Dependency: libtinfo.so.5()(64bit) for package: bash-4.2.46-34.el7.x86_64  
--> Processing Dependency: libdl.so.2()(64bit) for package: bash-4.2.46-34.el7.x86_64  
--> Package centos-release.x86_64 0:7-9.2009.0.el7.centos will be installed  
--> Package chkconfig.x86_64 0:1.7.6-1.el7 will be installed  
--> Processing Dependency: libpopt.so.0(LIBPOPT_0)(64bit) for package: chkconfig-1.7.6-1.el7.x86_64  
--> Processing Dependency: libsepol.so.1()(64bit) for package: chkconfig-1.7.6-1.el7.x86_64  
--> Processing Dependency: libselinux.so.1()(64bit) for package: chkconfig-1.7.6-1.el7.x86_64  
--> Processing Dependency: libpopt.so.0()(64bit) for package: chkconfig-1.7.6-1.el7.x86_64  
--> Package coreutils.x86_64 0:8.22-24.el7 will be installed  
--> Processing Dependency: ncurses for package: coreutils-8.22-24.el7.x86_64  
--> Processing Dependency: libcrypto.so.10(libcrypto.so.10)(64bit) for package: coreutils-8.22-24.el7.x86_64
```

## 9: Install compute node base meta-package

```
yum -y --installroot=$CHROOT install ohpc-base-compute
```

```
[root@master_wv ~]# yum -y --installroot=$CHROOT install ohpc-base-compute
Loaded plugins: fastestmirror, langpacks
Determining fastest mirrors
epel/x86_64/metalink
* base: mirrors.nxtgen.com
* epel: mirror.freedit.org
* extras: mirrors.nxtgen.com
* updates: mirrors.nxtgen.com
OpenHPC
openHPC-updates
base
extras
updates
```

## 10 : copy resolv file to chroot image path

```
cp -p /etc/resolv.conf $CHROOT/etc/resolv.conf
```

```
[root@master_wv ~]# cp -p /etc/resolv.conf $CHROOT/etc/resolv.conf
[Complete!]
```

## 12: install kernel package in iso

```
yum -y --installroot=$CHROOT install kernel
```

```
[root@master_wv ~]# yum -y --installroot=$CHROOT install kernel
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: mirror.freedit.org
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
--> Package kernel.x86_64 0:3.10.0-1160.119.1.el7 will be installed
--> Processing Dependency: linux-firmware >= 20190429-72 for package: kernel-3.10.0-1160.119.1.el7.x86_64
--> Processing Dependency: grub >= 8.28-2 for package: kernel-3.10.0-1160.119.1.el7.x86_64
--> Processing Dependency: /usr/sbin/new-kernel-pkg for package: kernel-3.10.0-1160.119.1.el7.x86_64
--> Processing Dependency: /usr/sbin/new-kernel-pkg for package: kernel-3.10.0-1160.119.1.el7.x86_64
--> Running transaction check
--> Package grub.x86_64 0:8.28-26.el7 will be installed
--> Package linux-firmware.noarch 0:20200421-83.git78c0348.el7_9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package           Arch      Version            Repository      Size
=====
Installing:
 kernel          x86_64    3.10.0-1160.119.1.el7      updates       52 M
Installing for dependencies:
```

## 13 : install modules user environment

```
yum -y --installroot=$CHROOT install lmod-ohpc
```

```
[root@master_wv ~]# yum -y --installroot=$CHR0OT install lmod-ohpc
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirrors.nxtgen.com
 * epel: mirror.freedif.org
 * extras: mirrors.nxtgen.com
 * updates: mirrors.nxtgen.com
Resolving Dependencies
--> Running transaction check
--> Package lmod-ohpc.x86_64 0:8.1.18-6.1.ohpc.1.3.9 will be installed
--> Processing Dependency: tcl for package: lmod-ohpc-8.1.18-6.1.ohpc.1.3.9.x86_64
--> Processing Dependency: lua-posix-ohpc for package: lmod-ohpc-8.1.18-6.1.ohpc.1.3.9.x86_64
--> Processing Dependency: lua-fs-ohpc for package: lmod-ohpc-8.1.18-6.1.ohpc.1.3.9.x86_64
--> Processing Dependency: /bin/csh for package: lmod-ohpc-8.1.18-6.1.ohpc.1.3.9.x86_64
--> Processing Dependency: libtcl8.5.so()(64bit) for package: lmod-ohpc-8.1.18-6.1.ohpc.1.3.9.x86_64
--> Running transaction check
--> Package lua-fs-ohpc.x86_64 0:1.6.3-4.1 will be installed
--> Package lua-posix-ohpc.x86_64 0:33.2.1-4.1 will be installed
--> Processing Dependency: lua-bit-ohpc for package: lua-posix-ohpc-33.2.1-4.1.x86_64
--> Package tcl.x86_64 1:8.5.13-8.el7 will be installed
--> Package tcsh.x86_64 0:6.18.01-17.el7_9.1 will be installed
--> Running transaction check
```

## 14 : Initialize warewulf database and ssh\_keys

wwinit database  
wwinit ssh\_keys

```
[root@master_wv ~]# wwinit database
database: Checking to see if RPM 'mysql-server' is installed NO
database: Checking to see if RPM 'mariadb-server' is installed OK
database: Activating and starting unit: mariadb
database: + /bin/systemctl -q enable mariadb.service OK
database: + /bin/systemctl -q restart mariadb.service OK
database: + mysql --defaults-extra-file=/tmp/0.1AfN759xZHS1/my.cnf OK
database: Database version: UNDEF (need to create database)
database: + mysql --defaults-extra-file=/tmp/0.1AfN759xZHS1/my.cnf were OK
database: + mysql --defaults-extra-file=/tmp/0.1AfN759xZHS1/my.cnf were OK
database: + mysql --defaults-extra-file=/tmp/0.1AfN759xZHS1/my.cnf were OK
database: Checking binstore kind SUCCESS
Done.
```

```
[root@master_wv ~]# wwinit ssh_keys
ssh_keys: Checking ssh keys for root OK
ssh_keys: Checking root .ssh config OK
ssh_keys: Checking for default ssh host key for nodes NO
ssh_keys: Creating default node ssh_host_rsa_key:
ssh_keys: + ssh keygen -d -t rsa -f /etc/warewulf/vnfs/ssh/ssh_host_rsa OK
ssh_keys: Checking for default RSA host key for nodes NO
ssh_keys: Creating default node ssh_host_dsa_key:
ssh_keys: + ssh keygen -d -t dsa -f /etc/warewulf/vnfs/ssh/ssh_host_dsa OK
ssh_keys: Checking for default ED25519 host key for nodes NO
ssh_keys: Creating default node ssh_host_ed25519_key: OK
ssh_keys: Checking for default Ed25519 host key for nodes NO
ssh_keys: Creating default node ssh_host_ed25519_key: OK
Done.
```

## 15 : Import files

wwsh file import /etc/passwd

wwsh file import /etc/group

wwsh file import /etc/shadow

**# (Optional) Include drivers from kernel updates; needed if enabling additional kernel modules on computes**

[sms]# export WW\_CONF=/etc/warewulf/bootstrap.conf

[sms]# echo "drivers += updates/kernel/" >> \$WW\_CONF

**# (Optional) Include overlayfs drivers; needed by Singularity**

[sms]# echo "drivers += overlay" >> \$WW\_CONF

**# Build bootstrap image**

[sms]# wwbootstrap `uname -r`

## Ansible virtual node file system:

```
wwvnfs --chroot $CHROOT
```

```
[root@master_wv ~]# wwsh file import /etc/passwd
[root@master_wv ~]# wwsh file import /etc/group
[root@master_wv ~]# wwsh file import /etc/shadow
[root@master_wv ~]# export WW_CONF=/etc/warewulf/bootstrap.conf
[root@master_wv ~]# echo "drivers += updates/kernel/" >> $WW_CONF
[root@master_wv ~]# echo "drivers += overlay" >> $WW_CONF
[root@master_wv ~]# wwbootstrap `uname -r`
Number of drivers included in bootstrap: 543
Number of firmware images included in bootstrap: 102
Building and compressing bootstrap
Integrating the Warewulf bootstrap: 3.10.0-1160.el7.x86_64
Including capability: provision-adhoc
Including capability: provision-files
Including capability: provision-selinux
Including capability: provision-vnfs
Including capability: setup-filesystems
Including capability: setup-ipmi
Including capability: transport-http
Compressing the initramfs
Locating the kernel object
Bootstrap image '3.10.0-1160.el7.x86_64' is ready
Done.
[root@master_wv ~]# wwvnfs --chroot $CHROOT
Using 'centos7.7' as the VNFS name
Creating VNFS image from centos7.7
Compiling hybridization link tree : 0.09 s
Building file list : 0.23 s
Compiling and compressing VNFS : █
```

16:

### # Set provisioning interface as the default networking device

```
[sms]# echo "GATEWAYDEV=${eth_provision}" > /tmp/network.$$
[sms]# wwsh -y file import /tmp/network.$$ --name network
[sms]# wwsh -y file set network --path /etc/sysconfig/network --mode=0644 --uid=0
```

### # Add nodes to Warewulf data store

```
wwsh -y node new <node name> --ipaddr=<ip for client> --hwaddr=<client > -D <enterface name>
```

### # Define provisioning image for hosts

```
wwsh -y provision set <node naem> --vnfs=centos7.7 --bootstrap=`uname -r` --
files=dynamic_hosts,passwd,group,shadow,network
```

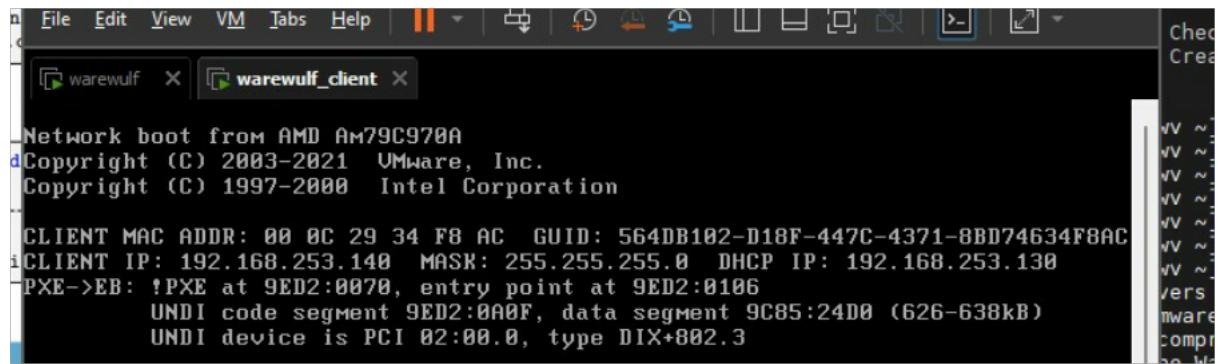
### # Restart dhcp / update PXE

```
[sms]# systemctl restart dhcpcd
[sms]# wwpxe update
```

```
[root@master_wv ~]# echo "GATEWAYDEV=ens34" > /tmp/network.$$
[root@master_wv ~]# wwsh -y file import /tmp/network.$$ --name network
[root@master_wv ~]# wwsh -y file set network --path /etc/sysconfig/network --mode=0644 --uid=0
About to apply 3 action(s) to 1 file(s):
SET: PATH          = /etc/sysconfig/network
SET: MODE          = 0644
SET: UID           = 0

Proceed?
[root@master_wv ~]# vim /etc/dhcp/dhc
dhclient.d/      dhclient-exit-hooks.d/ dhcpd6.conf          dhcpd.conf
[root@master_wv ~]# wwsh -y node new c1 --ipaddr=192.168.253.140 --hwaddr=00:0C:29:34:F8:AC -D ens34
[root@master_wv ~]# wwsh -y provision set c1 --vnfs=centos7.7 --bootstrap='uname -r' --files=dynamic_hosts,passw
d,group,shadow,network
[root@master_wv ~]# systemctl restart dhcpd
[root@master_wv ~]# wwsh pxe update
[root@master_wv ~]#
```

17 : restart client vm device make sure client vm has minimum of 8 gb ram and 60 gb hardisk and an interface configured in hostonly network same as host master vm .



Now Booting Warewulf...

```
Setting the hostname (c1):                                     OK
Loading drivers: uhci-hcd ohci-hcd ehci-hcd whci-hcd isp116x-hcd isp1362-hcd xhc
i-hcd s1811-hcd sd_mod                                      OK
Detecting hardware: ata_piix ata_piix pcnet32                  OK
Bringing up local loopback network:                            OK
Checking for network device: eth0 (ens34)                      OK
Configuring eth0 (ens34) statically: (192.168.253.140/255.255.255.0)  OK
Creating network initialization files: (ens34)                  OK
Trying to reach the master node at 192.168.253.130            OK
Probing for HW Address: (00:0c:29:34:f8:ac)                   OK
Starting syslogd:                                            OK
Getting base node configuration:                             OK
Starting the provision handler:
  * adhoc-pre                                              OK
  * ipmiconfig Auto configuration not activated           SKIPPED
  * filesystems                                           RUNNING
    * mounting /                                         OK
  * filesystems                                           OK
  * getvnfss                                             OK
```

```
CentOS Linux 7 (Core)
Kernel 3.10.0-1160.el7.x86_64 on an x86_64

c1 login: root
Password:
[root@c1 ~]# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
qdisc mq 0: lo
  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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UNKNOWN group default qlen 1000
  link/ether 00:0c:29:34:f8:ac brd ff:ff:ff:ff:ff:ff
  inet 192.168.253.140/24 brd 192.168.253.255 scope global eth0
    valid_lft forever preferred_lft forever
  inet6 fe80::20c:29ff:fe34:f8ac/64 scope link
    valid_lft forever preferred_lft forever
[root@c1 ~]#
```

## 18 : try doing ssh login to the client vm from host master

```
[root@master_wv ~]# nl /etc/hosts
 1 127.0.0.1 localhost.localdomain localhost4 localhost4.localdomain4
 2 ::1 localhost.localdomain localhost6 localhost6.localdomain6
 3 192.168.253.130 master
 4 ### ALL ENTRIES BELOW THIS LINE WILL BE OVERWRITTEN BY WAREWULF ####
 5 #
 6 # See provision.conf for configuration parameters

 7 # Node Entry for node: c1 (ID=7)
 8 192.168.253.140 c1.localdomain c1 c1-ens34.localdomain c1-ens34
[root@master_wv ~]# ssh root@c1
Warning: Permanently added 'c1,192.168.253.140' (ECDSA) to the list of known hosts.
Last login: Wed Jun 26 11:52:44 2024
[root@c1 ~]#
```

## WAREWULF

### About:

Warewulf is a Linux operating system provisioning platform that enables secure, scalable, turnkey cluster deployments while maintaining flexibility and simplicity.

Since its original release in 2001, Warewulf has grown to be the most popular open source and vendor-agnostic provisioning solution in the global HPC community. Warewulf is well-known for its huge scalability and easy administration of stateless (disk-optional) provisioning.

Warewulf uses a basic administrative approach that revolves around virtual node images that are used to provision cluster nodes. This means that hundreds or thousands of cluster nodes can boot and run from the same virtual node file system image.

Warewulf has gone through several modifications over the last two decades, but its design principles have always stayed the same: a simple, scalable, stateless (though earlier versions could furnish statefully), and extremely flexible operating system provisioning system for all types of clusters.

### FEATURES:

**Lightweight:** Warewulf provisions stateless operating system images and then gets out of the way.

**Simple:** Warewulf is used by hobbyists, researchers, scientists, engineers and systems administrators because it is easy, lightweight, and simple.

**Flexible:** Warewulf is highly flexible and can address the needs of any environment—from a computer lab with graphical workstations, to under-the-desk clusters, to massive supercomputing centers providing traditional HPC capabilities to thousands of users.

**Agnostic:** From the Linux distribution of choice to the underlying hardware, Warewulf is agnostic and standards compliant. From ARM to x86, Atos to Dell, Debian, SUSE, Rocky, CentOS, and RHEL, Warewulf can do it all.

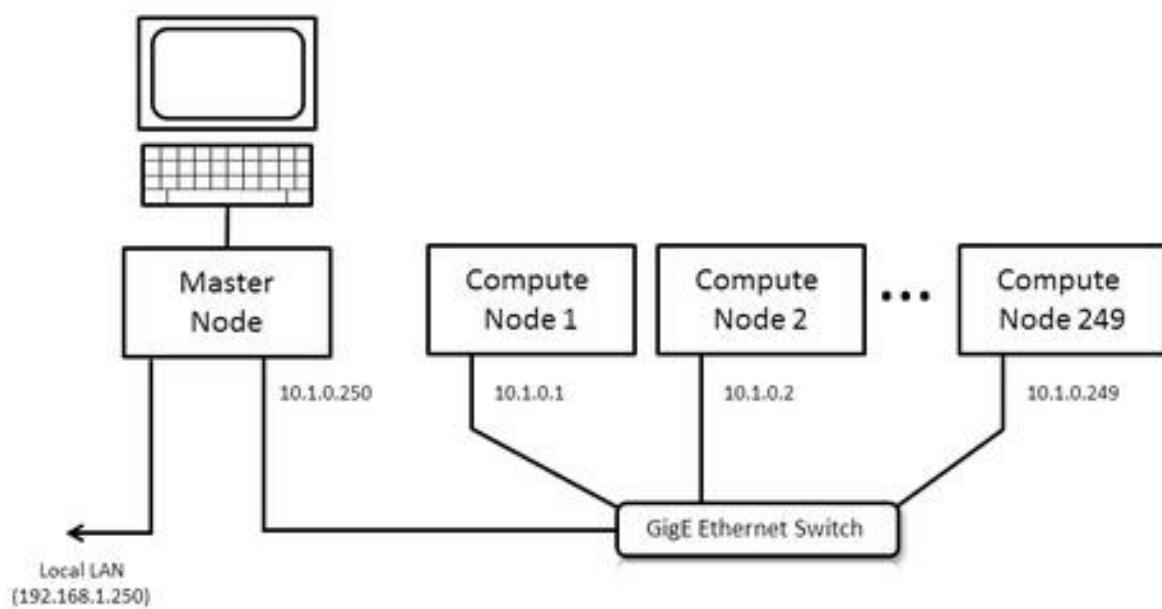
**Secure:** Warewulf is the only stateless provisioning system that will support SELinux out of the box. Just enable your node operating system container to support SELinux, and Warewulf do the rest!

**Open Source:** For the last 20 years, Warewulf has remained open source and continues to be the golden standard for cluster provisioning.

#### Prerequisites:

The basic list of packages you need on the master is,

- MySQL
- Apache (or similar web server)
- dhcp
- tftp-server
- Perl
- mod-perl



Network architecture

## Why is Provisioning Important

Clusters are collections of servers that work together to perform a specific job or set of tasks. While there are numerous use cases for clustering nowadays, Warewulf was initially created out of necessity. Scalability difficulties became apparent around 2000, as Linux clustering for HPC matured. Of course, in HPC, there are numerous scalability issues that must be addressed as clusters are scaled up. The "administrative scaling" concept, which states that full-time systems administrators can only manage so many servers, was introduced quite early on. While homogenous setups helped, we still had the issue of each deployed server becoming a point of administration, version drift, or debugging. The greater the cluster, the more difficult the challenge was to solve. Warewulf was intended to assist with just this.

## Warewulf and stateless Provisioning

The realization that you never need to install another computing node is known as stateless provisioning. Consider it similar to how a LiveOS or LiveISO is booted up on network nodes. This indicates that system administration is done as a single unit for the entire cluster rather than for each individual node.

Any individual nodes experiencing hardware issues are those in the cluster if every cluster node is booting from the same OS image (or collection of OS images). It is not necessary to do system administration and software debugging at isolated locations within a cluster. Version drift does not exist since nodes cannot become out of sync. It becomes identical to its neighbors with each reboot.

Warewulf provisions the operating system by default to system memory. There is no need for hard drives with Warewulf.

## iPXE

Warewulf uses iPXE for network boot. Typically iPXE is provided by the operating system; but the iPXE binaries can be built with scripts/build-ipxe.sh. This script accepts command-line arguments that are passed to the underlying make process. e.g.,

```
echo "#!ipxe
echo Tagging with vlan 1000
vcreate --tag 1000 net0 autoboot || shell" >vlan-1000.ipxe

sh scripts/build-ipxe.sh EMBED=$(readlink -f vlan-1000.ipxe)
```

By default, build-ipxe.sh will attempt to write iPXE builds to /usr/local/share/ipxe/. This path can be specified using the DESTDIR environment variable. Other supported environment variables include IPXE\_BRANCH and TARGETS.

### Warewulf vs Xcat

Features	Warewulf	Xcat
Licensing model	BSD License	Eclipse Public License
Supported Linux Distributions	RHEL 5/6(*), CentOS 5/6, Debian, Ubuntu	SLES 10/11(*), RHEL 5/6(*), CentOS 5/6, Fedora 8/9/12/13/14, AIX 5/6/7(*)
Scalability	10,000 nodes	100,000 nodes
Provisioning Method	Images, VNFS images created by Administrators	Images (default) and Packages
Support for third party software add-on	No	Yes
Command Line Interface Cluster Management Tool	wwsh	XCAT commands

### CONCLUSION:

High-performance computing (HPC) settings are the main use case for Warewulf, an open-source operating system management tools. It provides stateless and diskless computing node provisioning, enabling nodes to boot from network images instead of local storage. This methodology streamlines upkeep, minimizes expenses, and guarantees uniform settings throughout the cluster. With its emphasis on security features like SELinux support and per-node asset key-based provisioning, Warewulf is incredibly configurable and scalable. Because it's open source, the HPC community may collaborate and innovate with it.