# Warewulf Starter Kit

Geoff Galitz
CAOS Foundation
http://www.caosity.org
geoff@galitz.org

March 10, 2006

**Abstract**

Warewulf is a clustering framework. It's purpose is to provide the software infrastructure to build and maintain your cluster. It is designed to be simple and impose low overhead.

This documenation is current as of Warewulf version 2.6.

# 1 Requirements

Warewulf requires:

- Linux

- Perl

- Syslinux

- DHCP

- tftp

- PXE or Etherboot capable NIC for the nodes

- 2 NICs for the warewulf master, 1 for the cluster and 1 for general network access.

Recommendations:

- Caos Linux for better OS integration

- PXE capable network cards

## 2  Getting Warewulf

Warewulf is availble via svn:

```
# svn co https://runlevelzero.net/svn/warewulf
```

Or via yum on certain Linux distributions:

```
# yum install warewulf
```

## 3  Building Warewulf

If you are installing via yum, you can skip this section.

Warewulf code can be obtained via the svn repository. Checkout the code via svn and build it using either the mzbuild utility or via conventional means.

To build the RPM's via the mezzanine package (mzbuild):

```
$ mzbuild
```

If mezzanine does not exist for your distribution, you can still create the RPM's using the standard rpmbuild methodology.

Once the packages are built, it is best to create a local yum repository as yum will take care of any dependancies that may be required. To do so, you can create a file called local-repo in your /etc/yum.repos.d directory looking like this:

```
[Local Packages]
Name=Local Packages
url=file:////usr/local/packages
```
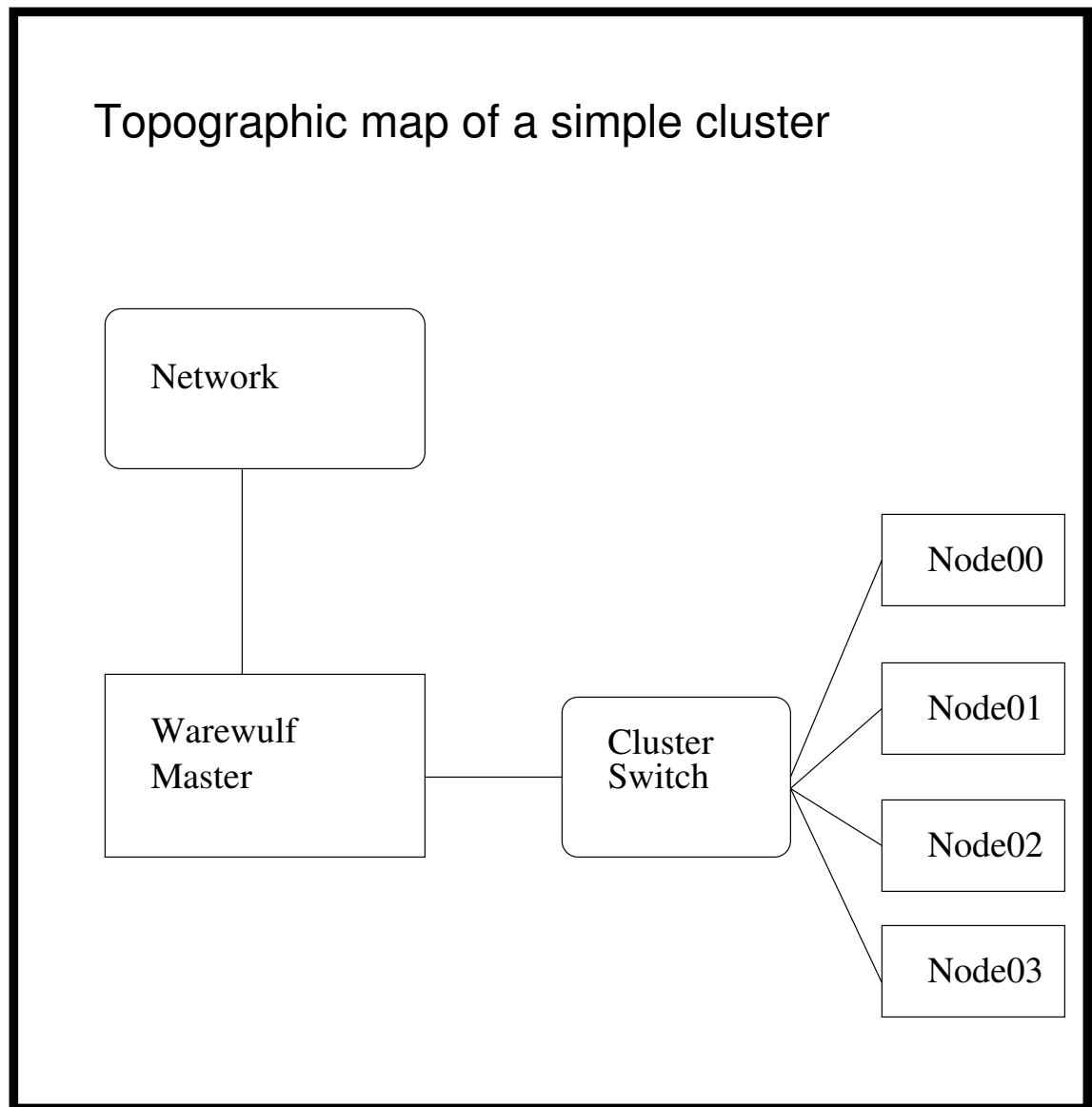
And of course you must create or update the repo database to reflect the added packages:

```
# createrepo /usr/local/packages
```

If that was succesful, installing the warewulf software should be a matter of:

```
# yum install warewulf
```

## 4   Configuring Warewulf

Topographic map of a simple cluster

Network

Warewulf
Master

Cluster
Switch

Node00

Node01

Node02

Node03

The warewulf master is configured via the wwinit command. Each step in the configuration process can be skipped in the event you need to make configuration changes for your system that a standard warewulf deployment would overwrite.

The first step in the process is configuring the network interfaces. The wwinit pro-

gram will query you for a boot network. This will be 10.128.0.0 unless you opt to change it. The purpose of this special network is to provide a safe and undefined network space for new nodes to come online and notify the master. Once the nodes are added, their identities will be defined by the MAC address and handled via the syslinux component.

The 10.128.0.0 network device will be configured as an alias attached to your primary cluster network device. This is normal and will not interfere with the normal operation of your cluster.

```
[root@wwmaster ~] # wwinit

======================= Warewulf configuration tool =========================

This program should be run on the master node of a Warewulf cluster after
the system has been completly configured (ie. drivers, network, etc..) and
the Warewulf programs and VNFS has been installed

Have you properly configured this system as a Warewulf master? [y/N]: y

======================= Auto boot device configuration =======================

Warewulf requires a specific non-conflicting network for booting the nodes.
when using the 'boot device = auto' paradigm. This will create a device alias
off of your admin network. Select an address range that is unique on the
network attached to eth1.

Specify the network address that should be used for the DHCP node
boot range [10.128.0.0]:

============================ Node address ranges ===========================

Below are the addresses and ranges that will be used for your Warewulf cluster
This information is calculated directly from the current network configuration
of your master node.

Network    Device    Nodes  Starting IP     Ending IP
BOOT       eth1:1    190    10.128.0.64     10.128.0.254
ADMIN      eth1      190    192.168.1.64    192.168.1.254
SHAREDFS   eth1      190    192.168.1.64    192.168.1.254
CLUSTER    eth1      190    192.168.1.64    192.168.1.254

Note: If you want to tweak the the above configuration, you should edit
        the network configuration on the master node.

Does this configuration look correct? [Y/n]:
```

```
=========================== Services configuration ==========================

Should I configure DHCP for you now? [Y/n]:
Internet Systems Consortium DHCP Server V3.0.1
Copyright 2004 Internet Systems Consortium.
All rights reserved.
For info, please visit http://www.isc.org/sw/dhcp/
Shutting down dhcpd:                                          [   OK   ]
Starting dhcpd:                                              [   OK   ]

Should I configure your NFS exports now? [Y/n]:
Stopping portmap:                                            [   OK   ]
Starting portmap:                                            [   OK   ]
Shutting down NFS mountd:                                    [   OK   ]
Shutting down NFS daemon:                                    [   OK   ]
Shutting down NFS quotas:                                    [   OK   ]
Shutting down NFS services:                                  [   OK   ]
Starting NFS services:                                       [   OK   ]
Starting NFS quotas:                                         [   OK   ]
Starting NFS daemon:                                         [   OK   ]
Starting NFS mountd:                                         [   OK   ]

Should I configure TFTP for you now? [Y/n]:
Stopping xinetd:                                             [   OK   ]
Starting xinetd:                                             [   OK   ]

Syslog is accepting remote connections.

======================= Building boot enviornment for nodes =================

Shutting down vnfsd:                                         [   OK   ]
Starting vnfsd:                                              [   OK   ]

Shutting down wwnewd:                                        [   OK   ]
Starting wwnewd:                                             [   OK   ]

Shutting down warewulfd:                                     [   OK   ]
Starting Warewulf:
   Warewulfd                                                 [   OK   ]
   PXE config files                                          [   OK   ]
   Adding hosts entries

Generating syncs for [nodegroup1]

[root@wwmaster ~] #
```

# 5 Configuring the VNFS and adding packages

The VNFS is virtual node filesystem for the nodes. You can have multiple VNFS images, but we won't cover that in this document.

There are build scripts for several Linux distributions in the /usr/share/warewulf/vnfs-scripts directory. Creating the VNFS is a matter of running the script for the OS you are running.

After running the VNFS creation script, you can add additional components via the yum command.

```
# yum --installroot=/vnfs/default install tcsh
```

After creating or updating the VNFS image, you need to build downloadable image:

```
# wwvnfs --build
```

# 6 Getting the nodes online

Once warewulf is installed and configured on the master, getting the nodes online should be straight-forward.

The nodes will require that PXE booting is enabled in BIOS. Once this is done, they will ask the master (via broadcast) for their identity. The master will assign the node an IP address when it comes online and assign it to the "new" nodegroup.

When added, each node is represented by a file in the /etc/warewulf/nodes directory structure. Each node is represented by a file named for that node. For example:

```
[root@wwmaster nodes]# ls new/
node00   node01   node02   node03   node04   node05
```

To make the nodes active, move the files into an active nodegroup directory (probably /etc/warewulf/nodes/nodegroup1) and restart the warewulf server software.

```
[root@wwmaster nodes]# mv new/node* nodegroup1
[root@wwmaster nodes]# service warewulf restart
```

# 7 Adding Users

Users are added to the config file for the relevant nodegroup. Add the users to the line called "user names" in the config file.

```
# What users should have access to the nodes in this group?

   user names            = foo,bar,fred,guest
```

The user configuration is then proprogated by the wwvnfs –sync command. At this point, the users should be able to connect to the nodes via rsh/rlogin. Note that the relevant shell needs to exist in the VNFS. If your preferred shell is tcsh or zsh, you will probably need to add it to the VNFS.