APRIL 25, OSTRAVA

Warewulf making cluster installations fast and reliable

Christian Goll <cgoll@suse.com>

SUSE

Introduction

Warewulf is tool for managing Beowulf clusters

Beowulf

- old british poem

Beowulf cluster

- became popular in the 90.
- use of the shelf hardware
  - 486 & Linux **not** Cray & Unix
- Warewulf is a typo of werewolf
- Caos Linux & Warewulf were designed for Beowulf clusters
- Caos Linux evolved into CentOS
  - Gregory Kurtzner started developement of Warewulf v4

Introduction

HPC landscape

Top five Supercomputers

1	Frontier	EPYC 64C	AMD MI250X	Slingshot-11
2	Aurora	Xeon 9470	Intel GPU Max	Slingshot-11
3	Eagle	Xeon 8480	NVIDIA H100	NVIDIA Infiniband
4	Fugaku	A64FX 48C 2.2GHz	-	Tofu interconnect D
5	LUMI	EPYC 64C 2GHz	AMD MI250X	Slingshot-11

- only Fugaku uses non standard CPU
- others are Beowulf clusters with GPUs attached

Introduction

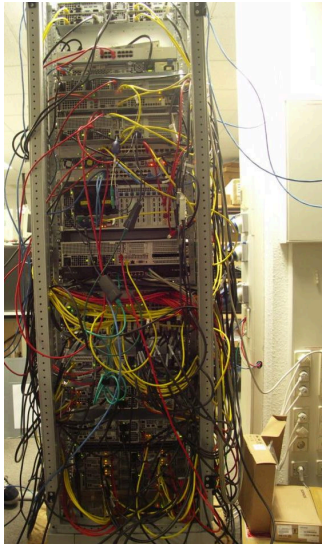
Beowulf cluster

Base components

- management node
- compute nodes
- management network
- network boot (PXE)

Optional components

- more compute nodes
- fast network interconnects
- central storage
- bmc/ipmi

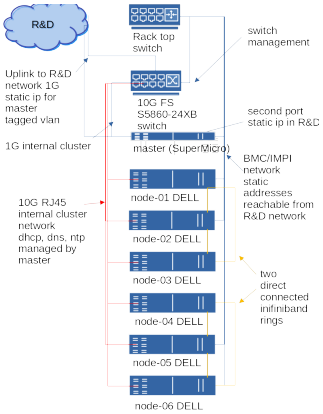


Introduction

Beowulf Cluster

Differences to data centers

- compute nodes are cattle
  - parallel/MPI require identical nodes
- hierarchical organization
- compute are not updated after boot process
- applications come from central storage
- applications are self compiled with tools like `spack` & `EasyBuild`



Warewulf description

Software stack

Warewulf components

Compute nodes

boot from network into tmpfs

Warewulfd delivers

- kernel & modules
- node image
- node configurations

`wwctl` cmd line tool

- manages node database
- manages node image

external components

Dhcp server

- ISC `dhcpd` server
- `dnsmasq`

tftp

- tftp from `kernel.org`
- dnsmasq

Optional

- nfs
- manage `/etc/hosts`

Warewulf configuration

database `/etc/warewulf/nodes.conf`

- plain yaml file
- easy backup
- can be version controlled
- external tools support
  - vim, ansible

Profiles

- stores identical values for collection of nodes
- values can be overridden on node basis

```
WW_INTERNAL: 45
nodeprofiles:
  default:
    comment: This profile is automatical
    container name: leap
    network devices:
      default:
        device: eth0
nodes:
  n01:
    profiles:
    - default
    network devices:
      default:
        hwaddr: 52:54:00:4e:cb:1d
        ipaddr: 172.16.130.101
  n02:
```

Warewulf configuration

command line database manipulation

add node

```
wwctl node add n01
    -I 10.10.10.1
```

modify node

```
wwctl node set n01
    --comment "
   Have fun"
```

list node

```
wwctl node list
   n01 -a
```

```
NODE FIELD                          PROFILE    VALUE
n01  Id                             --         n01
n01  Comment                        SUPERSEDED Have fun
n01  ContainerName                  default    leap
n01  Ipxe                           --         (default)
n01  RuntimeOverlay                 --         (generic)
n01  SystemOverlay                  --         (wwinit)
n01  Root                           --         (initramfs)
n01  Discoverable                   --         false
n01  Init                           --         (/sbin/init)
n01  Kernel.Args                    --         (quiet crashkernel
n01  Profiles                       --         default
n01  PrimaryNetDev                  --         (default)
n01  NetDevs[default].Type          --         (ethernet)
n01  NetDevs[default].OnBoot        --         (true)
n01  NetDevs[default].Device        default    eth0
n01  NetDevs[default].Hwaddr        --         52:54:00:4e:cb:1d
```

Warewulf configuration

Templates& Overlays

Configuration templates

- based on go templates
- {{.*foo*}} replaced with variable `foo`
- exported go function can be called

Configuration overlays

- rendered templates packed into overlay
- overlay put on top of node image

Listing 1: /etc/issue.ww → /etc/issue

```
1  Warewulf Node:       {{.Id}}
2  Container:           {{.Container}}
3  {{ if .Kernel.Version }}Kernel:
4  Kernelargs:          {{.Kernel.Args}}
5
6  Network:
7  {{- range $devname, $netdev := .NetDevs}}
8      {{$devname}}: {{$netdev.Device}}
9      {{$devname}}: {{$netdev.IpCIDR}}
10 {{if $netdev.Ipaddr6 }}     {{$devname}}:
11 {{if $netdev.Hwaddr }}     {{$devname}}:
12 {{end}}
13
14 Greetings from {{ .Tags.Location }}
```

Warewulf configuration

Rendered template

User defined variables

- every profile/node can have user defined variables
- extra namespace for networks

Add location tag

```
wwctl node set n01 --tagadd="
    Location=Ostrava"
```

Render template

```
wwctl overlay show --render n01
    wwinit /etc/issue.ww
```

Listing 2: renderd /etc/issue

```
backupFile: true
writeFile: true
Filename: /etc/issue
Warewulf Node:      n01
Container:          leap
Kernelargs:         quiet crashkernel=no

Network:
    default: eth0
    default: 172.16.130.101/24
    default: 52:54:00:4e:cb:1d

Greetings from Ostrava
```

Warewulf configuration

Overlays

warewulf defines two types of overlays

System overlay

- available on boot
- warewulf boot strap files
- static network configurations:
  - wicked
  - NetworkManager
  - other legacy scripts
- nfs mounts
- file system mounts

Runtime overlay

- updated regulary
- can be secured

User defined overlays

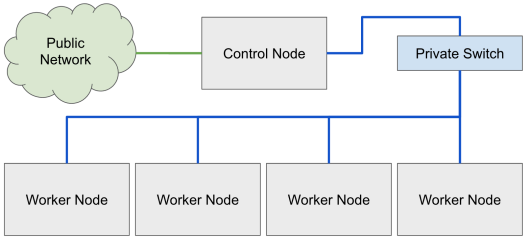
- users are encouraged to create own configuration templates
- can reside in system & runtime overlays

Warewulf configuration

Security

Assumptions

- private/cluster network is secure
- parallel filesystems & NFS
  → root on node means root everywhere



measurements

- node image & system overlay protected with Asset Tag
- system overlays **must** be downloaded from privileged port
- persistant malware installation is impossible as node images are ephermal

Warewulf configuration

Node images

ELements

- complete OS images
- called containers in warewulf
- must be imported from:
  - chroot directory
  - OCI registry
- multiple different node images can be imported
- node images independent from host OS

`registry.suse.com`

- SUSE SLE 15SP5

`registry.opensuse.org`

- openSUSE Tumleweed
- openSUSE Leap 15SP[3-5]

`ghcr.io`

- openSUSE Leap
- Rocky EL (8&9)
- Debian Bockworm

Warewulf configuration

Node image examples

Import SLE image from SUSE registry

```
ww4-host:# export WAREWULF_OCI_USERNAME=cgoll@suse.com
ww4-host:# export WAREWULF_OCI_PASSWORD=INTERNAL-USE-ONLY-xxxxxx
ww4-host:# wwctl container import docker://registry.suse.com/suse/hpc/
    warewulf4-x86_64/sle-hpc-node:latest sle-hpc
```

Import Leap image openSUSE registry

```
ww4-host:# wwctl container import docker://registry.opensuse.org/science/
    warewulf/leap-15.5/containers/kernel:latest leap
```

Execute shell in images

```
wwctl container shell sle-hpc
[sle-hpc] Warewulf>
```

- SLE registration from outer node is mounted into image

Warewulf configuration

disk management

Needs following elements:

- disks
- partitions (on disk)
- filesystem (on partition)

Implementation

- use `ignition`
  warewulf has its own service:
  ignition-ww4-disk.service
- **not** called in `initrd`
- **before** `sysroot.mount`

Single parition

```
wwctl node set n01 --diskname /dev/
    vda --diskwipe --partname scratch
     --partcreate --fsname scratch --
    fsformat btrfs --fspath /scratch
    --fswipe
```

Add swap

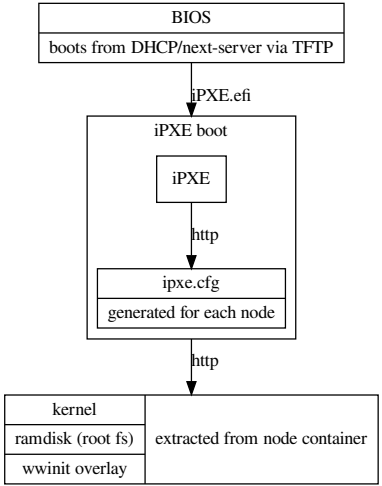
```
wwctl node set n01 --diskname /dev/
    vda --partname swap --partsize
    =1024 --partnumber 1 --fsname
    swap --fsformat swap --fspath
    swap
```

Warewulf boot

Boot process

boot with iPXE

- distribution iPXE binaries are used
- `tftp` transfers are small
- `kernel` is extracted from container/node image on the fly
- root fs is the container/node image
  configuration overlay added on top
- **no** secure boot

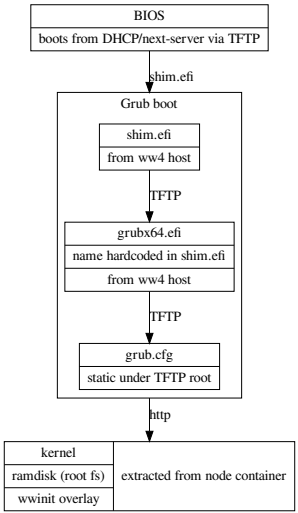


Warewulf boot

Secure boot

Boot with grub tftp

- grub & shim extracted from host
- secure boot only possible if host shim & grub can boot node kernel

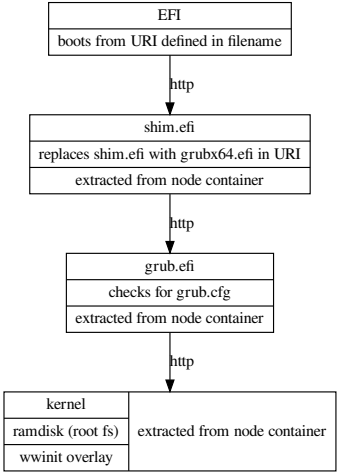


Warewulf secure boot

Cross product secure boot

Boot with grub http

- grub & shim extracted from node/container image
- secure boot with various distributions
- must be configured in BIOS



Warewulf

Development

Availiability

- part of SLE since 15SP5
- SLE node/container available

upstream

`github.com/warewulf/warewulf`
Rocky Linux Foundation project
Stakeholders:

- SUSE
- CIQ
- Intel/openHPC

Warewulf

Thank your, for you attention!