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

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

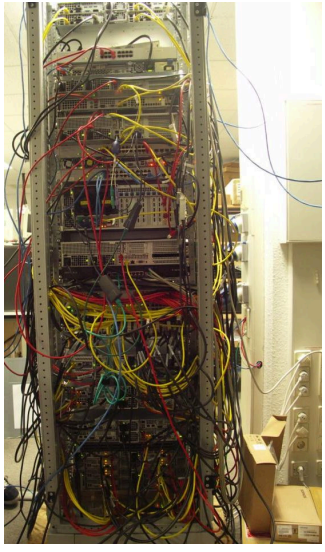
Beowulf cluster

base components

- management node
- compute nodes
- management network

optional components

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

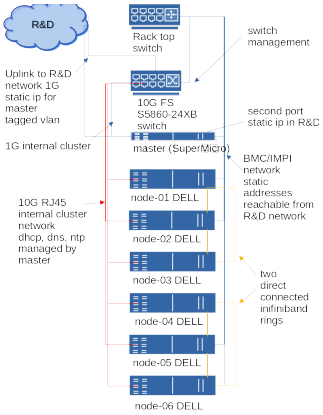


Introduction

Beowulf Cluster

differences to data centers

- compute nodes are cattle
- hierarchical organization
- compute are not updated after boot process
- application come from central storage
- applications are self compiled
- one application can run over several nodes



Warewulf description

software stack

warewulf components

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 ontop of node image

Listing 1: /etc/issue.ww

```
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}}
```

Warewulf configuration

rendered templates

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 ontop of node image

Listing 2: /etc/issue.ww

```
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}}
```

Warewulf configuration

overlays

warewulf defines two types of overlays

system overlay

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

runtime overlay

- updated on regular base
- can be secured

user defined overlays

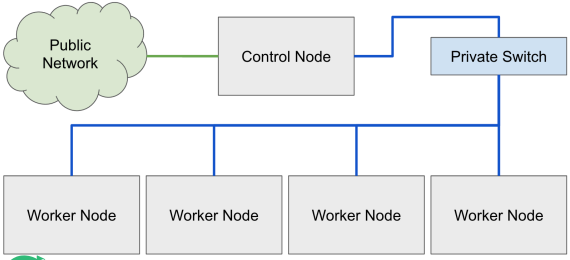
- users are encouraged to create own configuration templates
- can reside in system & runtime overlays

Warewulf configuration

security

assumptions

- private/cluster network is secure
- lateral movement isn't accounted
  - NFS mounts are common, when not **mandatory**



measurements

- node image & system overlay protected with BIOS UUID
- system overlays **must** be downloaded from privileged port

Warewulf configuration

node images

definition

- complete OS images
- called containers in warewulf
- must be imported from:
  - chroot directory
  - docker registry
  - local `dockerd`
- several different node images can be imported
- node images are vendor independent

`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 the SLE image

```
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 the Leap image

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

Warewulf configuration

node image examples

Execute shell in images

```
wwctl container shell sle-hpc
WARN   : Couldn`t mount /etc/SUSEConnect to /etc/SUSEConnect: no such
    file or directory
WARN   : Couldn`t mount /etc/zypp/credentials.d/SCCcredentials to /etc/
    zypp/credentials.d/SCCcredentials: no such file or directory
[sle-hpc] Warewulf>
```

- SLE registration from outer node is mounted into image

Warewulf configuration

disk management

Needs following elements:

- disks
- partitions needs parent disk
- filesystem needs partent partition

implementation

- call `ignition` with ignition-ww4-disk.service
- **not** in dracut
- **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

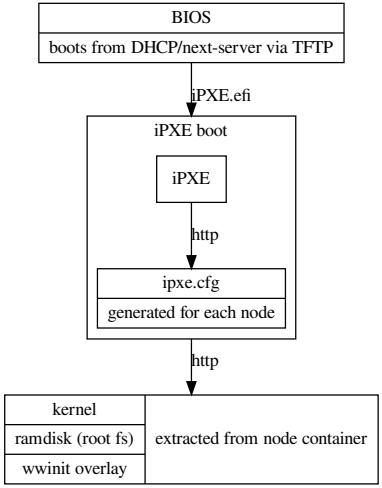
```
wwctl node set n01 --diskname /dev/vda --partname swap --partsize=1024 --partnumber 1 --fsname swap --fsformat swap --fspath swap
```

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

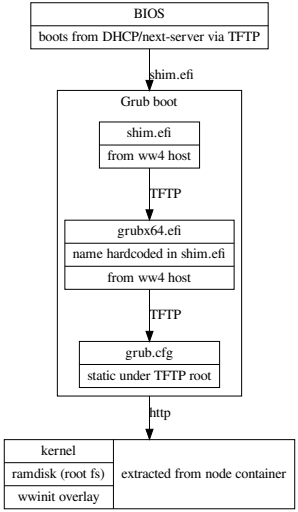


Warewulf boot

boot process

boot with grub tftp

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

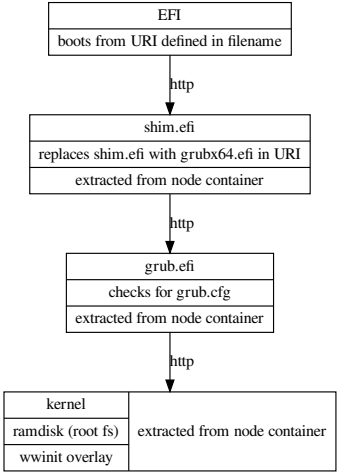


Warewulf boot

boot process

boot with grub http

- grub & shim xtracted 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 avaible

upstream

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

- SUSE
- CIQ
- Intel/openHPC

Warewulf

Thank your, for you attention!