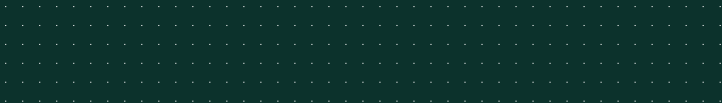


APRIL 25, OSTRAVA

Warewulf making cluster installations fast and reliable

Christian Goll <cgoll@suse.com>



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

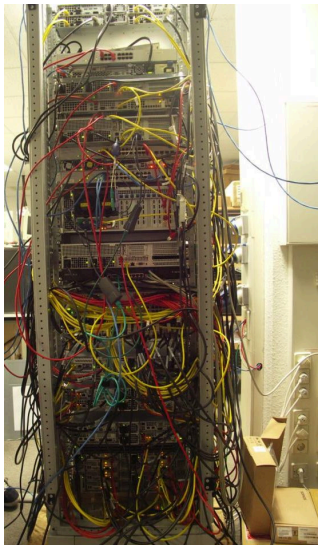
Beowulf cluster

base components

- management node
- compute nodes
- management network

optional components

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

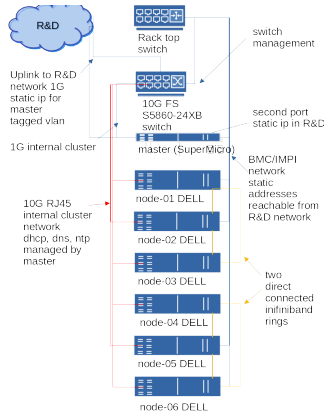


Introduction

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

- kernel tftp
- dnsmasq

optional

- nfs
- manage /etc/hosts



Warewulf description

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

```
1 WW_INTERNAL: 45
2 nodeprofiles:
3   default:
4     comment: This profile is automatically
5     container name: leap
6     network devices:
7       default:
8         device: eth0
9 nodes:
10  n01:
11    profiles:
12    - default
13    network devices:
14      default:
15        hwaddr: 52:54:00:4e:cb:1d
16        ipaddr: 172.16.130.101
17  n02:
```



Warewulf description

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"
```

1	NODE	FIELD	PROFILE	VALUE
2	n01	Id	--	n01
3	n01	Comment	SUPERSEDED	Have fun
4	n01	ContainerName	default	leap
5	n01	Ipxe	--	(default)
6	n01	RuntimeOverlay	--	(generic)
7	n01	SystemOverlay	--	(wwinit)
8	n01	Root	--	(initramfs)
9	n01	Discoverable	--	false
10	n01	Init	--	(/sbin/init)
11	n01	Kernel.Args	--	(quiet crashkernel
12	n01	Profiles	--	default
13	n01	PrimaryNetDev	--	(default)
14	n01	NetDevs[default].Type	--	(ethernet)
15	n01	NetDevs[default].OnBoot	--	(true)
16	n01	NetDevs[default].Device	default	eth0
17	n01	NetDevs[default].Hwaddr	--	52:54:00:4e:8c:b1:d
18	n01	NetDevs[default].Ipaddr	--	172.16.130.101



Warewulf description

node configuration

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: 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}}
```

