HPS
niklas.boelter@gwdg.de

Hendrik Nolte, Niklas Bölter

Node Provisioning Using Warewulf

Part II - Overlays

Table of contents

Discussion of First Tutorial Sheet

Time For Your Feedback!

Introduction: What are Overlays?

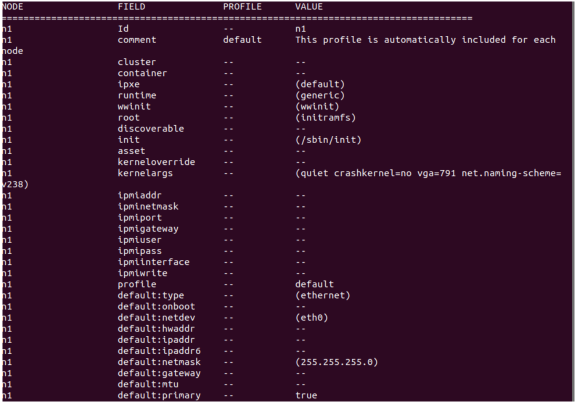
- During your hands-on session you have configured 2 nodes to boot into the same stateless image.
- Although this homogeneous state is a general advantage, sometimes there have to be node specific configurations, like
  - `hostname`
  - networks: Infiniband, Omni-Path, Ethernet, Integrated Platform Management Interface (IPMI)
  - sometimes: parallel file systems
- For this, Warewulf offers two different overlays:
  - A `System Overlay` which is called after the bootstrapping of the container is done
  - A `Runtime Overlay` which is called periodically during operation
- Both of these types can be templated
- If you have a template file you need to put a `.ww` suffix after the filename
  - Only then will Warewulf parse the variables and create individualized files for each node

Introduction: What are Overlays?

- Overlays are individual `cpio` archives for every node (think of .zip files)
- After the iPXE booted the kernel, the kernel calls `wwinit`, a script provided by the system overlay
- Only afterwards the container is booted via `init` (usually systemd)

Applying Overlays

- Overlays are stored in `/var/lib/warewulf/overlays`
- The default system overlay is called `wwinit`
- The default runtime overlay is called `generic`
- You can set the overlays as you set any other attribute



```
NODE                 FIELD              PROFILE      VALUE
=====================================================================================
n1                   Id                 --           n1
n1                   comment            default      This profile is automatically included for each
node
n1                   cluster            --           --
n1                   container          --           --
n1                   ipxe               --           (default)
n1                   runtime            --           (generic)
n1                   wwinit             --           (wwinit)
n1                   root               --           (initramfs)
n1                   discoverable       --           --
n1                   init               --           (/sbin/init)
n1                   asset              --           --
n1                   kerneloverride     --           --
n1                   kernelargs         --           (quiet crashkernel=no vga=791 net.naming-scheme=
v238)
n1                   ipmiaddr           --           --
n1                   ipminetmask        --           --
n1                   ipmiport           --           --
n1                   ipmigateway        --           --
n1                   ipmiuser           --           --
n1                   ipmipass           --           --
n1                   ipmiinterface      --           --
n1                   ipmiwrite          --           --
n1                   profile            --           default
n1                   default:type       --           (ethernet)
n1                   default:onboot     --           --
n1                   default:netdev     --           (eth0)
n1                   default:hwaddr     --           --
n1                   default:ipaddr     --           --
n1                   default:ipaddr6    --           --
n1                   default:netmask    --           (255.255.255.0)
n1                   default:gateway    --           --
n1                   default:mtu        --           --
n1                   default:primary    --           true
```

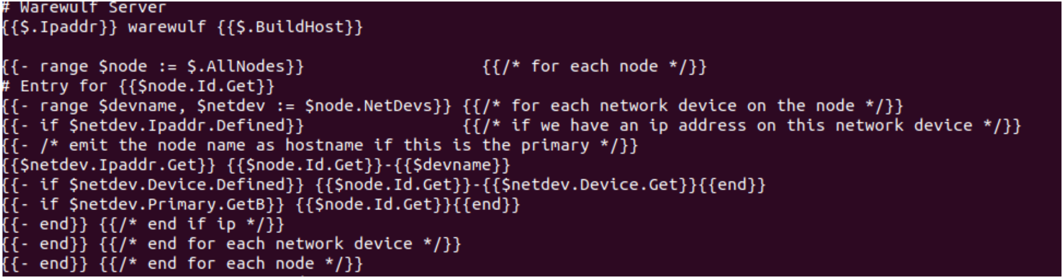
Figure: Warewulf node attributes

Templating

- Warewulf uses a simple `text/template` engine to convert dynamic, node specific content into static content
- Those files need to have a `.ww` suffix
- Those templates are used inside the overlays

Templating - Example

- Maybe you have wondered how you can resolve hostnames to IP addresses of the different compute nodes in your cluster?
- The answer is that an corresponding entry was done in `/etc/hosts`
- This file is used to resolve hostnames to ip addresses
- Lets have a look at `/var/lib/warewulf/overlays/generic/etc/hosts.ww`

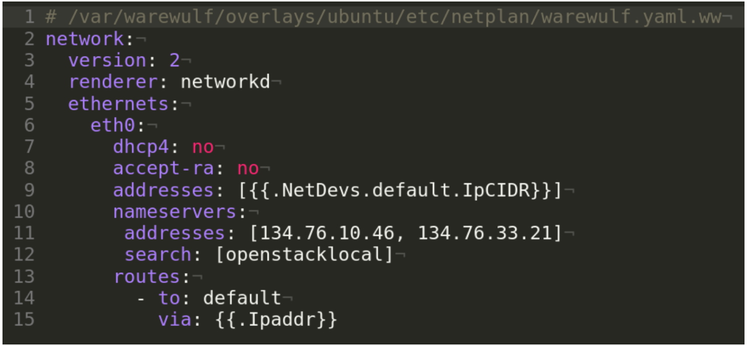


```
# Warewulf Server
{{$.Ipaddr}} warewulf {{$.BuildHost}}

{{- range $node := $.AllNodes}}                  {{/* for each node */}}
# Entry for {{$node.Id.Get}}
{{- range $devname, $netdev := $node.NetDevs}} {{/* for each network device on the node */}}
{{- if $netdev.Ipaddr.Defined}}                {{/* if we have an ip address on this network device */}}
{{- /* emit the node name as hostname if this is the primary */}}
{{$netdev.Ipaddr.Get}} {{$node.Id.Get}}-{{$devname}}
{{- if $netdev.Device.Defined}} {{$node.Id.Get}}-{{$netdev.Device.Get}}{{end}}
{{- if $netdev.Primary.GetB}} {{$node.Id.Get}}{{end}}
{{- end}} {{/* end if ip */}}
{{- end}} {{/* end for each network device */}}
{{- end}} {{/* end for each node */}}
```

Templating - Example 2

- How can we set the IP address for the nodes?
- On Ubuntu the network configuration for all interfaces is set via `netplan`
- Keep in mind a real compute node is connected to **multiple** networks



```
# /var/warewulf/overlays/ubuntu/etc/netplan/warewulf.yaml.ww
network:
  version: 2
  renderer: networkd
  ethernets:
    eth0:
      dhcp4: no
      accept-ra: no
      addresses: [{{.NetDevs.default.IpCIDR}}]
      nameservers:
       addresses: [134.76.10.46, 134.76.33.21]
       search: [openstacklocal]
      routes:
        - to: default
          via: {{.Ipaddr}}
```

Runtime Overlay - Systemd Integration

- Put /warewulf/runtime-overlay-deployed in the runtime overlay
- Create a new systemd service that has to start up **Before** services that have their configuration files in the runtime overlay (for security reasons)

```
[Service]
TimeoutStartSec=infinity
ExecStartPre=/usr/bin/bash -c "while \
        [ ! -f /warewulf/runtime-overlay-deployed ]; \
        do sleep 1; done; sleep 5"
ExecStart=/usr/bin/rm -f /warewulf/runtime-overlay-deployed
RestartSec=30s
Restart=always
```

Live Demo

Any Questions?

Live Demo