Writing a CNI - as easy as pie

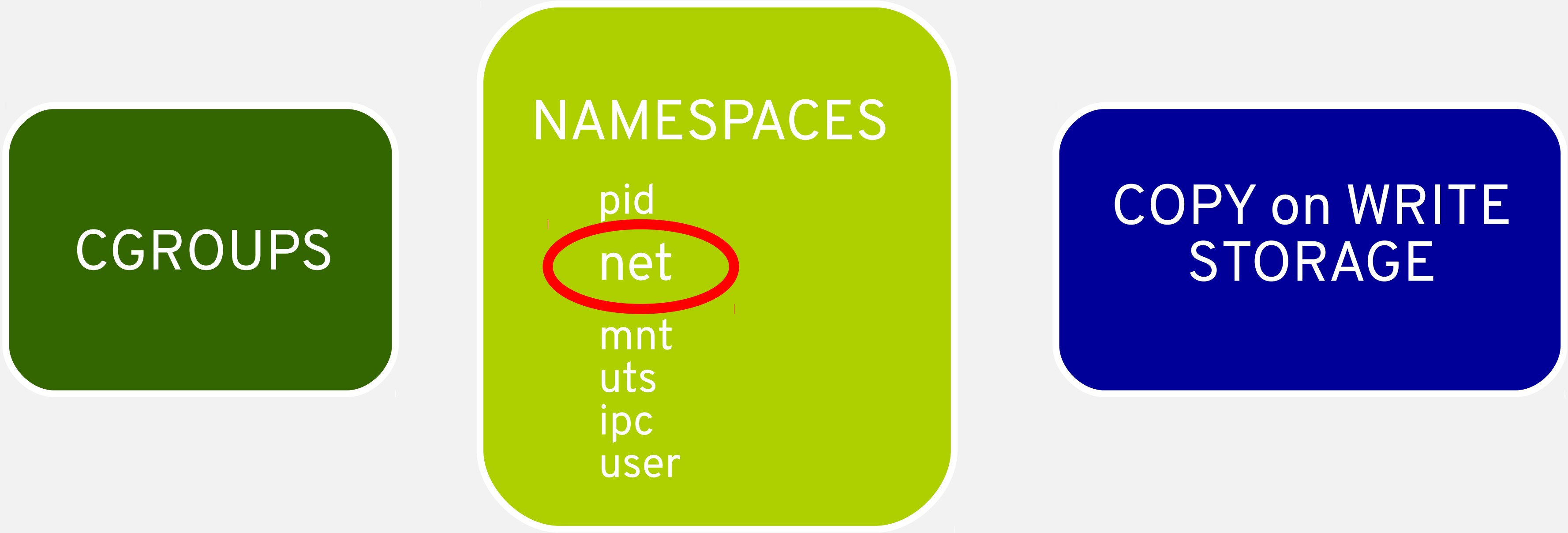
Marcin Mirecki
Software Engineer

FOSDEM 2019

TOC

1. Container networking
2. What is CNI?
3. Demo

Anatomy of a container



CGROUPS

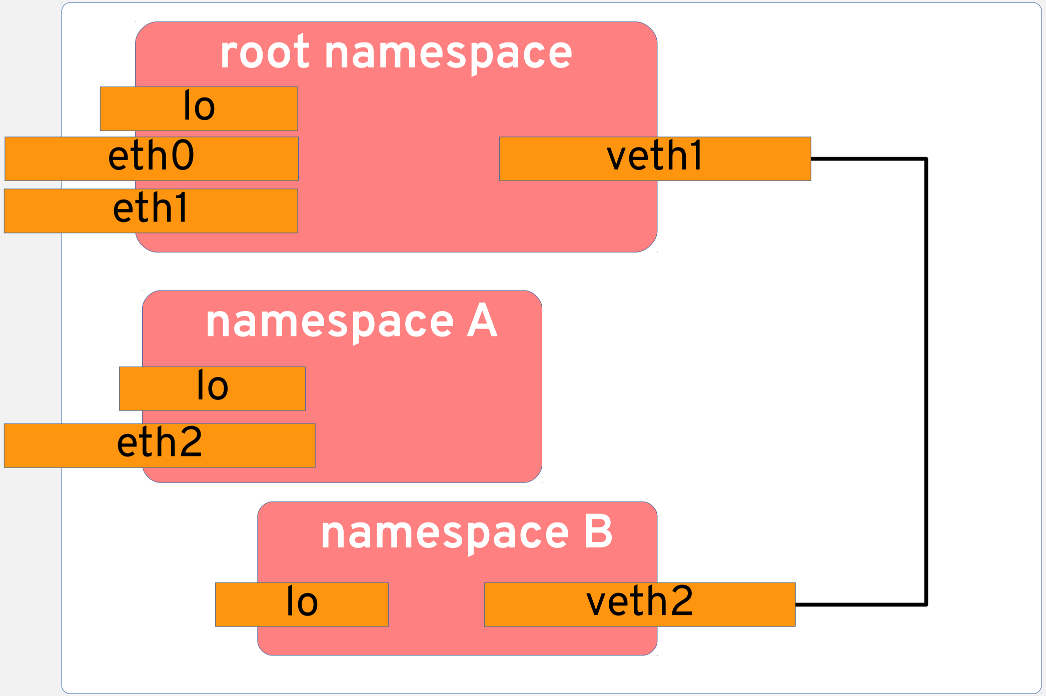
NAMESPACES

- pid
- net
- mnt
- uts
- ipc
- user

COPY on WRITE STORAGE

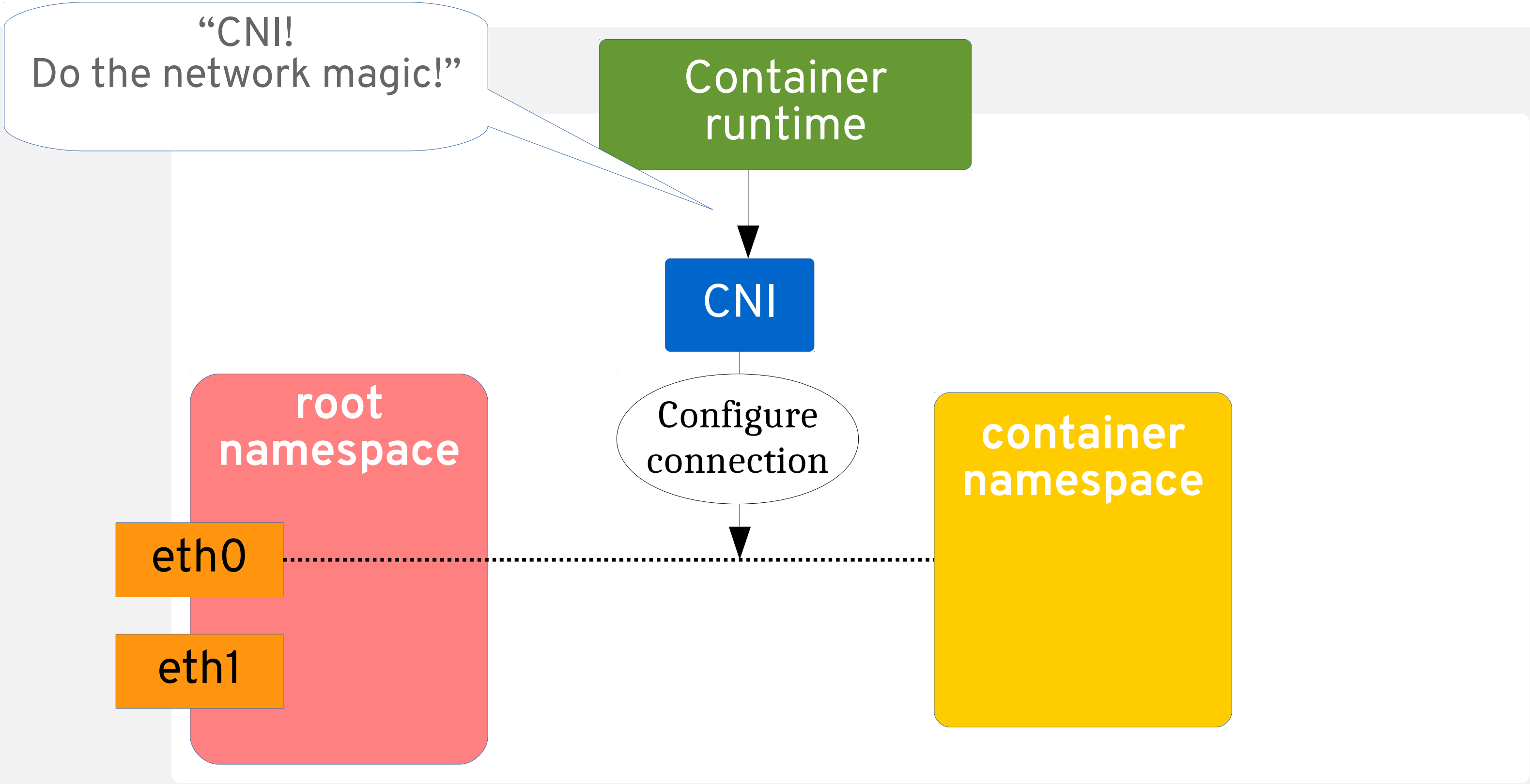
Network namespaces

- private network stack
- private network interfaces (lo included)
- private routing tables



Container networking

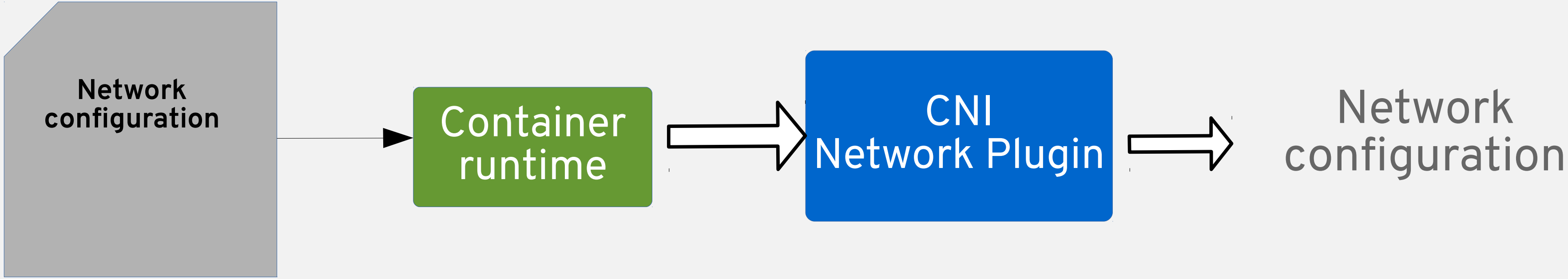
How a container is created



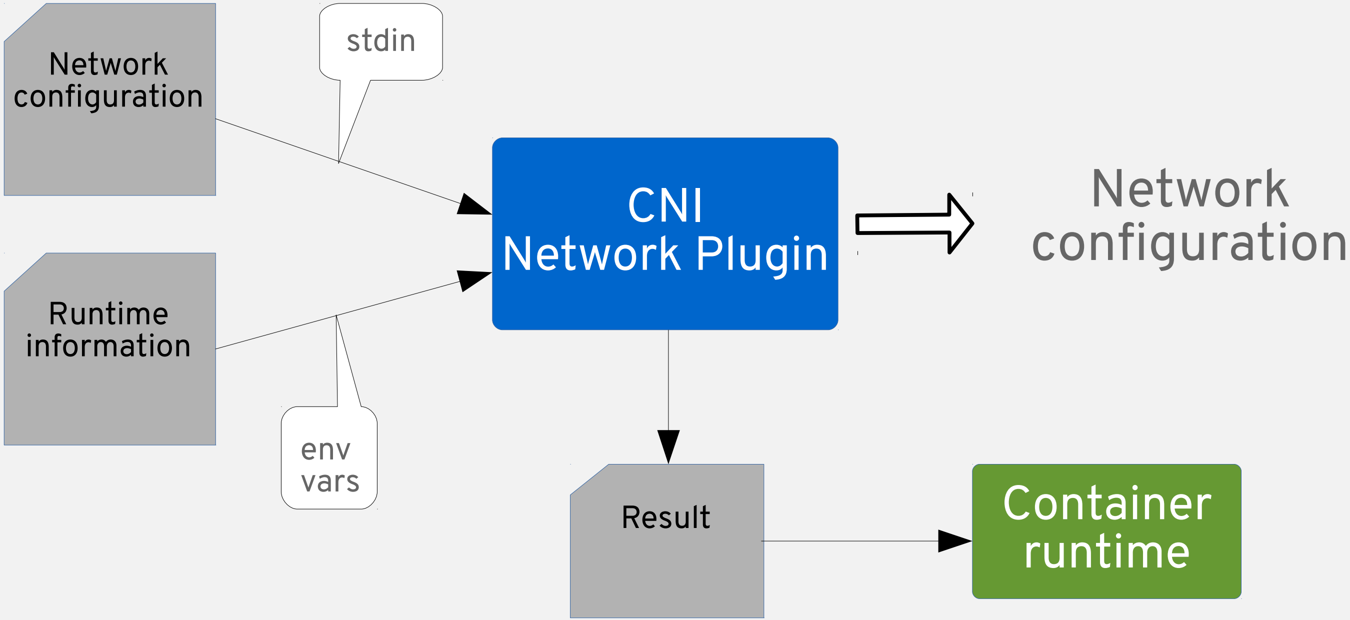
What’s a CNI?

- Short for: **Container Network Interface**
- Interface between container runtime and network implementation
- Consists of:
  - Specification
  - plugin implementations
  - plugin libraries
- Started as part of rkt
- Part of CNCF (Cloud Native Computing Foundation)

How a CNI works



CNI plugin invocation



CNI network configuration

```
{
   "cniVersion": "0.4.0",
   "name": "my name",
   "type": "demo",
   "ipam": { ... },
   "dns": { ... },
   "additonalArg1": ... ,

  ...
}
```

Name of the plugin binary

CNI runtime information

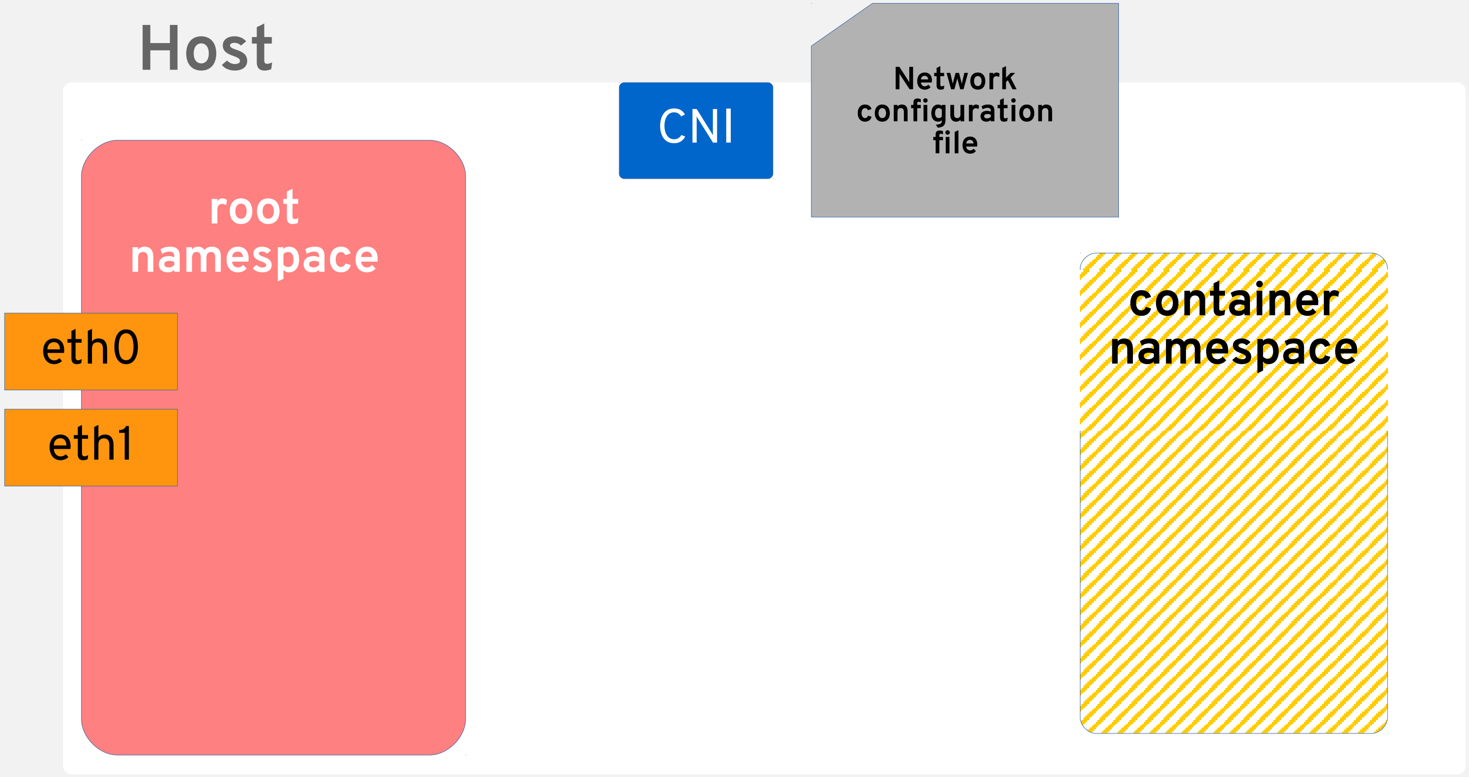
- **CNI_COMMAND** = ADD, DELETE, CHECK, VERSION
- **CNI_CONTAINERID** = <id>
- **CNI_NETNS** = */proc/*<pid>/ns/net
- **CNI_IFNAME** = eth0
- **CNI_PATH** = */opt/cni/bin*
- **CNI_ARGS**

CNI result

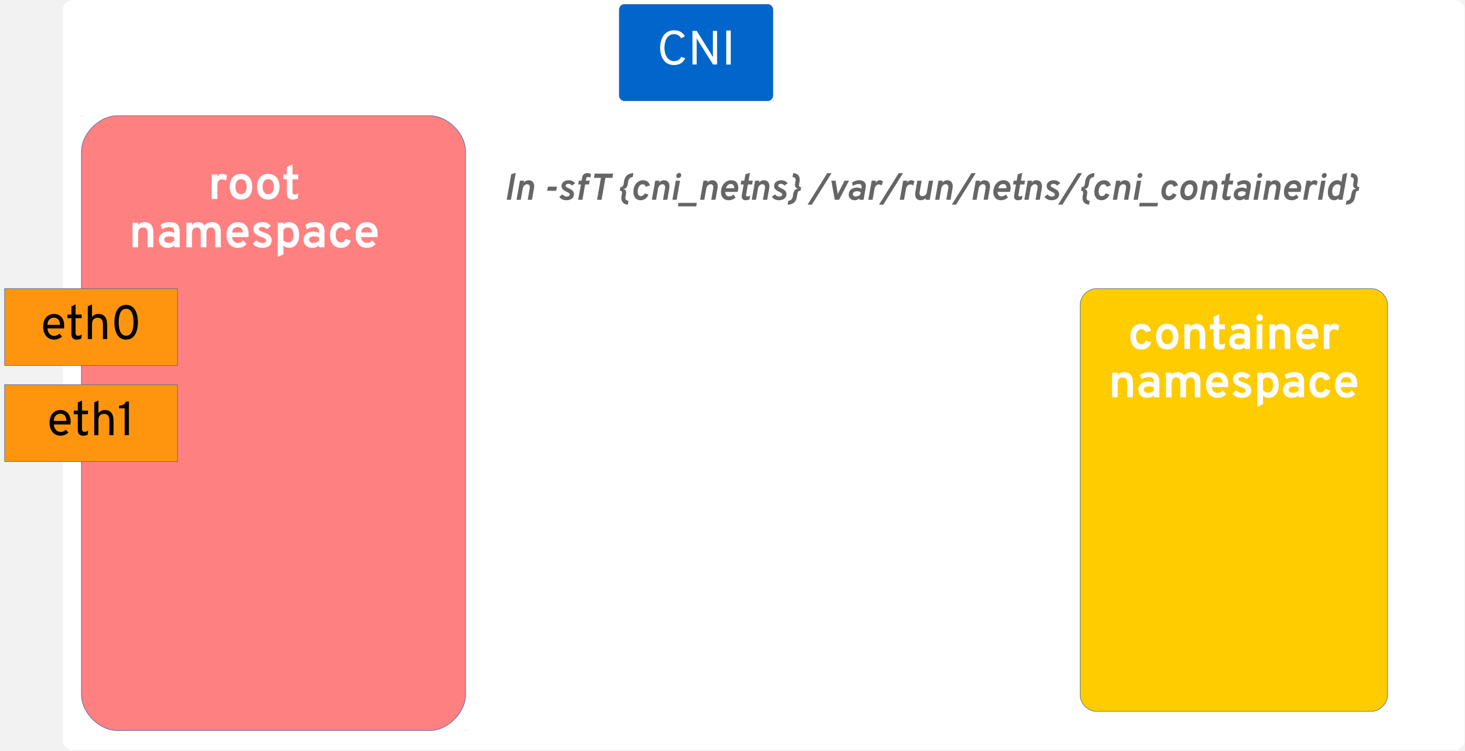
```
{
  "cniVersion": "0.4.0",
  "interfaces": [
      { ... },
  ],
  "ips": [
      { ... },
  ],
  "routes": [
      { ... },
  ]
  "dns": { … }
}
```

DEMO

DEMO

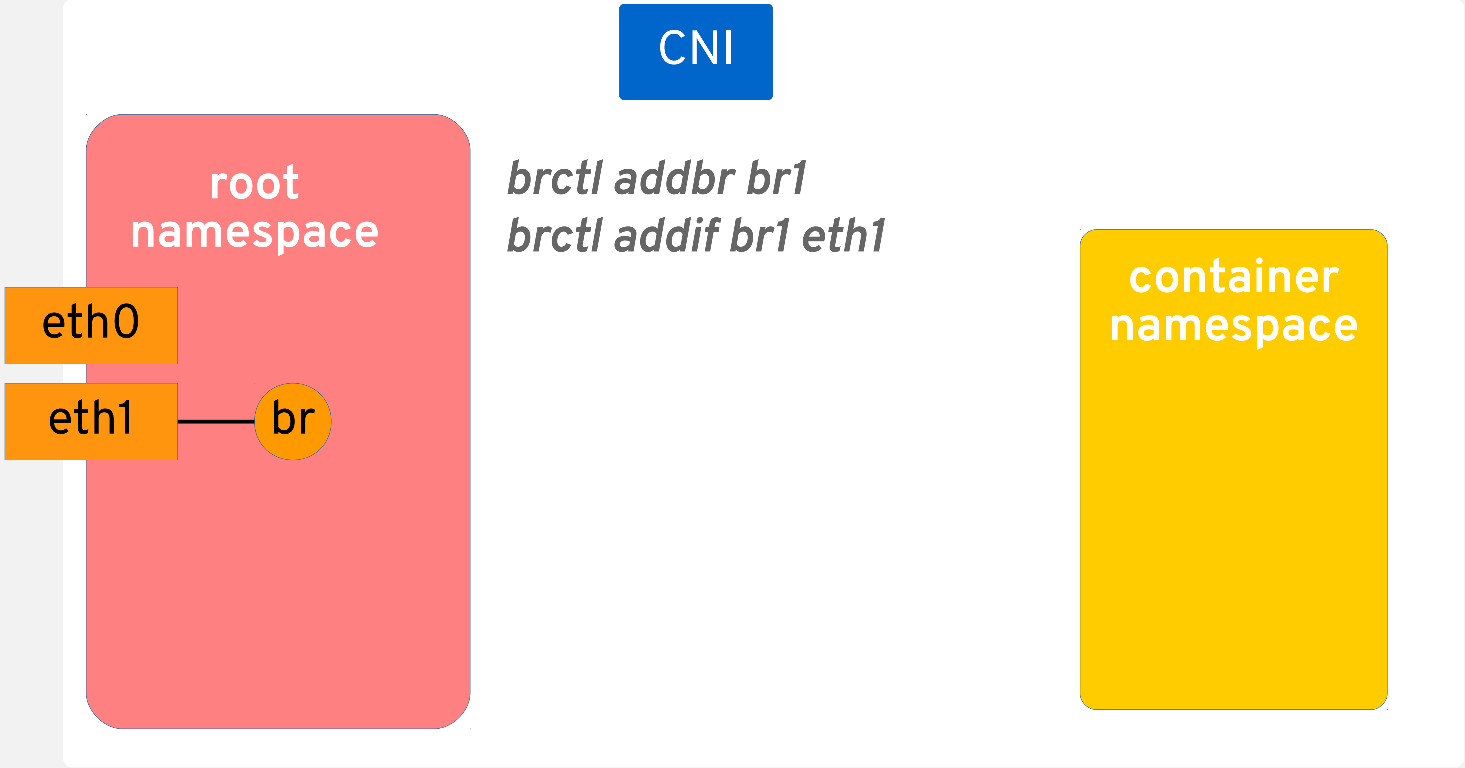

Host
CNI
Network configuration file
root namespace
eth0
eth1
container namespace

DEMO – Create a named network namespace



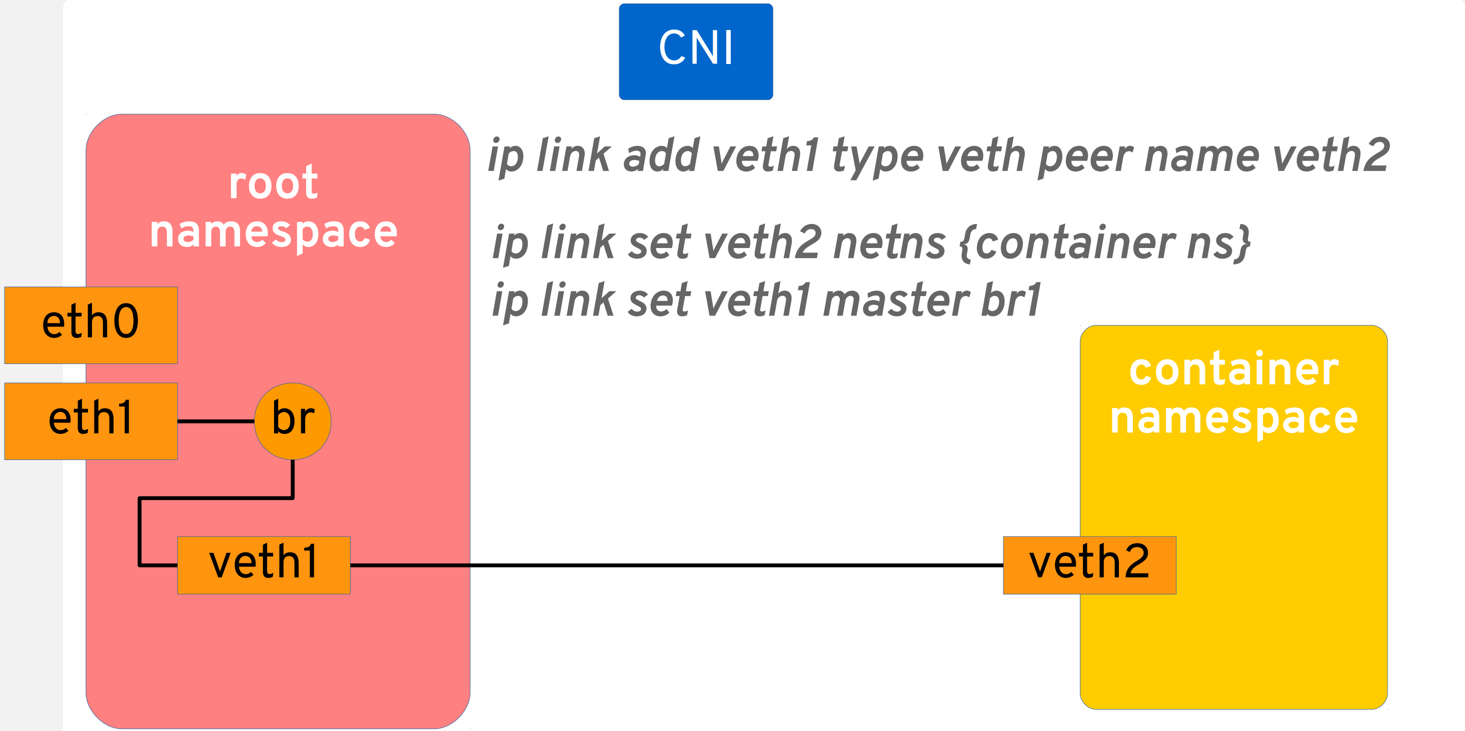
ln -sfT {cni_netns} /var/run/netns/{cni_containerid}

DEMO – Create a bridge



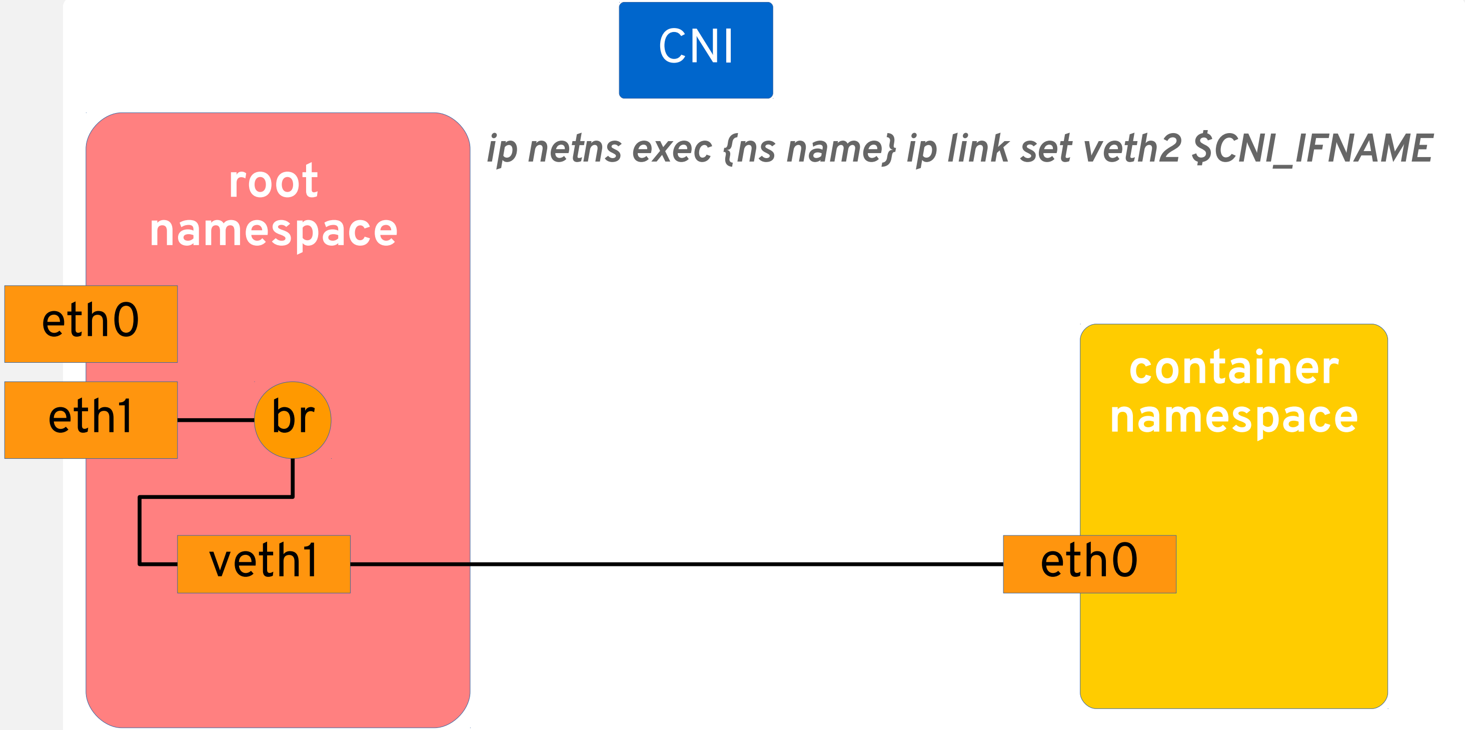
brctl addbr br1
brctl addif br1 eth1

DEMO - Connect the veth pair



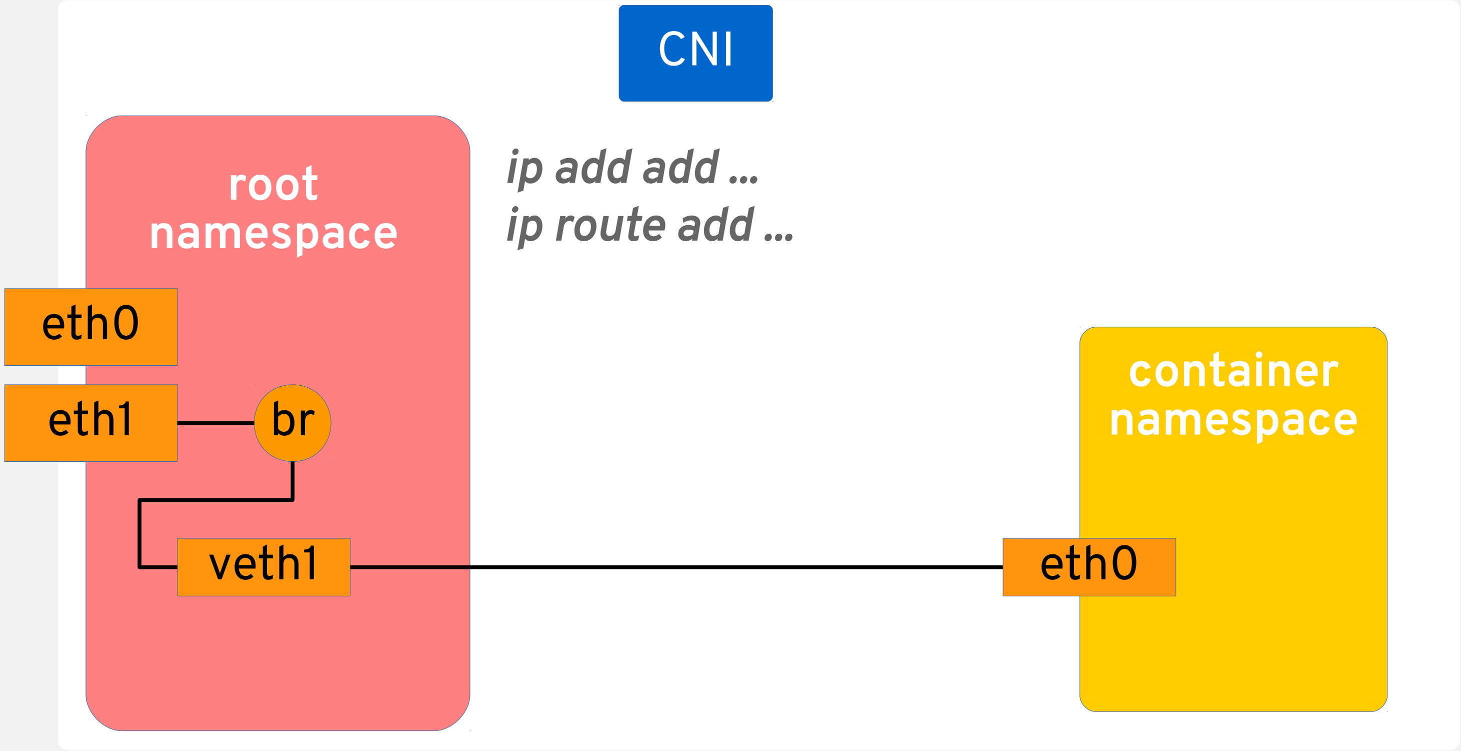
```
ip link add veth1 type veth peer name veth2

ip link set veth2 netns {container ns}
ip link set veth1 master br1
```

DEMO - Rename container interface



ip netns exec {ns name} ip link set veth2 $CNI_IFNAME

DEMO – Configure connection



```
ip add add ...
ip route add ...
```

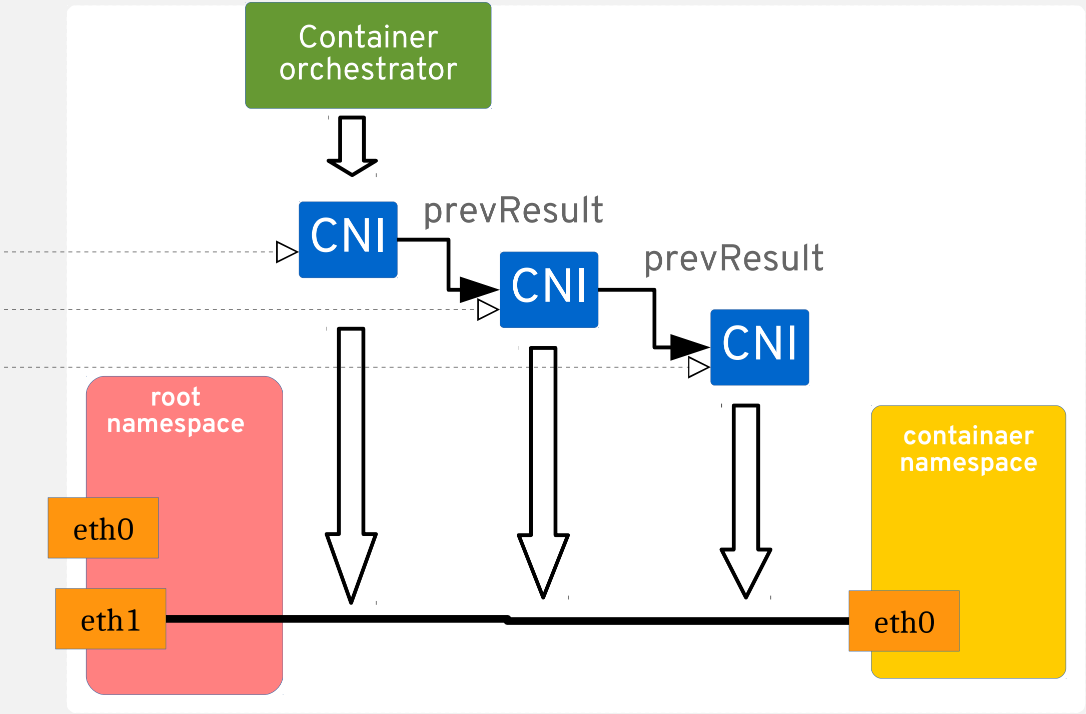
LIVE DEMO

Thank you

Backup slides

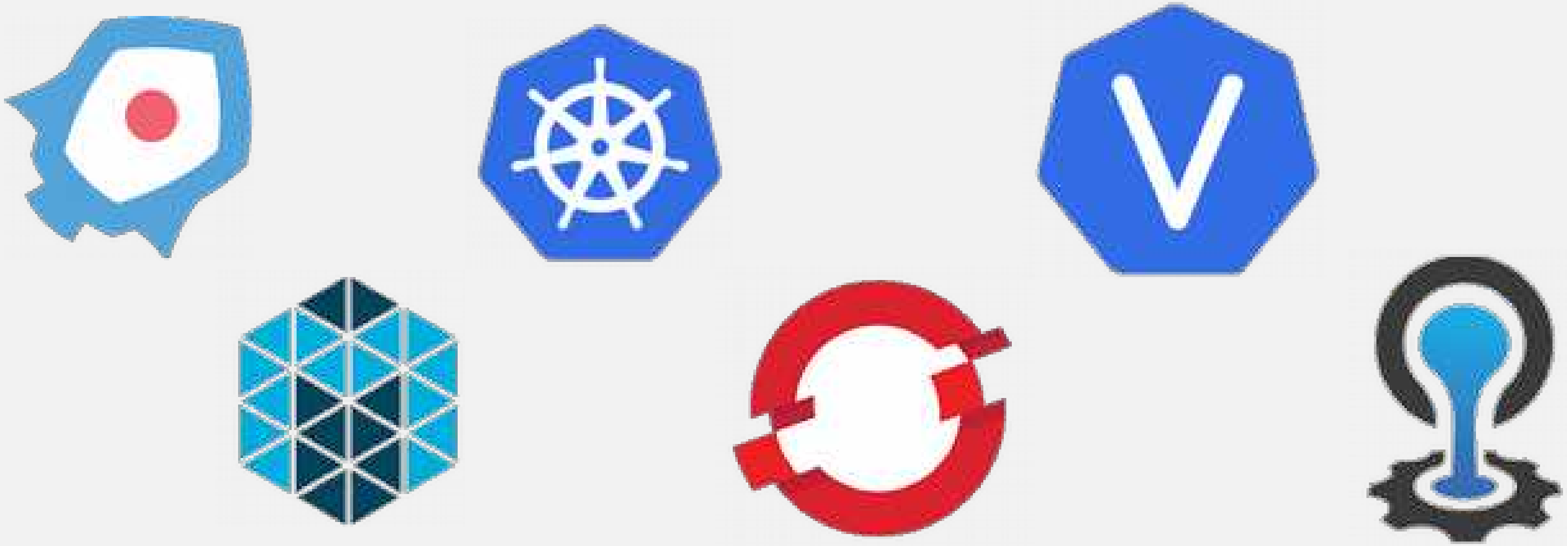
CNI plugin chaining

```
{
  ...
  "plugins": [
      {"type": A },
      {"type": B },
      {"type": C }
   ]
}
```



Why is it worth looking at?

Who is using CNI?



CNI network configuration – optional elements

```
{
    …
    "ipam": {
       "type": "plugin type"
    },
    "dns": {
       "nameservers": [],
       "domain": …,
       "search": [],
       "options": []
    }
}
```

Name of the IPAM plugin binary

CNI network configuration – example

```
{
    "cniVersion": "0.4.0",
    "name": "mynet",
    "type": "bridge",
    "bridge": "br0",
    "ipam": {
        "type": "ipam",
        "subnet": "10.1.0.0/16",
    },
    "dns": {
        "nameservers": [ "10.1.0.1" ]
    }
}
```

Plugin specific attributes

Sample net namespace magic …

ip netns add <name> - create a new net namespace named

ip link set dev eth0 netns <name> – move eth0 to namespace

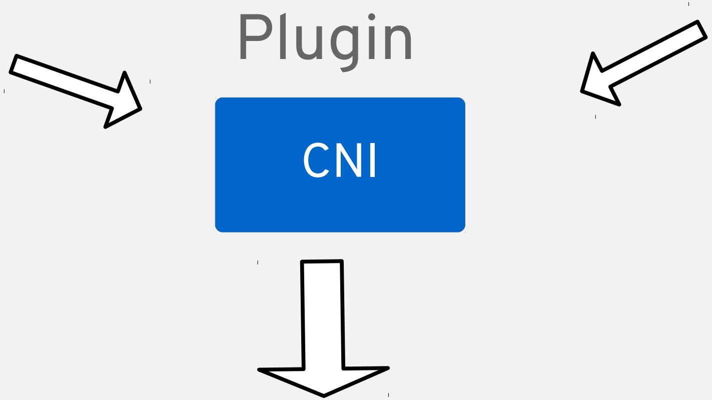
ip netns exec <name> ip link – list interfaces in namespace

ip netns exec <name> ip addr add …

CNI components

Network configuration

```
{
  "cniVersion": "0.4.0",
  "name": "myname",
  "type": "plugin type",
  "ipam": { ... },
  "dns": { ... },
  "additonalArg1": ... ,
  ...
}
```



Plugin

CNI

ACTION!

Runtime information

CNI_COMMAND
CNI_CONTAINERID
CNI_NETNS
CNI_IFNAME
CNI_ARGS
CNI_PATH