CNI Unleashed: how to deal with CNI plugin chains

February 5th, 2023, FOSDEM Network Dev Room

Daniel Mellado
d mellado@redhat.com
https://github.com/danielmellado

Miguel Duarte
mdbarroso@redhat.com
https://github.com/maiqueb
Agenda

- CNI intro
- CNI plugin chains
- Plugin chain use cases
- Demo
CNI basics
**CNI anatomy: from a Kubernetes perspective**

- **Kubernetes Node**
  - Container Runtime Implementation (crio/ocicni or docker/containerd)
    - libcni
    - CNI
  - Kubelet
  - CNI Binary Directory /opt/cni/bin
    - CNI Plugin
  - CNI Config Directory /etc/cni/net.d
    - CNI Config

- **Pod Sandbox**
  - Network namespace
    - eth0
  - Container
  - Container
## CNI operations

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ADD</strong></td>
<td>Add container to network, or apply modifications</td>
</tr>
<tr>
<td><strong>DEL</strong></td>
<td>Remove container from network, or un-apply modifications</td>
</tr>
<tr>
<td><strong>CHECK</strong></td>
<td>Check container's networking is as expected</td>
</tr>
<tr>
<td><strong>VERSION</strong></td>
<td>Check the spec for the exact format.</td>
</tr>
</tbody>
</table>

- **ADD**:
  - **ADD**
  - Add container to network, or apply modifications

- **DEL**:
  - **DEL**
  - Remove container from network, or un-apply modifications
  - Do garbage collection!

- **CHECK**:
  - **CHECK**
  - Check container's networking is as expected
  - Generally called right after pod creation succeeds. Exit non-zero if check doesn't succeed.

- **VERSION**:
  - **VERSION**
  - Probe plugin version support
  - Check the spec for the exact format.

[https://github.com/containernetworking/cni/blob/main/SPEC.md#cni-operations](https://github.com/containernetworking/cni/blob/main/SPEC.md#cni-operations)
CNI configuration

```json
{
    "cniVersion": "0.3.1",
    "name": "mynet",
    "type": "bridge",
    "bridge": "mynet0",
    "isDefaultGateway": true,
    "forceAddress": false,
    "ipMasq": true,
    "hairpinMode": true,
    "ipam": {
        "type": "host-local",
        "subnet": "10.10.0.0/16"
    }
}
```

- **Required.**
- **Plugin specific.**
- **IPAM is special, it's a "delegated plugin"**

It's all JSON.

It's "arbitrary" but required, may be helpful in logs.

This is a name of a binary on disk.
How does CNI find the binaries / configs?

Relevant parameters:

- `cni-conf-dir` => path to the CNI configuration
  - Defaults to `/etc/cni/net.d`
  - Smallest lexicographical order
- `cni-bin-dir` => path to CNI executables
  - Defaults to `/opt/cni/bin`

https://kubernetes.io/docs/concepts/extend-kubernetes/compute-storage-net/network-plugins/
CNI plugin chains
Chained plugins

- Adjust the configuration of an already-created interface
  - may need to create more interfaces to do so
- Available since CNI v0.3.0
- **Required** since CNI v1.0.0
- `.conflist` file extension when checking CNI configuration
  - `.conf` won’t work …
- When a meta plugin is passed a `prevResult`
  - **MUST** handle it: either passing it through, or modifying it appropriately
- **Delete considerations**
  - The list of plugins is executed in reverse order (add: x->y->z ; delete: z->y->x)
  - The previous result provided is always the final result of the add operation.
CNI configuration: chained plugins

```
{
   "cniVersion": "0.4.0",
   "name": "myconflist",
   "plugins": [
   {
      "name": "mynet",
      "type": "ptp",
      "ipam": {
         "type": "host-local",
         "subnet": "10.1.1.0/24"
      }
   },
   {
      "name": "mytuning",
      "type": "tuning",
      "sysctl": {
         "net.core.somaxconn": "500",
         "net.ipv4.conf.<IFNAME>.arp_filter": "1"
      }
   }
   ]
}
```
Use cases
Use cases

- **Tuning CNI**
  - Sysctl allow-list / sysctl button pusher
- **Bandwidth CNI**
  - Throttle ingress/egress traffic
- **Firewall**
  - allow traffic to/from container IP address
- **Port mapping**
- ...
Demo
CNI Tool: Your CNI swiss army knife

- Full tutorial / DIY workshop @
  https://dougbtv.com/nfvpe/2021/05/14/using-cnitool/
- It allows you to execute your plugins without having to launch a pod, cnitool calls your binary with the ENV variables and CNI configs.

```
./cnitool {...}
```

Environment variables

- CNI_COMMAND
- CNI_CONTAINERID
- CNI_IFNAME

STDIN

```
{
  "cniVersion": "0.3.1",
  "name": "mynet",
  "type": "bridge",
  "bridge": "mynet",
  "ipam": {}
}
```
https://github.com/maiqueb/fosdem2023-cni-unchained#debug-cni
Conclusions

● Plugins only useful when used in addition to other plugins => meta plugins
● Meta-plugins **enable** plenty of use cases
  ○ Prevent IP spoofing / bandwidth throttle / port-forward / configure sysctls /…
● Meta-plugins **must** handle the result of previous plugins in the chains
● Plugin chains are the **only allowed** CNI configuration from CNI v1.0.0
● Know your `prevResult`
Thank you! Questions ?...
CNI config example - calico

```json
{
    "name": "any_name",
    "cniVersion": "0.1.0",
    "type": "calico",
    "kubernetes": {
        "kubeconfig": "/path/to/kubeconfig"
    },
    "ipam": {
        "type": "calico-ipam"
    }
}
```

https://docs.projectcalico.org/reference/cni-plugin/configuration
CNI config example - calico

```
{
  "name": "any_name",
  "cniVersion": "0.1.0",
  "type": "calico",
  "kubernetes": {
    "kubeconfig": "/path/to/kubeconfig"
  },
  "ipam": {
    "type": "calico-ipam"
  }
}
```

$ podman exec node01 "ls -lah /opt/cni/bin"

```
-rw-r-xr-x. 1 root root  35M Nov 15 09:12 calico
-rw-r-xr-x. 1 root root  35M Nov 15 09:12 calico-ipam
```

https://docs.projectcalico.org/reference/cni-plugin/configuration
“Full” bandwidth

```json
{
  "cniVersion": "0.4.0",
  "name": "full-steam-ahead",
  "plugins": [
    {
      "type": "bridge",
      "bridge": "mynet0",
      "isDefaultGateway": true,
      "capabilities": { "ips": true },
      "ipam": {
        "type": "static"
      }
    }
  ]
}
```
Throttled bandwidth

```json
{
  "cniVersion": "0.4.0",
  "name": "limited-bandwidth",
  "plugins": [
    {
      "type": "bridge",
      "bridge": "mynet0",
      "isDefaultGateway": true,
      "capabilities": { "ips": true },
      "ipam": {
        "type": "static"
      }
    },
    {
      "type": "bandwidth",
      "ingressRate": 500000,
      "ingressBurst": 500000
    }
  ]
}
```