
KubeCon
CloudNativeCon
North America 2019

CNI Intro

Bryan Boreham, Weaveworks

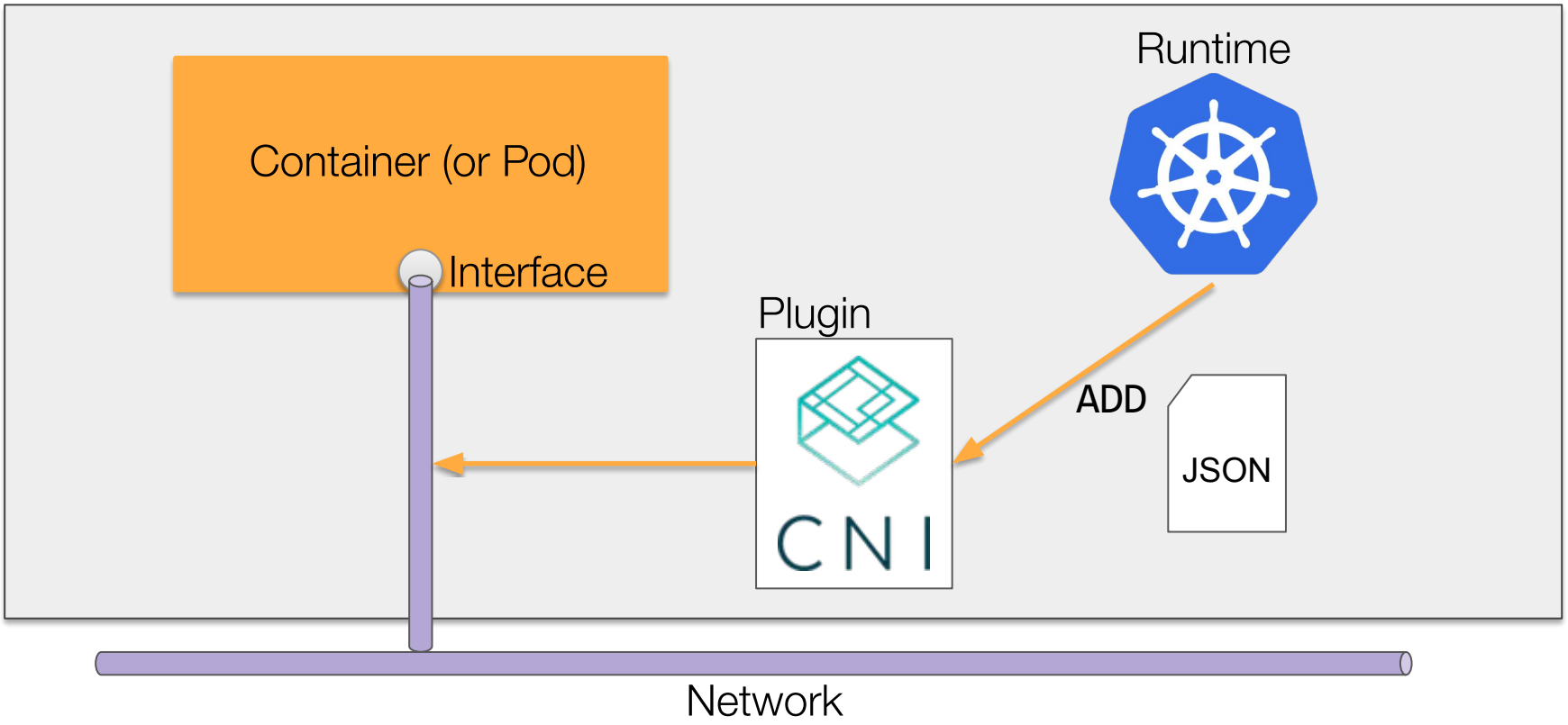
Dan Williams, Red Hat

Show of hands

Outline

- What is the CNI project?
- Who is the CNI project?
- How does the project work?
- What's new?
- How can I get involved?

How CNI fits in



What is the CNI project?

The CNI project has two major parts:

1. The CNI specification documents
   - libcni, a CNI runtime implementation
   - skel, a reference plugin implementation
   - github.com/containernetworking/cni
2. A set of reference and example plugins
   - Interface plugins: ptp, bridge, macvlan,...
   - “Chained” plugins: portmap, bandwidth, tuning
   - github.com/containernetworking/plugins

Specification

1. A vendor-neutral specification - not just for Kubernetes
2. Also used by Mesos, CloudFoundry, podman, CRI-O
3. Defines a basic execution flow & configuration format for network operations
4. Attempts to keep things simple and backwards compatible

Configuration Format

1. JSON-based configuration
2. Both standard keys and plugin-specific ones
3. Configuration fed to plugin on stdin for each operation
4. Stored on-disk or by the runtime

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

Execution Flow

1. Basic commands: ADD, DEL, CHECK and VERSION
2. Plugins are executables
3. Spawned by the runtime when network operations are desired
4. Fed JSON configuration via stdin
5. Also fed container-specific data via stdin
6. Report structured result via stdout

Reference and example plugins

- A set of common plugins that need a home.
- Main: bridge, loopback, vlan, macvlan, ipvlan, host-device, ptp, Windows bridge, Windows overlay
- IPAM: host-local, DHCP, static
- Meta: bandwidth, firewall, flannel, portmap, source-based routing, tuning

Who is the CNI project?

Seven maintainers:

- Bruce Ma (Alibaba)
- Bryan Boreham (Weaveworks)
- Casey Callendrello (IBM Red Hat)
- Dan Williams (IBM Red Hat)
- Gabe Rosenhouse (Pivotal)
- Matt Dupre (Tigera)
- Piotr Skamruk (CodiLime)

Lots of contributors!

What happened recently?

- Lots of work!
  - libcni 0.7.0 (April), 0.7.1 (June)
  - plugins 0.7.5 (March), 0.8.0 (May), 0.8.1 (June), 0.8.2 (Aug)
  - 275 commits, 47 contributors
- New maintainers!
- CHECK function
- Config and Result caching
- New plugins: bandwidth, firewall, sbr, static (IP address)
- Windows support: win-bridge, win-overlay
- Tuning plugin extended (promisc, mac, mtu)

How does the project work?

Spec:

- Actively maintained, but slow cadence
- Trying to hit 1.0 next year

Plugins:

- Faster release cadence
- Lots of contributors

Kubenet, kube-proxy, etc:

- Are in a different project. CNI works with, but is independent of Kubernetes

What's next?

- 1.0 is now feature-complete; the spec is stable
- We still need:
  - Complete test coverage
  - Spec review for clarity
  - Signed release binaries

How can I get involved?

- Slack - https://slack.cncf.io - topic #cni
- Github - for issues with the library or project plugins
  - Or for PRs!
- Make your own plugin! We can host a link to it.

Questions!

We will be at the booths from time to time:

Red Hat **#D1**

Weaveworks **#S51**