# CNI Deep Dive

CloudNativeCon December 2018

Bryan Boreham, Weaveworks
Dan Williams, Red Hat

# Outline

- What is the CNI project?
- Who is the CNI project?
- How does the project work?
- What’s being changed?
- How can I get involved?

# 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 “base” plugins
   - Interface plugins: ptp, bridge, macvlan,...
   - “Chained” plugins: portmap, bandwidth, tuning
   - github.com/containernetworking/plugins

# What is the CNI project: Specification

1. A vendor-neutral specification - not just for Kubernetes
2. Also used by Mesos & CloudFoundry
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, 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

# What is the CNI project: “base” plugins

A set of common plugins that need a home.

Main: bridge, macvlan, ipvlan, host-device, ptp, Windows

IPAM: host-local, DHCP, static

Meta: bandwidth, flannel, portmap, tuning

# Who is the CNI project?

Five maintainers:

- Bryan Boreham (Weaveworks)
- Casey Callendrello (Red Hat / CoreOS)
- Dan Williams (Red Hat)
- Gabe Rosenhouse (Pivotal)
- Matt Dupre (Tigera)

Lots of contributors!

# 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

# What happened recently?

Since libcni 0.6.0 (Aug 2017) and plugins 0.7.0 (Feb 2018):

- Added GET function
- Removed GET function
- Added CHECK function
- Traffic shaping plugin
- Static IP plugin
- Windows improvements

# What’s next?

- CHECK function
- 1.0:
  - Conformance test suite for CNI plugins (both reference and 3rd party)
  - Stable SPEC
  - Complete test coverage
  - Signed release binaries
- Exploring a gRPC interface

# How can I get involved

- Github
- Slack
- IRC
- KubeCon

# Questions!