KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon

North America 2019

CNI Deep Dive

Bryan Boreham, Weaveworks

Dan Williams, Red Hat

Outline

- Intro - brief recap of the CNI project
- Deep dives
  - CVE-2019-9946
  - CHECK function
  - Config and Result caching
- What’s being changed?
- Looking forward - CNI 2.0
- 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

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!

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

A story about a CVE

When you put many components together they can interact in unexpected ways.

CNI PortMap plugin inserted its iptables rules

Thus they go in front of firewall rules intended to block other traffic.

A story about CHECK

Check: the latest CNI_COMMAND. Since spec 0.4.0

- Asks the plugin: “Is your container's network healthy?”
- Hard-learned knowledge: if you kill Kubelet or Docker at exactly the wrong time, it will kill your CNI plugin partway through!
    - Dockershim currently assumes networking is up iff the pod has an IP on eth0
    - But what if you were killed between setting the IP and adding routes? This is a real-world bug.
    - Some end-users **love** to restart docker + kubelet constantly. Try not to do that.

How to get it into Kubernetes (etc)

- CHECK support already in CRI-O
- Need to add it to Kubernetes dockershim and CRI-ContainerD

Check details

Hey, you, $CNI_PLUGIN. I asked you to set up Pod A.

Do you think Pod A is still correctly set up?

- What does “correct” mean?
- What if a chained plugin changed stuff?

Check details

The spec language:

- The plugin should return an error if a resource included in the CNI Result type (interface, address or route) was created by the plugin, and is listed in prevResult, but is missing or in an invalid state.
- The plugin should return an error if other resources not tracked in the Result type, such as the following, are missing or are in an invalid state:
  - Firewall rules
  - Traffic shaping controls
  - IP reservations
  - External dependencies such as a daemon required for connectivity
  - etc.

Check details

Putting it together:

- On CHECK, runtime passes the Result from ADD. Validate against that.
- MUST handle case where a chained plugin “messed with stuff”

- There is a “skipCheck” safety valve for people doing really odd things
  - Use it at your peril!

Result caching

- Question: How can you CHECK if you don’t know what to CHECK for?
- Answer: pass the Result from ADD so you can CHECK against it
- Corollary: How can you DEL if you don’t know what to DEL?

Config caching

Configuration caching

- Question: how can you DEL if you changed your configuration after the ADD?
- Answer: use the original config from ADD so you know what to DEL

Use case

- Admins want to change the initial network plugin of a cluster
- Many CNI plugins are partly run as containers themselves
- This means changing the CNI plugin and configuration at runtime
- What if networked containers are already running?

Config and Results caching

- Caching of both config and Result done in libcni
- Offloads responsibility for doing this from runtimes like Kubernetes/dockershim, CRI-O, Multus, etc
- libcni handles reading the cached Result at the right times
- libcni also provides an interface for reading cached configuration

What’s next?

- 1.0:
  - 1.0 is now feature-complete
  -
  - Stable SPEC
  - Complete test coverage
  - Signed release binaries
  - Conformance test suite for CNI plugins (optional)
- 2.0
  - gRPC interface
  - Configuration using something better than files
  - Plugin drop-ins
  - Garbage-collection
  - What do you want to see?

How can I get involved

- Github
- Slack
- IRC
- KubeCon

Questions!

Thanks to Casey Callendrello for contributions to this presentation