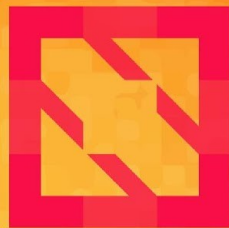




KubeCon



CloudNativeCon

North America 2019





KubeCon



CloudNativeCon

North America 2019

Multiple Networks for Kubernetes Workloads

Piotr Skamruk
CodiLime

Doug Smith
Red Hat



Agenda



KubeCon



CloudNativeCon

North America 2019

- Rationale for this presentation, short introduction to Kubernetes networking
- Review of CNI multiplexers:
 - CNI-Genie
 - Multus CNI
- Other approaches:
 - Nokia DANM
 - Tungsten Fabric, ZTE Knitter
- Few words about Network Attachments
- Few words about proposal of native Network object in Kubernetes

Why multiple networks?



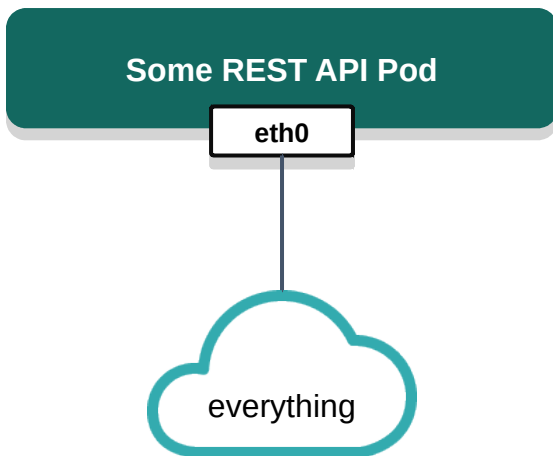
KubeCon



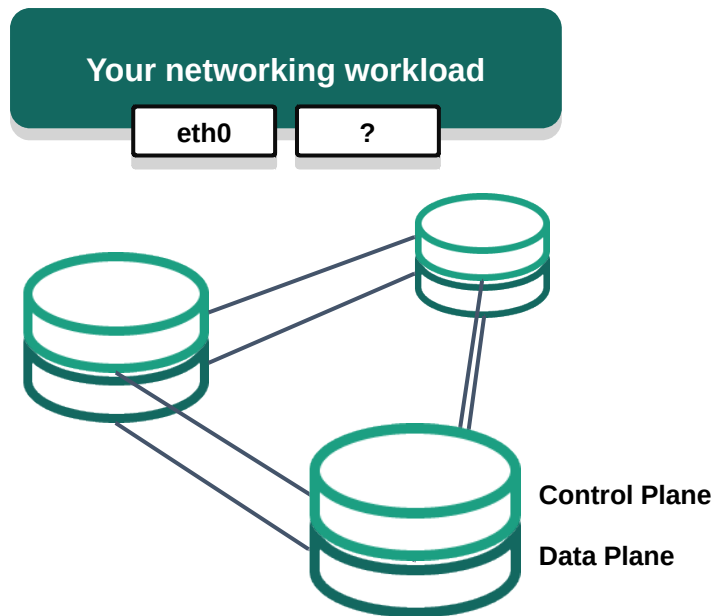
CloudNativeCon

North America 2019

It's straightforward for web scale.



But what about for performance networking workloads?



Kubernetes networking



KubeCon

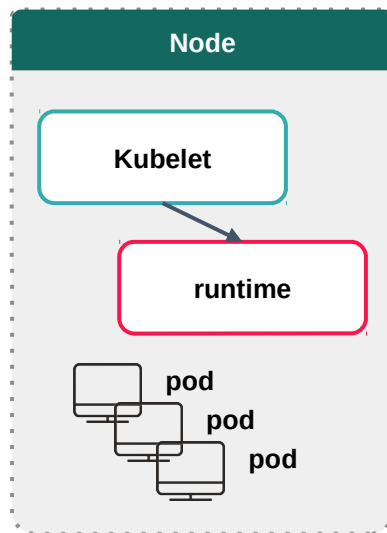
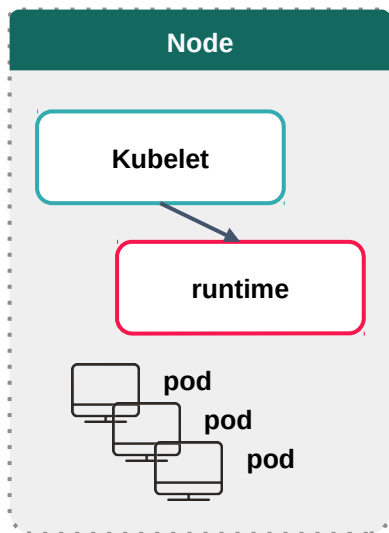


CloudNativeCon

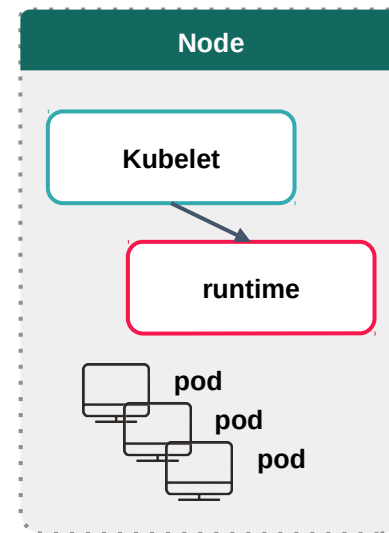
North America 2019



Control Plane



...



Kubernetes networking



KubeCon



CloudNativeCon

North America 2019

Assumptions:

- Each pod has own IP address
- Pod on a node can communicate with other pods/services in cluster without any NAT
- System services (kubelet, daemonsets) on a node can communicate with all pods on that node
- No extorsion for a particular solution - operator can decide which solution will fulfill his own assumptions
- Networking can be based on kubenet (using CNI plugin Bridge), or on CNI

A short introduction to CNI

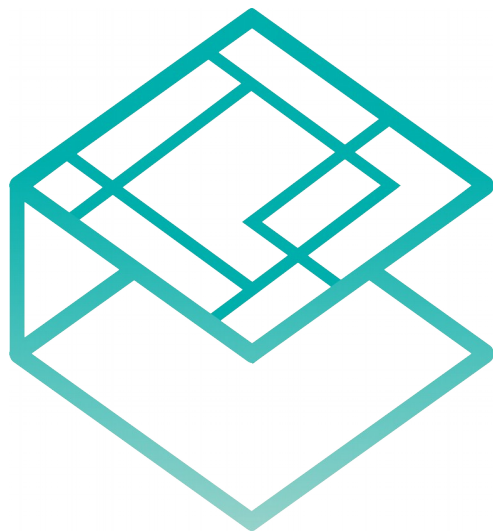


KubeCon



CloudNativeCon

North America 2019



- CNI (Container Network Interface) is a [Cloud Native Computing Foundation](#) project, consisting of a specification and libraries for writing plugins to configure network interfaces in Linux container
- CNI is container runtime agnostic -- it doesn't require Kubernetes.
- CNI is easy to implement as a developer -- it's basically as simple as having your application read/write STDOUT.
- CNI simplifies networking for Kubernetes, Kubernetes doesn't have to know the ins-and-outs of your networking -- you can have your plugins keep that knowledge.

Example CNI plugin: Bridge

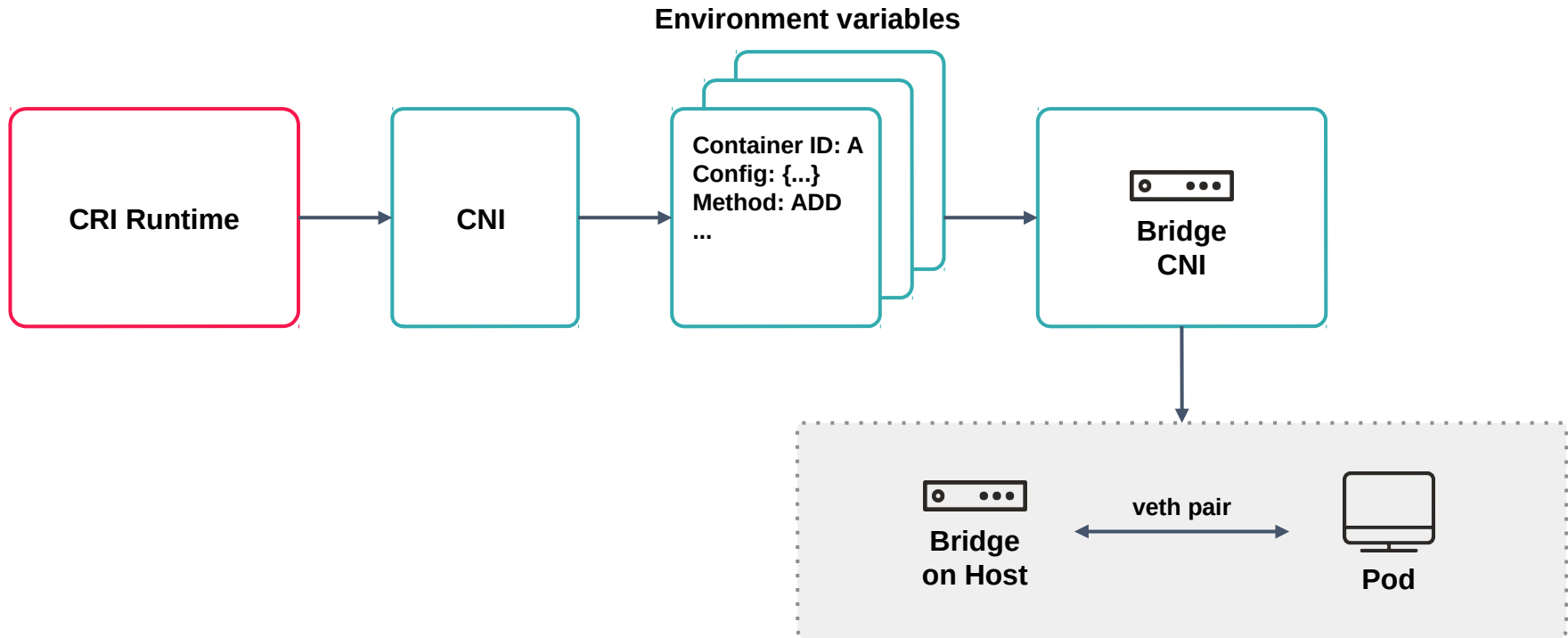


KubeCon



CloudNativeCon

North America 2019



A dissection of CNI configurations



KubeCon



CloudNativeCon

North America 2019

```
{
  "cniVersion": "0.4.0",
  "name": "dbnet",
  "type": "bridge",
  // type (plugin) specific
  "bridge": "cni0",
  "ipam": {
    "type": "host-local",
    // ipam specific
    "subnet": "10.1.0.0/16",
    "gateway": "10.1.0.1"
  },
  "dns": {
    "nameservers": [ "10.1.0.1" ]
  }
}
```

Type refers to the file name of the CNI plugin binary on disk that will be called.

Plugins can implement any arbitrary field name that's not already reserved, here the bridge CNI asks for the desired name of a bridge created on each node.

An optional secondary CNI plugin used for IP Address Management (IPAM) to customize how IP addresses are allocated.

CNI Multiplexers



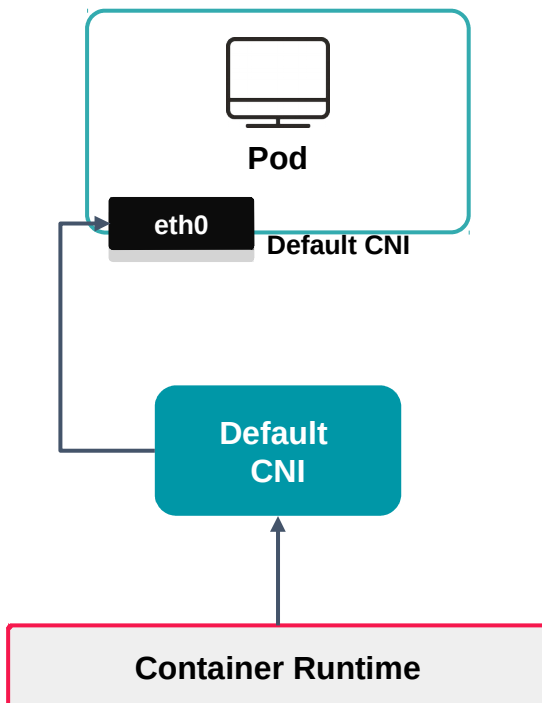
KubeCon



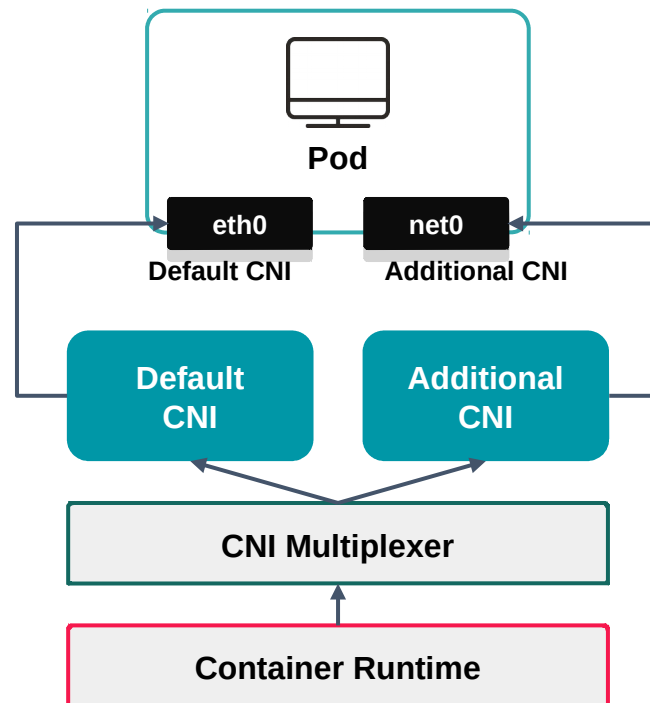
CloudNativeCon

North America 2019

Pod without Multiplexer



Pod with Multiplexer



Quick refresher: Custom Resources

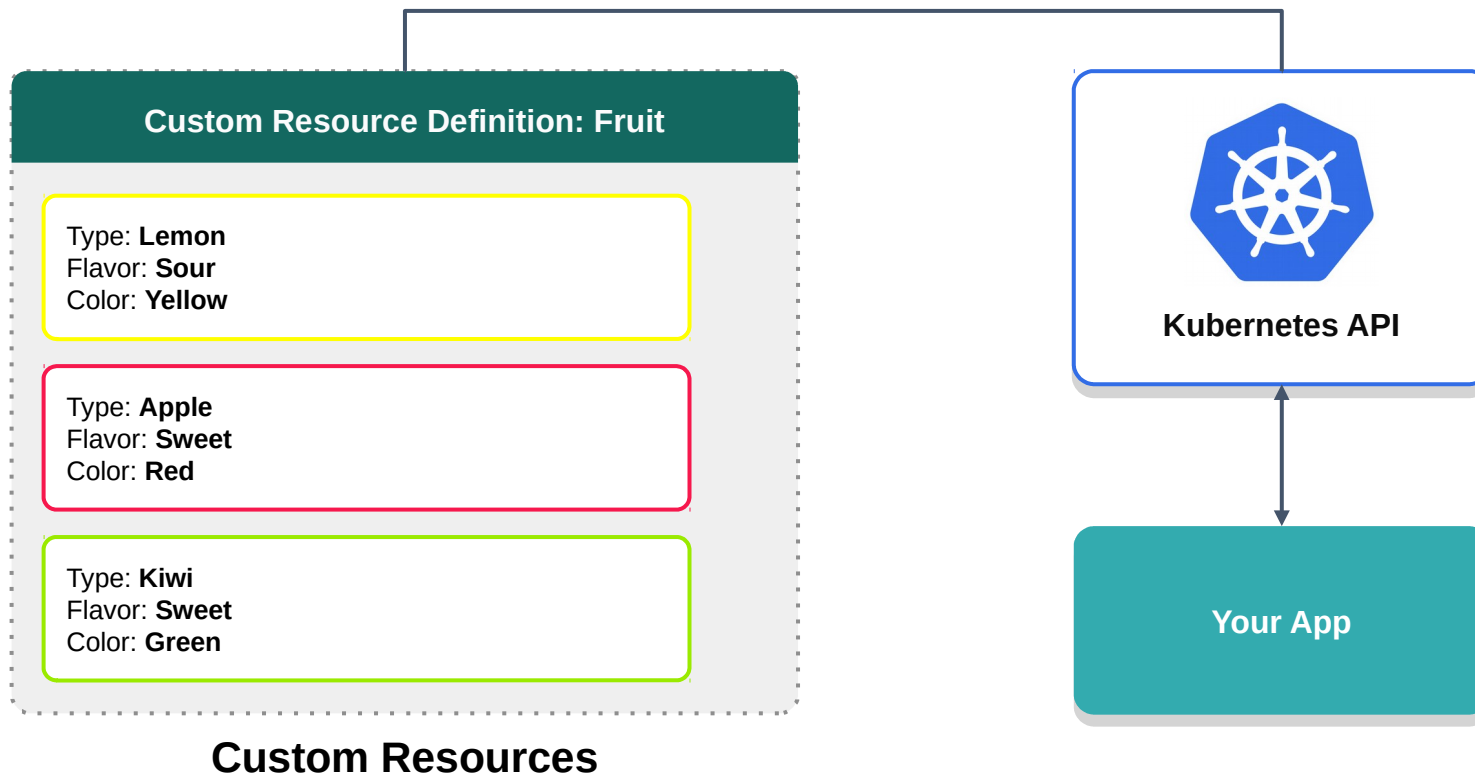


KubeCon



CloudNativeCon

North America 2019



NetworkAttachmentDefinitions



KubeCon



CloudNativeCon

North America 2019

The Network Plumbing Working Group's primary goal is to define a [Spec](#) that creates a standardized Custom Resource to express how you attach multiple networks to pods in Kubernetes.

Having a standardized custom resource helps to normalize the user experience should users change between technologies, or if platforms change the deployed technology to achieve multiple network attachments.

Learn more @ <https://github.com/k8snetworkplumbingwg/community>

Multus CNI



KubeCon



CloudNativeCon

North America 2019

```
apiVersion: "k8s.cni.cncf.io/v1"
kind: NetworkAttachmentDefinition
metadata:
  name: macvlan-conf
spec:
  config: '{
    "cniVersion": "0.3.0",
    "type": "macvlan",
    "master": "eth0",
    "mode": "bridge",
    "ipam": {
      "type": "host-local",
      "subnet": "192.168.1.0/24",
      "rangeStart": "192.168.1.200",
      "rangeEnd": "192.168.1.216",
      "routes": [
        { "dst": "0.0.0.0/0" }
      ],
      "gateway": "192.168.1.1"
    }
  }'
```



Is a custom resource, and uses NetworkAttachmentDefinitions as defined by the Network Plumbing Working Group

Annotations refer to this name

Contains a CNI configuration packaged in the spec.config field.

CNI-Genie - Simple case



KubeCon



CloudNativeCon

North America 2019

```
apiVersion: v1
kind: Pod
metadata:
  name: app-on-multiple-interfaces
  annotations:
    cni: "flannel,weave"
    multi-ip-preferences: |
      [
        "multi entry": 0,
        "ips": {
          "": {
            "ip": "",
            "interface": ""
          }
        }
      ]
```

```
spec:
  ...
```

Optional “cni” annotation - can hold list of configuration names, if empty - one would be chosen in “smart” way using cAdvisor data; if not defined - default plugin(s) will be used

Optional “multi-ip-preferences” annotation - will store information about ip addressing per selected configuration

CNI-Genie - Network attachments case



KubeCon

CloudNativeCon

North America 2019

```
apiVersion: v1
kind: Pod
metadata:
  name: app-on-multiple-interfaces
  annotations:
    k8s.v1.cni.cncf.io/networks:
flannel@eth1, customns/weavenet@eth2
    k8s.v1.cni.cncf.io/network-status: |-
  [
    ...
  ]
spec:
  ...
```

“networks” annotation - contains list of network attachments to select. Can be in form of string, or json list

“networks-status” annotation - will contain detailed info of attachments, including default attachment

Nokia DANM



KubeCon



CloudNativeCon

North America 2019



```
apiVersion: apps/v1beta1
kind: Deployment
metadata:
  name: test-deployment
  namespace: default
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: test-deployment
      annotations:
        danm.k8s.io/interfaces: |
          [
            {"tenantNetwork":"management",
"ip":"dynamic"},
            {"clusterNetwork":"external",
"ip":"dynamic"},
            {"tenantNetwork":"internal", "ip":"dynamic"}
          ]
    spec:
      containers:
        - name: busybox
          image: busybox:latest
          args:
            - sleep
            - "1000"
```

Higher level expression of what you'd like the interfaces to do.

More opinionated. Has more built in, but, limited in terms of flexibility.

Example use/configuration for <https://github.com/nokia/danm>

Other solutions



KubeCon



CloudNativeCon

North America 2019



Tungsten Fabric

- Main purpose: connecting multiple orchestration stacks like Kubernetes, Mesos/SMACK, OpenShift, OpenStack and VMware.
- It's also using network attachments, but not as meta plugin calling other normal plugins - but only knowing how to handle tungsten logic.



ZTE Knitter - it's speaking CNI protocol, it allows to connect multiple interfaces to a single pod, but its CNI responses are [faked, predefined, with same content all the time](#).

Network Plumbing Working Group



KubeCon



CloudNativeCon

North America 2019

- Formed @ Kubecon North America 2017
- Originally formed to address the “multi-networking problem” in Kubernetes in an out-of-tree fashion (that is, without changing the core Kubernetes code base)
- Meets every other week (opposite SIG-Network meetings) at 9:30 AM Eastern Standard Time
- Please come and join the effort!

Native K8S Network object



KubeCon



CloudNativeCon

North America 2019

- Proposal prepared and left for comments on NPWG
- What we can gain by adding to kubernetes new core object - Network
- Fallback to current simple model - “default” network
- How it can be handled on runtime level (name of CNI config).
- What can be easily improved later (exposing by CRI networks available for particular the node, exposing this info by kubelet e.g. in node status as a list of strings, using this info by scheduler during pods allocation on nodes)

- <https://github.com/intel/multus-cni/>
- <https://github.com/cni-genie/CNI-Genie>
- <https://github.com/nokia/danm>
- <https://github.com/ZTE/Knitter>
- <https://tungsten.io/>
- <https://www.youtube.com/watch?v=-lZzcq9aZg4>
- [Network Attachments Spec](#)





KubeCon



CloudNativeCon

North America 2019

Questions & Answers





KubeCon



CloudNativeCon

North America 2019

Thank You

