

# Practical CNI

Introduction to the project and how to get hands dirty with it

**Lorenzo Fontana - @fntlnz**

DevOps @ Kiratech

Docker Captain - Docker Maintainer

<https://fntlnz.wtf>

**Tuesday**, June 20

 **LINUXCON**

**containercon**

 **CLOUDOPEN**

CHINA 2017

# A brief introduction

- A runtime independent platform for container networking
- Part of CNCF (well, since May 2017) <sup>1</sup>
- An effort held by multiple entities: CoreOS, Red Hat OpenShift, Apache Mesos, Cloud Foundry, Kubernetes, Kurma and rkt.
- Designed around a minimal specification



# The SPEC<sup>1</sup> - and its purpose

- Defines the interactions between “runtimes” and “plugins”
- The interactions are regulated by known fields or by custom fields by following conventions <sup>2</sup>
- The **interactions** are drove by two important definitions:
  - **Container:** in this context, can be considered a synonymous of a linux network namespace, the real unit of definition depends on the **particular runtime implementation**
  - **Network:** refers to a group of entities that are uniquely addressable and can communicate amongst each other, like: network devices, a container, etc..

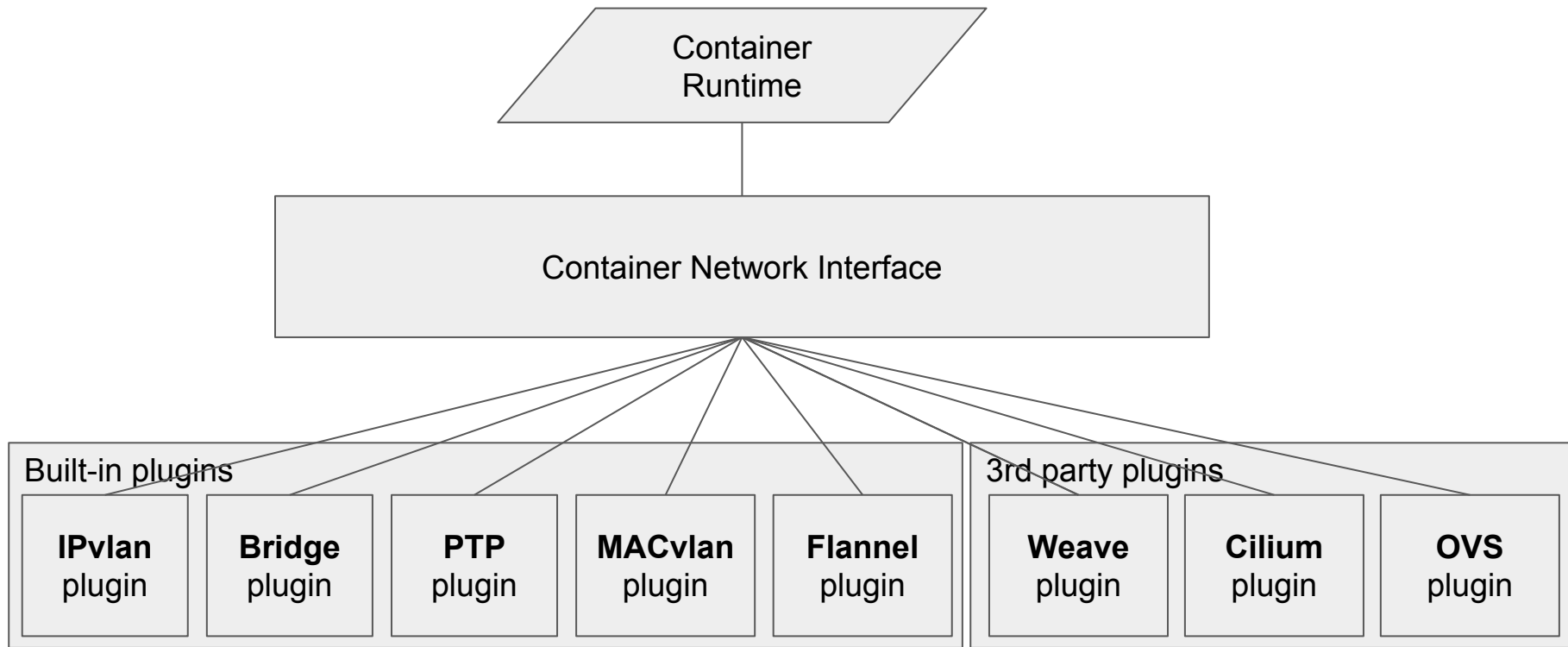
<sup>1</sup> <https://github.com/containernetworking/cni/blob/master/SPEC.md>

<sup>2</sup> <https://github.com/containernetworking/cni/blob/master/CONVENTIONS.md>

# Wait, Plugins ?

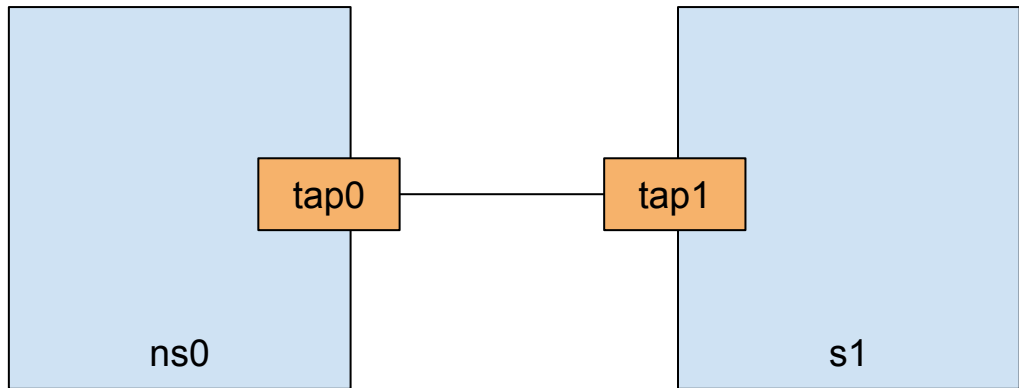
- Plugins are executables **invoked** by the *container runtime*
- Plugins are **responsible** for:
  - IPAM
  - Connecting **VETH Pairs**
  - Adding necessary network components, like **bridges**

# CNI Overview



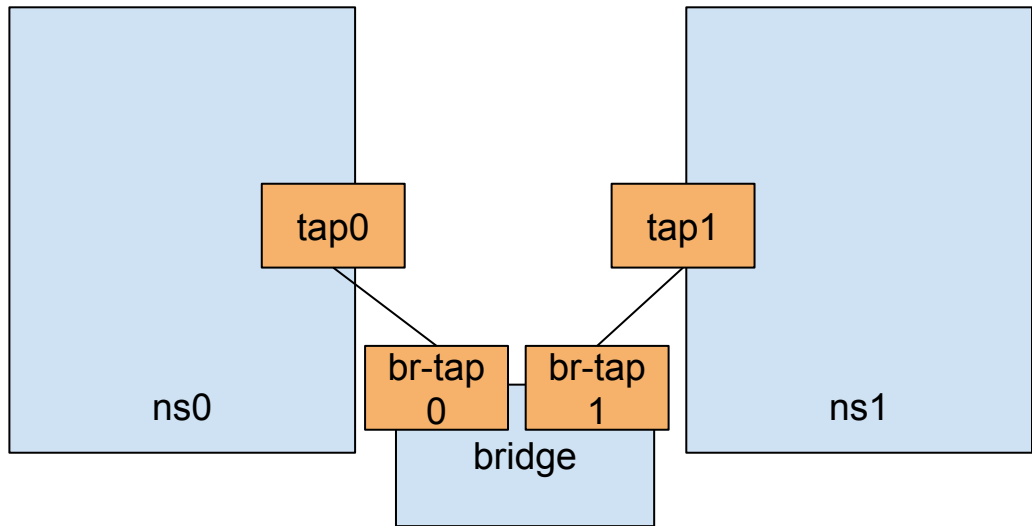
# The “ptp” Plugin

```
{  
  "name": "mynet",  
  "type": "ptp",  
  "ipam": {  
    "type": "host-local",  
    "subnet": "10.1.1.0/24"  
  },  
  "dns": {  
    "nameservers": [ "10.1.1.1", "8.8.8.8" ]  
  }  
}
```



# The “bridge” Plugin

```
{  
  "name": "imthebestnetever",  
  "type": "bridge",  
  "bridge": "myawesomebr",  
  "isDefaultGateway": true,  
  "ipMasq": true,  
  "hairpinMode": true,  
  "ipam": {  
    "type": "host-local",  
    "subnet": "10.22.0.0/16"  
  }  
}
```



# Lifecycle of a CNI Bridge

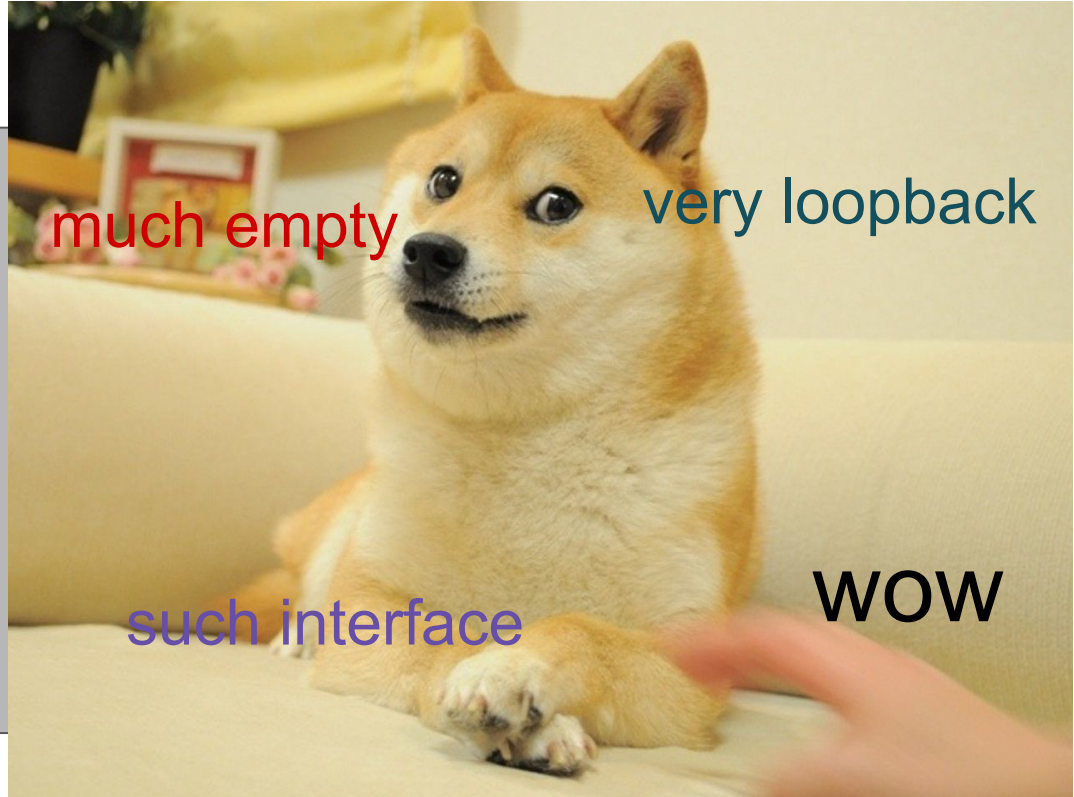
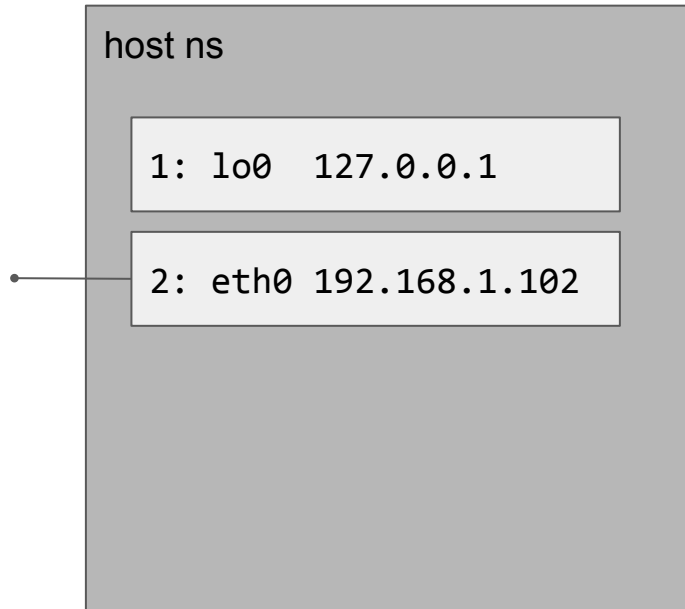
host ns

1: lo0 127.0.0.1

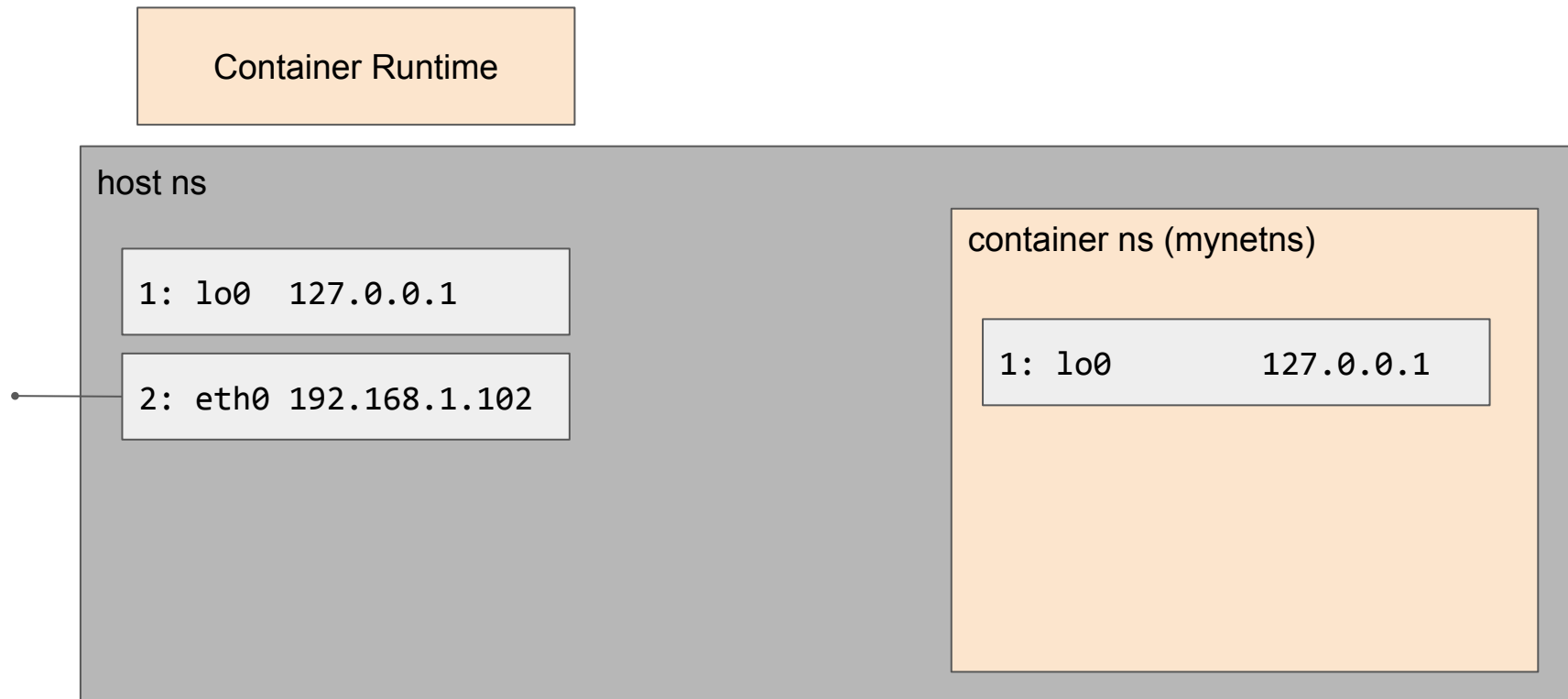
2: eth0 192.168.1.102



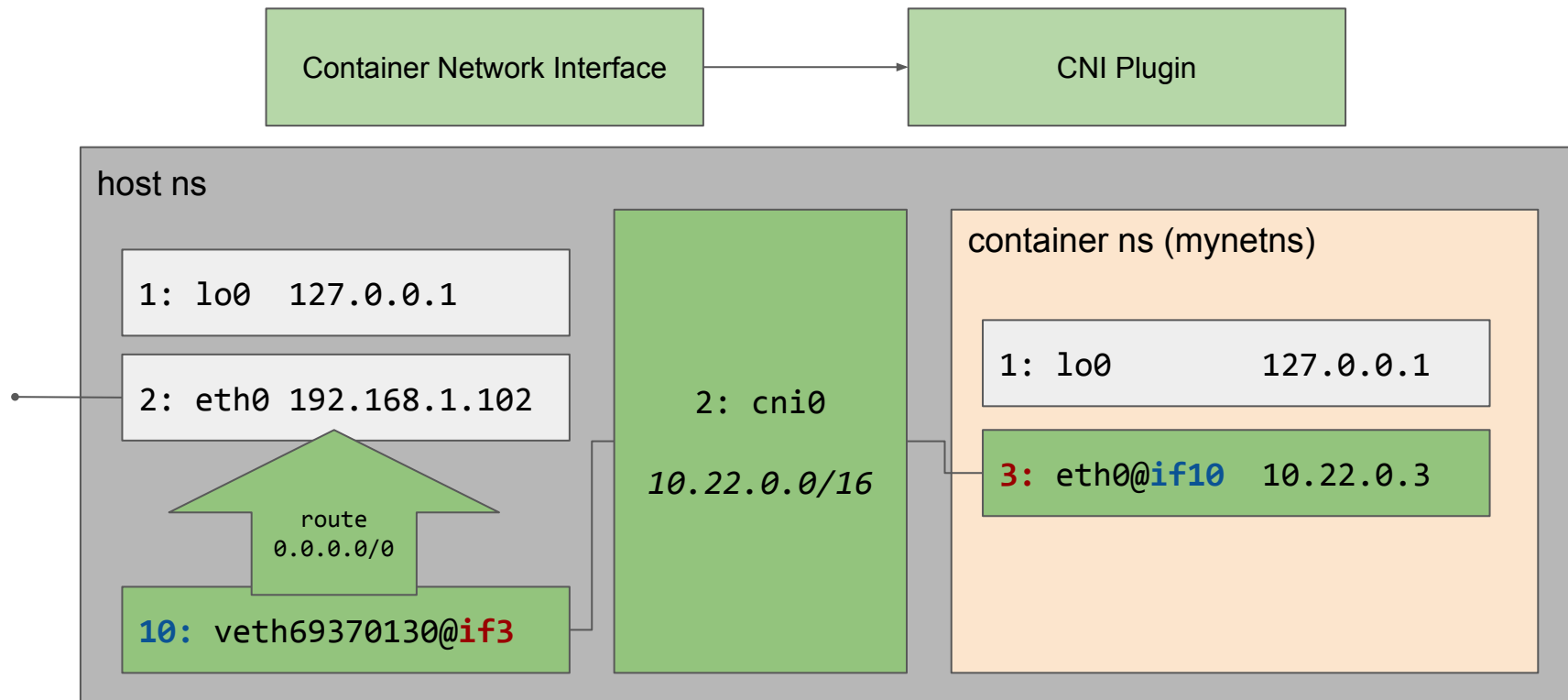
# Lifecycle of a CNI Bridge



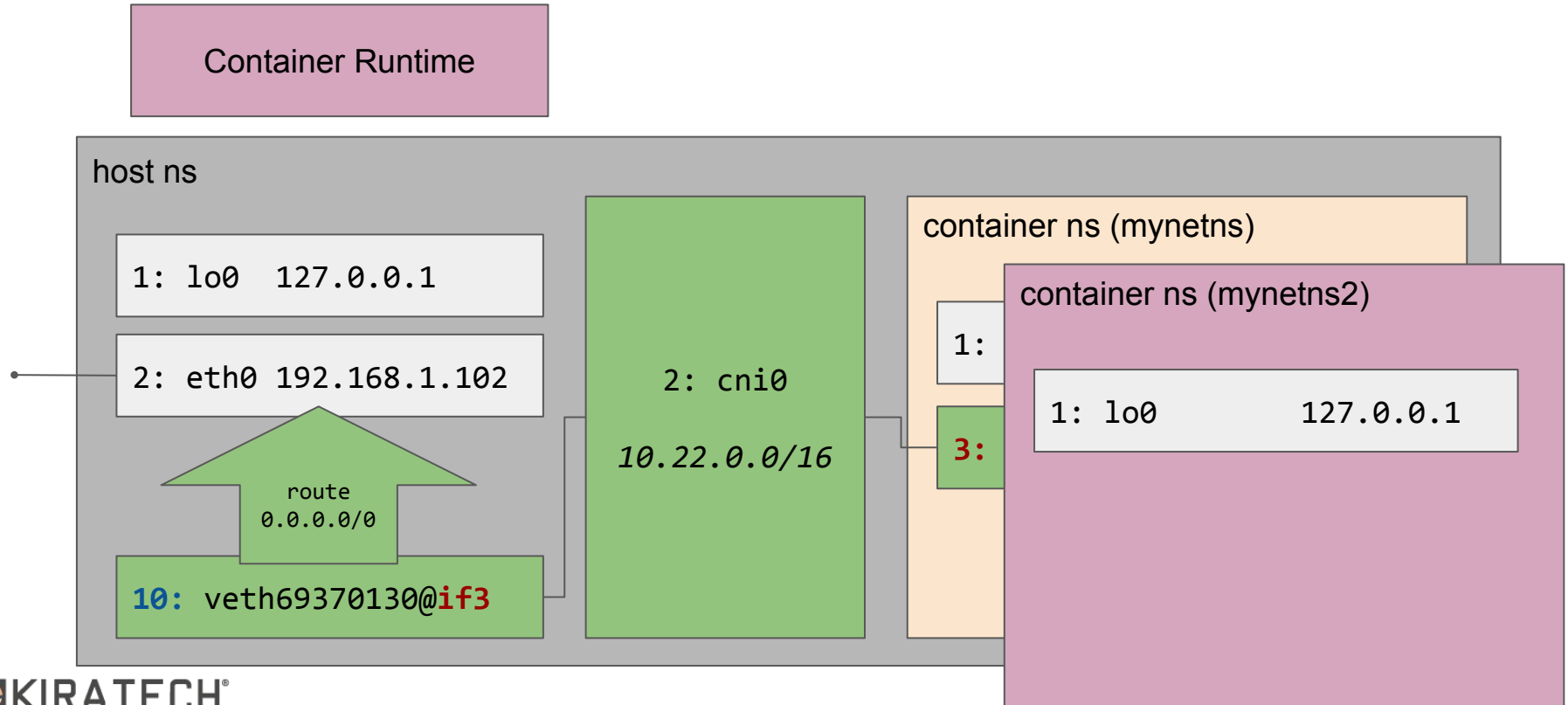
# Lifecycle of a CNI Bridge



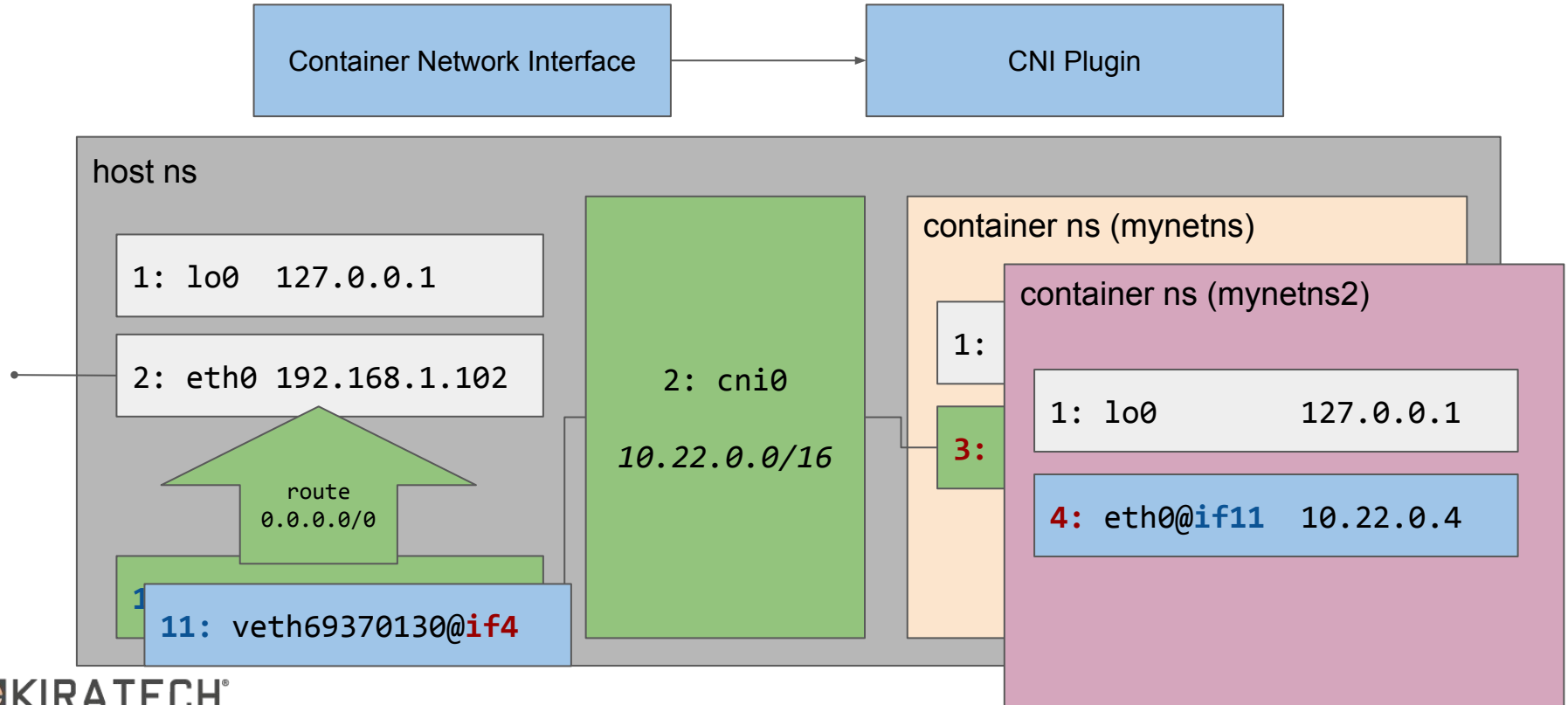
# Lifecycle of a CNI Bridge



# Lifecycle of a CNI Bridge

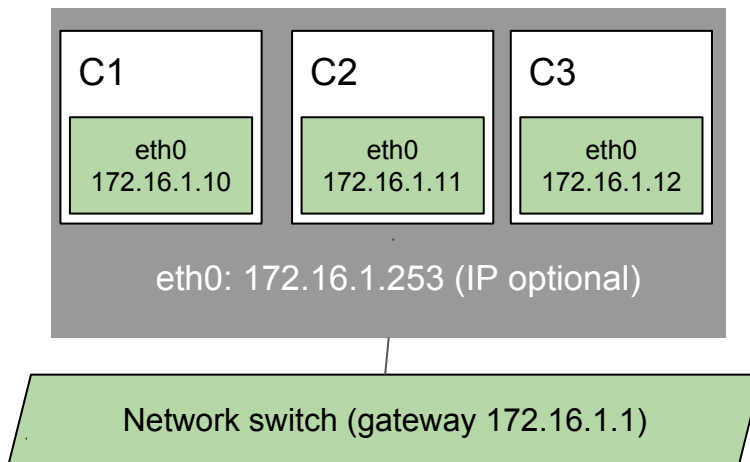


# Lifecycle of a CNI Bridge



# The “IPvlan” Plugin

```
{  
  "name": "mynet",  
  "type": "ipvlan",  
  "master": "eth0",  
  "ipam": {  
    "type": "host-local",  
    "subnet": "10.1.2.0/24",  
  }  
}
```



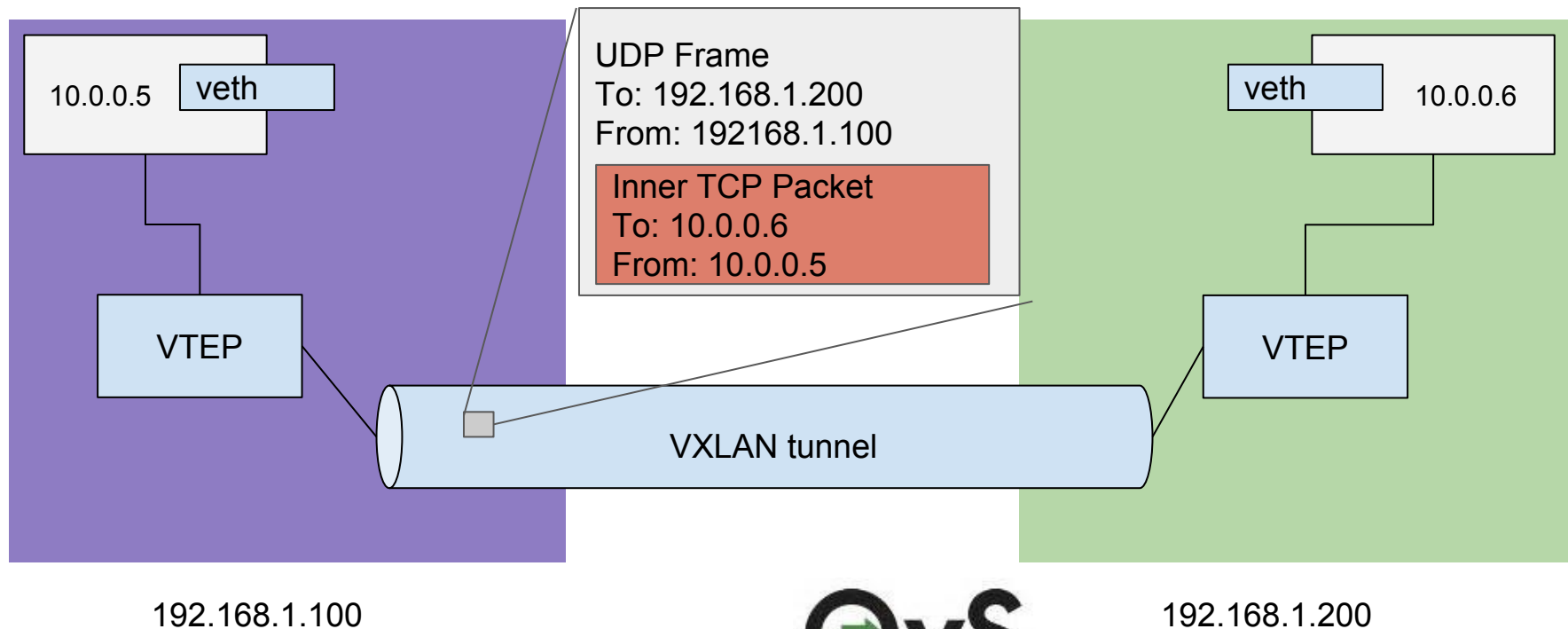
And you will need at least a PTP too, since IPvlan does not allow you to connect to the host otherwise

<https://github.com/torvalds/linux/tree/master/drivers/net/ipvlan>

<https://github.com/containernetworking/cni/blob/master/Documentation/ipvlan.md>

<https://lwn.net/Articles/620087/>

# Overlay networking



# OVN

- L2/L3 network virtualization for Open vSwitch
- Integrated with Container Orchestrators <sup>1</sup>
- Integrated with hypervisors (KVM, XEN, Hyper-V)
- Integrated with OpenStack
- Integrated DHCP
- Integrated QoS
- Trunking
- Scalable
- Multiple overlay technologies (Geneve, STT and VXLAN)
- Means hybrid networks between infrastructures (hybrid / cloud) and different virtualization technologies (OS Level: containers and Full: Virtual machines)



# Thank you!



Find me on

<https://fntlnz.wtf>

<https://github.com/fntlnz>

<https://twitter.com/fntlnz>

<http://www.kiratech.it/china>

lo@linux.com

 **LINUXCON**

containercon

 **CLOUDOPEN**

CHINA 2017