Leveraging Kubernetes for NFV

José María Roldán Gil

14/03/2019

t3chfest

Agenda

- 5G Networks
- 5G-CORAL project
- Kubernetes for NFV
- Demo

Who am I: José María Roldán Gil

- Telecommunications Engineer

Internet of Things
Sensing Assets

Telcaria

jmroldangil@gmail.com
josrolgil

Next technological revolution is coming: 5G

Two keys in 5G networks

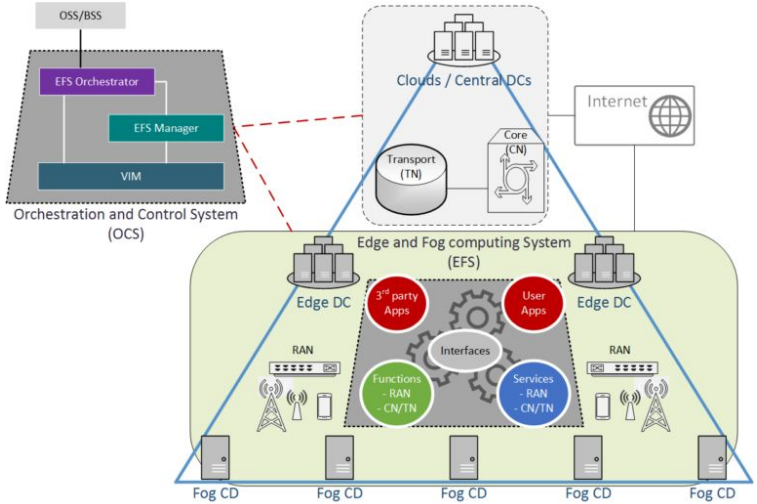
Software Defined Networks

Network Function Virtualization

5G-CORAL

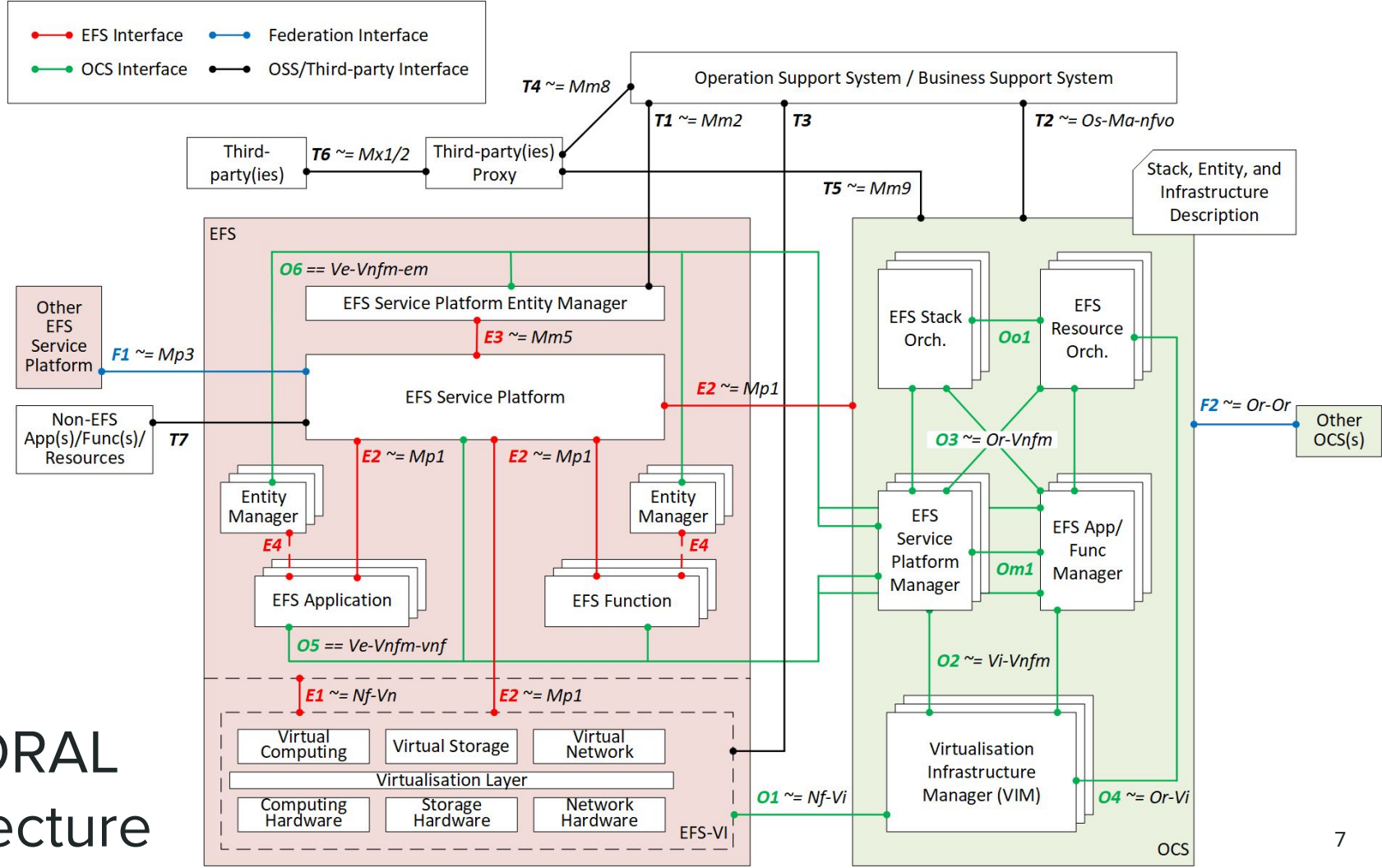
A 5G Convergent Virtualized Radio Access Network Living at the Edge

- From cloud computing to fog computing



Convergent 5G multi-RAT access through an integrated virtualized edge and fog solution

5G-CORAL Architecture

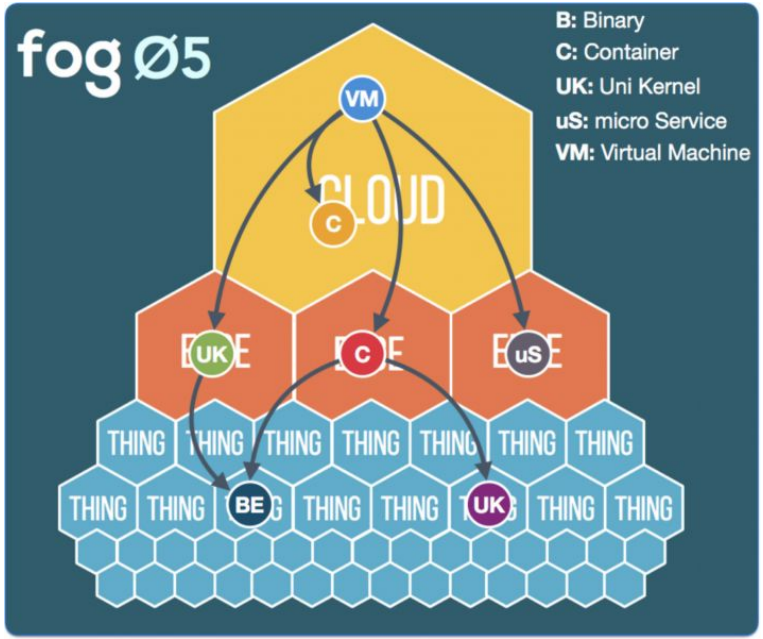


5G-CORAL OCS

Implementation option I: fog05

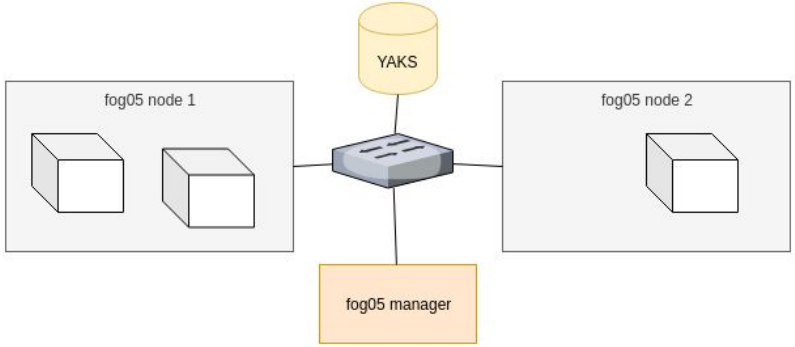
https://github.com/eclipse/fog05

Fog computing: a system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum



5G-CORAL OCS

Implementation option I: fog05




```
{“name”: “function”,
  “version”: 1,
  “type”: “lxd”,
  “uuid”: “690c0d9f-e88b-41d3-837b-eca2d32c7859”,
  “entity_data”: {
      “name”: “ubuntu-host”,
      “uuid”: “ccae85ec-3f87-497b-bd6f-cf1990b5577a”,
      “cpu”: 1,
      “memory”: 512,
      “disk_size”: 4,
      “base_image”: “file:///etc/host.tar.gz”
      “networks”: [{“intf_name: “eth0”, “br_name”:”lan01”}]
    }
}
```

```
api = API(endpoint=’10.10.1.2’)
manifest = json.loads(read_file(path))
...
api.entity.define(manifest, node1, wait=True)
api.entity.configure(euuid, iuuid, wait=True)
api.entity.run(euuid, node1, iuuid, wait=True)
...
api.entity.stop(euuid, node1, iuuid, wait=True)
api.entity.clean(euuid, node1, iuud, wait=True)
api.entity.undefine(euuid, node1, iuud, wait=True)
```

5G-CORAL OCS

Implementation option II: Kubernetes

Kubernetes is an open source system for

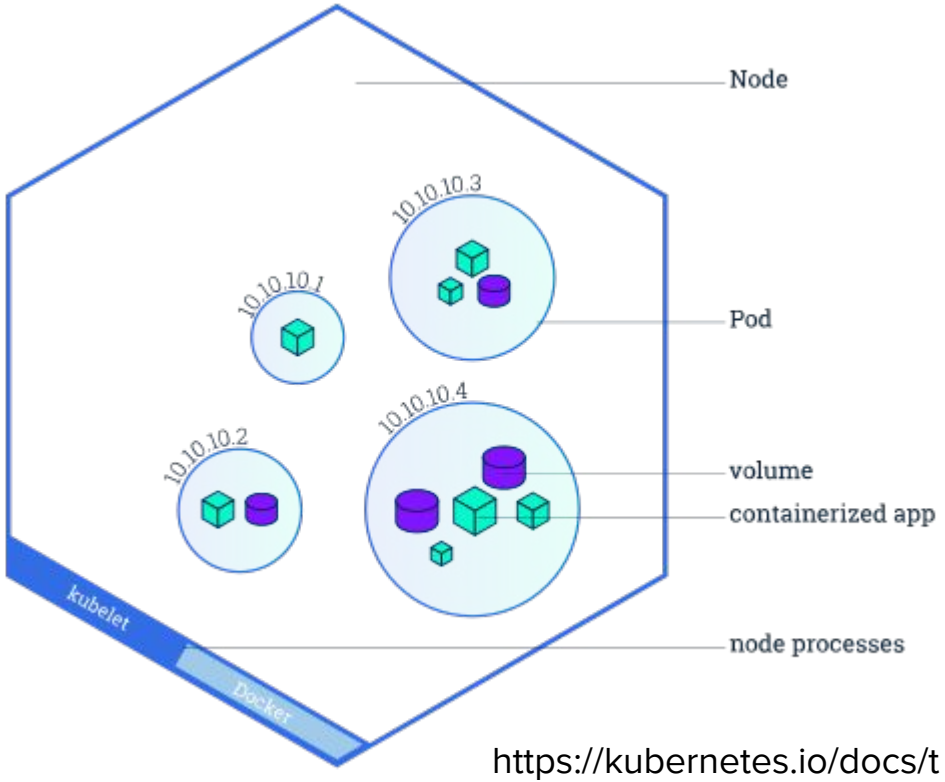
automatic deployment, scaling and management

of containerized applications.

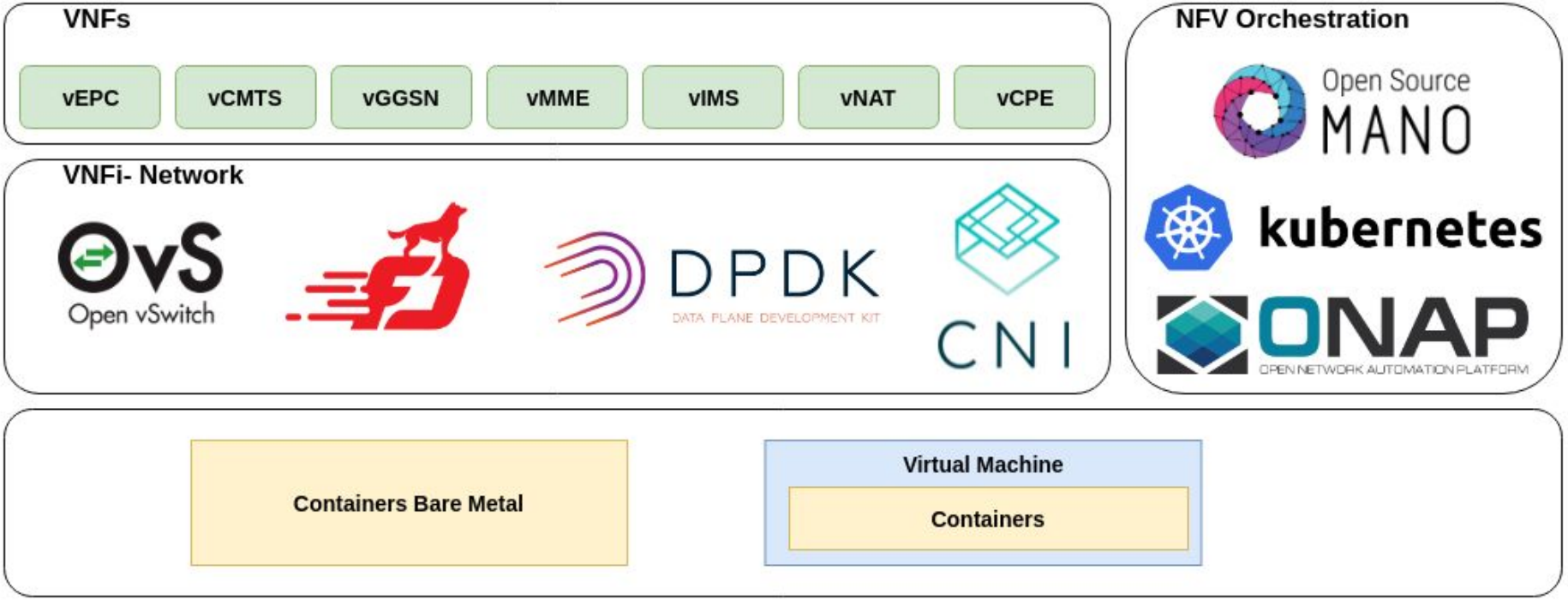
Intended for microservices

Kubernetes basics

One IP address assigned to a pod
The container has one network interface



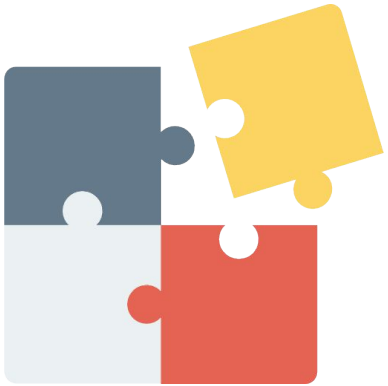
https://kubernetes.io/docs/tutorials/kubernetes-basics/explore/explore-intro/


VNFs
vEPC
vCMTS
vGGSN
vMME
vIMS
vNAT
vCPE
NFV Orchestration
Open Source
MANO
VNFi- Network
OvS
Open vSwitch
DPDK
DATA PLANE DEVELOPMENT KIT
CNI
kubernetes
ONAP
OPEN NETWORK AUTOMATION PLATFORM
Containers Bare Metal
Virtual Machine
Containers

Kubernetes for NFV

When using a container as a virtualized network function:

- Kubernetes supports only one network interface
- Need of different network interfaces
    - Separation of control and data plane
    - Network segregation and security
    - Link aggregation for redundancy

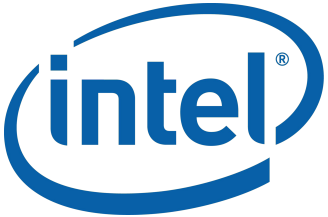


Multus

Multus CNI is a container network interface (CNI) plugin for Kubernetes that enables attaching multiple network interfaces to pods.

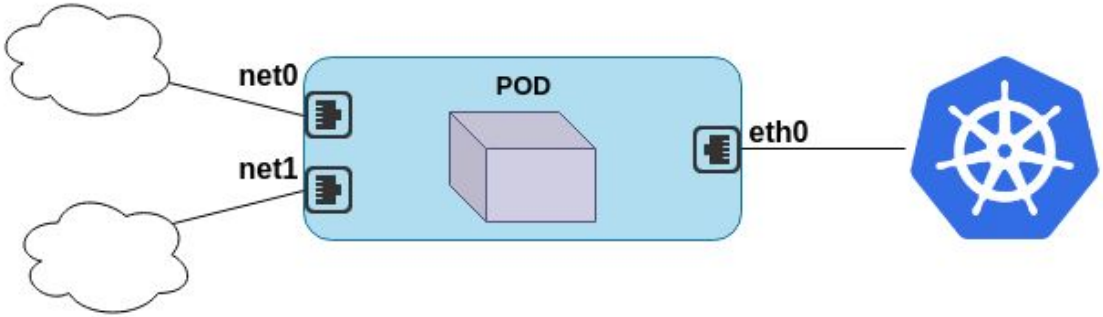
It’s a CNI plugin than can

call other CNI plugins



https://github.com/intel/multus-cni

Multus



Author: multus

Container Network Interface

CNI is a Cloud Native Computing Foundation project

Specification and libraries for writing plugins to configure network, along with a number of supported plugins

https://github.com/containernetworking

Some standard network plugins

Bridge

- The container is plugged to a virtual bridge

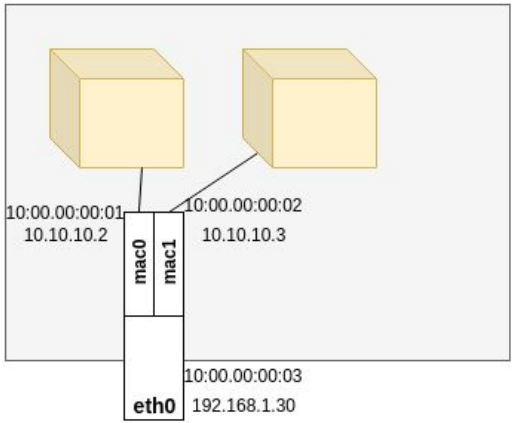
```
{
    "name": "mynet",
    "type": "bridge",
    "bridge": "mynet0",
    "isDefaultGateway": true,
    "forceAddress": false,
    "ipMasq": true,
    "hairpinMode": true,
    "ipam": {
           "type": "host-local",
           "subnet": "10.10.0.0/16"
    }
}
```

Some standard network plugins II

Macvlan

- A host interface gets enslaved with the virtual interface
- Same physical device but distinct MAC address

```
{
        "name": "mynet",
        "type": "macvlan",
        "master": "eth0",
        "ipam": {
                "type": "dhcp"
        }
}
```

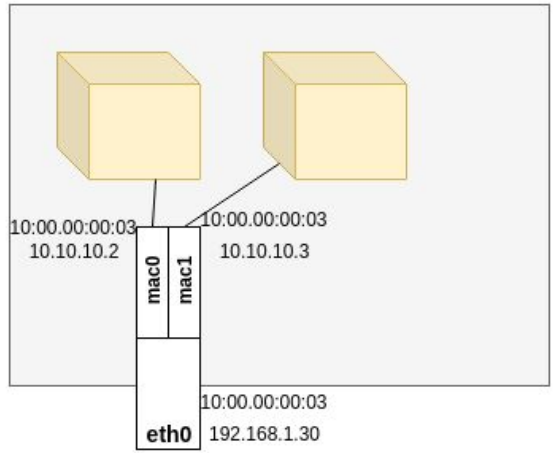


Some standard network plugins II

IPvlan

- A host interface gets enslaved with the virtual interface
- Same physical device and same MAC address
- The kernel driver inspect packages when deciding which interface should process it

```
{
        "name": "mynet",
        "type": "ipvlan",
        "master": "eth0",
        "ipam": {
                "type": "host-local",
                "subnet": "10.10.10.0/24"
        }
}
```



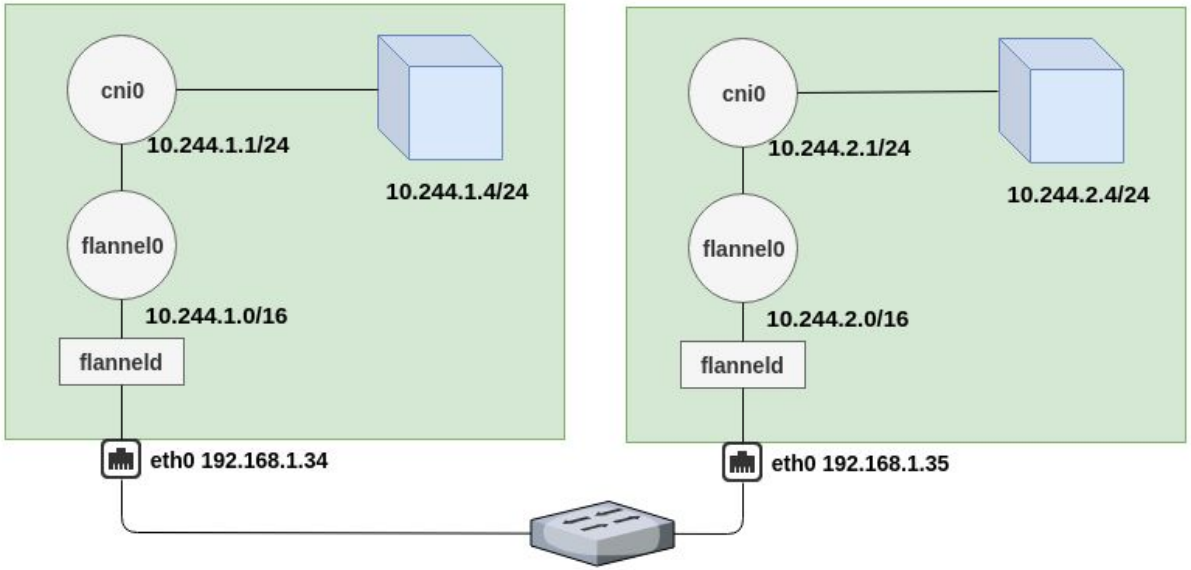
More network plugins

Flannel

- Provides a layer 3 IPv4 network between multiple nodes in a cluster
- On each host:
  - flanneld
  - allocates a subnet lease out of an address space
- Packets fowarding using backends
  - VXLAN, host-gw, UDP,...
- Uses docker0; flannel0 (tunnel) and daemon to encapsulate

```
{
        "name": "mynet",
        "type": "bridge",
        "mtu": 1472,
        "ipMasq": false,
        "isGateway": true,
        "ipam": {
                "type": "host-local",
                "subnet":
"10.1.17.0/24"
        }
  }
```

More network plugins



- Builds an overlay network
- Flanneld creates some kernel routes
- Flannel0 TUN, linux kernel interface

More network plugins

Calico

- Apache project
- Layer 3 networking, leveraging linux kernel
- Use BGP: routes distribution
- Separate policy decisions from routing
- Distributed firewall
- Millions of nodes

Fosters scalability and simplicity: what overlay model suffer

More network plugins

Weave net

- Virtual network to connect containers across hosts
- Service discovery: micro DNS on each host
- No external store
- VXLAN between hosts
- Traffic encryption

More network plugins

Romana

- Endpoints receive real routable addresses, no overlays or tunnels
- Addresses blocks instead of individual ones

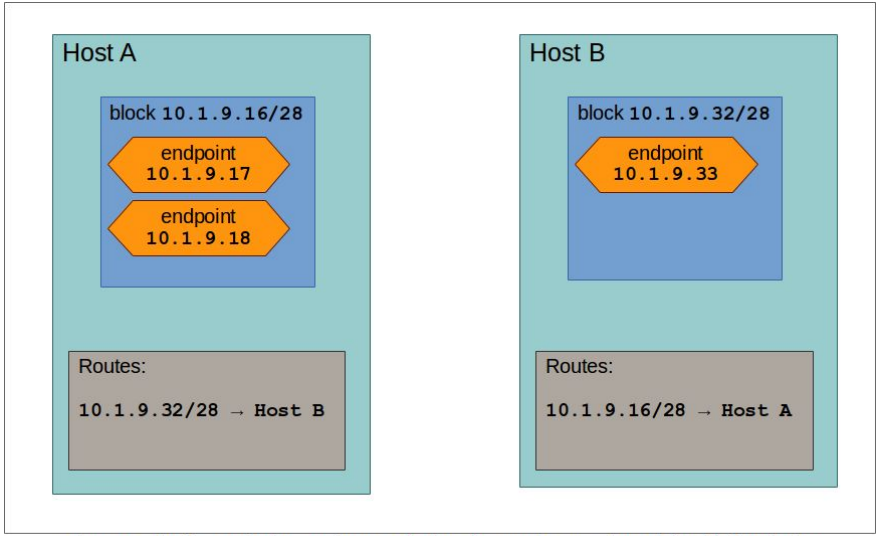


Figure 1: Blocks, endpoints and routes in the cluster after creation of the third endpoint

Author: romana

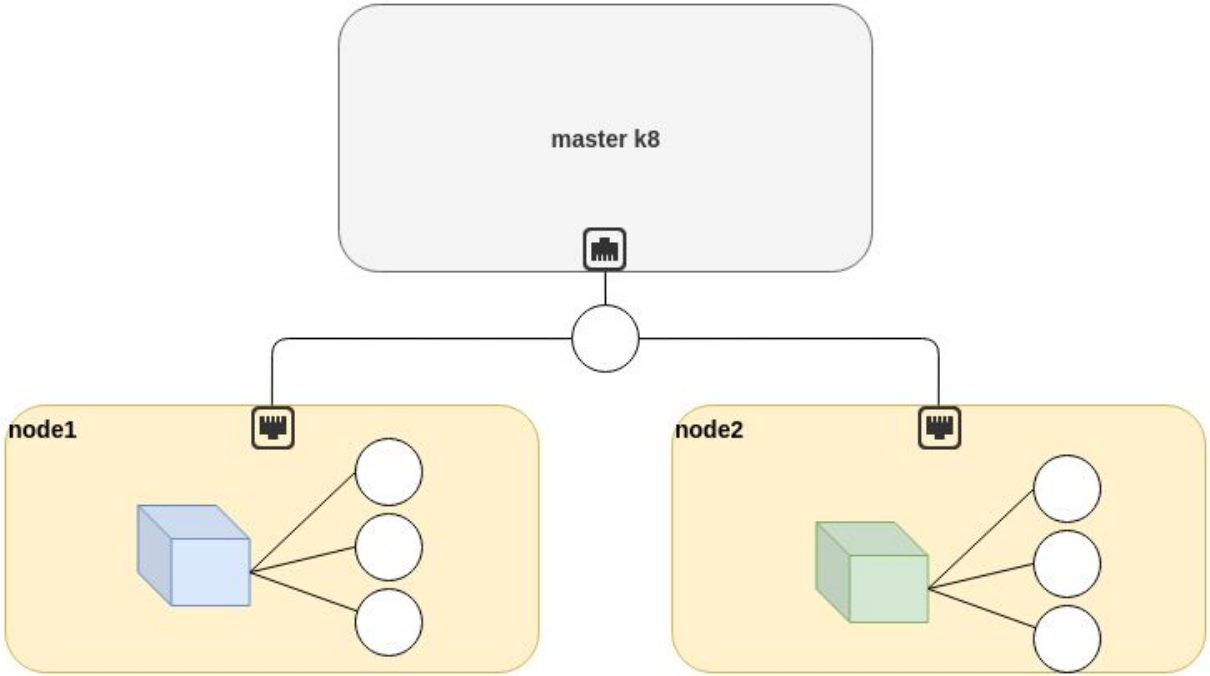
And more

- Cillium
- Kube router

...

Demo time!

Scenario



CHARLA · JUEVES 14 · 18:30h

Leveraging Kubernetes for NFV

José María Roldán
@josrolgil

Thanks for coming!

t3chfest

14 Y 15 DE MARZO DE 2019