7th SIG-NOC

Special Interest Group Network Operation Control
hosted by CSUC - Barcelona

GÉANT

# Kubernetes Container Networking

NmaaS service cluster

**Frédéric LOUI / RENATER**

frederic.loui@renater.fr

Barcelona, Spain
19,20 April 2018

# Agenda

- NMaaS service in a nutshell
- NMaaS under the hood
- NMaaS overall architecture/workflow
- Kubernetes core concept from the networking perspective
- Typical Kubernetes cluster design
- Kubernetes cluster architecture
- Key take-away

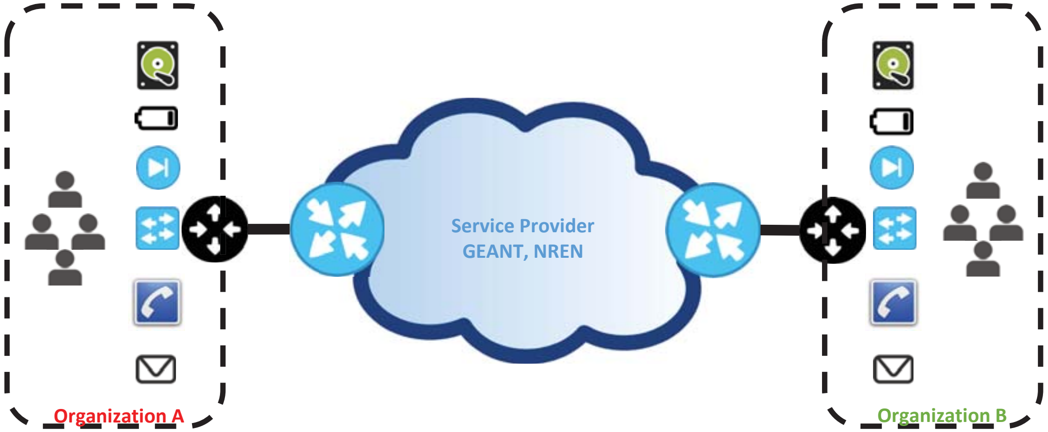
# NMaaS in a Nutshell

# NMaaS under the hood

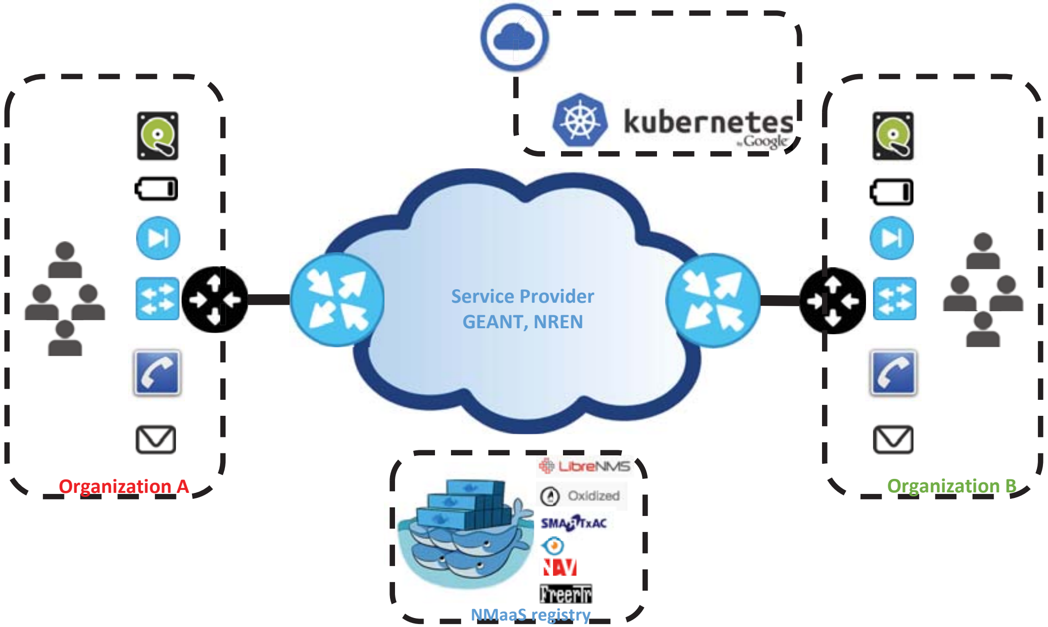
GÉANT

- Portal
  - Bootstrap
  - spring by Pivotal
- Network automation
  - ANSIBLE
  - OPENCONFIG
- Services cluster
  - docker
  - kubernetes by Google

# Network Management as a Service
## NM service deployment workflow



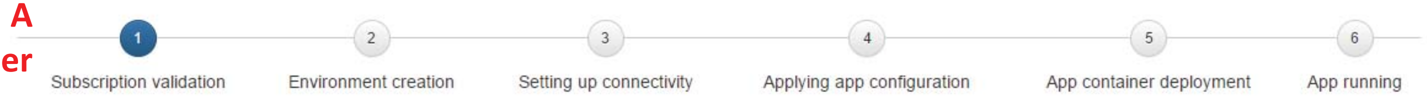
# Network Management as a Service
# NM service deployment workflow



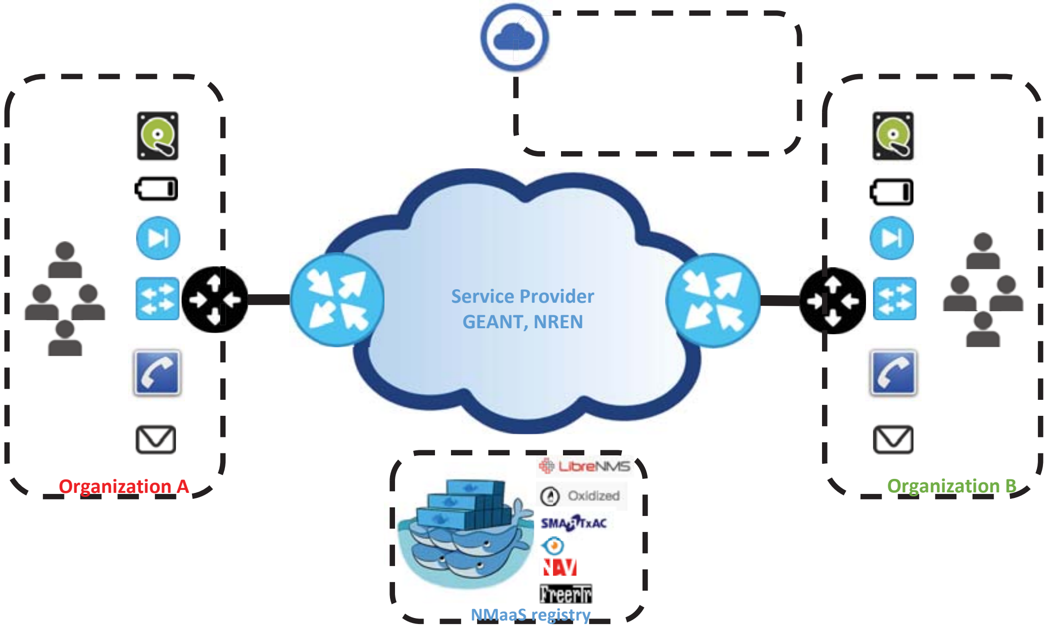
# Network Management as a Service
# NM service deployment workflow

GÉANT

**Organization A customer**



1 Subscription validation
2 Environment creation
3 Setting up connectivity
4 Applying app configuration
5 App container deployment
6 App running



# Network Management as a Service
## NM service deployment workflow

**Organization A customer**

1 Subscription validation

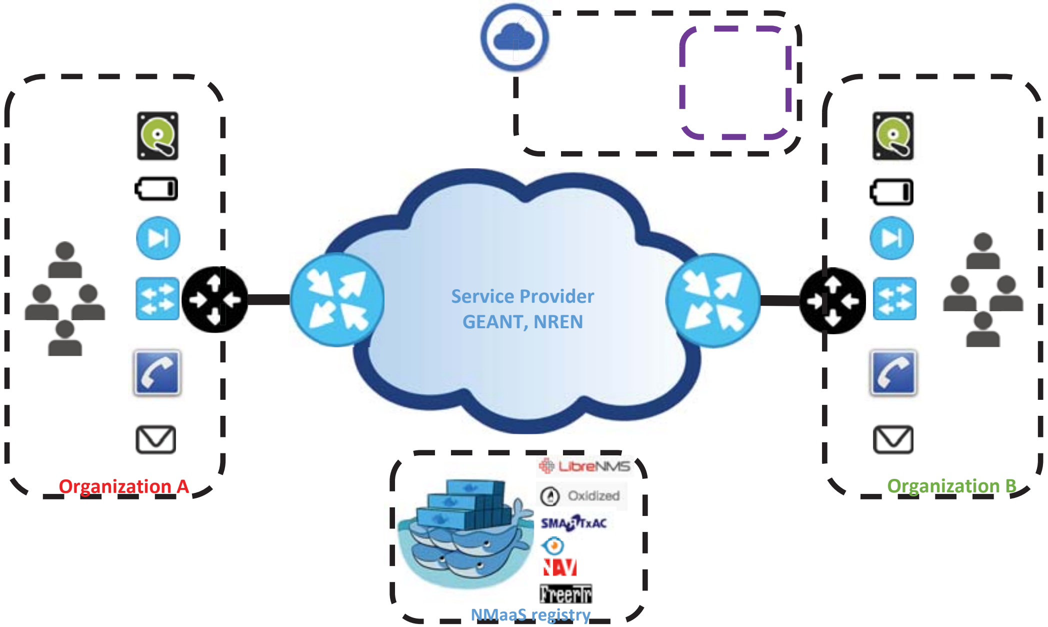
2 Environment creation

3 Setting up connectivity

4 Applying app configuration

5 App container deployment

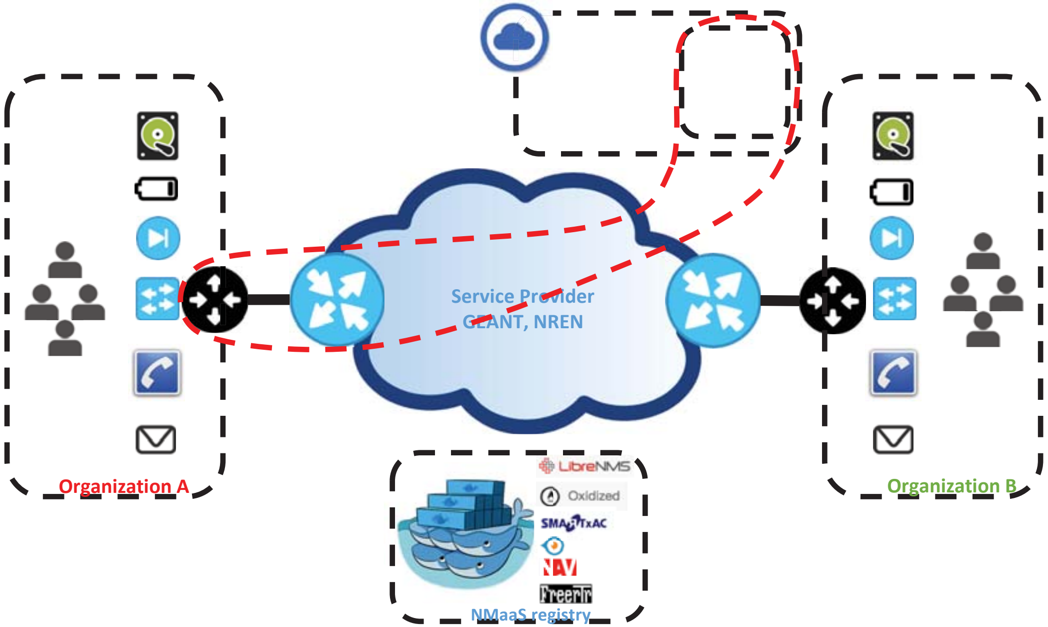
6 App running



# Network Management as a Service
## NM service deployment workflow

Organization A customer

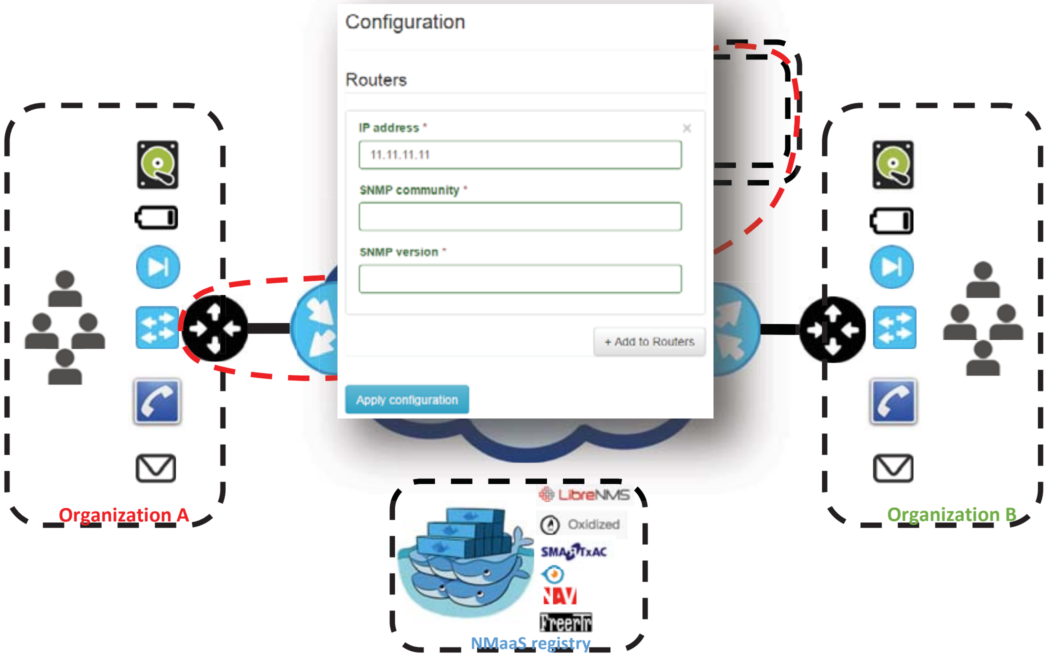
1 Subscription validation
2 Environment creation
3 Setting up connectivity
4 Applying app configuration
5 App container deployment
6 App running



# Network Management as a Service
## NM service deployment workflow

Organization A customer

1 Subscription validation — 2 Environment creation — 3 Setting up connectivity — 4 Applying app configuration — 5 App container deployment — 6 App running



Configuration

Routers

IP address *

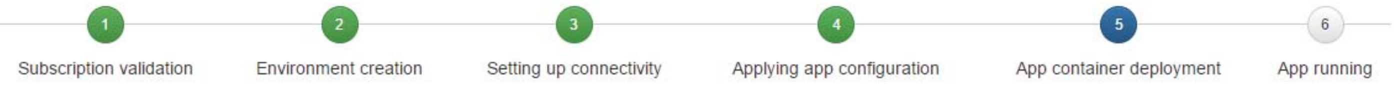
11.11.11.11

SNMP community *

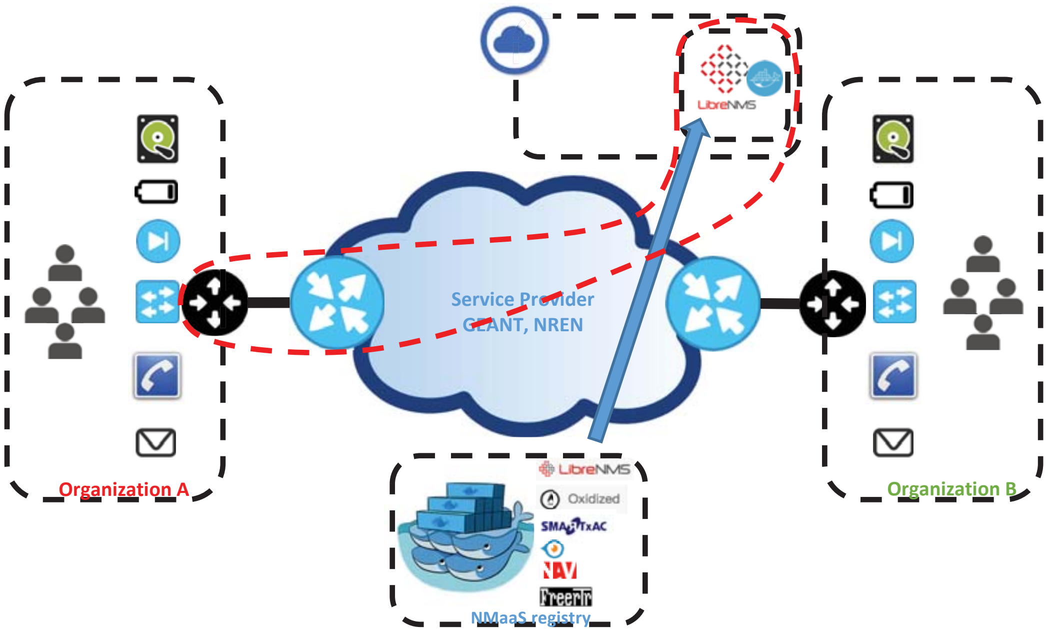
SNMP version *

# Network Management as a Service
# NM service deployment workflow

Organization A customer



1 Subscription validation
2 Environment creation
3 Setting up connectivity
4 Applying app configuration
5 App container deployment
6 App running

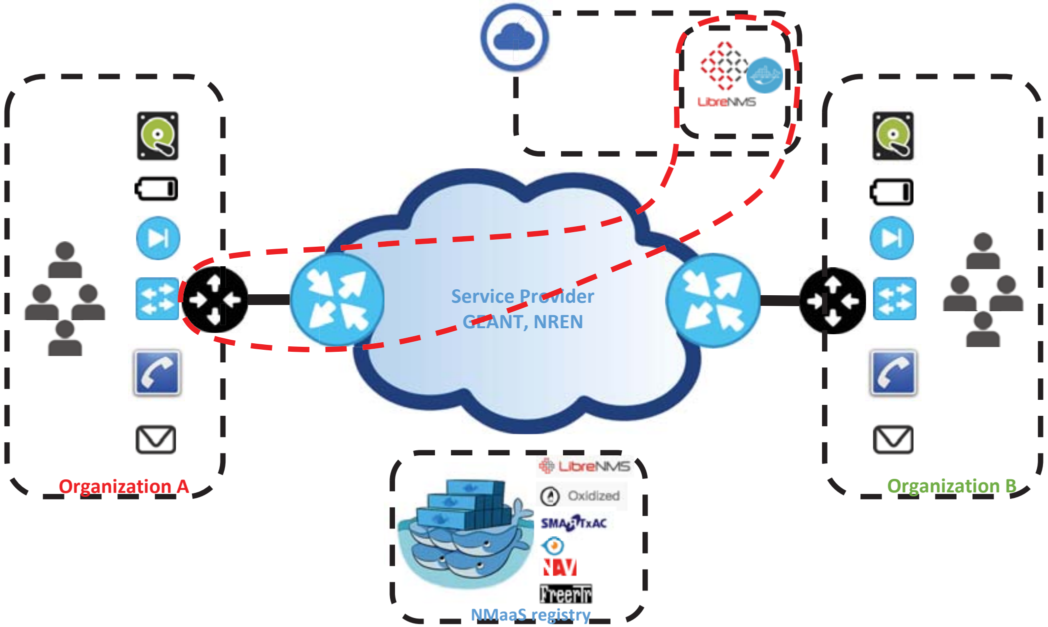


# Network Management as a Service
## NM service deployment workflow

GÉANT

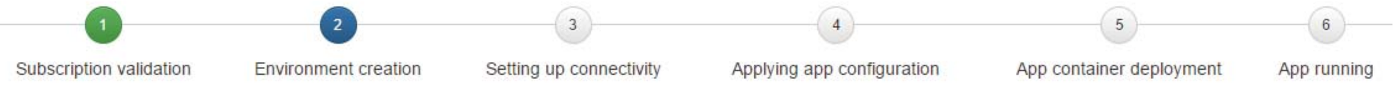
**Organization A customer**

1. Subscription validation
2. Environment creation
3. Setting up connectivity
4. Applying app configuration
5. App container deployment
6. App running

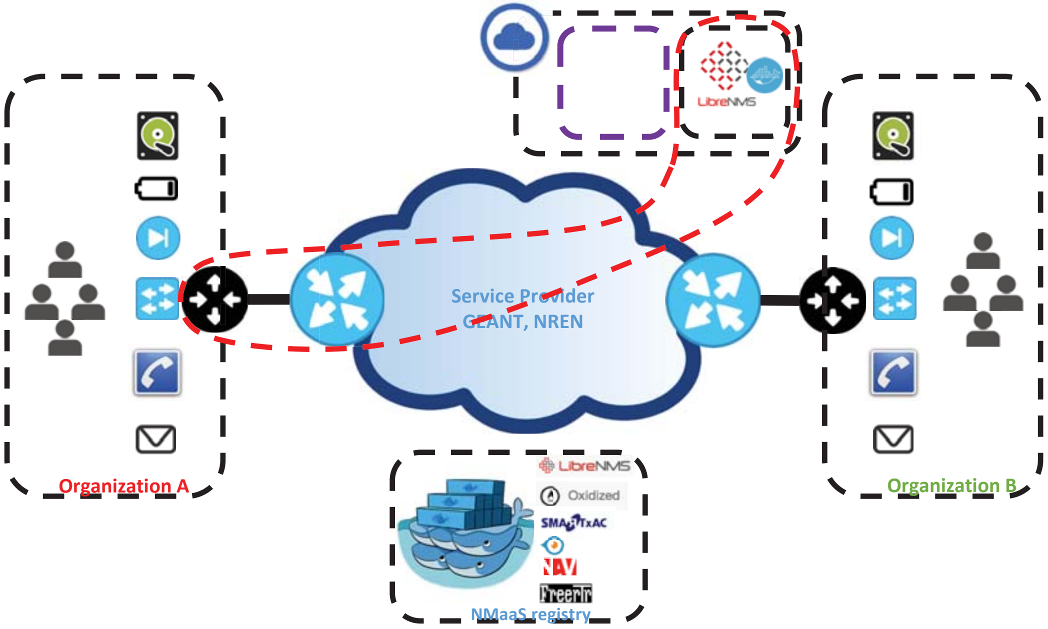


# Network Management as a Service
## NM service deployment workflow

Organization B customer



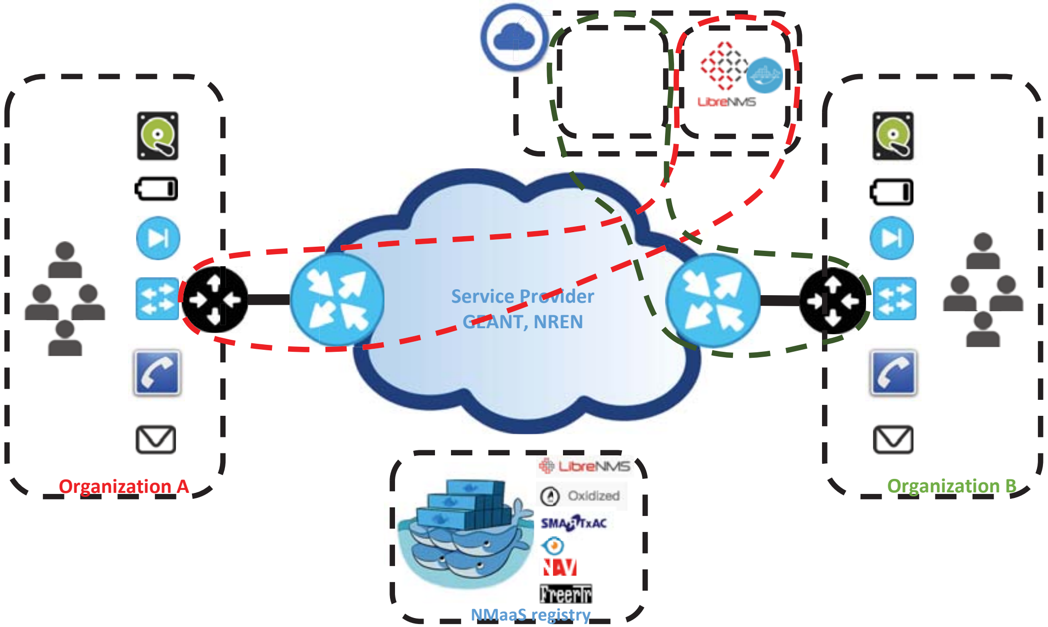
1 Subscription validation
2 Environment creation
3 Setting up connectivity
4 Applying app configuration
5 App container deployment
6 App running



# Network Management as a Service
# NM service deployment workflow

## Organization B customer

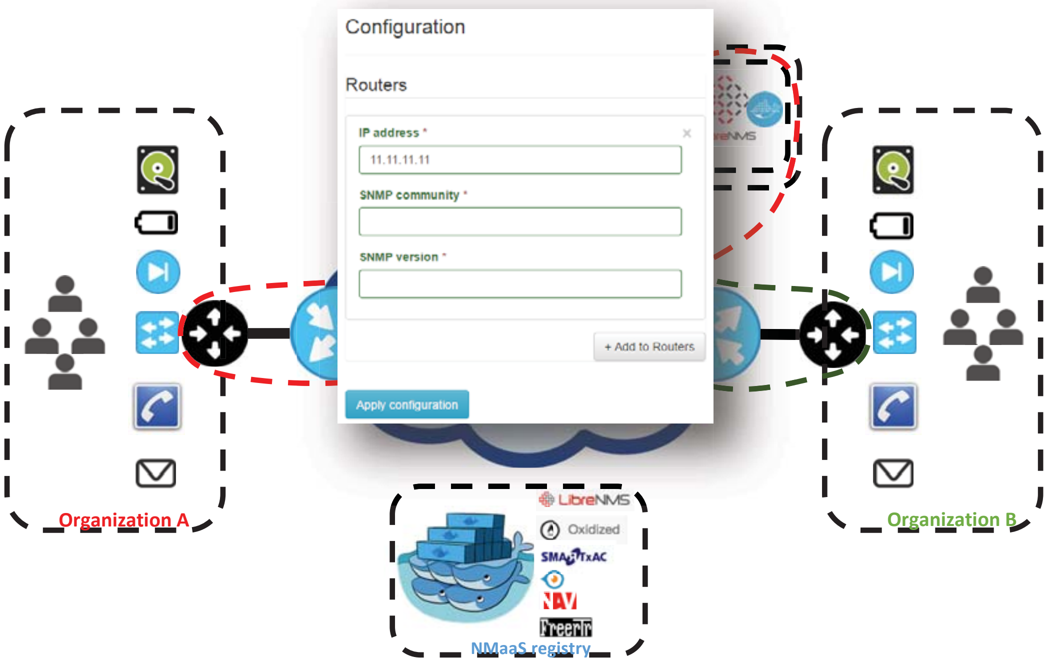
1. Subscription validation
2. Environment creation
3. Setting up connectivity
4. Applying app configuration
5. App container deployment
6. App running



# Network Management as a Service
# NM service deployment workflow

**Organization B customer**

1. Subscription validation
2. Environment creation
3. Setting up connectivity
4. Applying app configuration
5. App container deployment
6. App running



Configuration

Routers

IP address *

11.11.11.11

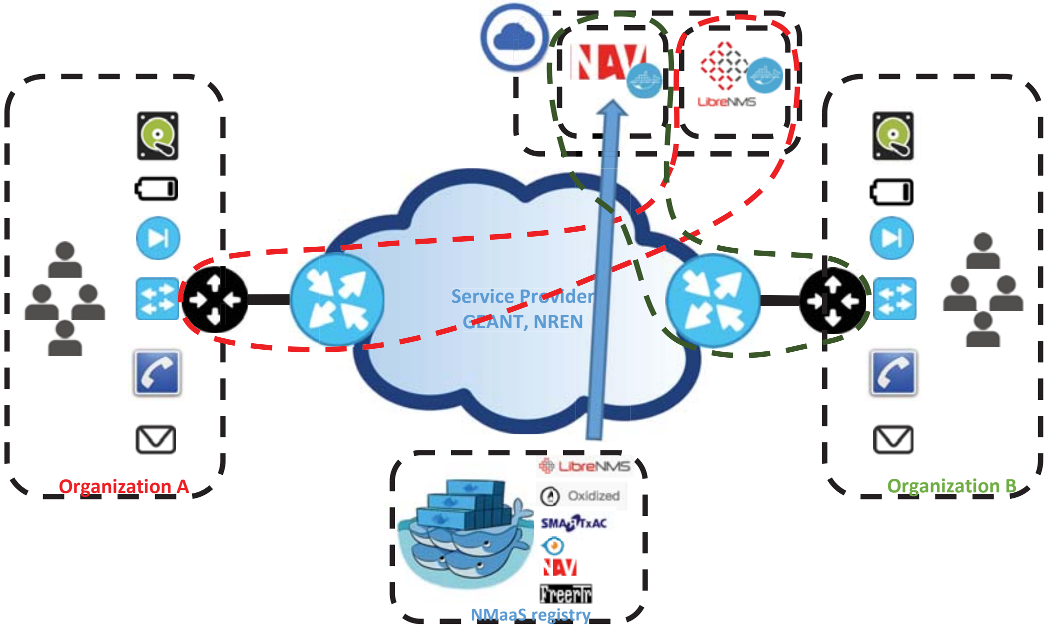
SNMP community *

SNMP version *

# Network Management as a Service
## NM service deployment workflow

Organization B customer

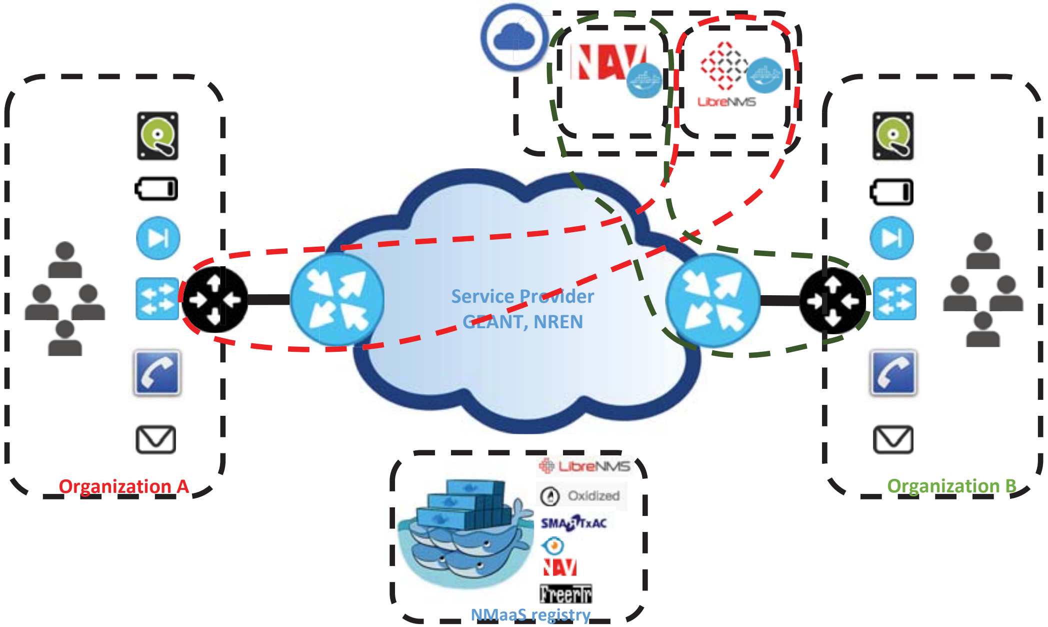
1 Subscription validation
2 Environment creation
3 Setting up connectivity
4 Applying app configuration
5 App container deployment
6 App running



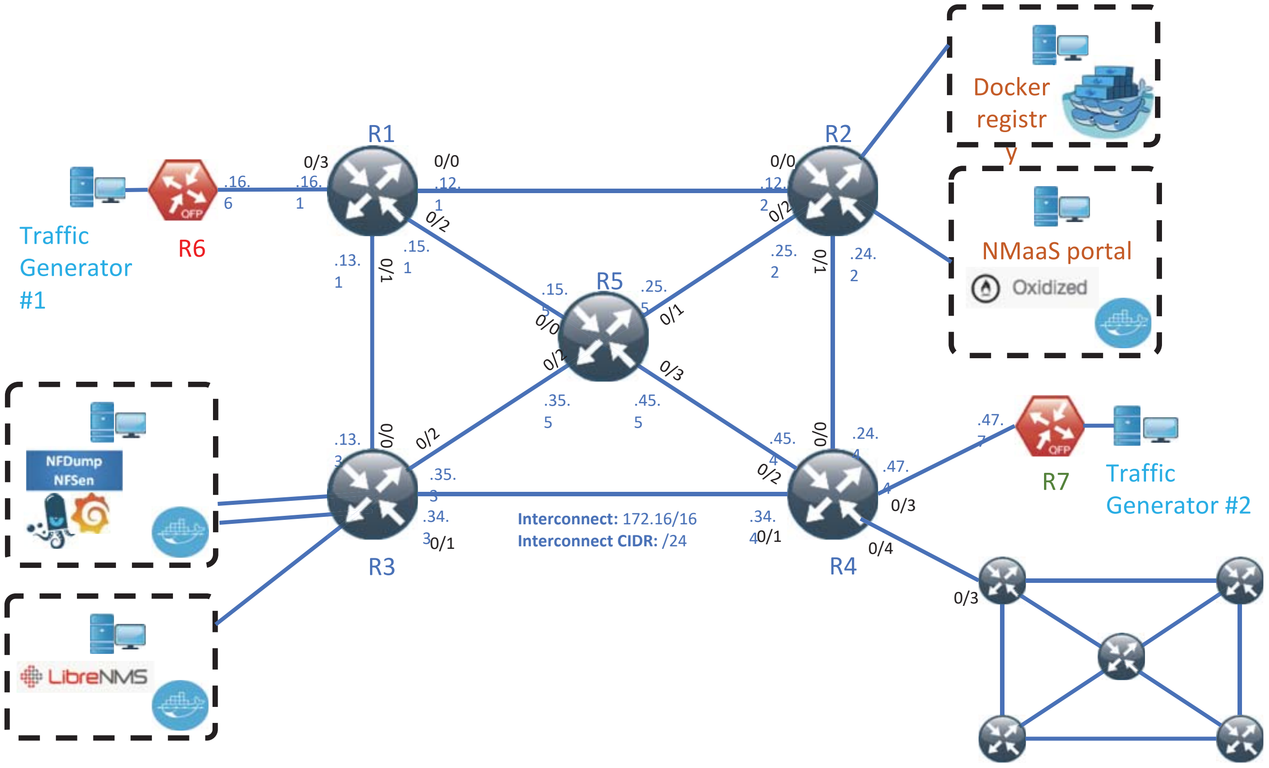
# Network Management as a Service
## NM service deployment workflow

**Organization B customer**

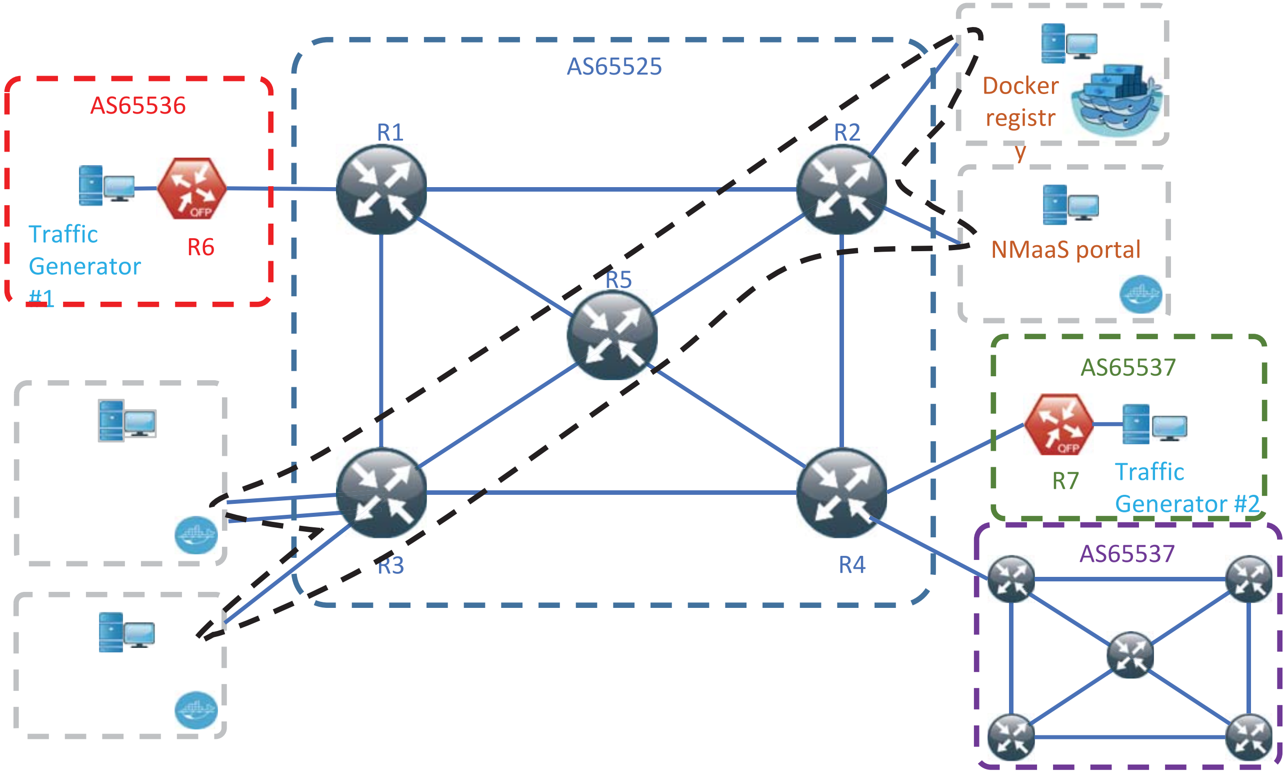
1. Subscription validation
2. Environment creation
3. Setting up connectivity
4. Applying app configuration
5. App container deployment
6. App running



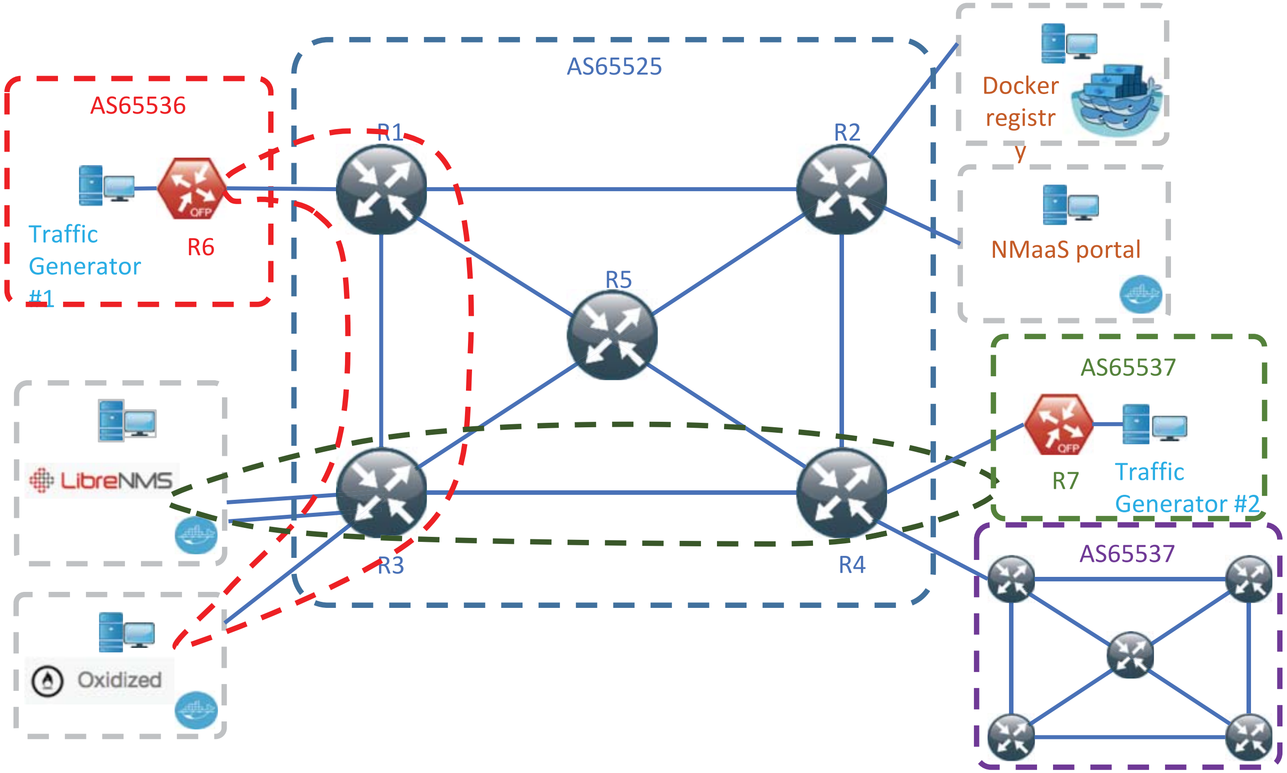
# Network Management as a Service
# Deployment in the testbed



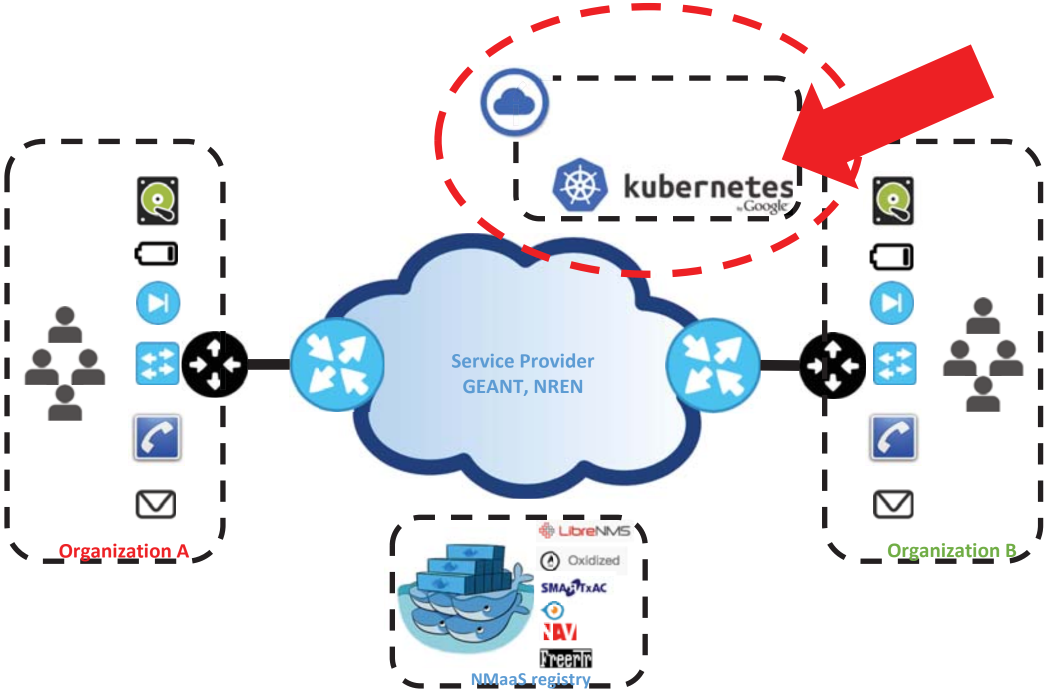
# Network Management as a Service
# Deployment in the testbed



# Network Management as a Service
## Deployment in the testbed

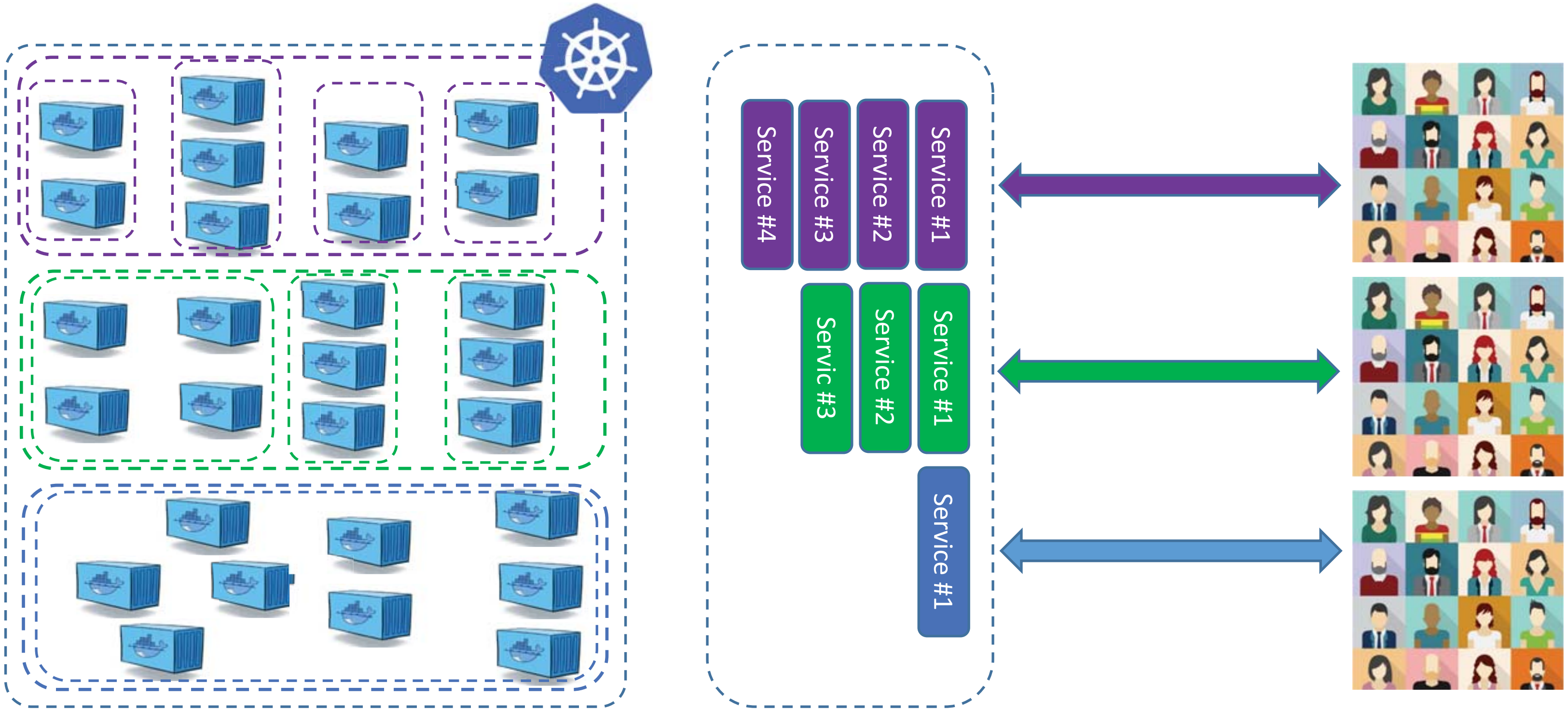


# Network Management as a Service
# NM service deployment workflow



# Kubernetes

Container based micro-service orchestrator and scheduler



# Few numbers

- Example of small companies in US
  - Large K8s deployment
    - 25 clusters with 7500 nodes
    - Plan to move to 40K nodes by Q4 2017
- Google’s lesson’s learned
  - **Kubernetes Scaling and Performance Goals**
    - https://github.com/kubernetes/community/blob/master/sig-scalability/goals.md
    - Max core per cluster 200 000
    - Max pod per core 10
    - Management overhead per node Goal: <5%, with a minimum of 0.5 core, 1GB RAM
    - Management overhead per cluster Goal: <1%, with a minimum of 2 cores, 4GB RAM
- Have you played Pokemon GO ?
  - If yes, read this :

    *https://cloudplatform.googleblog.com/2016/09/bringing-Pokemon-GO-to-life-on-Google-Cloud.html*

# Kubernetes core components and POD concepts

- Node (VM or physical)

K8s
« master »

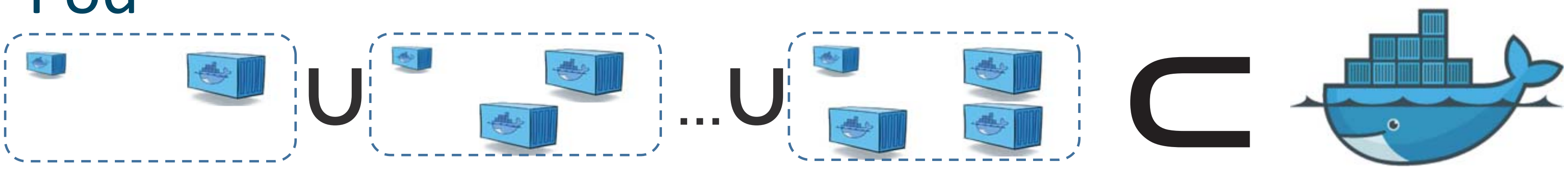
Etcd
« K8s brain storage »

Container engine
« worker »

Routing process
« Router + network plugin »

- Container + Container Engine

- Pod



# Kubernetes POD

Container engine
« worker »

- PODs are deployed on worker nodes
- PODs are manipulated by K8s
- PODs are dynamic in essence
  - Can be moved from one worker to another dynamically by K8s
  - Have a short lifetime
  - Have then a dynamic IP !
  - Like container, PODs are immutable
- **NEVER REFER TO POD IP OR BIND A DNS RECORD TO POD IP IN ORDER TO PROPOSE A SERVICE TO A CUSTOMER !**

# Kubernetes Deployment

## Kubernetes deployment defines:

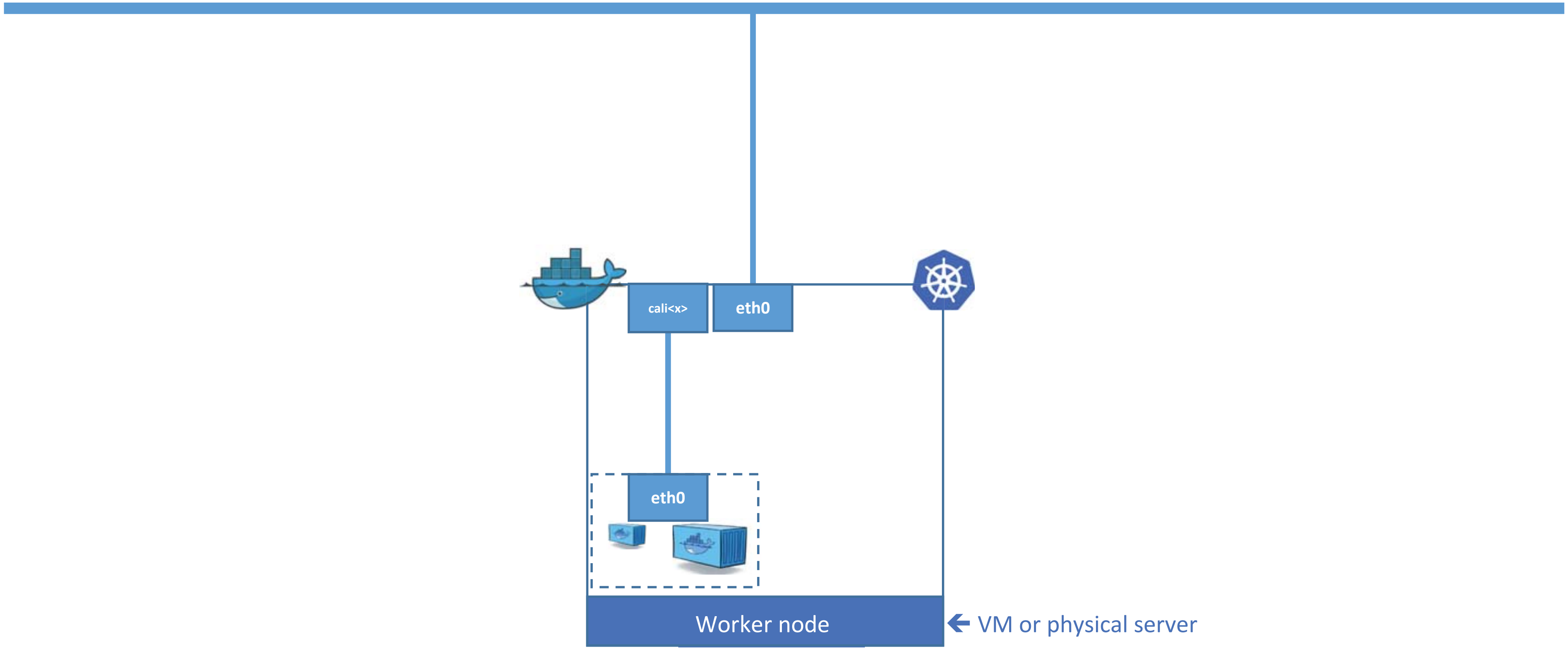
- **How a set of PODs is deployed**
  - Indicate which container from registry is used
  - Attach Storage volume
  - How much replica etc
  - Which network ports are exposed
  - Etc.

# Kubernetes Service

Kubernetes services construct binds:

- Typically a set of PODs deployment
- To a well know and user defined service IP address
- This service is bound to a DNS record by a K8s DNS

# Kubernetes: POD networking



# Kubernetes POD manifest: sig-noc-pod-bastion.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: sig-noc-pod-bastion
spec:
  containers:
  - name: sig-noc-tiny-netutils
    image: floui/tiny-net-tools
    command: [ "/bin/sh" ]
    args: ["-c", "while true; do { echo -e 'HTTP/1.1 200 OK\r\n’;
          echo 'Hello 7TH SIG-NOC@Barcelona !'; } | nc -l -p  8080; done"]
```

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl create -f ./sig-noc-pod-bastion.yaml
pod "sig-noc-pod-bastion" created
kubeadm@kube2-6:~/7TH-SIG-NOC$
```

# Kubernetes POD manifest: sig-noc-pod-bastion.yaml

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl get pod -o wide | egrep "NAME|sig-noc-pod"
NAME                                                                  READY     STATUS    RESTARTS   AGE       IP                 NODE
sig-noc-pod-bastion                                                   1/1       Running   0          1m        192.168.18.129     172.16.1.7
kubeadm@kube2-6:~/7TH-SIG-NOC$
```

```
kubeadm@kube2-7:~$ sudo docker ps | grep sig-noc-pod
41070131bb73        floui/tiny-net-tools@sha256:f2089f227a19a6e880c63503abb678b53a9bcfbca3851c8d7de6dac1f716e2fd
"/bin/sh -c 'while tr"   5 minutes ago       Up 5 minutes                            k8s_sig-noc-tiny-netutils_sig-noc-pod-
bastion_default_46615117-3f27-11e8-93b9-5254002cd33f_0
fac58734b72d        gcr.io/google_containers/pause-amd64:3.0
"/pause"                 5 minutes ago       Up 5 minutes                            k8s_POD_sig-noc-pod-bastion_default_46615117-3f27-11e8-
93b9-5254002cd33f_0
kubeadm@kube2-7:~$
```

```
kubeadm@kube2-7:~$ sudo docker inspect 41070131bb73 | grep NetworkMode
            "NetworkMode": "container:fac58734b72d300af5652ad013b3781a361a8c4722104d738d777a929f45e856",
kubeadm@kube2-7:~$ sudo docker inspect fac58734b72d | grep NetworkMode
            "NetworkMode": "none",
```

```
kubeadm@kube2-7:~$ sudo docker inspect 41070131bb73 | grep "Pid\""
            "Pid": 29504,
kubeadm@kube2-7:~$ sudo docker inspect fac58734b72d | grep "Pid\""
            "Pid": 29326,
```

# Kubernetes POD manifest: sig-noc-pod-bastion.yaml

```
kubeadm@kube2-7:~$ sudo nsenter -t 29504 -n ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
4: eth0@if233: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default
    link/ether 86:6e:fa:c7:64:06 brd ff:ff:ff:ff:ff:ff link-netnsid 0
kubeadm@kube2-7:~$ sudo nsenter -t 29326 -n ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN mode DEFAULT group default qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
4: eth0@if233: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default
    link/ether 86:6e:fa:c7:64:06 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

```
kubeadm@kube2-7:~$ sudo nsenter -t 29504 -n ifconfig
eth0      Link encap:Ethernet  HWaddr 86:6e:fa:c7:64:06
          inet addr:192.168.18.129  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: fe80::846e:faff:fec7:6406/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:648 (648.0 B)  TX bytes:558 (558.0 B)

kubeadm@kube2-7:~$ sudo nsenter -t 29326 -n ifconfig
eth0      Link encap:Ethernet  HWaddr 86:6e:fa:c7:64:06
          inet addr:192.168.18.129  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: fe80::846e:faff:fec7:6406/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:648 (648.0 B)  TX bytes:558 (558.0 B)
```

# Kubernetes POD manifest: sig-noc-pod-bastion.yaml

```
kubeadm@kube2-7:~$ sudo nsenter -t 29504 -n ifconfig
eth0      Link encap:Ethernet  HWaddr 86:6e:fa:c7:64:06
          inet addr:192.168.18.129  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: fe80::846e:faff:fec7:6406/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:648 (648.0 B)  TX bytes:558 (558.0 B)

kubeadm@kube2-7:~$ sudo nsenter -t 29326 -n ifconfig
eth0      Link encap:Ethernet  HWaddr 86:6e:fa:c7:64:06
          inet addr:192.168.18.129  Bcast:0.0.0.0  Mask:255.255.255.255
          inet6 addr: fe80::846e:faff:fec7:6406/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:1 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:648 (648.0 B)  TX bytes:558 (558.0 B)
```

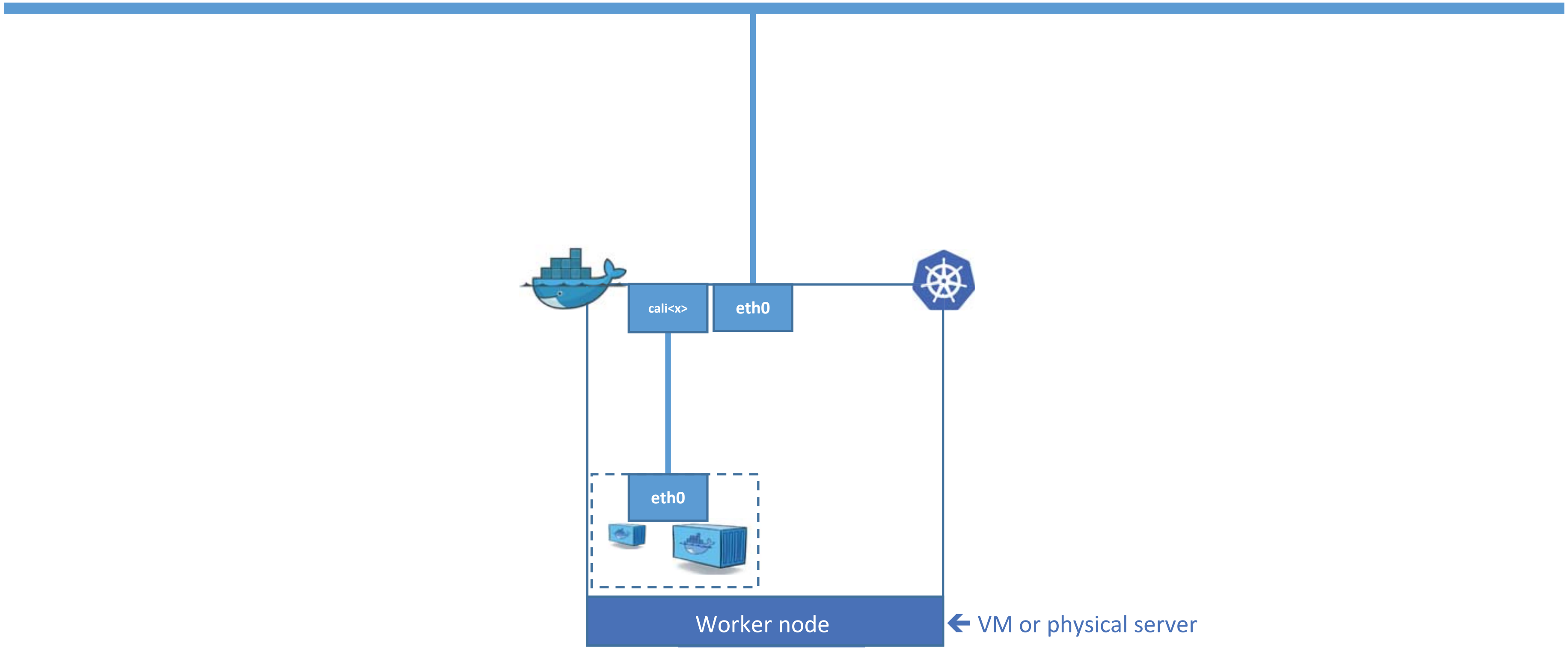
# Kubernetes POD manifest: sig-noc-pod-bastion.yaml

```
kubeadm@kube2-7:~$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:9a:b1:2a brd ff:ff:ff:ff:ff:ff
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:f8:7f:6b brd ff:ff:ff:ff:ff:ff
4: eth2: <BROADCAST,MULTICAST,ALLMULTI,PROMISC,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:f7:12:31 brd ff:ff:ff:ff:ff:ff
5: eth2.30@eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:f7:12:31 brd ff:ff:ff:ff:ff:ff
6: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN mode DEFAULT group default
    link/ether 02:42:a2:9e:5e:62 brd ff:ff:ff:ff:ff:ff
8: tunl0@NONE: <NOARP,UP,LOWER_UP> mtu 1440 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 1
    link/ipip 0.0.0.0 brd 0.0.0.0
222: cali703c6192aa8@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default
    link/ether d6:3a:5c:ca:6e:3b brd ff:ff:ff:ff:ff:ff link-netnsid 0
223: cali8ea7ff0c6ac@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default
    link/ether be:51:25:d3:88:1a brd ff:ff:ff:ff:ff:ff link-netnsid 1
224: cali8fb48e24e9f@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default
    link/ether 06:9d:d4:45:24:e1 brd ff:ff:ff:ff:ff:ff link-netnsid 7
233: calid73200a0875@if4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default
    link/ether 56:a6:8f:bf:29:6a brd ff:ff:ff:ff:ff:ff link-netnsid 2
kubeadm@kube2-7:~$
```

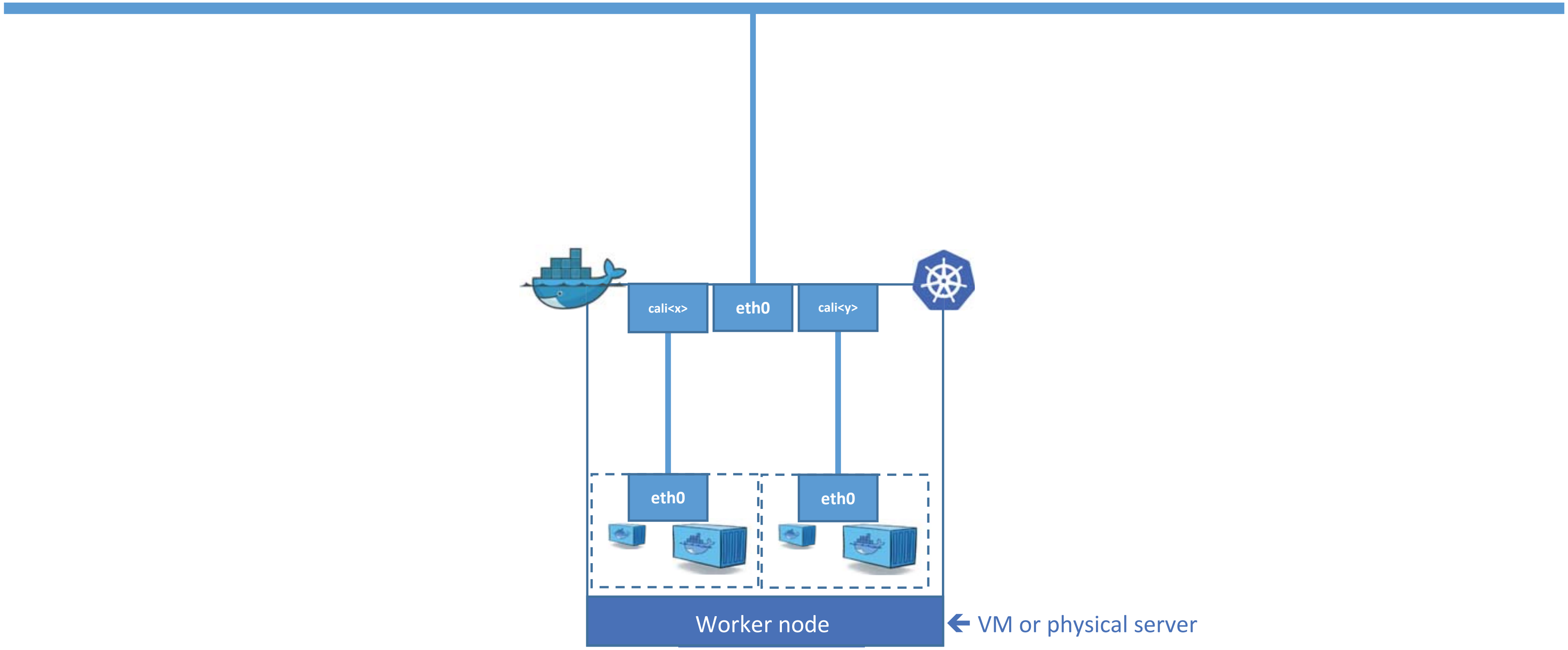
# Kubernetes POD manifest: sig-noc-pod-bastion.yaml

```
kubeadm@kube2-7:~$ ip route show
default via 10.134.1.13 dev eth0 onlink
10.1.0.0/24 dev eth2.30  proto kernel  scope link  src 10.1.0.7
10.128.0.0/9 dev eth0  proto kernel  scope link  src 10.134.241.7
172.16.1.0/24 dev eth1  proto kernel  scope link  src 172.16.1.7
172.17.0.0/16 dev docker0  proto kernel  scope link  src 172.17.0.1 linkdown
192.168.11.0/24 via 172.16.1.13 dev eth1
192.168.11.66 via 10.1.0.8 dev tunl0  proto bird onlink
blackhole 192.168.18.128/26  proto bird
192.168.18.129 dev calid73200a0875  scope link
192.168.18.184 dev cali8ea7ff0c6ac  scope link
192.168.18.187 dev cali703c6192aa8  scope link
192.168.18.188 dev cali8fb48e24e9f  scope link
192.168.72.64/26 via 10.1.0.8 dev tunl0  proto bird onlink
192.168.127.192/26 via 10.1.0.10 dev tunl0  proto bird onlink
192.168.165.128/26 via 10.1.0.9 dev tunl0  proto bird onlink
192.168.229.192/26 via 10.1.0.6 dev tunl0  proto bird onlink
kubeadm@kube2-7:~$
```

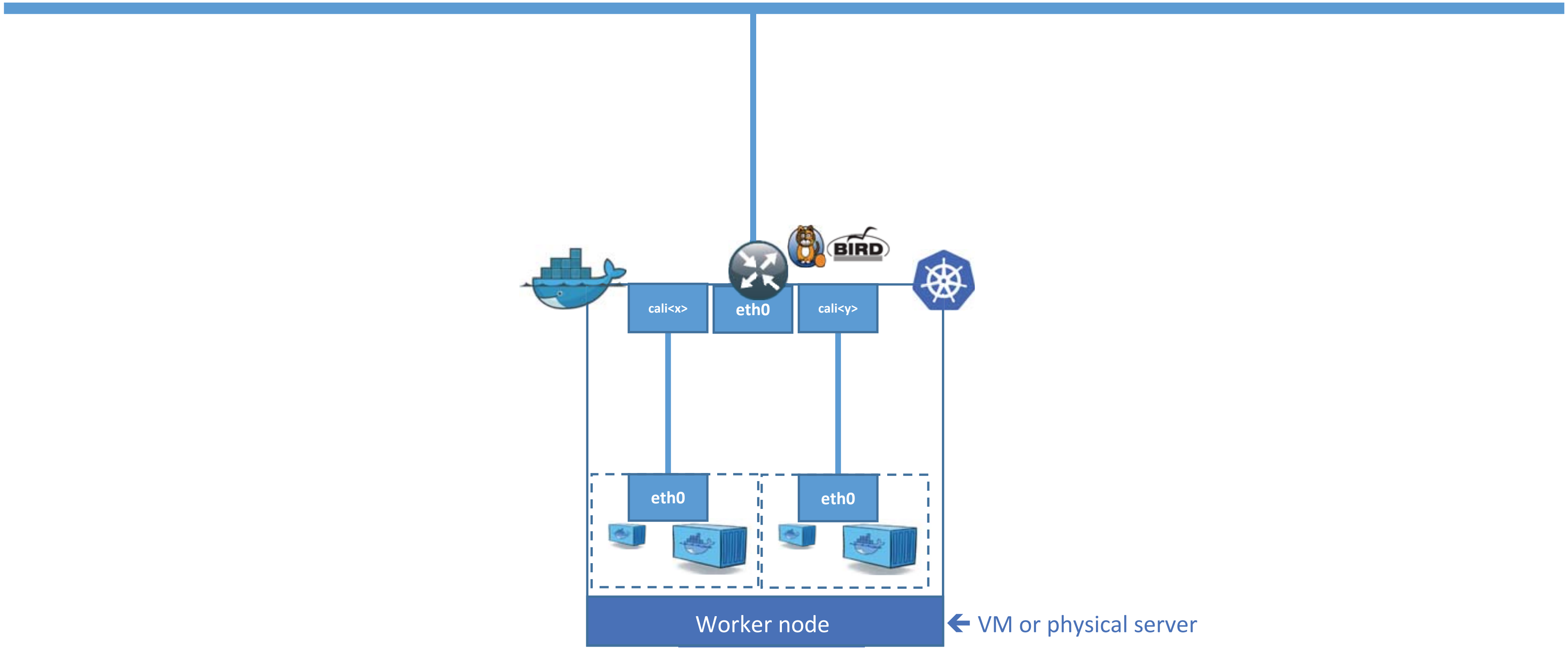
# Kubernetes: POD networking



# Kubernetes: POD to POD networking within the same Worker



# Kubernetes: POD network plugin: Calico → CNI compliant



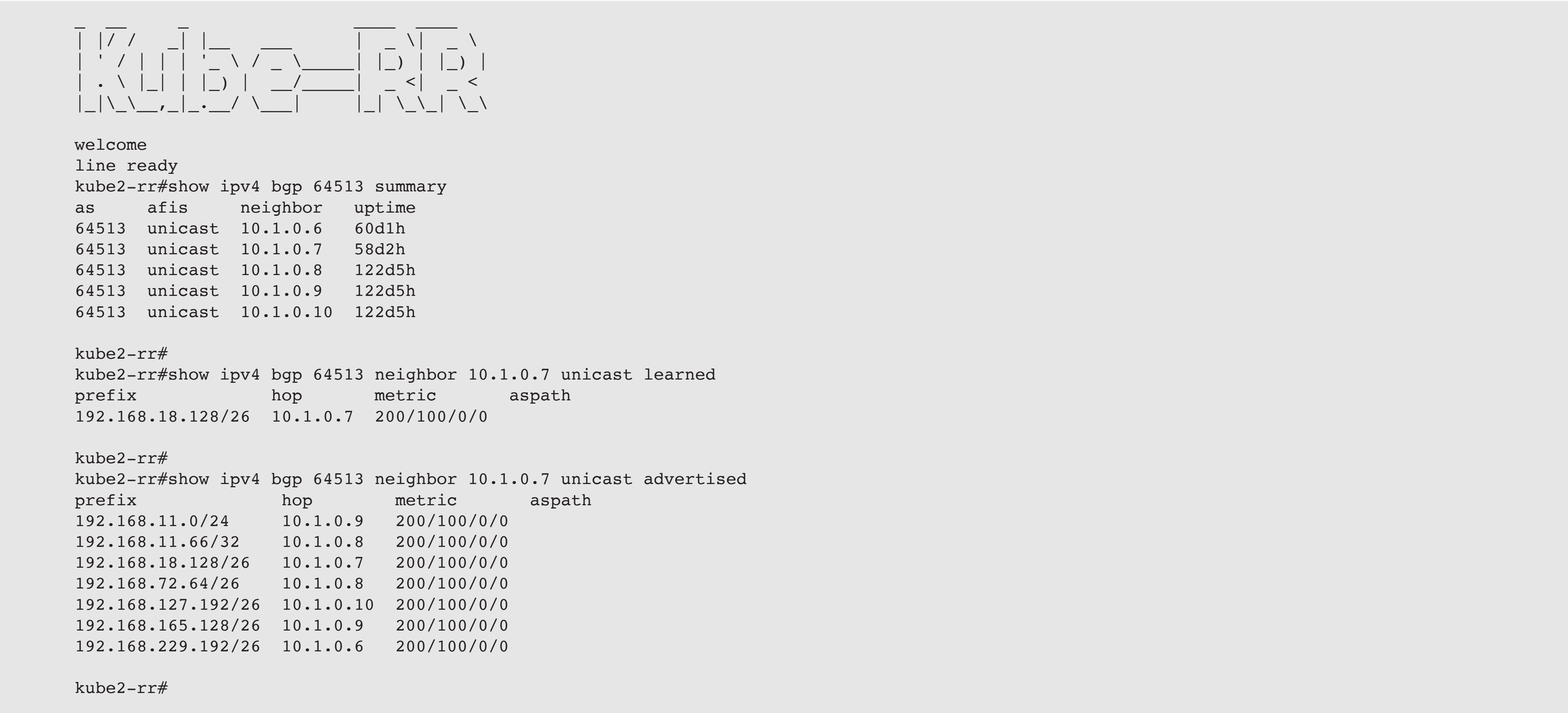
# Kubernetes network plugin CNI - Calico

```
kubeadm@kube2-7:~$ ip route show
default via 10.134.1.13 dev eth0 onlink
10.1.0.0/24 dev eth2.30  proto kernel  scope link  src 10.1.0.7
10.128.0.0/9 dev eth0  proto kernel  scope link  src 10.134.241.7
172.16.1.0/24 dev eth1  proto kernel  scope link  src 172.16.1.7
172.17.0.0/16 dev docker0  proto kernel  scope link  src 172.17.0.1 linkdown
192.168.11.0/24 via 172.16.1.13 dev eth1
192.168.11.66 via 10.1.0.8 dev tunl0  proto bird onlink
blackhole 192.168.18.128/26  proto bird
192.168.18.129 dev calid73200a0875  scope link
192.168.18.184 dev cali8ea7ff0c6ac  scope link
192.168.18.187 dev cali703c6192aa8  scope link
192.168.18.188 dev cali8fb48e24e9f  scope link
192.168.72.64/26 via 10.1.0.8 dev tunl0  proto bird onlink
192.168.127.192/26 via 10.1.0.10 dev tunl0  proto bird onlink
192.168.165.128/26 via 10.1.0.9 dev tunl0  proto bird onlink
192.168.229.192/26 via 10.1.0.6 dev tunl0  proto bird onlink
kubeadm@kube2-7:~$
kubeadm@kube2-7:~$ sudo ifconfig tunl0
tunl0     Link encap:IPIP Tunnel  HWaddr
          inet addr:192.168.18.128  Mask:255.255.255.255
          UP RUNNING NOARP  MTU:1440  Metric:1
          RX packets:1215334 errors:0 dropped:0 overruns:0 frame:0
          TX packets:948472 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:219535377 (219.5 MB)  TX bytes:464217710 (464.2 MB)
```

# Kubernetes network plugin CNI - Calico

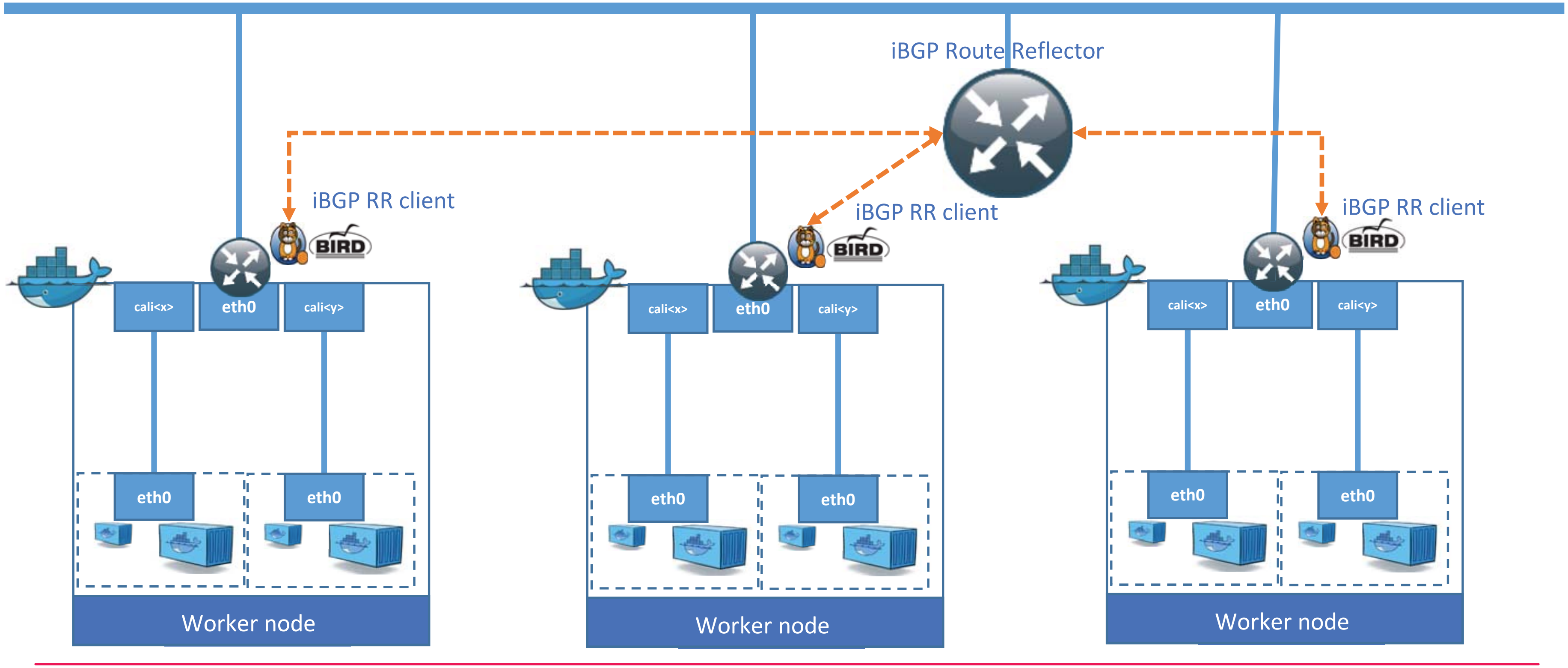
```
sudo ETCD_CA_CERT_FILE=/var/lib/kubernetes/ca.pem ETCD_ENDPOINTS=https://172.16.1.6:2379 calicoctl node status
Calico process is running.

IPv4 BGP status
+--------------+-----------+-------+------------+-------------+
| PEER ADDRESS | PEER TYPE | STATE |   SINCE    |    INFO     |
+--------------+-----------+-------+------------+-------------+
| 10.1.0.113   | global    | up    | 2018-02-14 | Established |
+--------------+-----------+-------+------------+-------------+

IPv6 BGP status
No IPv6 peers found.
```

# Kubernetes network plugin CNI - Calico

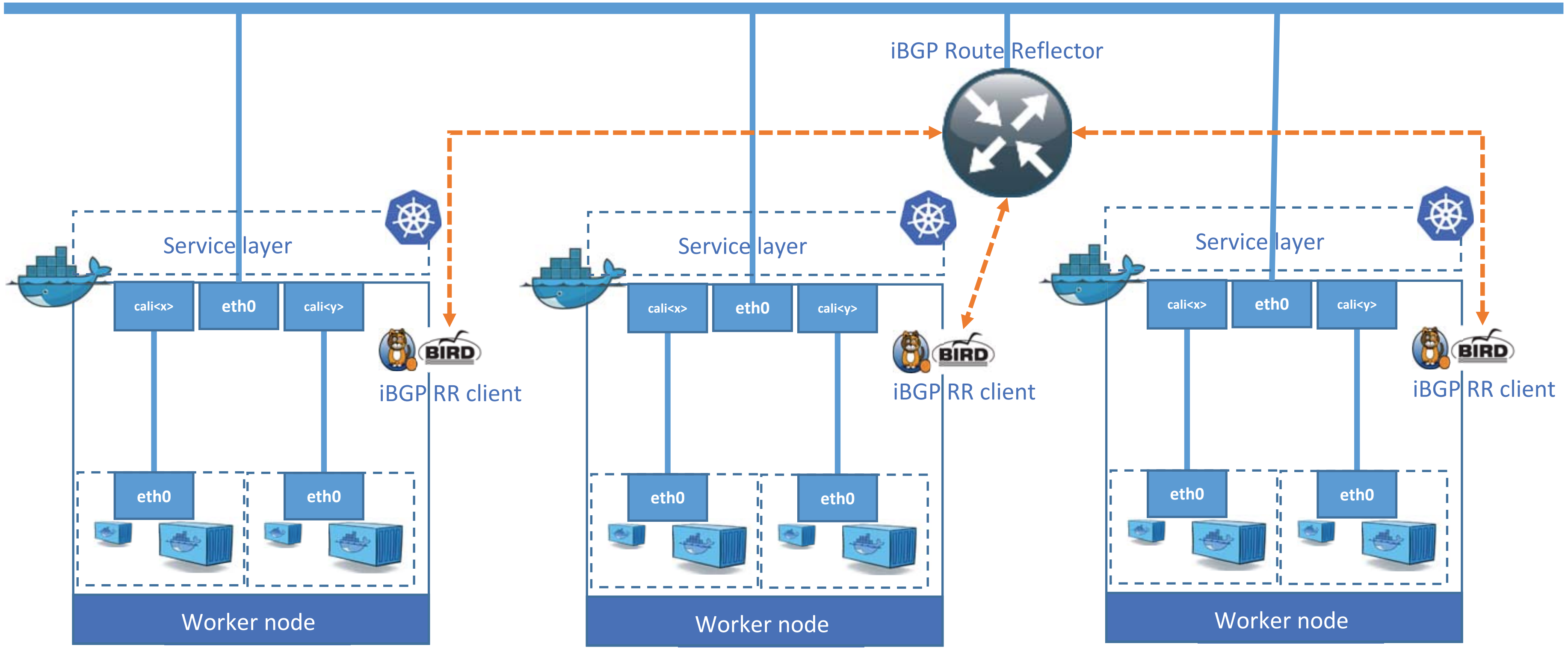


```
_  __     _                 ____  ____
| |/ /   _| |__   ___       |  _ \|  _ \
| ' / | | | '_ \ / _ \_____| |_) | |_) |
| . \ |_| | |_) |  __/_____|  _ <|  _ <
|_|\_\__,_|_.__/ \___|     |_| \_\_| \_\

welcome
line ready
kube2-rr#show ipv4 bgp 64513 summary
as     afis     neighbor   uptime
64513  unicast  10.1.0.6   60d1h
64513  unicast  10.1.0.7   58d2h
64513  unicast  10.1.0.8   122d5h
64513  unicast  10.1.0.9   122d5h
64513  unicast  10.1.0.10  122d5h

kube2-rr#
kube2-rr#show ipv4 bgp 64513 neighbor 10.1.0.7 unicast learned
prefix             hop       metric       aspath
192.168.18.128/26  10.1.0.7  200/100/0/0

kube2-rr#
kube2-rr#show ipv4 bgp 64513 neighbor 10.1.0.7 unicast advertised
prefix              hop         metric       aspath
192.168.11.0/24     10.1.0.9    200/100/0/0
192.168.11.66/32    10.1.0.8    200/100/0/0
192.168.18.128/26   10.1.0.7    200/100/0/0
192.168.72.64/26    10.1.0.8    200/100/0/0
192.168.127.192/26  10.1.0.10   200/100/0/0
192.168.165.128/26  10.1.0.9    200/100/0/0
192.168.229.192/26  10.1.0.6    200/100/0/0

kube2-rr#
```

# Kubernetes: POD inter-node networking



# Kubernetes: Service



# Kubernetes deployment manifest: sig-noc-www-deployment-no-ha.yaml

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ cat sig-noc-www-deployment-no-ha.yaml
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: sig-noc-www-deployment-no-ha
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: sig-noc-www-front-end
        version: v1
        availability: no-replica
    spec:
      containers:
      - name: sig-noc-www-ctn
        image: floui/tiny-net-tools
        command: [ "/bin/sh" ]
        args: ["-c", "while true; do { echo -e 'HTTP/1.1 200 OK\r\n'; echo 'Hello
7TH SIG-NOC@Barcelona !'; } | nc -l -p  8080; done"]
```

# Kubernetes service manifest: sig-noc-www-service-no-ha.yaml

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ cat sig-noc-www-service-no-ha.yaml
apiVersion: v1
kind: Service
metadata:
  name: sig-noc-www-service-no-ha
  namespace: default
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: signoc-www-port
  selector:
    app: sig-noc-www-front-end
    version: v1
    availability: no-replica
  sessionAffinity: None
  type: ClusterIP
status:
  loadBalancer: {}
```

# Kubernetes service

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl create -f ./sig-noc-www-service-no-ha.yaml
service "sig-noc-www-service-no-ha" created

kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl describe svc sig-noc-www-service-no-ha
Name:                   sig-noc-www-service-no-ha
Namespace:              default
Labels:                 <none>
Annotations:            <none>
Selector:               app=sig-noc-www-front-end,availability=no-replica,version=v1
Type:                   ClusterIP
IP:                     10.13.158.214
Port:                   <unset> 80/TCP
Endpoints:              192.168.18.135:8080
Session Affinity:       None
Events:                 <none>
```

# Kubernetes service

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl get deploy | egrep "NAME|sig-noc-www"
NAME                                                     DESIRED   CURRENT   UP-TO-DATE   AVAILABLE   AGE
sig-noc-www-deployment-no-ha                             1         1         1            1           5m
kubeadm@kube2-6:~/7TH-SIG-NOC$
```

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl describe deploy sig-noc-www-deployment-no-ha
Name:                   sig-noc-www-deployment-no-ha
Namespace:              default
CreationTimestamp:      Fri, 13 Apr 2018 17:45:45 +0200
Labels:                 app=sig-noc-www-front-end
                        availability=no-replica
                        version=v1
Annotations:            deployment.kubernetes.io/revision=1
Selector:               app=sig-noc-www-front-end,availability=no-replica,version=v1
Replicas:               1 desired | 1 updated | 1 total | 1 available | 0 unavailable
StrategyType:           RollingUpdate
MinReadySeconds:        0
RollingUpdateStrategy:  1 max unavailable, 1 max surge
Pod Template:
  Labels:       app=sig-noc-www-front-end
                availability=no-replica
                version=v1
  Containers:
   sig-noc-www-ctn:
    Image:      floui/tiny-net-tools
    Port:       8080/TCP
    Command:
      /bin/sh
    Args:
      -c
      while true; do { echo -e 'HTTP/1.1 200 OK
'; echo 'Hello 7TH SIG-NOC@Barcelona !'; } | nc -l -p  8080; done
    Environment:        <none>
    Mounts:             <none>
  Volumes:              <none>
Conditions:
  Type          Status  Reason
  ----          ------  ------
  Available     True    MinimumReplicasAvailable
OldReplicaSets: <none>
NewReplicaSet:  sig-noc-www-deployment-no-ha-4934366 (1/1 replicas created)
Events:
  FirstSeen     LastSeen        Count   From                    SubObjectPath   Type            Reason                  Message
  ---------     --------        -----   ----                    -------------   --------        ------                  -------
  6m            6m              1       deployment-controller                   Normal          ScalingReplicaSet       Scaled up replica set
sig-noc-www-deployment-no-ha-4934366 to 1
```

# Kubernetes service

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl get pod -o wide | egrep "NAME|sig-noc-www"
NAME                                                              READY     STATUS    RESTARTS   AGE       IP                 NODE
sig-noc-www-deployment-no-ha-4934366-mm665                        1/1       Running   0          17m       192.168.18.135     172.16.1.7
kubeadm@kube2-6:~/7TH-SIG-NOC$
```

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl describe pod sig-noc-www-deployment-no-ha-4934366-mm665
Name:           sig-noc-www-deployment-no-ha-4934366-mm665
Namespace:      default
Node:           172.16.1.7/172.16.1.7
Start Time:     Fri, 13 Apr 2018 17:45:45 +0200
Labels:         app=sig-noc-www-front-end
                availability=no-replica
                pod-template-hash=4934366
                version=v1
Annotations:    kubernetes.io/created-
by={"kind":"SerializedReference","apiVersion":"v1","reference":{"kind":"ReplicaSet","namespace":"default","name":"sig-noc-www-deployment-no-
ha-4934366","uid":"baa81c6a-3f31-11e8-...
Status:         Running
IP:             192.168.18.135
Controllers:    ReplicaSet/sig-noc-www-deployment-no-ha-4934366
Containers:
  sig-noc-www-ctn:
    Container ID:       docker://833667758e7126564574345f783378444ccfcc63bc24126398ef6260d35dc466
    Image:              floui/tiny-net-tools
    Image ID:           docker-pullable://floui/tiny-net-tools@sha256:f2089f227a19a6e880c63503abb678b53a9bcfbca3851c8d7de6dac1f716e2fd
    Port:               8080/TCP
    Command:
      /bin/sh
    Args:
      -c
      while true; do { echo -e 'HTTP/1.1 200 OK
'; echo 'Hello 7TH SIG-NOC@Barcelona !'; } | nc -l -p  8080; done
    State:              Running
      Started:          Fri, 13 Apr 2018 17:45:48 +0200
    Ready:              True
    Restart Count:      0
    Environment:        <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from default-token-6hbsr (ro)


…
… <output omitted for clarity>
```

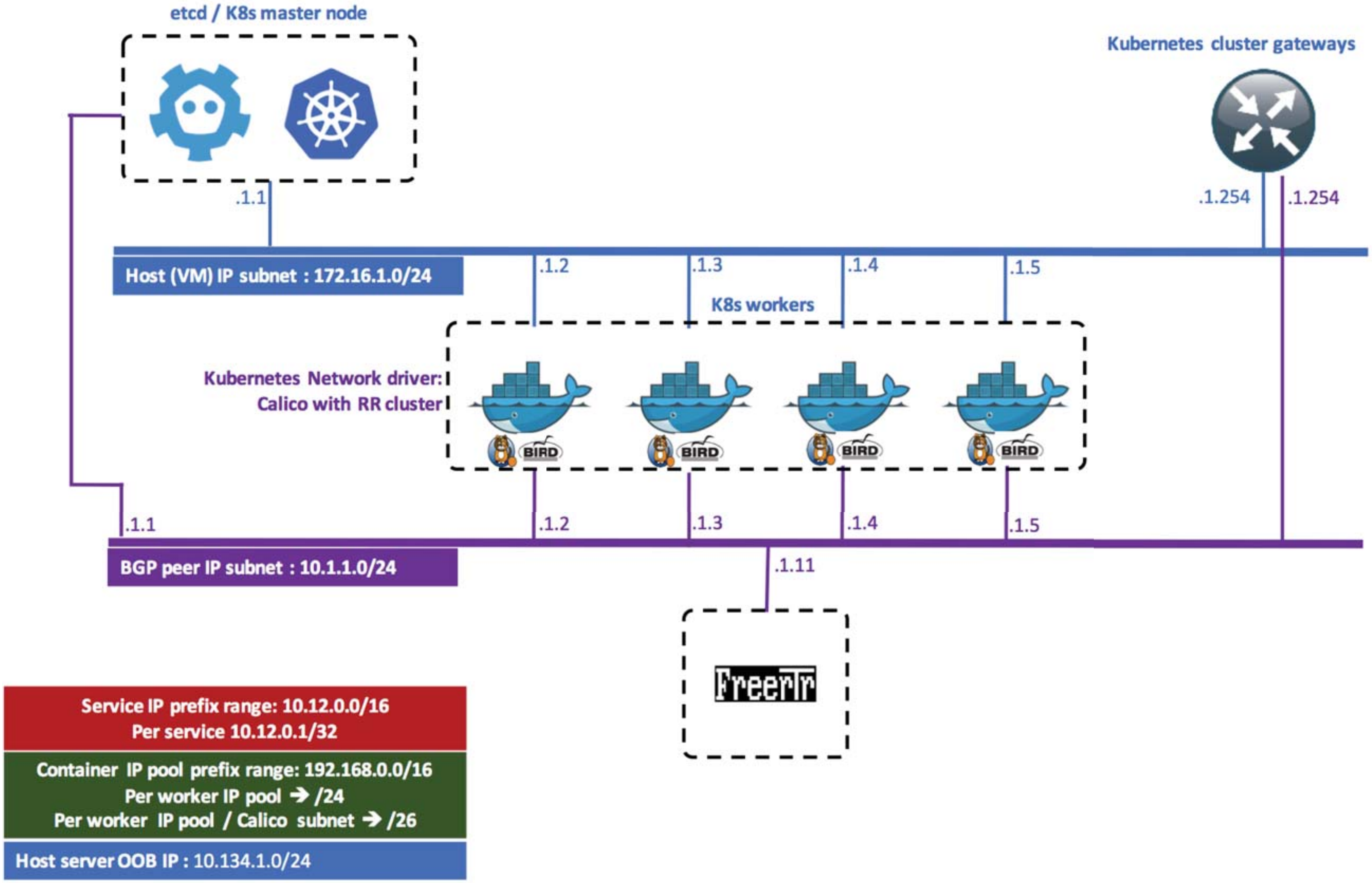
# Kubernetes service

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl exec -it sig-noc-pod-bastion curl http://sig-noc-www-service-no-ha
Hello 7TH SIG-NOC@Barcelona !
kubeadm@kube2-6:~/7TH-SIG-NOC$
```

```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl exec -it sig-noc-pod-bastion curl http://10.13.158.214:80
Hello 7TH SIG-NOC@Barcelona !
kubeadm@kube2-6:~/7TH-SIG-NOC$
```

```
kubeadm@kube2-7:~$ sudo iptables-save | grep sig-noc-www
-A KUBE-SEP-V6XS3NIQUIQ4JZQR -s 192.168.18.135/32 -m comment --comment "default/sig-noc-www-service-no-ha:" -j KUBE-MARK-MASQ
-A KUBE-SEP-V6XS3NIQUIQ4JZQR -p tcp -m comment --comment "default/sig-noc-www-service-no-ha:" -m tcp -j DNAT --to-destination
192.168.18.135:8080
-A KUBE-SERVICES -d 10.13.158.214/32 -p tcp -m comment --comment "default/sig-noc-www-service-no-ha: cluster IP" -m tcp --dport 80 -j KUBE-
SVC-HUEFJ5RUVO2FDIQ4
-A KUBE-SVC-HUEFJ5RUVO2FDIQ4 -m comment --comment "default/sig-noc-www-service-no-ha:" -j KUBE-SEP-V6XS3NIQUIQ4JZQR
kubeadm@kube2-7:~$
```

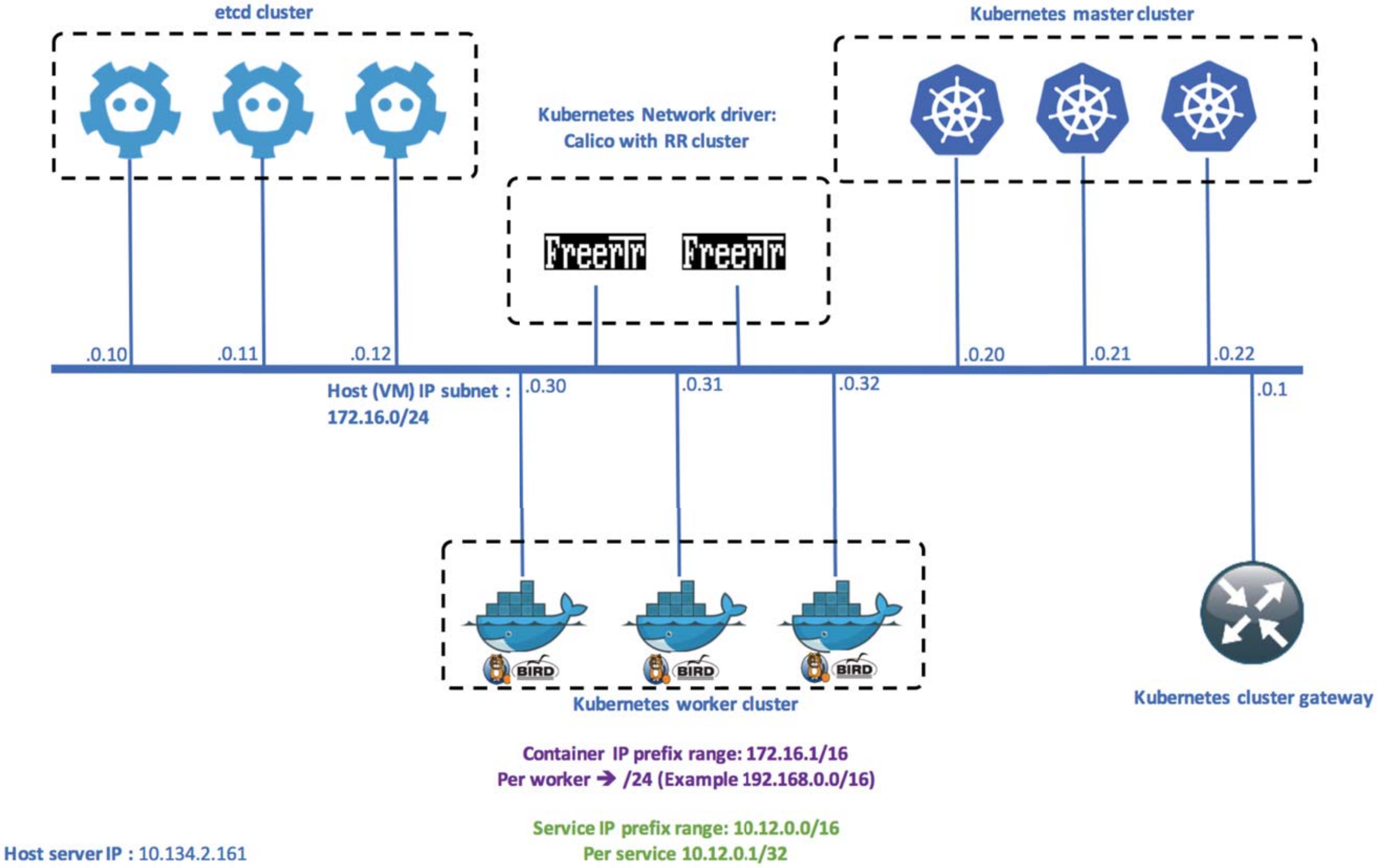
```
kubeadm@kube2-6:~/7TH-SIG-NOC$ kubectl get pod -o wide | egrep "NAME|sig-noc-www"
NAME                                                              READY     STATUS    RESTARTS   AGE       IP                NODE
sig-noc-www-deployment-no-ha-4934366-mm665                        1/1       Running   0          17m       192.168.18.135    172.16.1.7
kubeadm@kube2-6:~/7TH-SIG-NOC$
```



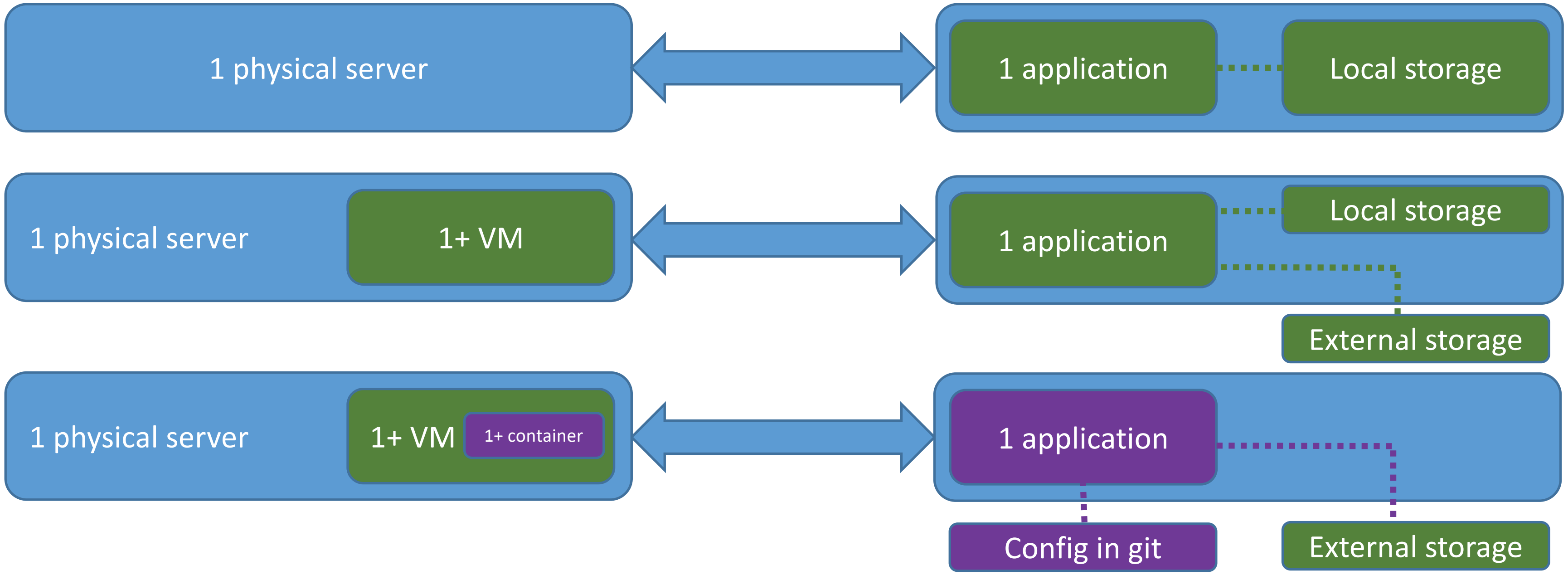
Service IP prefix range: 10.12.0.0/16
Per service 10.12.0.1/32

Container IP pool prefix range: 192.168.0.0/16
Per worker IP pool → /24
Per worker IP pool / Calico subnet → /26

Host server OOB IP : 10.134.1.0/24


etcd cluster
Kubernetes Network driver:
Calico with RR cluster
Kubernetes master cluster
FreeRtr
FreeRtr
.0.10
.0.11
.0.12
.0.20
.0.21
.0.22
Host (VM) IP subnet :
172.16.0/24
.0.30
.0.31
.0.32
.0.1
BIRD
BIRD
BIRD
Kubernetes worker cluster
Kubernetes cluster gateway
Container IP prefix range: 172.16.1/16
Per worker ➔ /24 (Example 192.168.0.0/16)
Service IP prefix range: 10.12.0.0/16
Per service 10.12.0.1/32
Host server IP : 10.134.2.161

# Conclusion


1 physical server
1 application
Local storage
1 physical server
1+ VM
1 application
Local storage
External storage
1 physical server
1+ VM
1+ container
1 application
Config in git
External storage

# Key take away

GÉANT

- Example of small companies in US
  - Large K8s deployment
    - 25 clusters with 7500 nodes
    - Plan to move to 40K nodes by Q4 2017
- Google's lesson's learned
  - **Kubernetes Scaling and Performance Goals**
    - https://github.com/kubernetes/community/blob/master/sig-scalability/goals.md
    - Max core per cluster 200 000
    - Max pod per core 10
    - Management overhead per node Goal: <5%, with a minimum of 0.5 core, 1GB RAM
    - Management overhead per cluster Goal: <1%, with a minimum of 2 cores, 4GB RAM

# Key take away

- Type of network architecture:
  - 1 AS per rack design
  - 1 AS per node design
  - Horizontal scaling by adding rack or node design
  - ToR switch as RR within the cluster and in datapath
- Managing Kubernetes  clusters
  - Require solid expertise already in place within NREN
  - DCI impact on network backbone equipment
- Kubernetes 1.6
  - Federation
  - Taint/Affinity features
- Impact on NREN organization
  - Learning curve
  - Process change
  - IT landscape drastic transformation

## Key take away

# Get interests ?

# Join GN4-2 JRA2-T5 the effort !

- Tell us what you think
- Port **YOUR** application our platform
- **Register** to be a pilot ?
- Start working with GN4-2 JRA2-T5 ?

**Join us during GN4-3 ?**

# 7th SIG-NOC

Special Interest Group Network Operation Control
hosted by CSUC - Barcelona

GÉANT

## Thank you

This work is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 731122 (GN4-2).