



MIRANTIS

Agnostic intro to K8s without BS
PART 2

Fabrizio Soppelsa

28.03.2017

Part 2 AGENDA

- K8s recap
- Form a K8s cluster
- Form an etcd HA cluster
- Form an HA K8s cluster
- K8s trends and criticalities

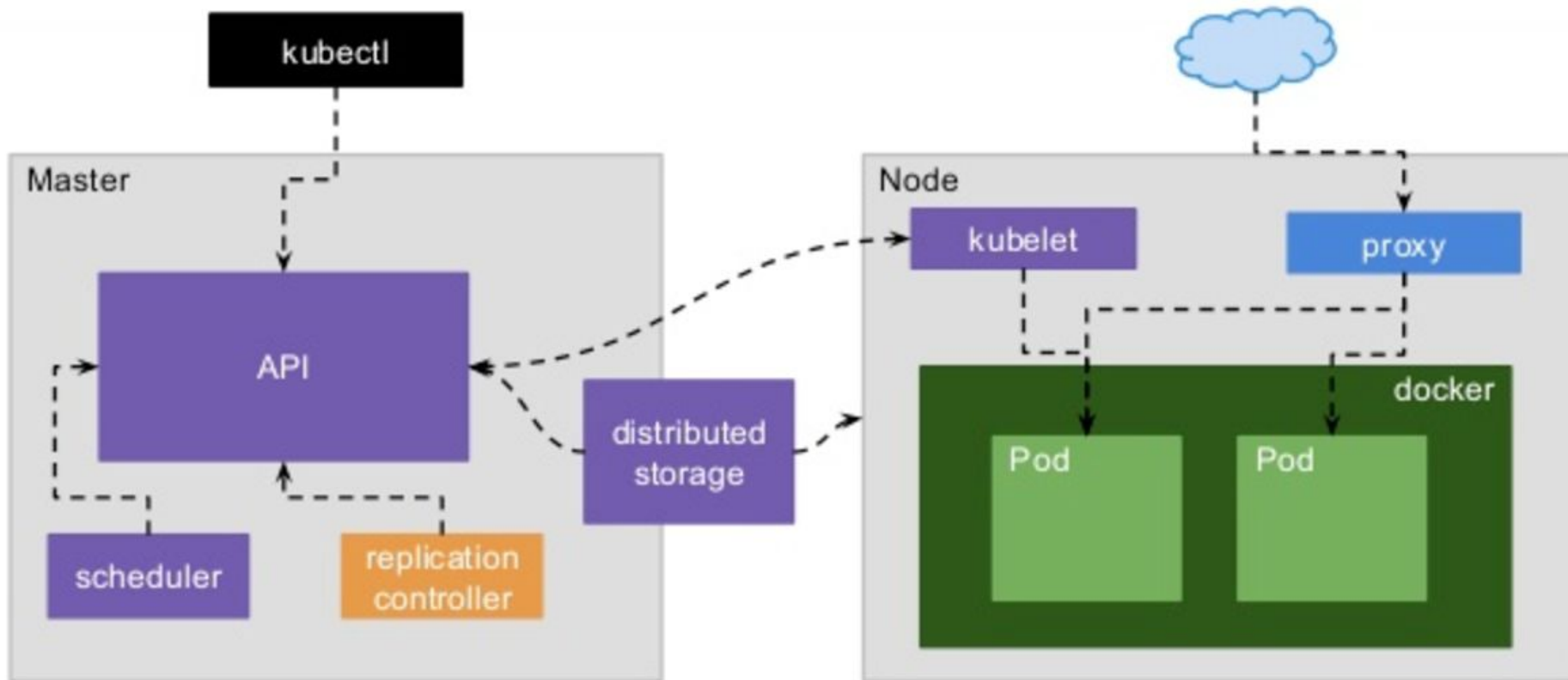
Kubernetes structure

- Kubernetes **master**
- Kubernetes **nodes**
- **etcd**
- Some SDN
- Linux as a base host
- Typically uses Docker (but...)

Recap - Kubernetes concepts

- **Pods** are the smallest deployable unit
- A **replication controller** governs pods replicas and scale
- A **service** may describe a multi-tier app
- YAML everywhere
- `kubectl [get, create, delete, describe, ...]`

Kubernetes in a picture



Form a K8s cluster

Manual setup

How to form a K8s cluster

1. Provision the nodes with a Linux distro
 - a. 1 etcd server
 - b. 1 master
 - c. 2 nodes
2. Install etcd on the etcd server
3. Install and configure a **SDN** on master and nodes (requires Etcd)
4. Install the K8s master
 - a. **kube-apiserver**
 - b. **kube-scheduler**
 - c. **kube-controller-manager**
5. Install the 2 K8s nodes
 - a. **Docker daemon**
 - b. **kubelet**
 - c. **kube-proxy**
6. Configuration

Etcd concepts

Etcd is a highly available, distributed, and consistent key-value store that is used for shared configuration and service discovery.

- **Highly available**
- **Distributed**
- **Key-value store**
- **RAFT**
 - Consensus (value agreement)
 - Leader election
 - Replicated state machine
- **Splits tolerance**

RAFT in practice

1. Get a Docker compose file
 - a. <https://github.com/fsoppelsa/raftexample/blob/master/raftexample.yaml>
 - b. Uses fsoppelsa/raftexample
2. Create a local RAFT
 - a. `docker-compose -f raftexample.yaml up`
3. Set values, kill containers, restart, observe

```
curl -L http://127.0.0.1:9121/testkey -XPUT -d value
```

```
curl -L http://127.0.0.1:9121/testkey
```

Form a K8s cluster - a look to the SDN

1. Net communication between containers is the hardest part
2. 1 host: Trivial: Docker bridge
3. But K8s has many nodes
4. Solution: Use (VxLAN) **overlay**

```
docker network create -d overlay ...
```

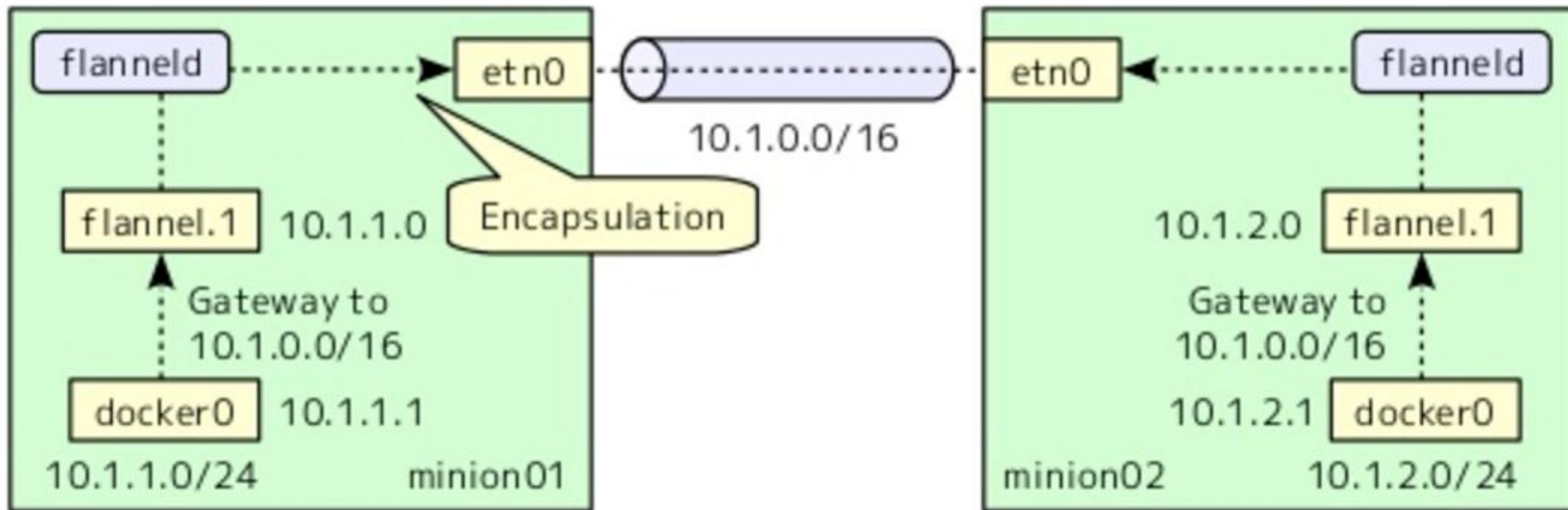
Or use overlay containers technologies:

- Flannel by CoreOS
- Calico (used to isolate/secure)
- Open vSwitch

Flannel (or Calico)

Idea

- **Minions** communicate on a physical network
- **Containers/Pods** communicate on the overlay network: Flannel assigns non-overlapping subnets and gateways the Docker bridge through an interface *flannel.1*. Packets are transferred using VxLAN.



Form a K8s cluster - Configurations

1. Flannel - [interface via etcdctl](#)
2. Master node
3. Nodes

I find [this document](#) very useful

Form an HA K8s cluster

Manual setup
Kargo

Form an etcd HA cluster/1

<https://coreos.com/etcd/docs/latest/op-guide/clustering.html>

1. Provision 3 Linux bare or virtual nodes
2. Install etcd
 - `yum install etcd`
3. Initialize the cluster
 - Modify `/etc/etcd/etcd.conf` on each node
4. Start **etcd1**
 - `systemctl start etcd`
5. Start **etcd2**
6. Start **etcd3**
7. Verify the cluster health
 - `etcdctl cluster-health`

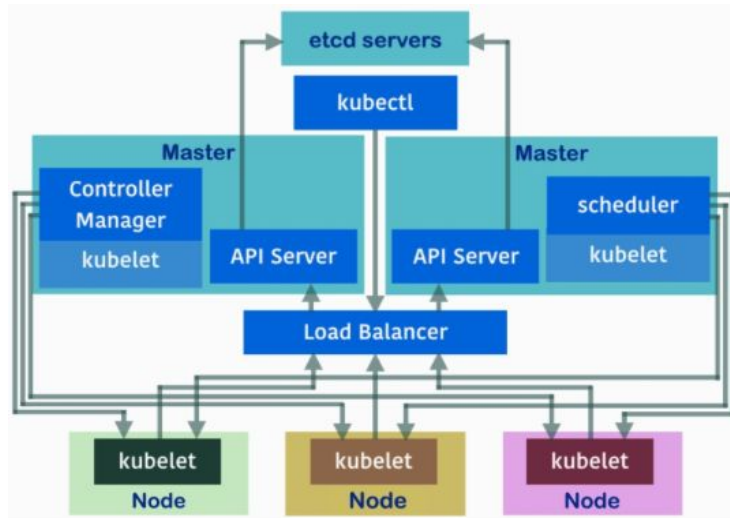
Form an etcd HA cluster/2

/etc/etcd/etcd.conf on node etcd1

```
ETCD_NAME=etcd1
ETCD_DATA_DIR="/var/lib/etcd/etcd1.etcd"
ETCD_LISTEN_PEER_URLS="http://172.28.128.7:2380"
ETCD_LISTEN_CLIENT_URLS="http://0.0.0.0:2379"
ETCD_INITIAL_ADVERTISE_PEER_URLS="http://172.28.128.7:2380"
ETCD_INITIAL_CLUSTER="etcd1=http://172.28.128.7:2380,etcd2=http://172.28.128.8:2380,etcd3=http://172.28.128.9:2380"
ETCD_INITIAL_CLUSTER_STATE="new"
ETCD_INITIAL_CLUSTER_TOKEN="etcd-cluster"
ETCD_ADVERTISE_CLIENT_URLS="http://172.28.128.7:2379"
```

Form a Kubernetes HA cluster

- More masters
- Big news: no quorum, but need of an HTTP **Load Balancer** (i.e. Nginx)
- It's completely fine to have 2 masters
- Every master runs its own API server
- But there is **one and only** one Scheduler and Controller Manager running at one moment
- Trend: Install Kubelets on Masters too - to containerize Controller Manager and Scheduler



Kargo and around

- Kargo has/had Mirantis and Other company which I don't really like headquartered in Mountain View traction
- In **kubernetes-incubator**
- Installer based on Ansible
- What happened: Proliferation of wrappers and tools
- So: People started using just Ansible plays...
- So: People do things manually for now
- The next big thing: Helm (check it out [on our corporate blog](#))

K8s “criticalities” at the day

- No official installers (but...)*
- Etcd to be run as an external service (but...)
- Heavy-fast-developed. Now we're like around OpenStack Essex (but...)
- Complex architecture (but...)
- Eating the orchestration market (Fleet dead; only Docker Swarm can resist)
- K8s and OpenStack: COE in Magnum (dead), object in Murano (dead)

Thanks

Q&A