CHINA OpenStack Days

Topic: The Future of Containers in OpenStack

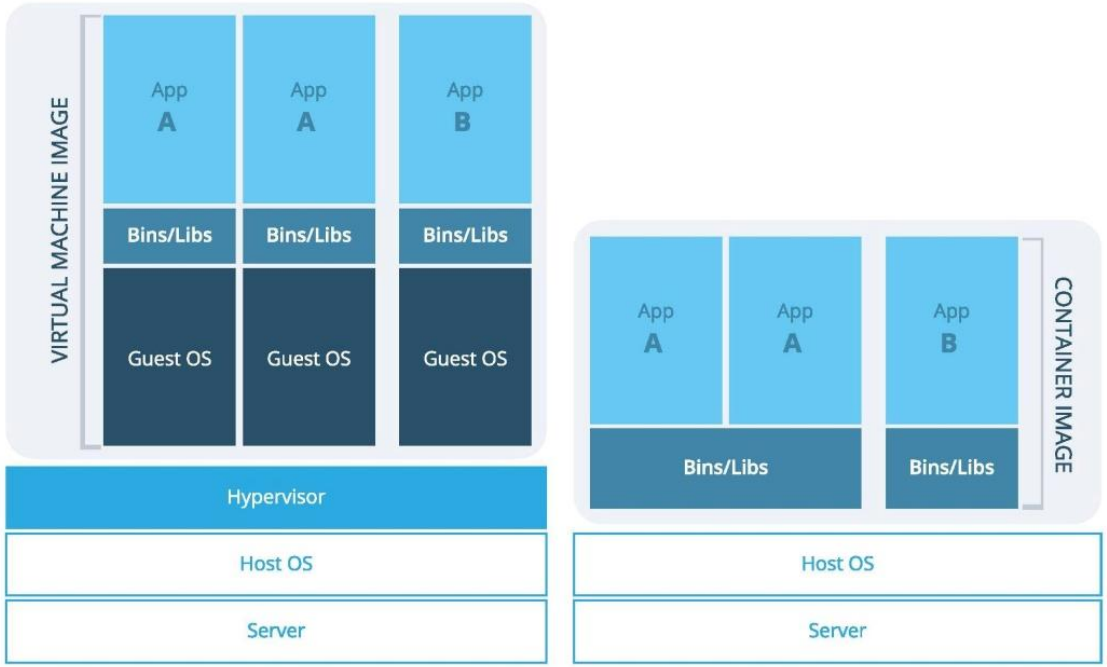
Name: Hongbin Lu
Title: OpenStack Zun PTL

Agenda

- Introduction
- Containers on OpenStack
- Solutions
  - Magnum
  - Zun
  - Kuryr
- The Future

Introduction – What is Container

- Provide an isolated environment for running applications
- Like a VM, but not a VM
  - Split compute resources
  - Share kernel
- Enabling technologies
  - Linux namespaces
  - Cgroups

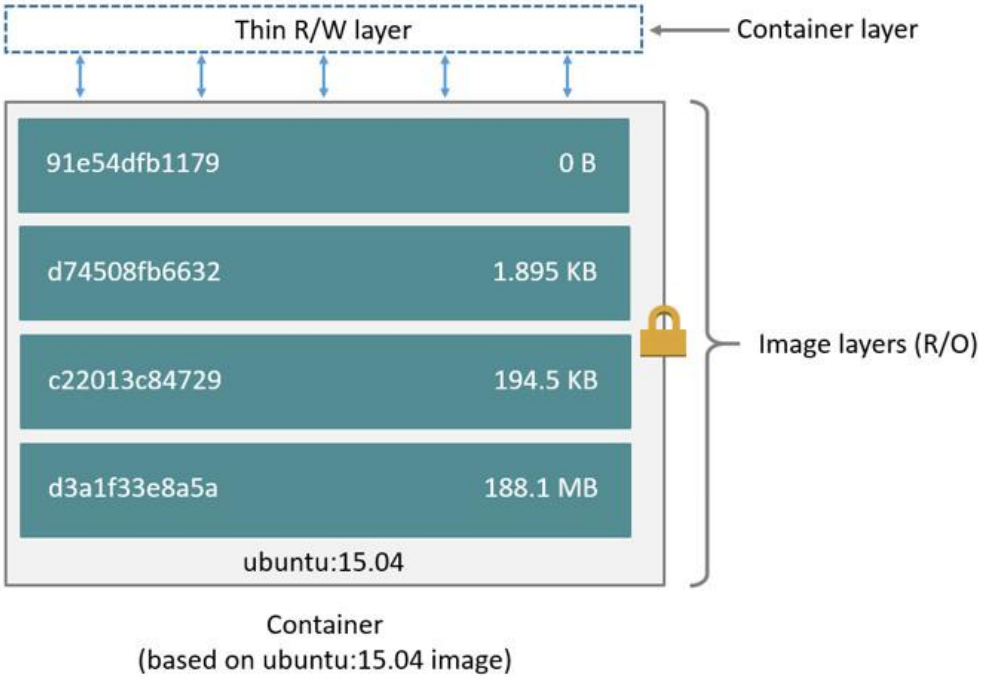


Introduction – Why containers

- Lightweight
  - Small in size
  - Boot very fast
  - Less resource consumption
- Portable
  - Container can run on almost anywhere with a Linux OS.

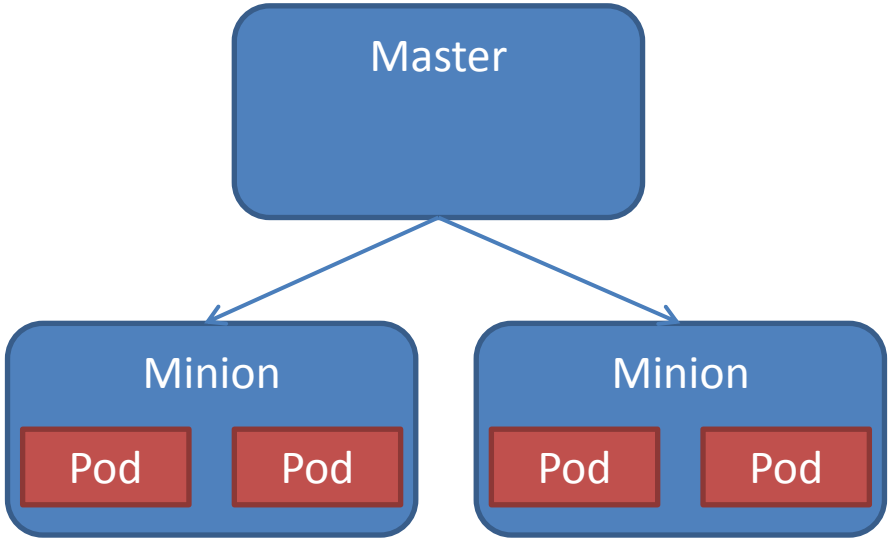
Introduction - Docker

- A tool
  - Package application
  - Deploy the application
- DockerHub
  - Marketplace of packages
- Docker image
  - A tar ball
  - Contain root file system of container
  - Layering
  - Shared by all containers
  - Read-only
  - Read from a layer
  - Copy-on-write



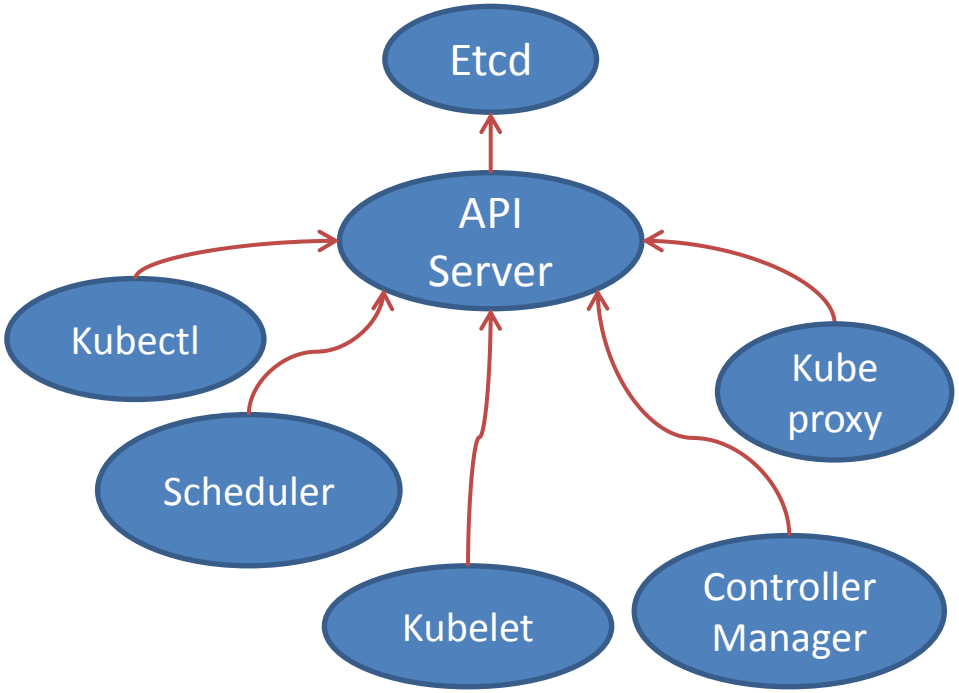
Introduction – Kubernetes

- A platform to deploy containerized applications
- Abstractions to describe applications
- Manage application lifecycle
- Key concepts
  - Pod
  - Master node
  - Minion node



Introduction – Kubernetes Architecture

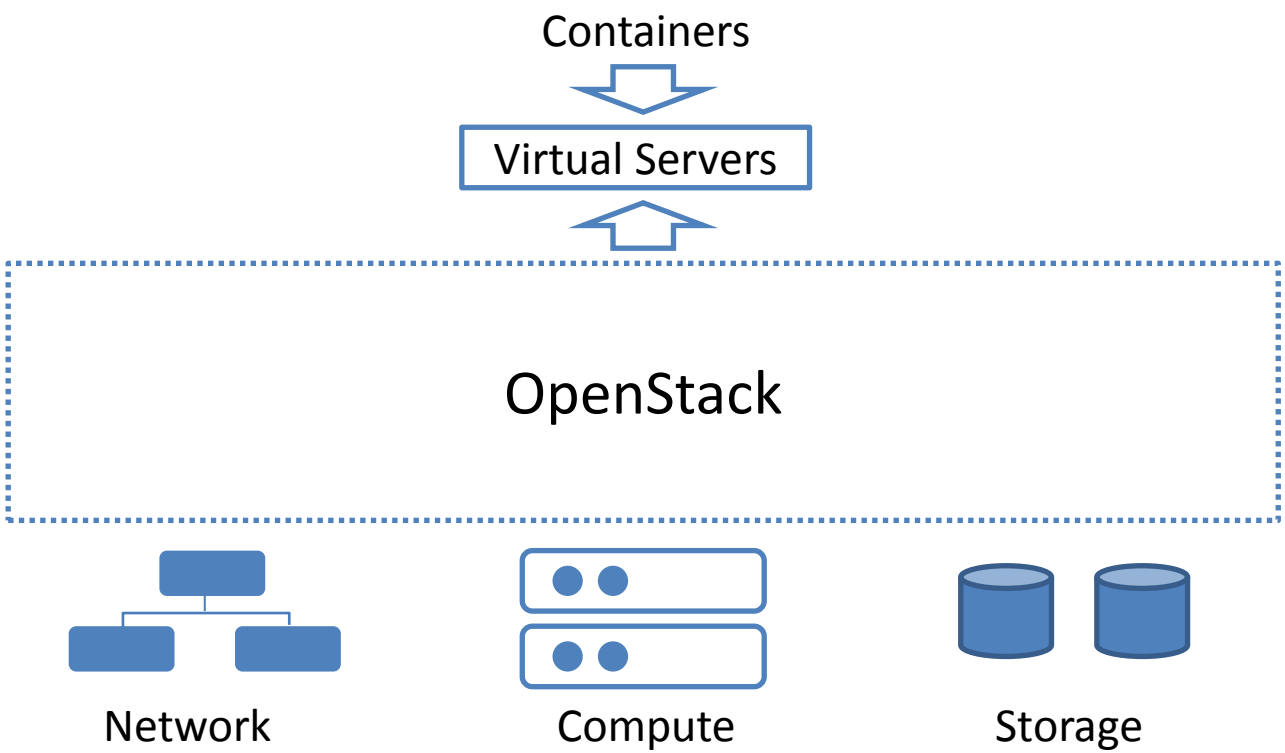
- Etcd is the cluster store
- API Server is a proxy to access etcd
- End-users use kubectl to access the API
- Scheduler watches API for unscheduled pods
- Kubelet watches API for pods scheduled to its host
- Controller Manager watches API and perform business logic
- Kube-proxy realizes the service concept



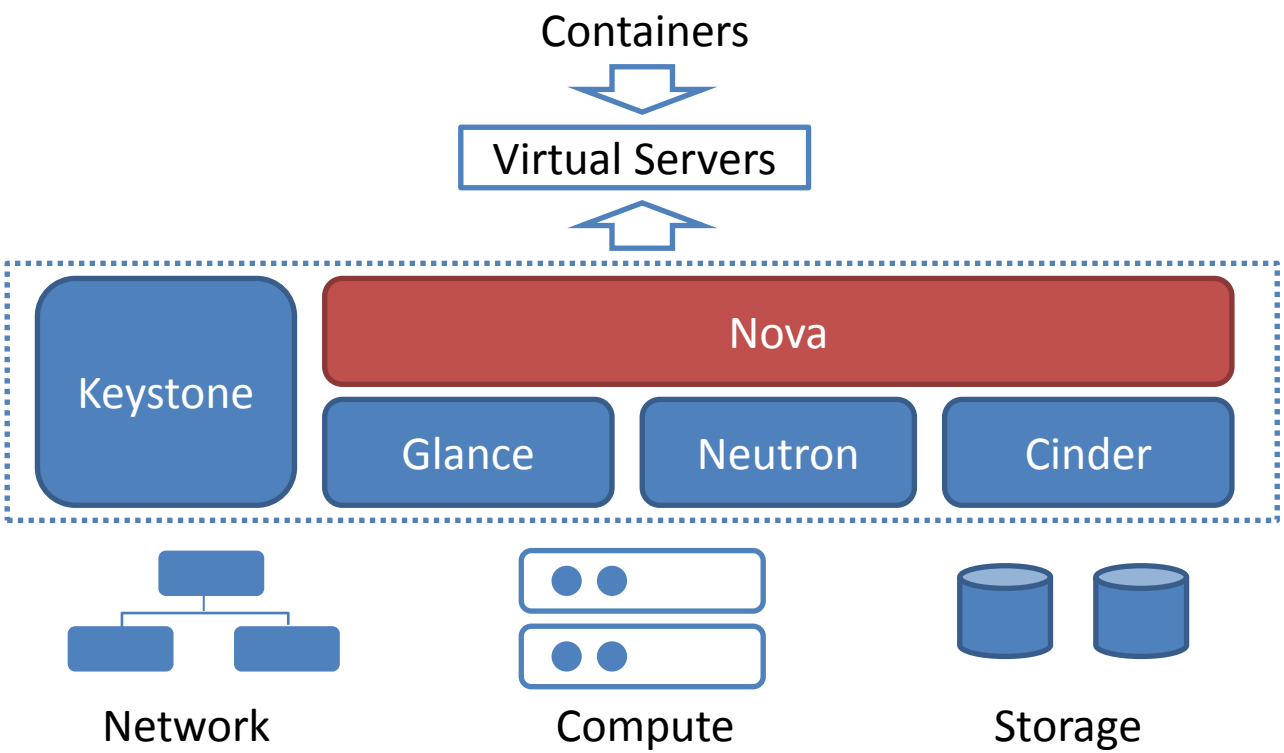
Agenda

- Introduction
- Containers on OpenStack
- Solutions
  - Magnum
  - Zun
  - Kuryr
- The Future

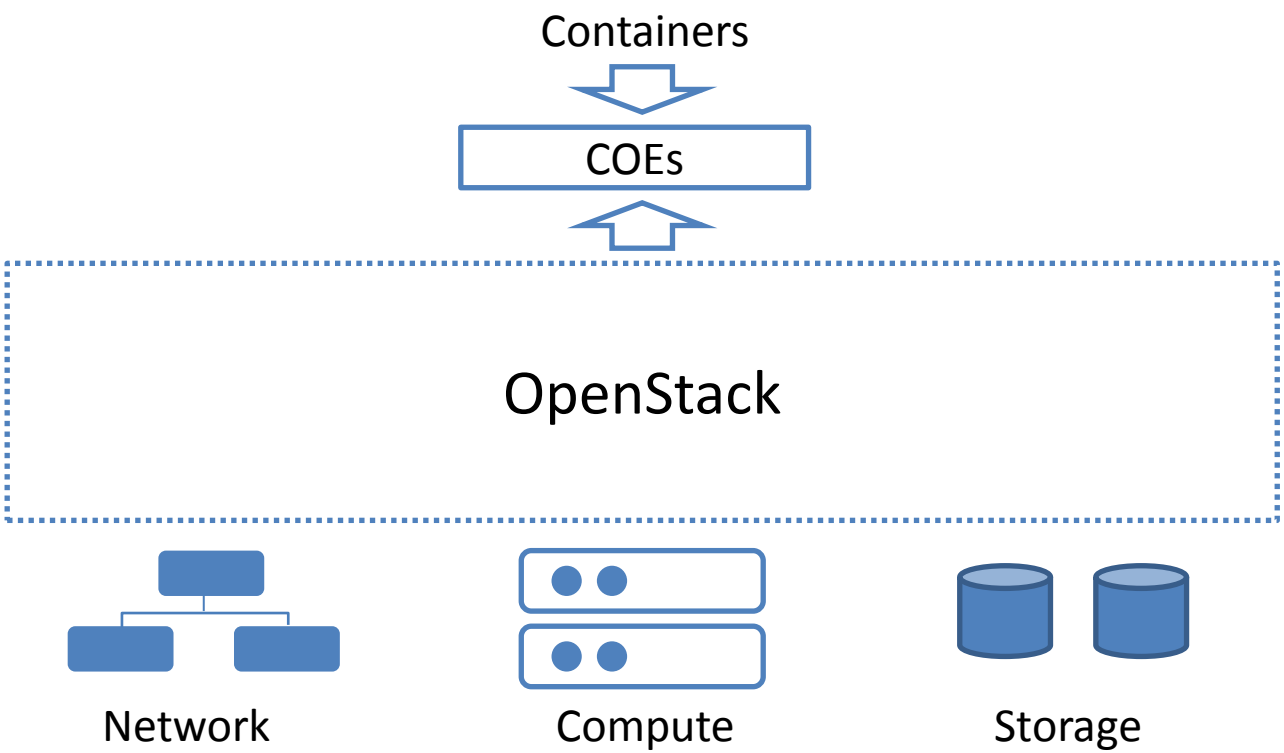
Containers on OpenStack - 1



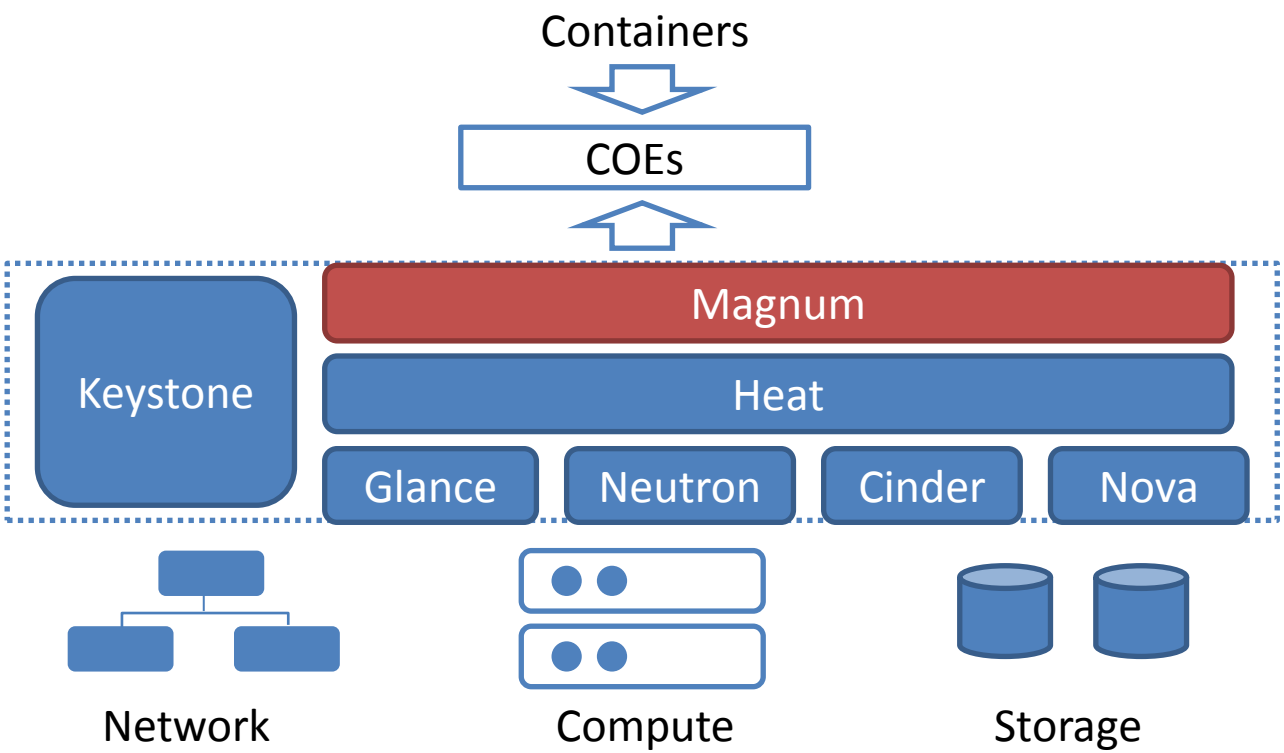
Containers on OpenStack - 1



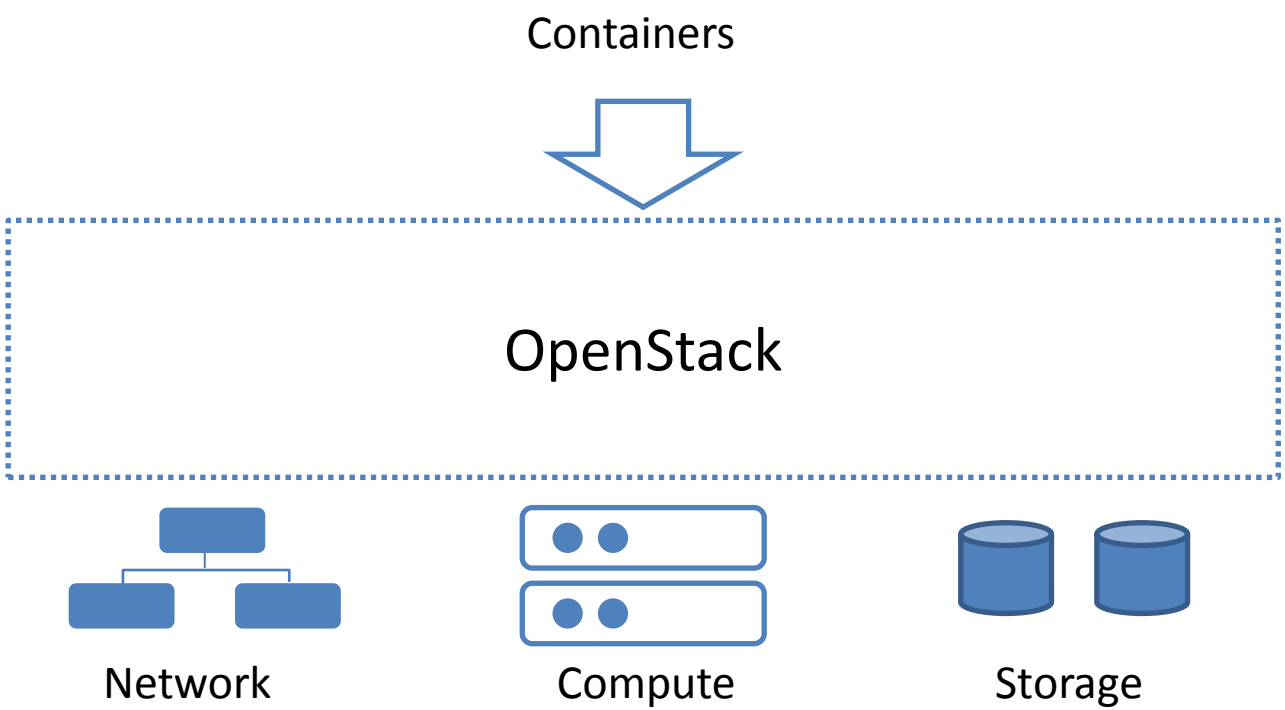
Containers on OpenStack - 2



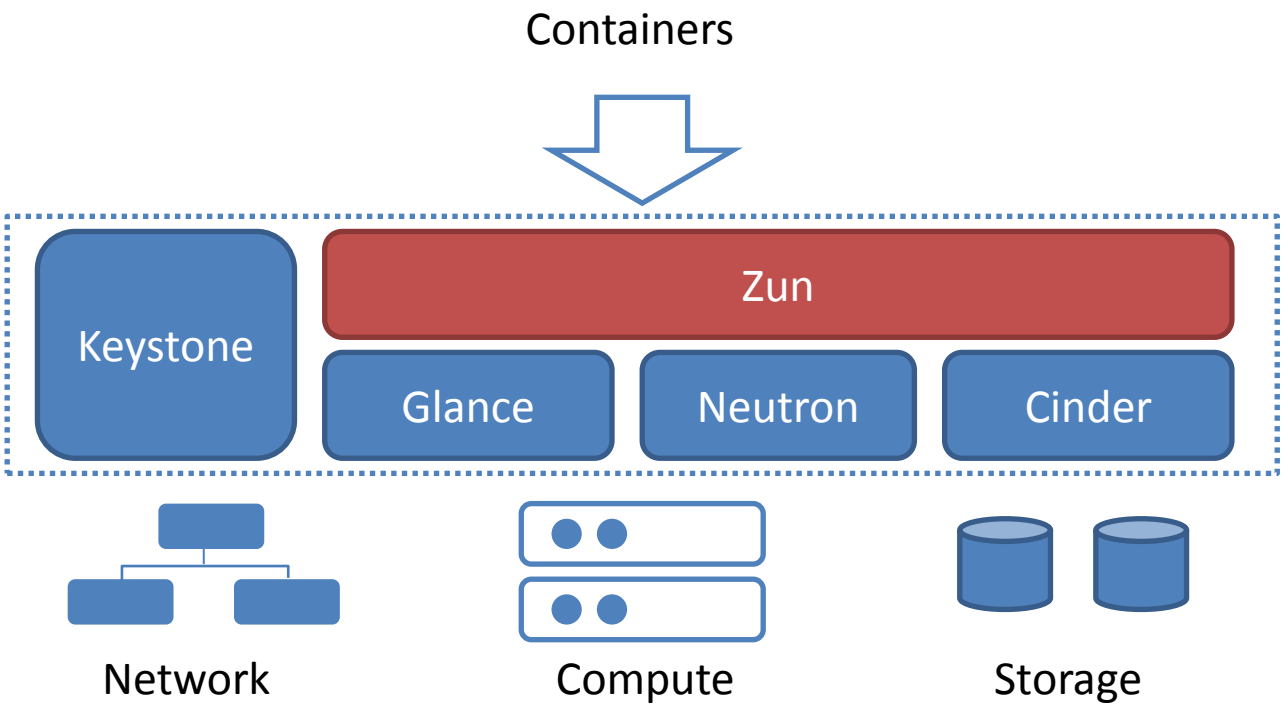
Containers on OpenStack - 2



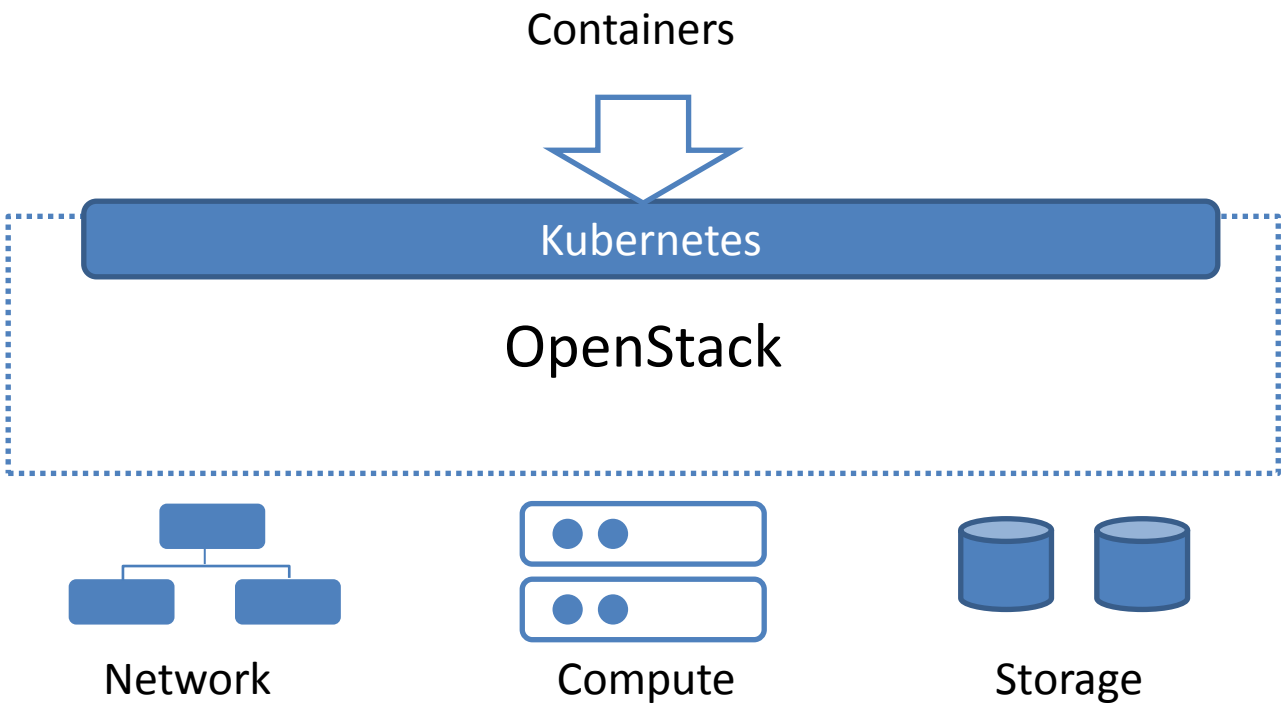
Containers on OpenStack - 3



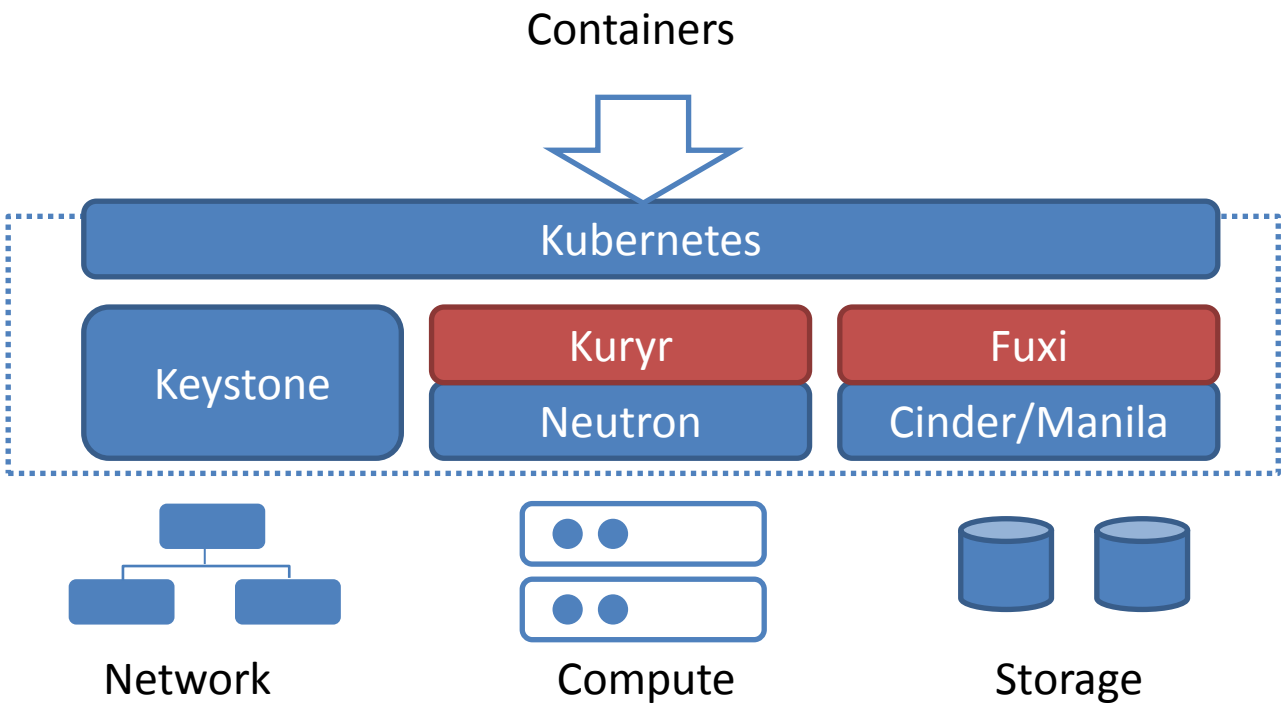
Containers on OpenStack - 3



Containers on OpenStack - 4



Containers on OpenStack - 4

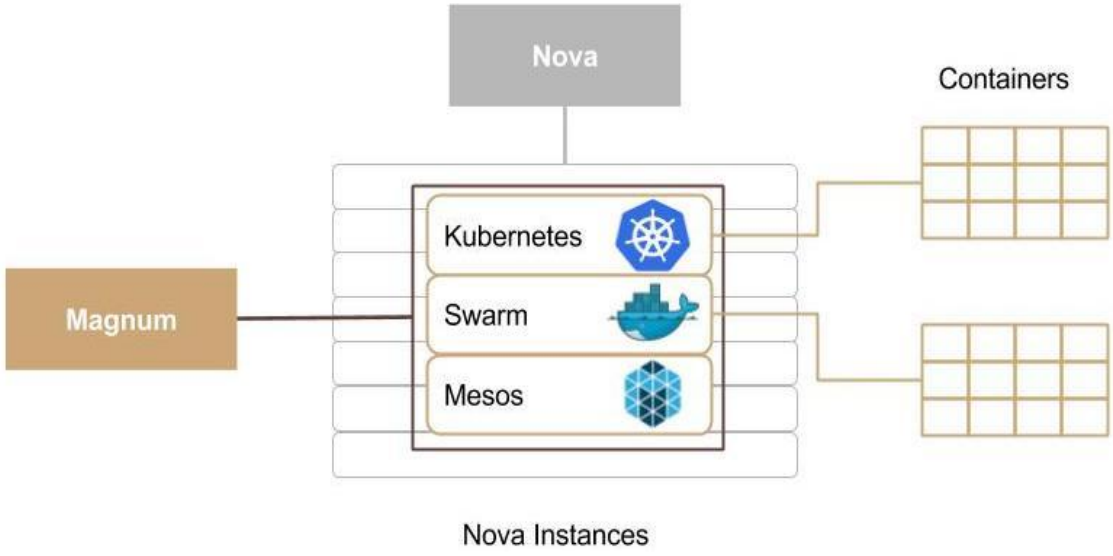


Agenda

- Introduction
- Containers on OpenStack
- Solutions
  - Magnum
  - Zun
  - Kuryr
- The Future

Magnum - Overview

- First container project in OpenStack
- Provide an API service for provisioning COEs on Nova instances
- Supported COEs:
  - Kubernetes
  - Swarm
  - Mesos
- Containers are managed by COEs

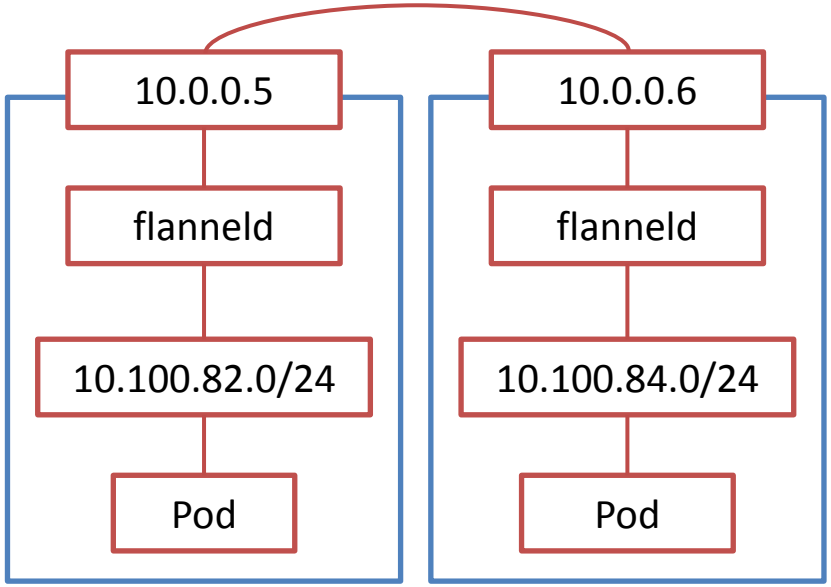


Magnum - Workflow

```
$ # Create a k8s cluster
$ magnum cluster-create k8s-cluster \
        --cluster-template k8s-cluster-template \
        --node-count 1

$ # Configure kubectl to consume to the cluster
$ magnum cluster-show k8s-cluster
$ magnum ca-sign --cluster k8s-cluster --csr client.csr > client.crt
$ magnum ca-show --cluster k8s-cluster > ca.crt
$ eval $(magnum cluster-config k8s-cluster)

$ # Use the k8s cluster
$ kubectl create -f ./redis-master.yaml
```

Magnum - Network

- Overlay network solution
- Flannel for pod-to-pod communications
  - IP-per-pod
  - Double-encapsulation
- Neutron LBaaS for external-to-pod communications
  - Expose Pod via Kubernetes service
  - Create a Neutron load balancer for each Kubernetes Service



Magnum - Storage

- Kubernetes
  - **PersistentVolumeClaim (PVC)**: A request for storage by a user
  - **PersistentVolume (PV)**: A piece of storage in the cluster that has been provisioned by an administrator
- OpenStack integration
  - Cinder volume as PV
  - Provision Cinder volumes
  - Mount Cinder volumes to pods

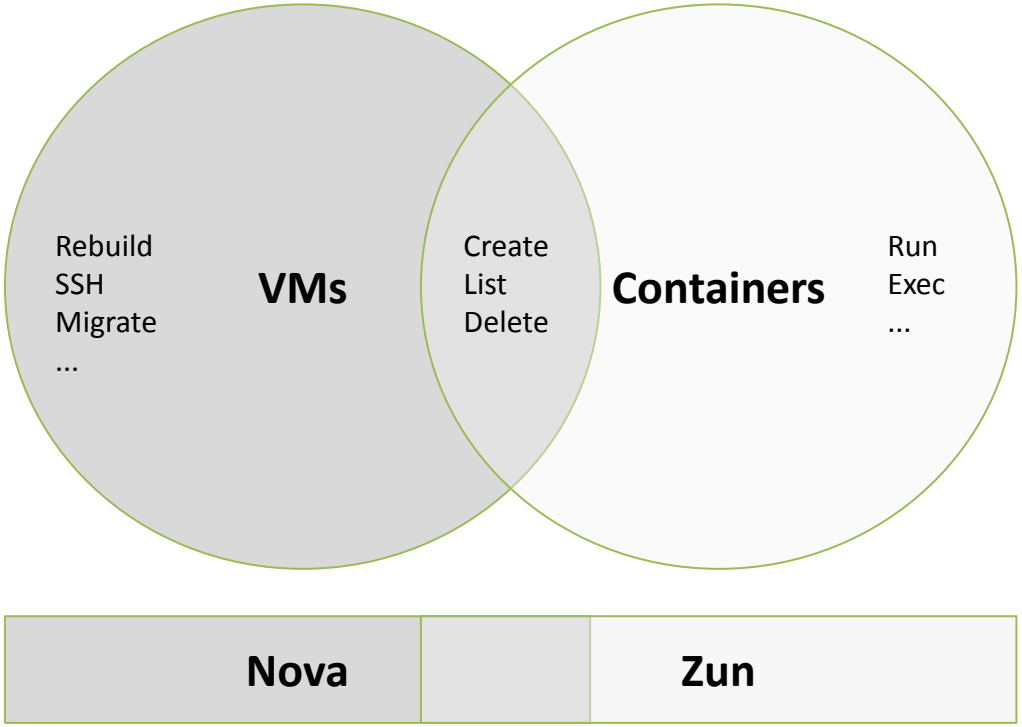
Agenda

- Introduction
- Containers on OpenStack
- Solutions
  - Magnum
  - Zun
  - Kuryr
- The Future

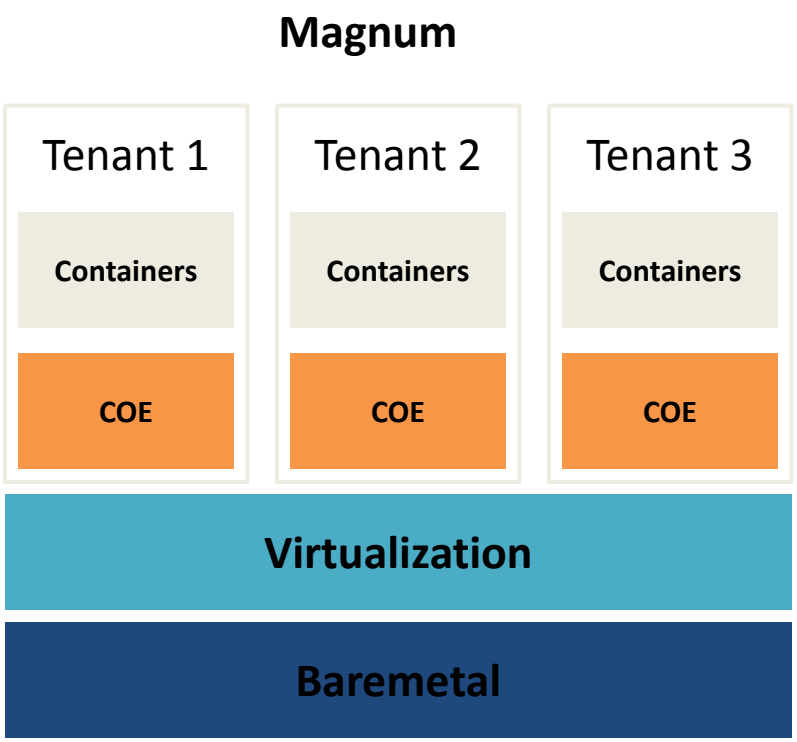
Zun – Overview

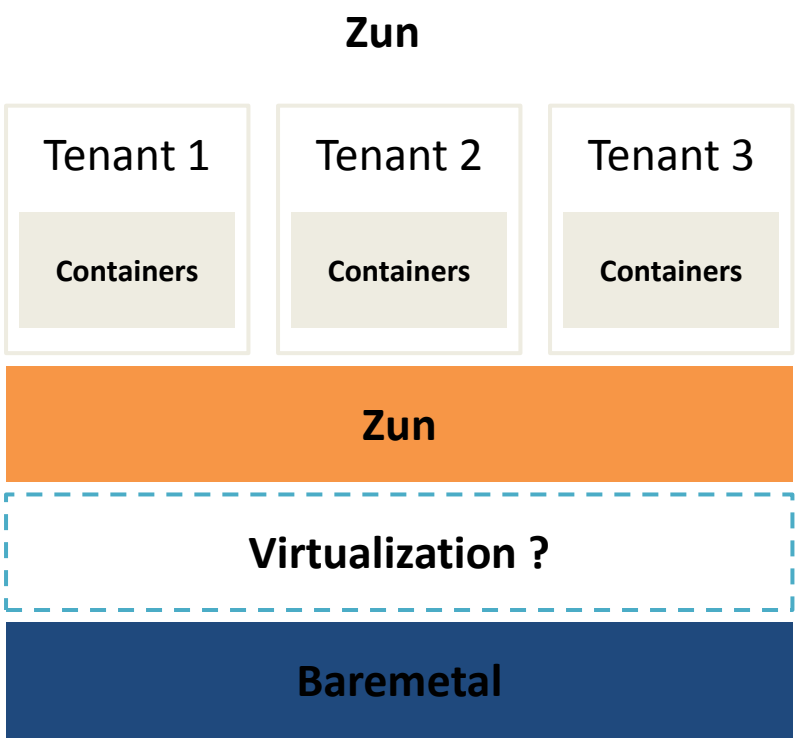
- A new container project
- Provide an API service for provisioning containers on OpenStack
- Container engine
  - Docker
  - (more to come)
- Container Host
  - Baremetal
  - Nova instance (coming soon)
  - COE (coming soon)
- Container network: Neutron
- Container volume: Cinder/Manila

Why Zun


Rebuild
SSH
Migrate
...
VMs
Create
List
Delete
Containers
Run
Exec
...
Nova
Zun

Why Zun


Magnum
Tenant 1
Tenant 2
Tenant 3
Containers
Containers
Containers
COE
COE
COE
Virtualization
Baremetal


Zun
Tenant 1
Tenant 2
Tenant 3
Containers
Containers
Containers
Zun
Virtualization ?
Baremetal

Zun - Workflow

```
$ # Run a container
$ docker search cirros
$ zun run --name test-container cirros ping 8.8.8.8

$ # Retrieve the log of the container
$ zun logs test-container
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: seq=0 ttl=40 time=25.513 ms
64 bytes from 8.8.8.8: seq=1 ttl=40 time=25.348 ms
64 bytes from 8.8.8.8: seq=2 ttl=40 time=25.226 ms
64 bytes from 8.8.8.8: seq=3 ttl=40 time=25.275 ms

$ # Execute another command
$ zun exec test-container ls -a
...
```

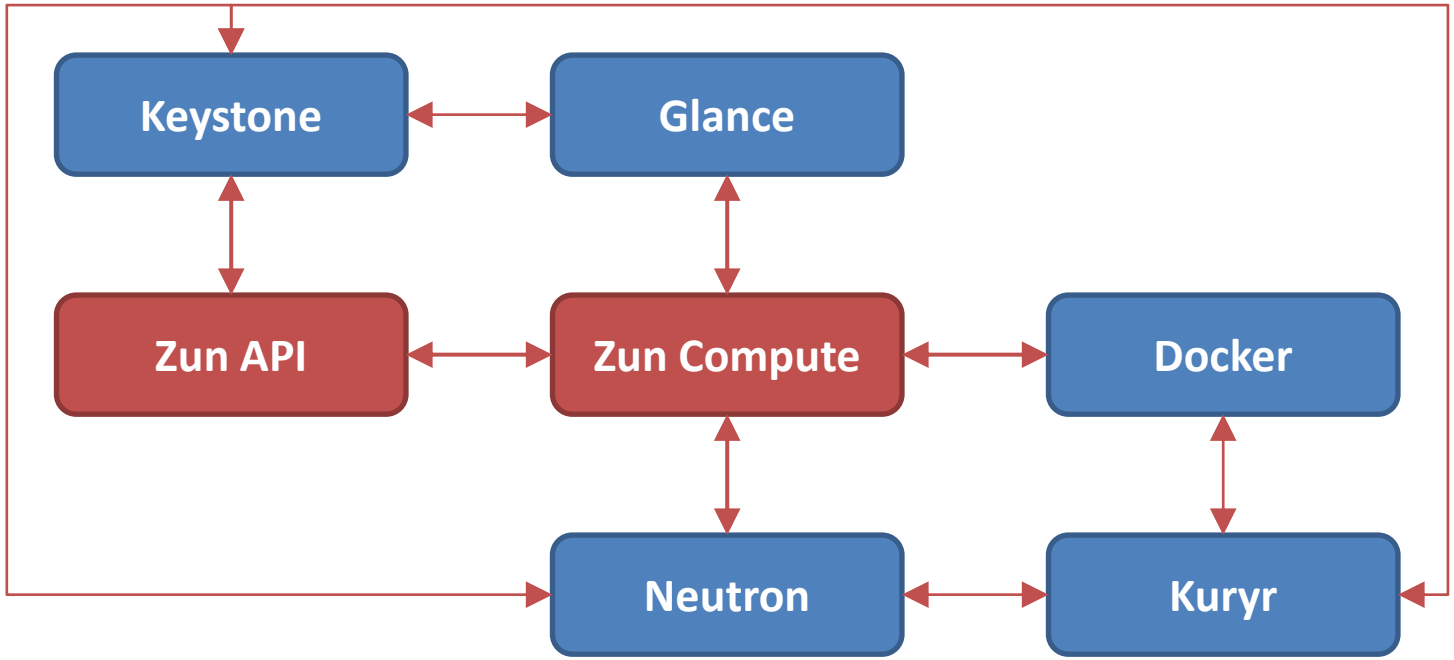
Zun – Practical Example

```
$ # Run a database
$ zun run --name mydb \
      --environment MYSQL_DATABASE=wordpress \
      --environment MYSQL_ROOT_PASSWORD=rootpass \
      mysql:latest

$ # Retrieve the IP address of the database
$ zun show mydb

$ zun run \
      --environment WORDPRESS_DB_HOST=<mydb_ip_address> \
      --environment WORDPRESS_DB_USER=root \
      --environment WORDPRESS_DB_PASSWORD=rootpass \
      wordpress:latest
```

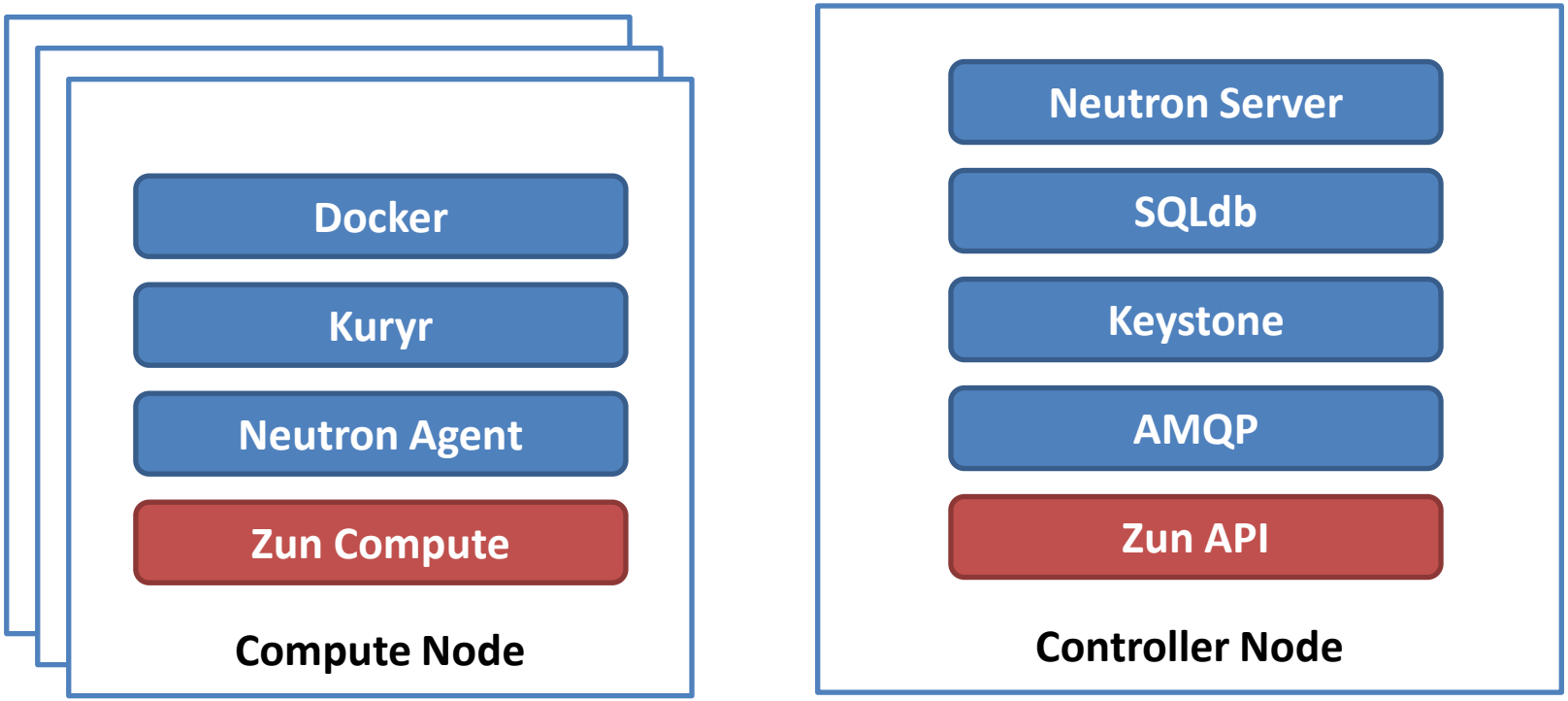
Zun – Heat Integration

```yaml
resources:
 db:
   type: OS::Zun::Container
   properties:
     image: mysql
     environment:
       MYSQL_ROOT_PASSWORD: rootpass
       MYSQL_DATABASE: wordpress
 wordpress:
   type: OS::Zun::Container
   properties:
     image: "wordpress:latest"
     environment:
       WORDPRESS_DB_HOST: {get_attr: [db, addresses, private, 0, addr]}
       WORDPRESS_DB_USER: root
       WORDPRESS_DB_PASSWORD: rootpass
```

Zun - Architecture



Zun - Deployment



Agenda

- Introduction
- Containers on OpenStack
- Solutions
  - Magnum
  - Zun
  - Kuryr
- The Future

Kuryr

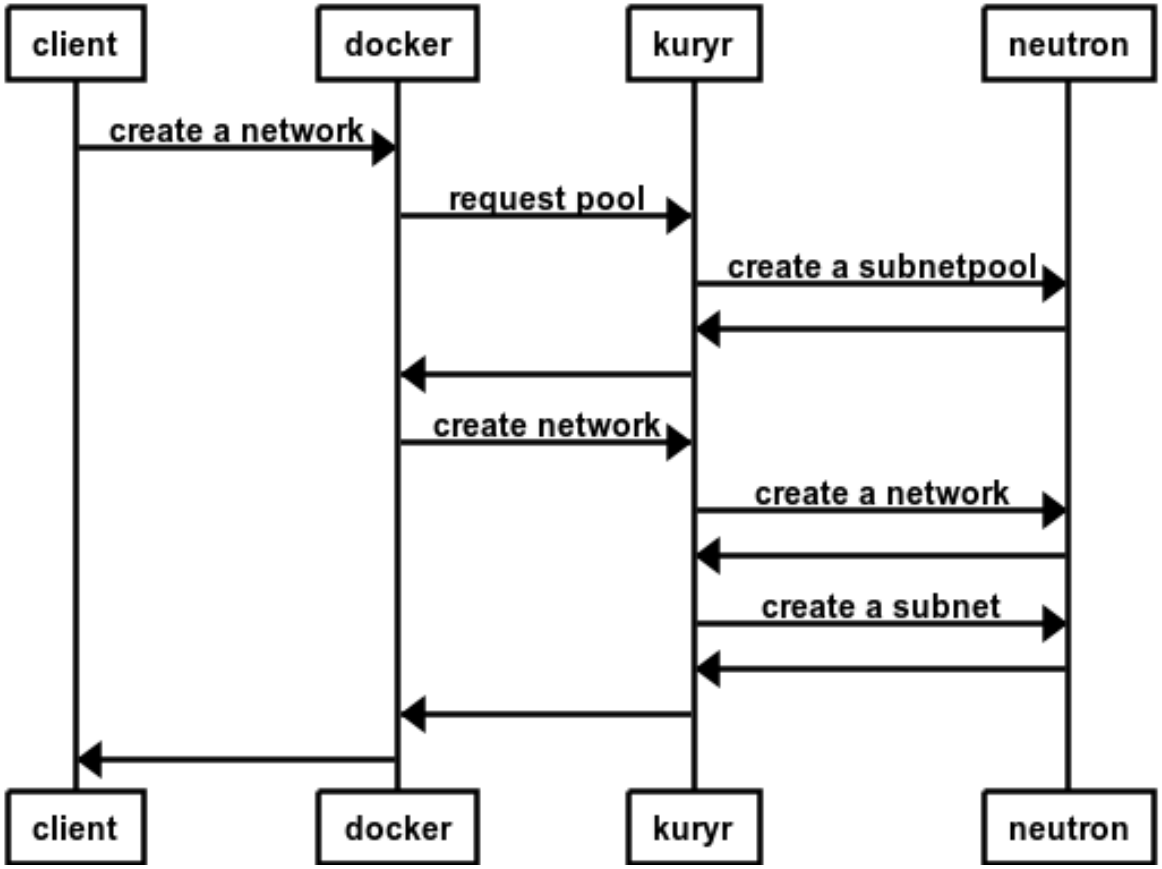
- Not an API service for end-users
- Provides plugins for Docker and Kubernetes
  - Kuryr-libnetwork: Implement Docker network model by using Neutron
  - Fuxi: Provide data volume to Docker by using Cinder/Manila
  - Kuryr-Kubernetes: A network controller and a CNI driver
  - Fuxi-Kubernetes: A volume provisioner and a FlexVolume plugin

Kuryr-libnetwork

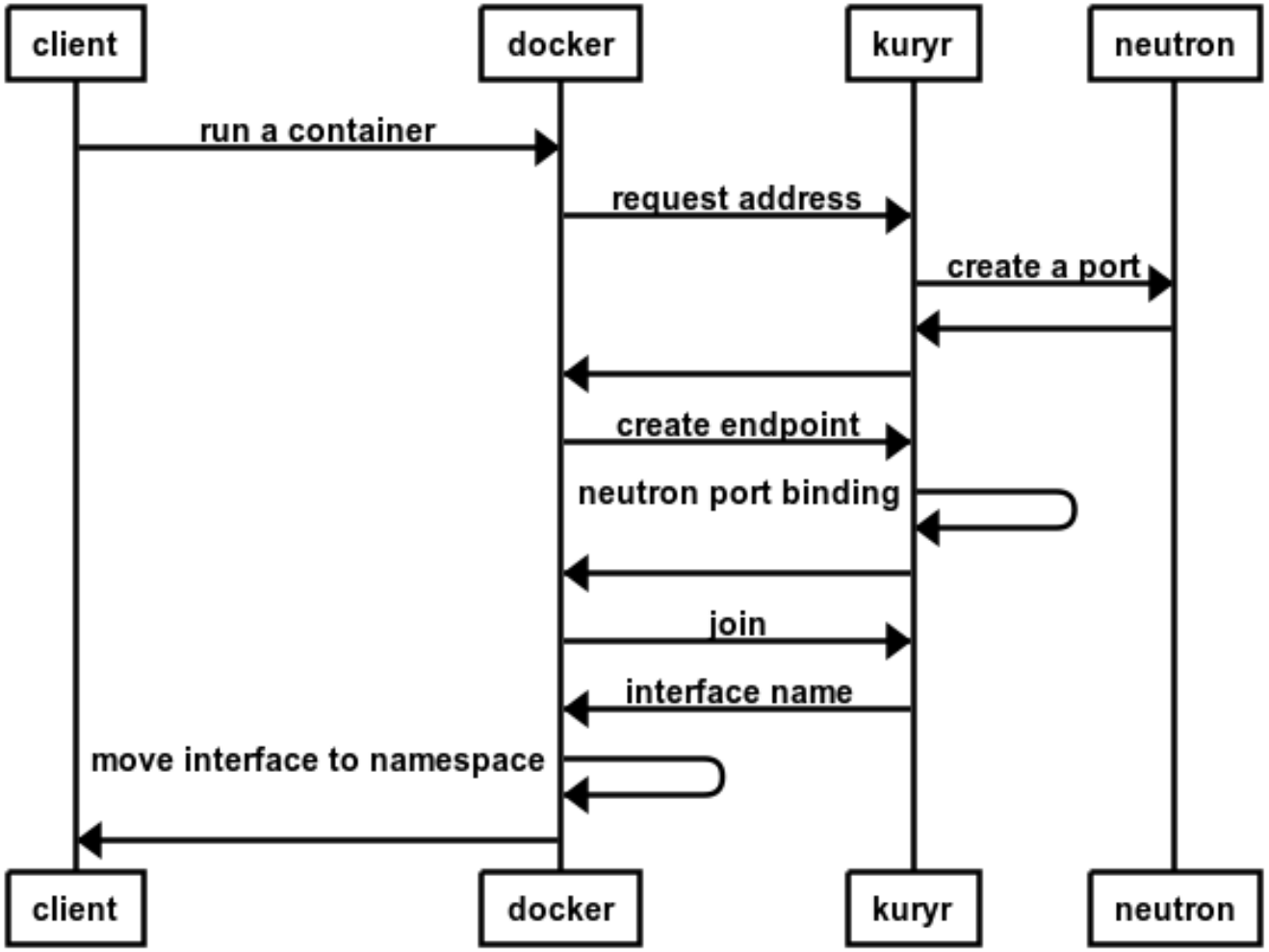
- A docker network plugin
- End-users use native Docker API
  - Create network
  - Connect containers to networks
- Docker calls Kuryr-libnetwork
- Kuryr-libnetwork fulfill network request
  - Create Neutron network
  - Create a neutron port
  - Bind the neutron port to the container

```
$ # Create network
$ docker network create \
    --driver=kuryr --ipam-driver=kuryr \
    --subnet 10.10.0.0/16 --gateway 10.10.0.1 \
    foo

$ # Run container
$ docker run -it --net=foo alpine
```

Create Network



Run Container



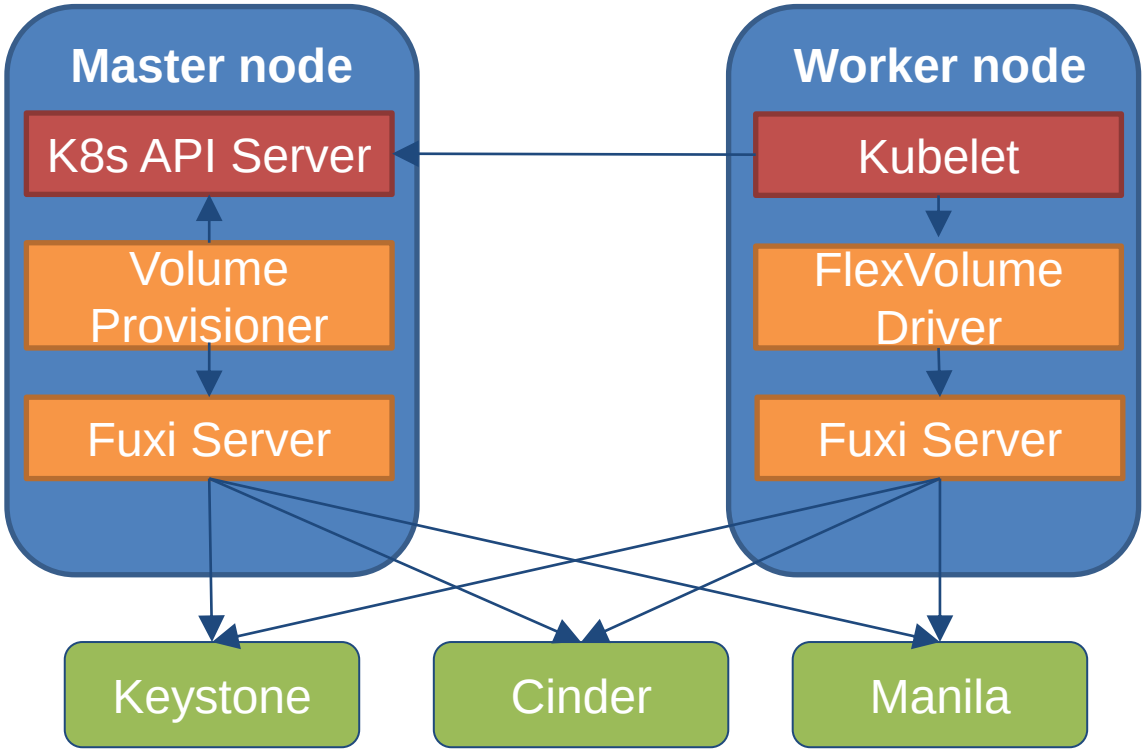
Fuxi

- A docker volume plugin
- End-users use native Docker API
  - Create volume
  - Bind-mount volumes to containers
- Fuxi fulfill volume request
  - Create Cinder volume or Manila share
  - Connect to the volume
  - Mount volume to host

```
$ # Create cinder volume
$ docker volume create --driver fuxi \
    --name my_vol \
    --opt size=1 \
    --opt fstype=ext4 \
    --opt multiattach=true \
    --opt volume_provider=cinder

$ # Create generic manila volume
$ docker volume create --driver fuxi \
    --name my_vol \
    --opt volume_provider=manila

$ # Run container with volume
$ docker run -v my_vol:/var/www httpd
```

Fuxi-Kubernetes

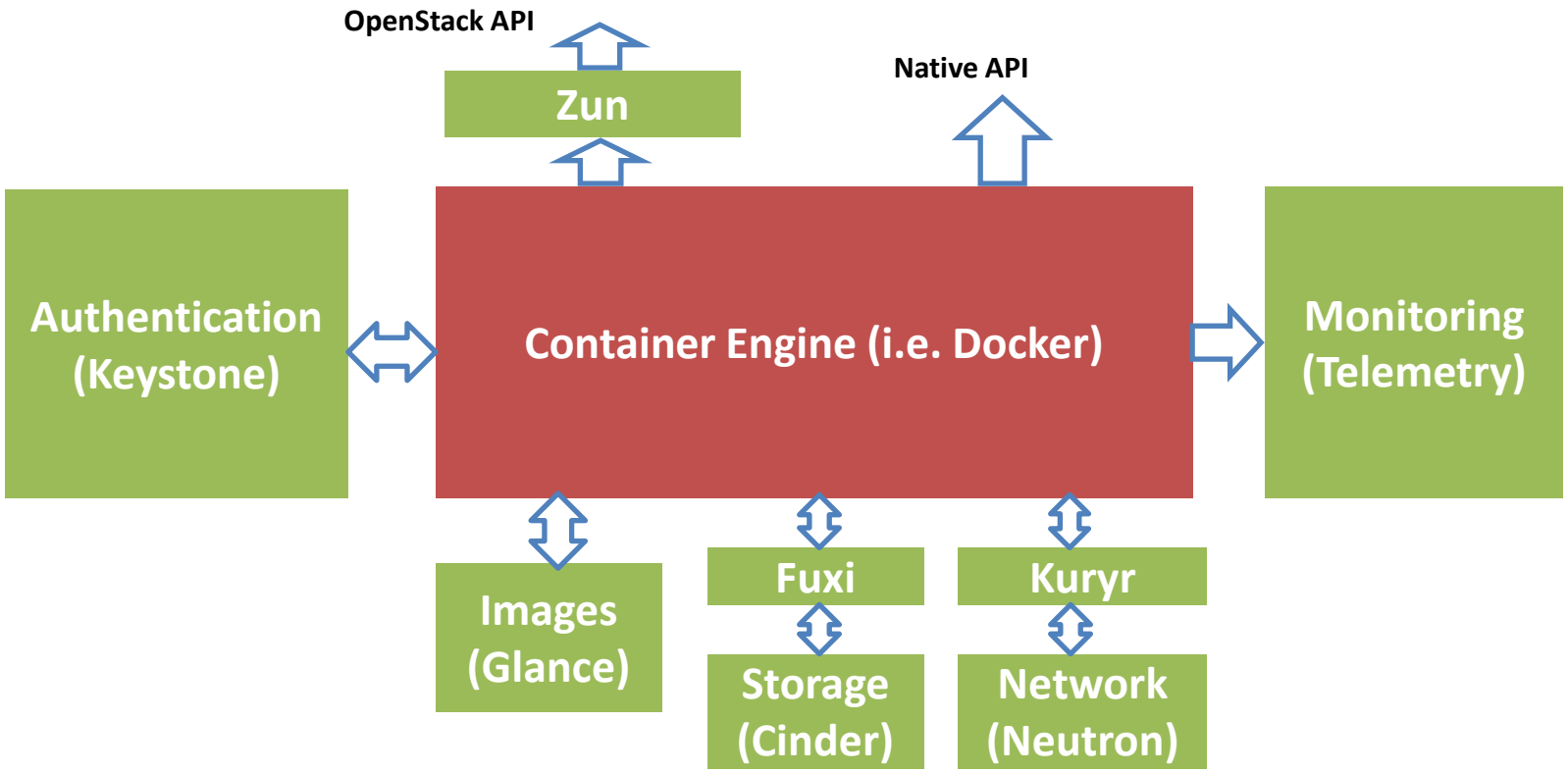
- Volume Provisioner: Watch k8s API Server for PVCs and create PVs
- FlexVolume Driver: Mount volumes to pods.
- Fuxi Server: Manage storage resources in Cinder/Manila



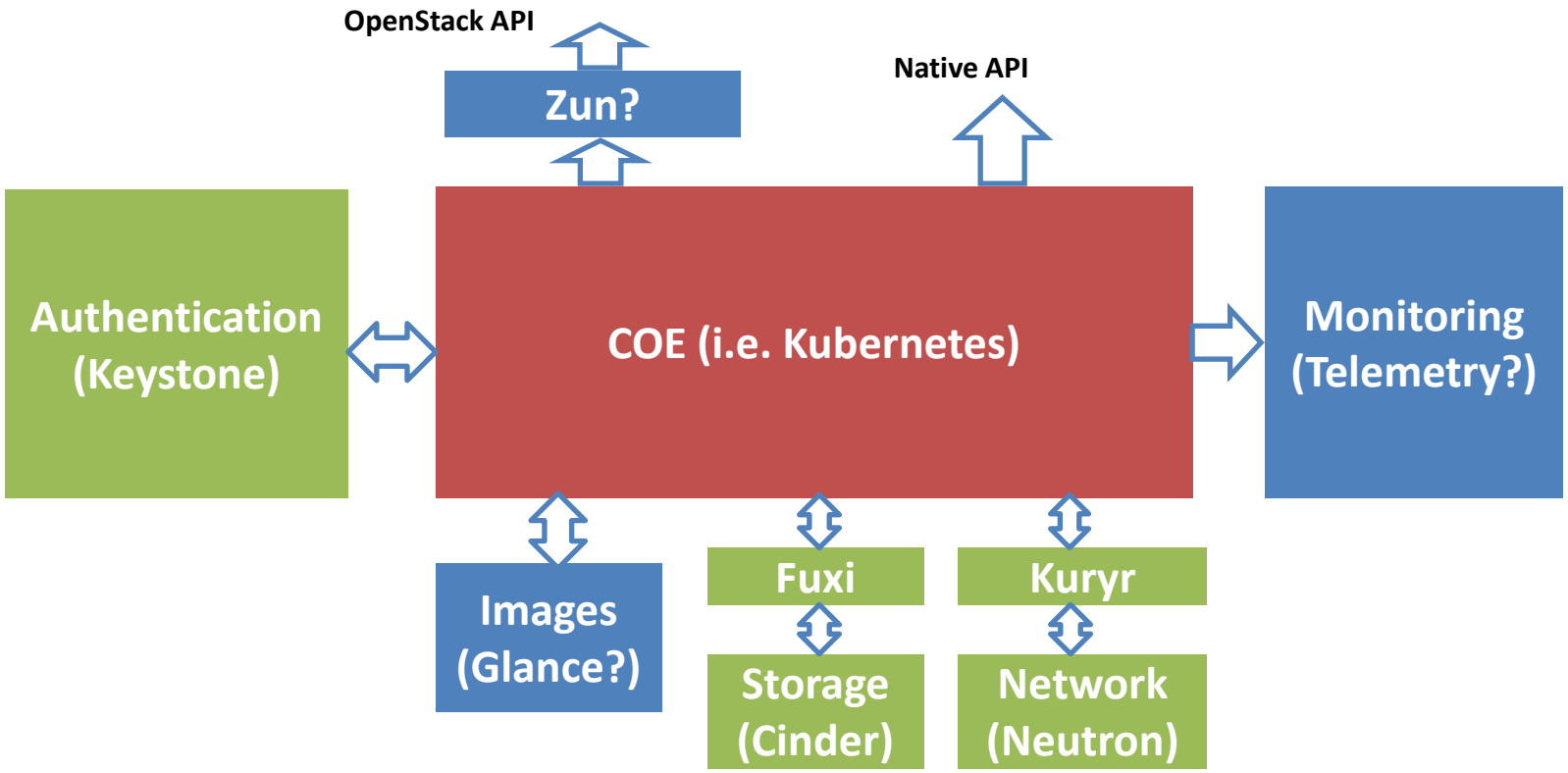
Agenda

- Introduction
- Containers on OpenStack
- Solutions
  - Magnum
  - Zun
  - Kuryr
- The Future

The Future - Container Engine



The Future - COE



THANK YOU