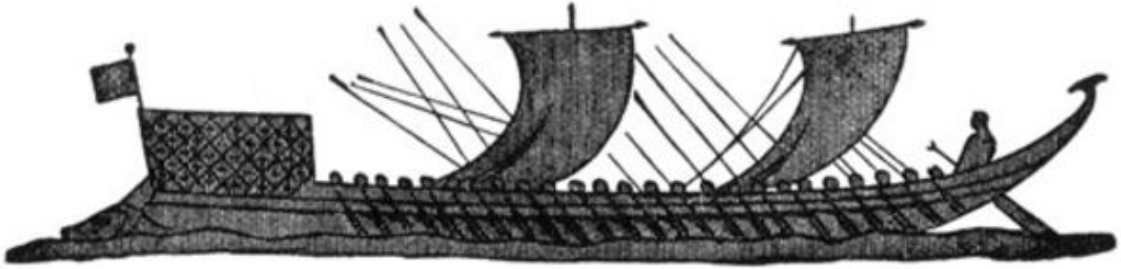
Payload orchestration: Kubernetes e Spark

D. Ciangottini, INFN sez. Perugia

Kubernetes overview

What Does “Kubernetes” Mean?

Greek for “pilot” or
“Helmsman of a ship”



What Does Kubernetes do?

- Known as the linux kernel of distributed systems.
- Abstracts away the underlying hardware of the nodes and provides a uniform interface for workloads to be both deployed and consume the shared pool of resources.
- Works as an engine for resolving state by converging actual and the desired state of the system.

What is Kubernetes?

Project that was spun out of Google as an open source container orchestration platform.

Built from the lessons learned in the experiences of developing and running Google’s Borg and Omega.

Designed from the ground-up as a loosely coupled collection of components centered around deploying, maintaining and scaling workloads

Decouples Infrastructure and Scaling

All services within Kubernetes are natively Load Balanced.

Can scale up and down dynamically.

Used both to enable self-healing and seamless upgrading or rollback of applications.

Self Healing

Kubernetes will ALWAYS try and steer the cluster to its desired state.

Me: “I want 3 healthy instances of redis to always be running.”

Kubernetes: “Okay, I’ll ensure there are always 3 instances up and running.”

Kubernetes: “Oh look, one has died. I’m going to attempt to spin up a new one.”

What can Kubernetes REALLY do?

Autoscale Workloads

Blue/Green Deployments

Fire off jobs and scheduled cronjobs

Manage Stateless and Stateful Applications

Provide native methods of service discovery

Easily integrate and support 3rd party apps

Most Importantly...

Use the SAME API

across bare metal and EVERY cloud provider!!!

Who “Manages” Kubernetes?

The CNCF is a child entity of the Linux Foundation and operates as a vendor neutral governance group.

CLOUD NATIVE
COMPUTING FOUNDATION

Project Stats

Over 90,000 stars on Github

3000+ Contributors to K8s Core

Most discussed Repository by a large margin

100,000+ users in Slack Team

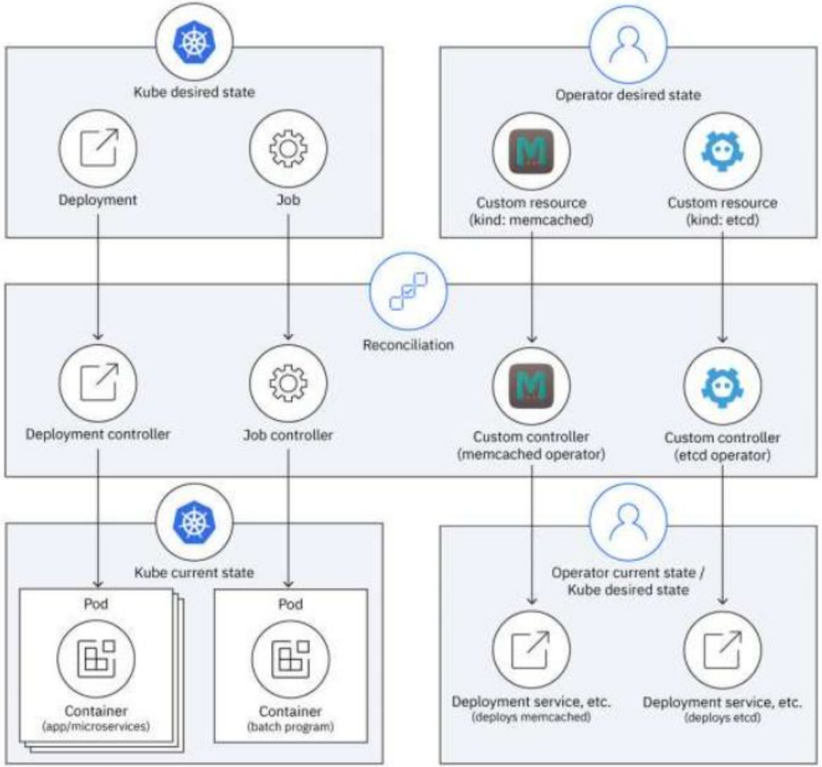
Kubernetes key concepts

Tell me what you want…
and leave me alone

Kubernetes fundamentals are the reconcile cycles

The user provides a desired state

The system works to keep the infrastructure on that state continuously

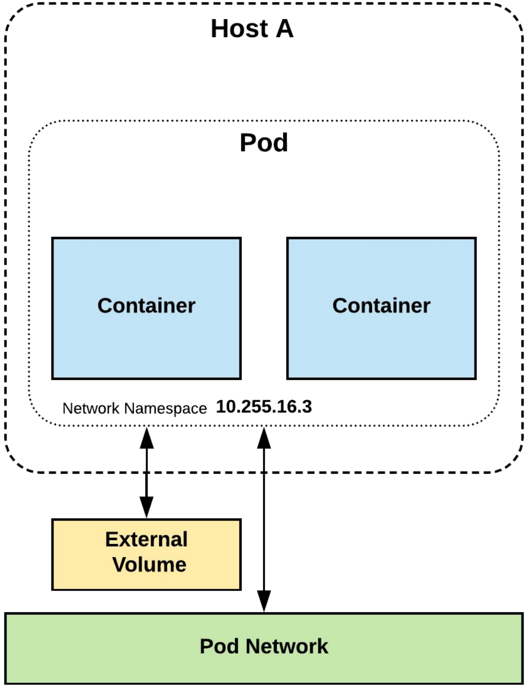


Some basic resource types: POD

Atomic unit or smallest “unit of work”of Kubernetes.

Pods are one or MORE containers that share volumes, a network namespace, and are a part of a single context.

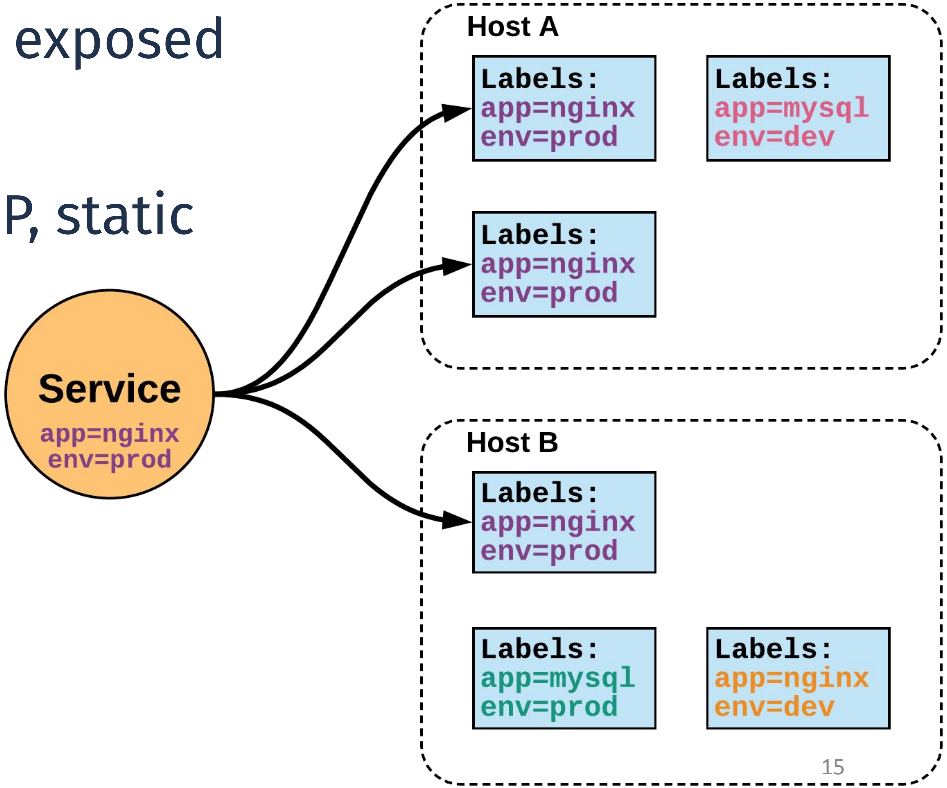
They are ephemeral!



Some basic resource types: Services

Unified method of accessing the exposed workloads of Pods.

Durable resource: static cluster IP, static namespaced DNS name



Some basic resource types: Ingress – Name Based Routing

An API object that manages external access to the services in a cluster

Provides load balancing, SSL termination and name/path-based virtual hosting

Gives services externally-reachable URLs

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: name-virtual-host-ingress
spec:
  rules:
  - host: first.bar.com
    http:
      paths:
      - backend:
          serviceName: service1
          servicePort: 80
  - host: second.foo.com
    http:
      paths:
      - backend:
          serviceName: service2
          servicePort: 80
  - http:
      paths:
      - backend:
          serviceName: service3
          servicePort: 80
```

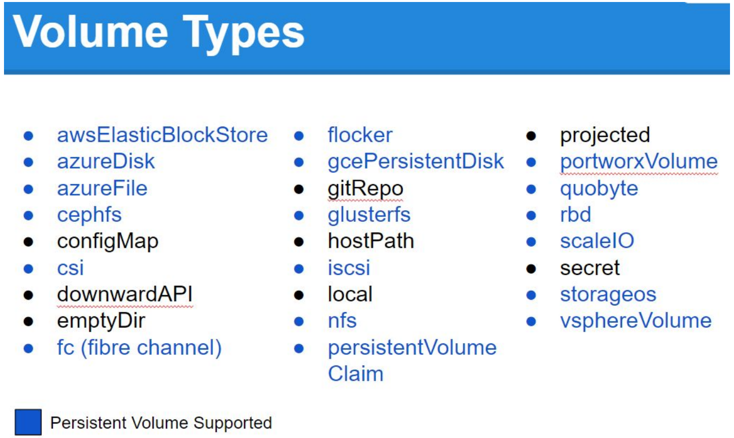
Some basic resource types: Volumes

Storage that is tied to the Pod’s Lifecycle.

A pod can have one or more types of volumes attached to it.

Can be consumed by any of the containers within the pod.

Survive Pod restarts; however their durability beyond that is dependent on the Volume Type.



Volume Types

- awsElasticBlockStore
- azureDisk
- azureFile
- cephfs
- configMap
- csi
- downwardAPI
- emptyDir
- fc (fibre channel)
- flocker
- gcePersistentDisk
- gitRepo
- glusterfs
- hostPath
- iscsi
- local
- nfs
- persistentVolume Claim
- projected
- portworxVolume
- quobyte
- rbd
- scaleIO
- secret
- storageos
- vsphereVolume

■ Persistent Volume Supported

What is under a K8s button @INFNCloud?

- Ingress service provided through nginx
  - By default is capable of leveraging the dedicated wildcard DNS:
    - *.<public ip>.myip.cloud.infn.it
- Monitoring instance
- Kubernetes Dashboard
- A kubeconfig file for access to the cluster from CLI

There is no default storage class. We will see some of the easiest options for getting persistent volumes today.

Time for the hands-on session