Taller Online

# Docker & Kubernetes

Sesión #6 | @inicopaez

# Agenda

Initial questions & expectations

Patterns: sidecars & microservices

Scalability

Helm Packaging

Deploy Pipeline

Kubenetes flavors & OpenShift

# Dudas iniciales

> Arquitectura de Kubernetes propia en Cloud vs con servicios de Cloud providers. (EKS)

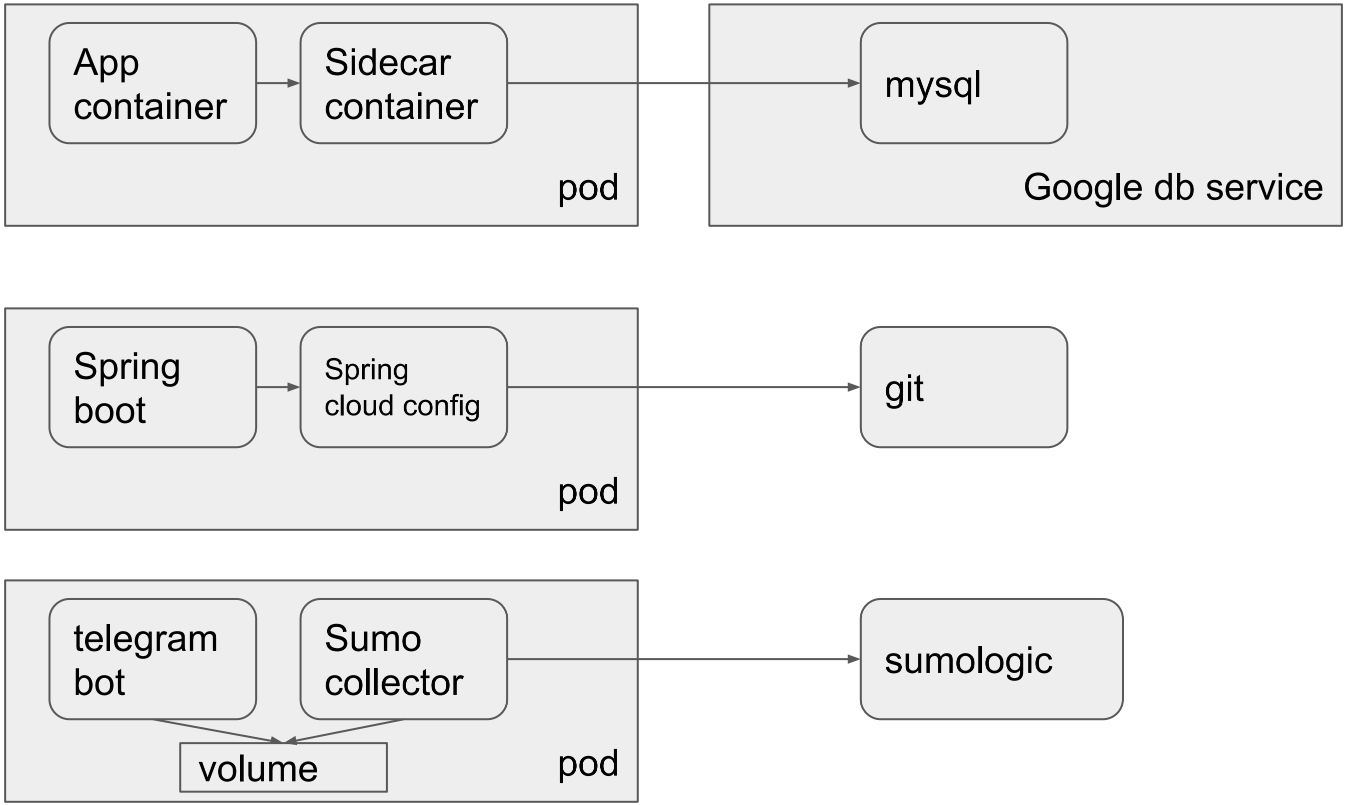
> Buenas practicas en entornos corporativos y casos de exitos en empresas similares a Balanz.

> Integracion de Kubernetes con otras tools

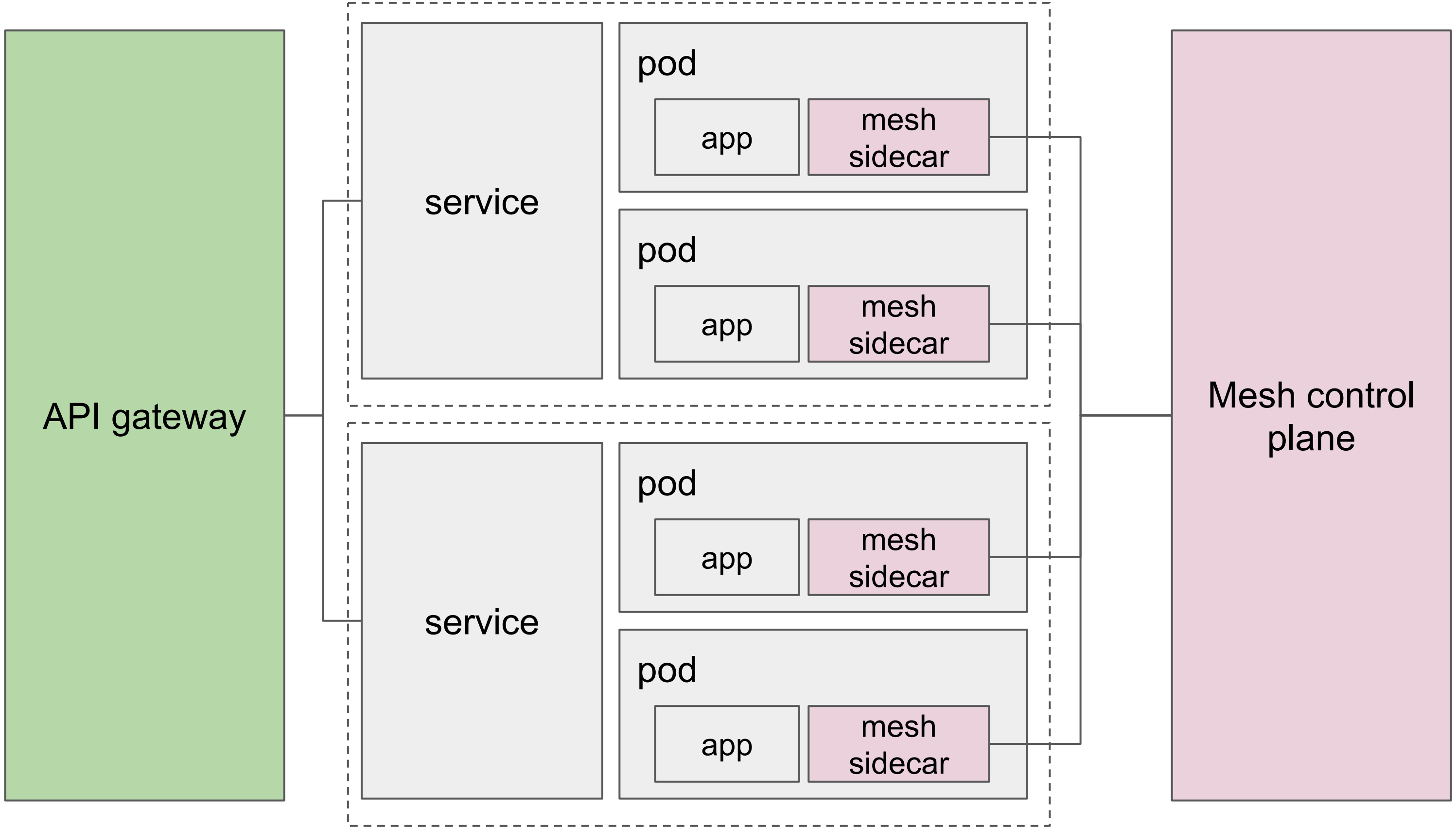
> HA y FT en docker teniendo en cuenta que se corre onpremise.

> “...despleagar mi proyecto de microservicios java (springboot) en instancias de docker…”

# Sidecar Pattern



# Microservices @ K8s



# Microservices @ K8s

Routing / Communication

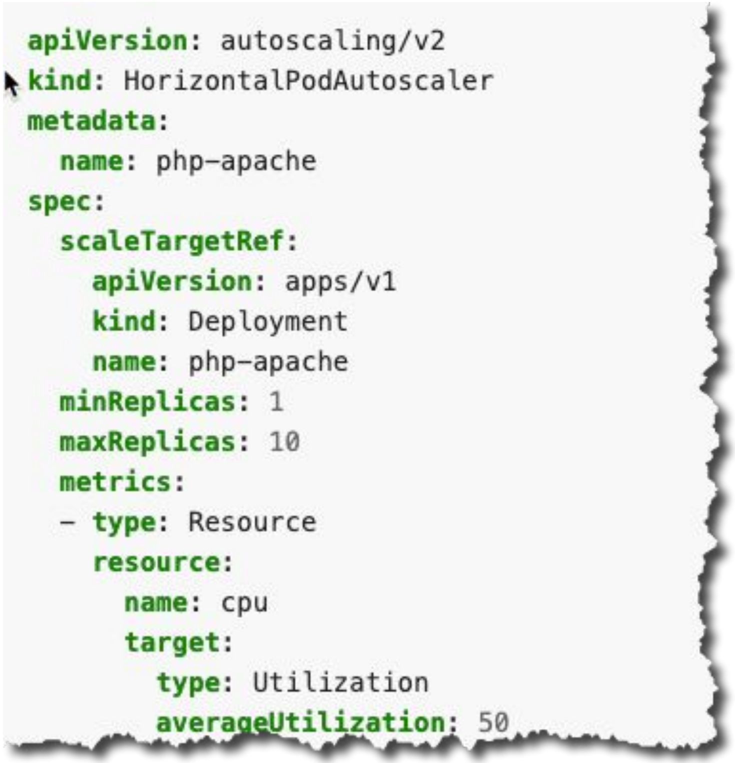
Monitoring / Observability

Rate limiting

Security

# Scalability:
# pods vs nodes

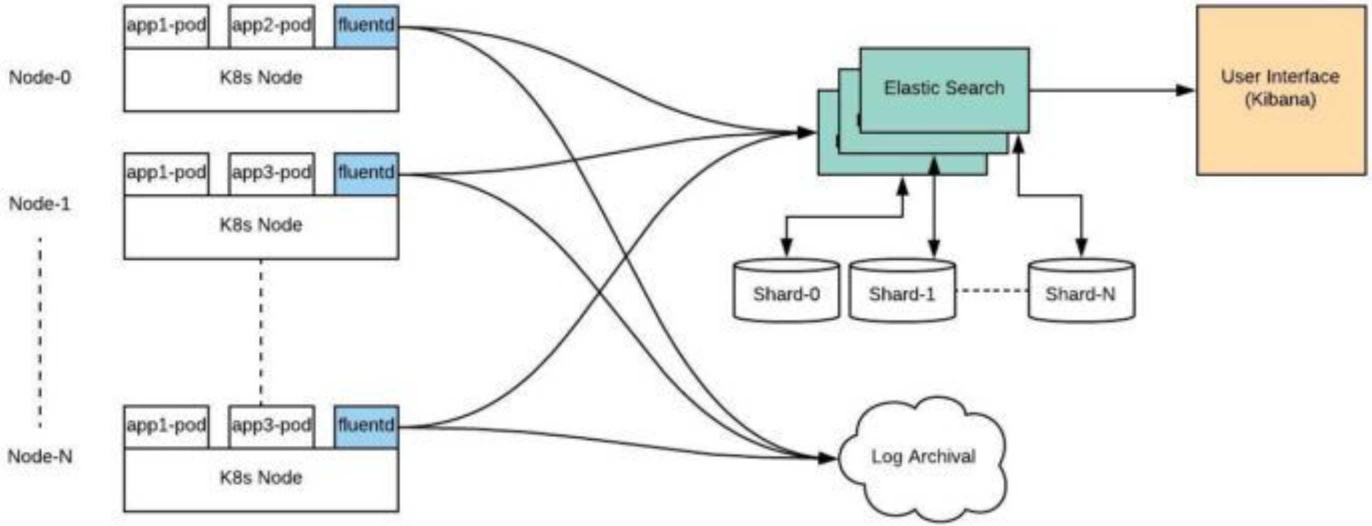
# Horizontal Pods Autoscaling



```yaml
apiVersion: autoscaling/v2
kind: HorizontalPodAutoscaler
metadata:
  name: php-apache
spec:
  scaleTargetRef:
    apiVersion: apps/v1
    kind: Deployment
    name: php-apache
  minReplicas: 1
  maxReplicas: 10
  metrics:
  - type: Resource
    resource:
      name: cpu
      target:
        type: Utilization
        averageUtilization: 50
```

https://kubernetes.io/docs/tasks/run-application/horizontal-pod-autoscale-walkthrough/

# Kubernetes Monitoring (caso ELK)



# Kubernetes Monitoring

Built-in: Kubenetes dashboard, kubectl top

Stand-alone: Prometheus, Grafana, ELK

Cloud-Services: New Relic, Datadog

Package Manager for Kubernetes

# Helm quickstart

```
helm repo add bitnami https://charts.bitnami.com/bitnami

helm search repo bitnami

helm install jenkins1 bitnami/jenkins --set service.type=NodePort
```

# Kubernetes flavors

Minikube

https://microk8s.io/

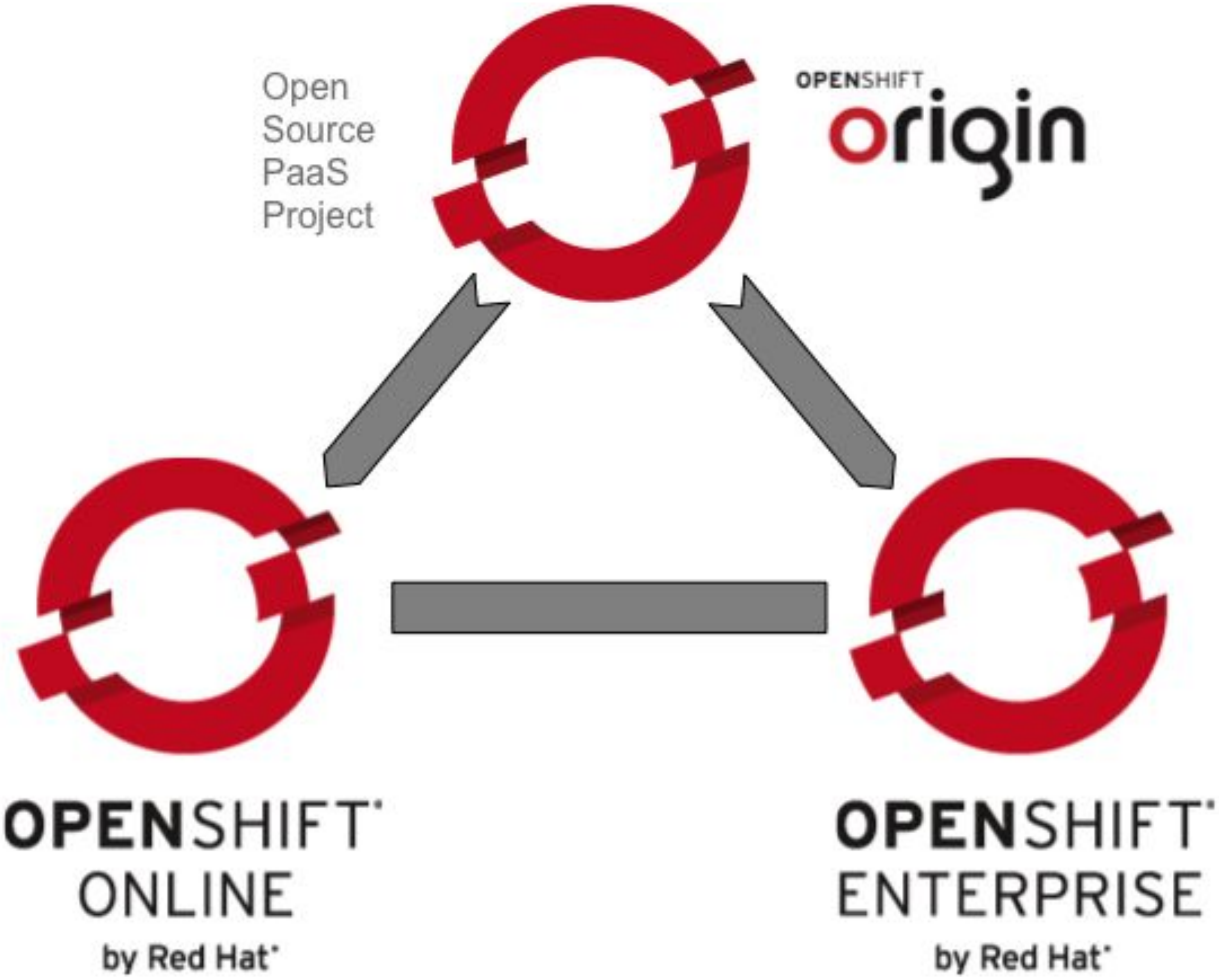
OpenShift / Minishift

https://k3s.io/

Rancher

OPENSHIFT®

by Red Hat®



Open Source PaaS Project

OPENSHIFT origin

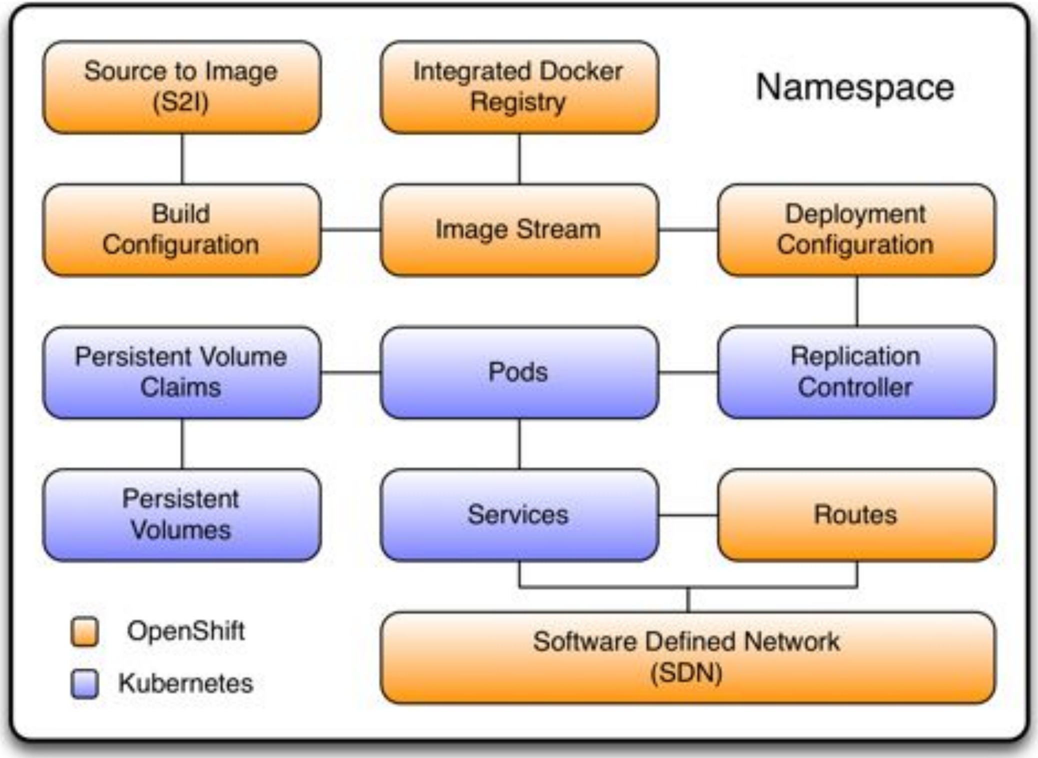
Public PaaS Service

OPENSHIFT® ONLINE by Red Hat®

On-premise or Private PaaS Software

OPENSHIFT® ENTERPRISE by Red Hat®

# Kubernetes / Openshift



Source to Image (S2I)

Integrated Docker Registry

Namespace

Build Configuration

Image Stream

Deployment Configuration

Persistent Volume Claims

Pods

Replication Controller

Persistent Volumes

Services

Routes

Software Defined Network (SDN)

🟧 OpenShift

🟦 Kubernetes

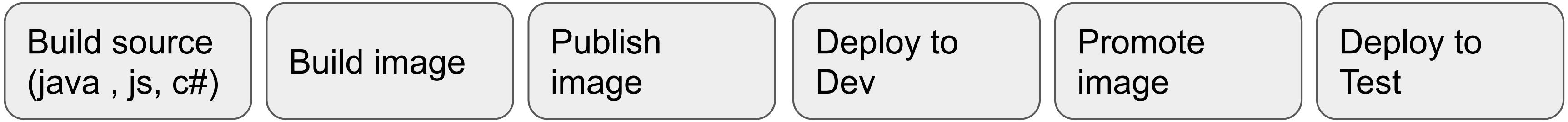
Source 2 Image

S2i git(java) images2ijava

# Deployment Pipeline

(https://gitlab.com/fiuba-memo2/ejemplos/telegram-bot-example/-/blob/master/.gitlab-ci.yml)

# CI / CD Pipeline



Build source (java , js, c#)

Build image

Publish image

Deploy to Dev

Promote image

Deploy to Test

```
Image promotion strategy
    App:1.0.0
    App:1.0.0-rc
    App:1.0.0-ga
```

https://gitlab.com/fiuba-memo2/ejemplos/telegram-bot-example/-/blob/master/.gitlab-ci.yml

# CI / CD for Containers

# End of Session 6