Sesión 2 – Objetos en Kubernetes

MitoCode Network

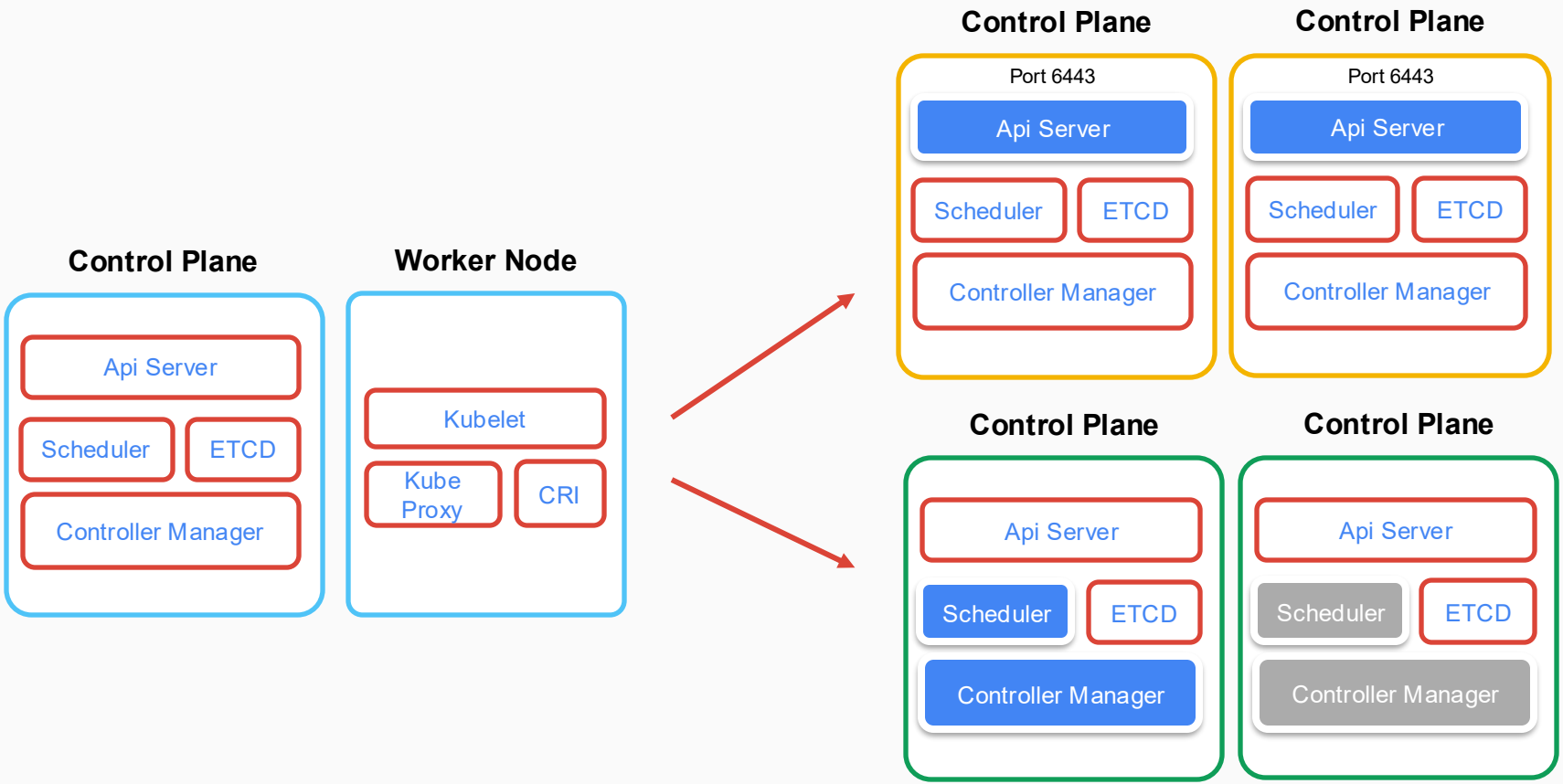
Por: Juan Carlos Salvador García

MitoCode

Agenda

1. Alta Disponibilidad en Kubernetes
2. Upgrade de Cluster de Kubernetes
3. Objetos en Kubernetes: Pods
4. Objetos en Kubernetes: Replicasets
5. Objetos en Kubernetes: Namespace
6. Objetos en Kubernetes: Deployments

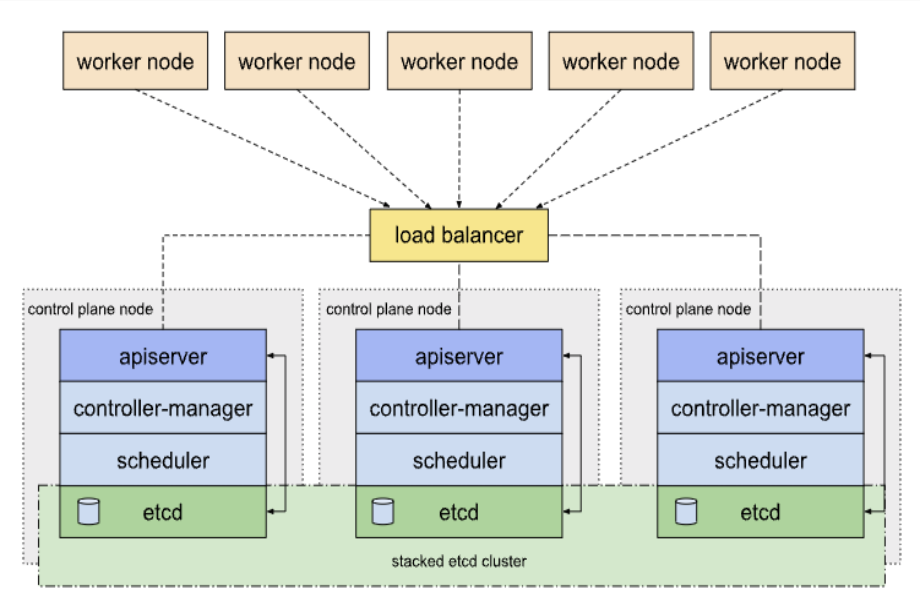
Alta Disponibilidad en Kubernetes



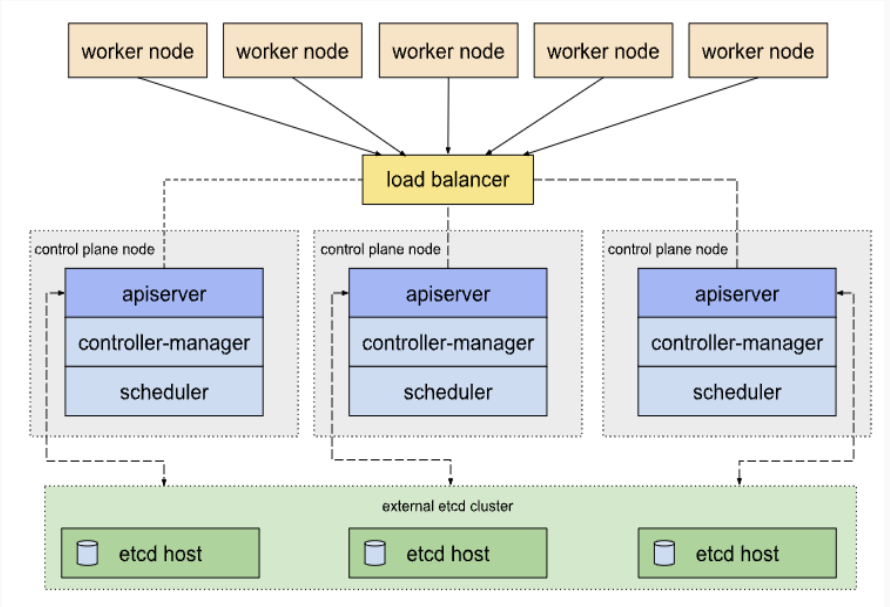
MitoCode

Alta Disponibilidad en Kubernetes

Stacked ETCD



External ETCD



MitoCode

Upgrade de Cluster de Kubernetes

Arquitectura

- [x] 1 Control Plane
- [x] 1 Worker Node

Objetivos

- [x] Actualizar versión de Kubernetes

Procedimiento Control Plane y Worker Node

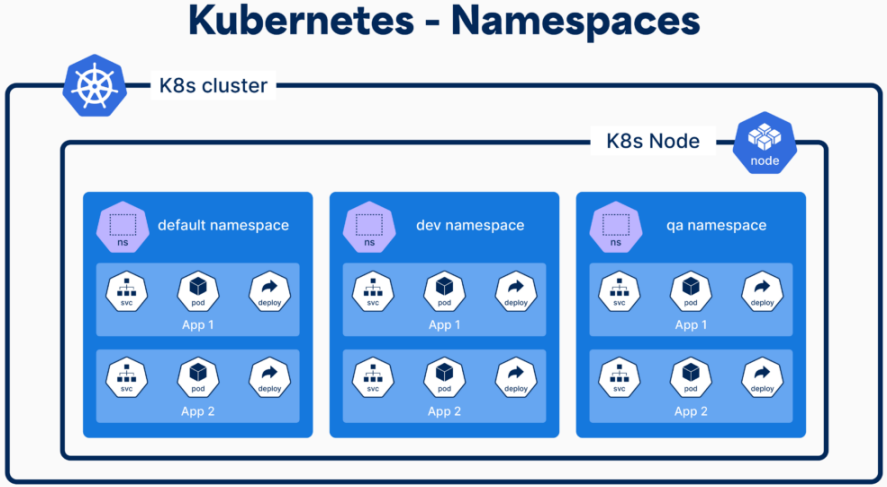
- [x] Actualizar repositorios de kubernetes.
- [x] Actualizamos versión de cluster.
- [x] Actualizar kubeadmn.
- [x] Actualizamos versión de kubelet y kubectl.

MitoCode

Objetos en Kubernetes

Namespaces

Permite isolar un conjunto de recursos dentro del cluster de Kubernetes.



MitoCode

Namespaces Iniciales

Default

Namespace que da kubernetes para poder empezar a usar el cluster.

Kube-system

Aloja objetos creados por el sistema de kubernetes.

Kube-public

Cualquier usuario puede acceder a este namespace, es reservado para uso interno del cluster.

Kube-nose-lease

Permite verificar la salud de cada nodo.

MitoCode

Comandos Namespaces

Comandos:

- [x] kubectl create namespace <name>
- [x] kubectl get namespace
- [x] kubectl api-resources --namespaced=true
- [x] kubectl api-resources --namespaced=false

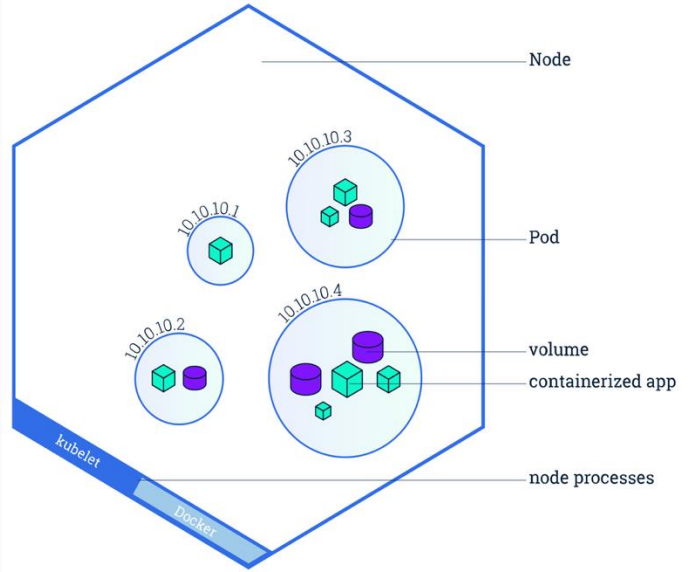
MitoCode

Objetos en Kubernetes

Pods

Características

- Es la unidad más pequeña de Kubernetes.
- Puede alojar de 1 a más contenedores.
- Puede ser creados de manera imperativa o declarativa.
- Son efimeros si se administran de manera independiente.



MitoCode

Ciclo de Vida de los Pods

Pending

Los pods aún no han sido programados ya sea porque están descargando la imagen o no se reunen los recursos necesarios.

Running

Al menos un contenedor esta ejecutandose y el pod ya ha sido ubicado en un nodo.

Succeeded

Todos los contenedores han terminado sus procesos y no se están reiniciando.

Failed

Todos los contenedores han terminado sus procesos y al menos uno está retornando error.

Unknown

No se puede reconocer el estado del Pod posiblemente por una comunicación entre el Pod y el nodo donde se ejecuta.

ImagePullBackOff

Error al descargar la imagen del contenedor.

CrashLoopBackOff

El pod se mantiene reiniciandose posiblemente por algún error en una dependencia.

MitoCode

Despliegue de Pods

Despliegue Imperativo:

- kubectl run NAME --image=image [--env="key=value"] [--port=port] [--dry-run=server|client] [--overrides=inline-json] [--command] -- [COMMAND] [args...] [options]
- kubectl run server –image nginx
- kubectl get pods
- kubectl run backend --image=nginx --dry-run=client –o yaml

MitoCode

Despliegue de Pods

Despliegue Declarativo:

kubectl apply -f pod.yaml

pod.yaml

```yaml
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```

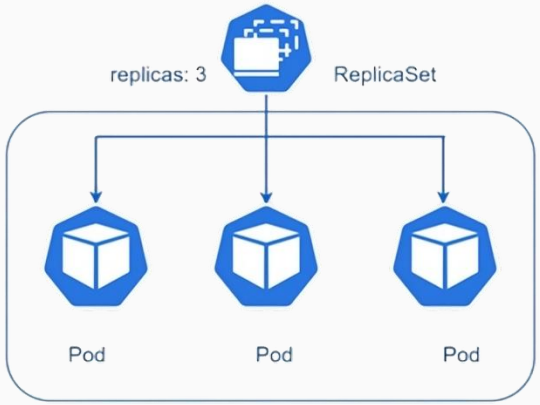
MitoCode

Objetos en Kubernetes

Replicasets

Características

- Garantiza la disponibilidad de un conjunto de réplica de Pods.
- Solo se debe usar si no se va a administrar una actualización de pods.



MitoCode

Despliegue YAML

Despliegue Declarativo:

kubectl apply -f replicaset.yaml

replicaset.yaml

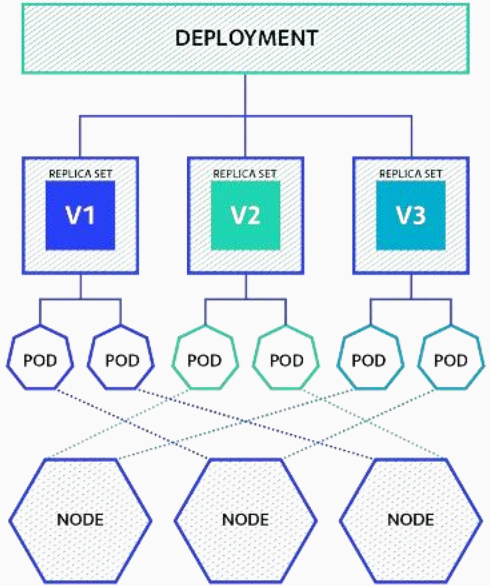
```yaml
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: nginx
  labels:
    app: backend
    server: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      server: nginx
  template:
    metadata:
      labels:
        server: nginx
    spec:
      containers:
      - name: nginx
        image: nginx

```

MitoCode

Objetos en Kubernetes

Deployments

Permite gestionar el ciclo de vida y escalabilidad de los pods.



MitoCode

Despliegue YAML

Despliegue Declarativo:

kubectl apply -f deployment.yaml

deploymentyaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.14.2
        ports:
        - containerPort: 80
```

MitoCode


EJERCICIOS
MitoCode

Ejercicios

1. Lista la cantidad de Pods que se encuentran dentro del namespace kube-system.
2. Lista todos los Pods que se encuentren dentro del cluster.
3. Crea un Pod de manera imperativa que use la imagen ubuntu:20.
4. Usando la CLI de Kubernetes genera el YAML que te permita crear un Pod que use la imagen redis.
5. Despliega el YAML pregunta 5 y deja el Pod en estado running.
6. Crea un namespace con el formato <primera letra nombre><primera letra apellido>-dev.
7. Despliega un replicaset de nombre server-rs en el nuevo namespace que use la imagen de nginx y tenga 5 replicas.
8. Elimina un pod del replicaset server-rs y explica que sucede.
9. Crea un deployment con el yaml pregunta9.yaml y logra que se despliegue correctamente.

MitoCode

MitoCode Network

www.mitocode.com