Kubernetes

Unidad Didáctica 01: Introducción

http://cursosdedesarrollo.com/

Índice de contenidos

- Introducción
- Instalación
- Pod
- Volume
- Replica Set
- Service
- Deployment
- Conclusiones

http://cursosdedesarrollo.com/

Introducción

Kubernetes es un software libre de orquestación de contenedores software en cluster

http://cursosdedesarrollo.com/

Introducción

La base de funcionamiento de Kubernetes (k8s) son los contenedores de Docker

http://cursosdedesarrollo.com/

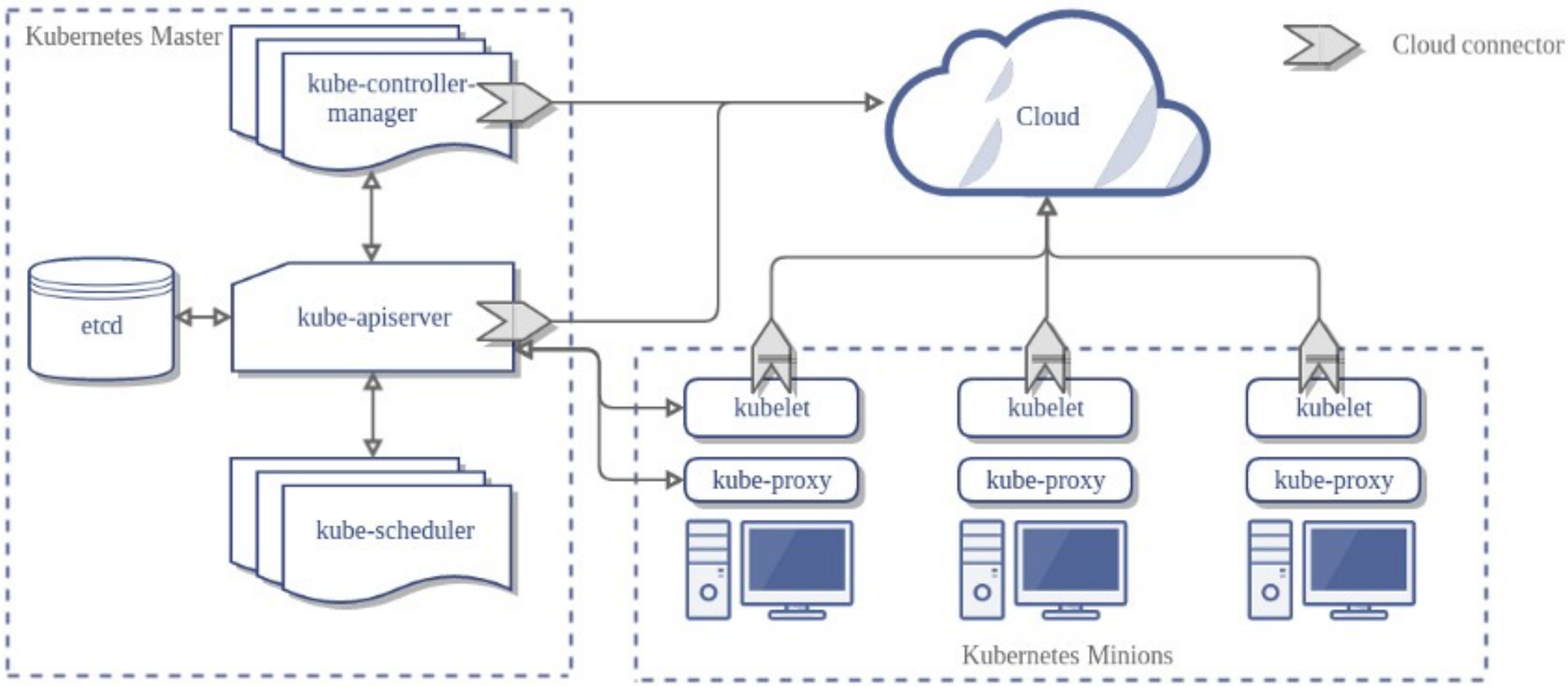
Introducción

La arquitectura principal de k8s es de tipo cluster con mínimo 3 servidores: 1 master y 2 nodes (o minions)

(Sí, son así de frikis)

http://cursosdedesarrollo.com/

Introducción



http://cursosdedesarrollo.com/

Introducción

Se distinguen varios componentes:

- Cloud Controller Manager (CCM): Controla nodos, rutas y servicios

- Kubernetes Controller manager: el controlador de volúmes se dejó fuera del CCM por su nivel de complejidad

Introducción

Cada nodo de k8s tiene instalado un kubelet y un kube-proxy

Los nodos son los encargados de ejecutar los contenedores

Introducción

El kubelet inicializa un nodo sin información especifica del proveedor de servicios. Sin embargo, añade un artefacto (taint) al nodo recién creado de forma que este no esté disponible para el planificador hasta que el CCM completa el nodo con la información específica del proveedor. Sólo entonces elimina el taint y el nodo se vuelve accesible

Introducción

Nodos

Kubelet

El controlador de nodos incluye la funcionalidad del kubelet que es dependiente de la nube. Previa a la introducción de CCM, el kubelet era responsable de inicializar un nodo con detalles específicos al proveedor como direcciones IP, etiquetas de región/zona y tipo de instancia

La introducción de CCM transfiere esta inicialización del kubelet al CCM

Introducción

El kube-proxy es un proxy de redes instalado en cada nodo

Se encarga del stream forwarding y del round robin de UDP, TCP, SCTP de los nodos reflejando la configuración de Kubernetes API

Existe un addo que puede llegar a manejar el DNS de las ip del clúster

http://cursosdedesarrollo.com/

Introducción

CCM

Controlador de Nodos

El controlador de nodos es responsable de inicializar un nodo obteniendo información del proveedor de servicios sobre los nodos ejecutándose en el clúster. El controlador de nodos lleva a cabo las siguientes funciones:

- Inicializa un nodo con etiquetas de región y zona específicas del proveedor.
- Inicializa un nodo con detalles de la instancia específicos del proveedor, como por ejemplo, el tipo o el tamaño.
- Obtiene las direcciones de red del nodo y su hostname.
- En caso de que el nodo deje de responder, comprueba la nube para ver si el nodo ha sido borrado. Si lo ha sido, borra el objeto nodo en Kubernetes.

Introducción

CCM

Controlador de Rutas

El controlador de Rutas es responsable de configurar rutas en la nube para que contenedores en diferentes nodos dentro de un clúster kubernetes se puedan comunicar entre sí

http://cursosdedesarrollo.com/

Introducción

CCM

Controlador de Servicios

El controlador de servicios es responsable de monitorizar eventos de creación, actualización y borrado de servicios. Basándose en el estado actual de los servicios en el clúster Kubernetes, configura balanceadores de carga del proveedor (como Amazon ELB, Google LB, or Oracle Cloud Infrastructure Lb) de forma que estos reflejen los servicios definidos en Kubernetes

Adicionalmente, se asegura de que los sistemas de apoyo de servicios para balanceadores de carga en la nube se encuentren actualizados.

http://cursosdedesarrollo.com/

Introducción

Los siguientes proveedores de servicios en la nube han implementado CCMs integrables en k8s:

Digital Ocean

Oracle

Azure

GCP

AWS

BaiduCloud

Linode

Instalación

Veamos cómo hacer la instalación de k8s en Ubuntu

https://vitux.com/install-and-deploy-kubernetes-on-ubuntu/

Instalación

Primero deberemos tener instalado Docker de la manera habitual

http://cursosdedesarrollo.com/

Instalación

Después deberemos instalar k8s desde los repositorios

$ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -

$ sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial main"

$ sudo apt install kubeadm

$ kubeadm version

http://cursosdedesarrollo.com/

Instalación

En Windows y MacOSX debería estar ya instalado con Docker for Desktop

http://cursosdedesarrollo.com/

Instalación

Cuando ya está instalado y configurado el servidor de kubernetes podemos ejecutar los comandos:

kubectl cluster-info

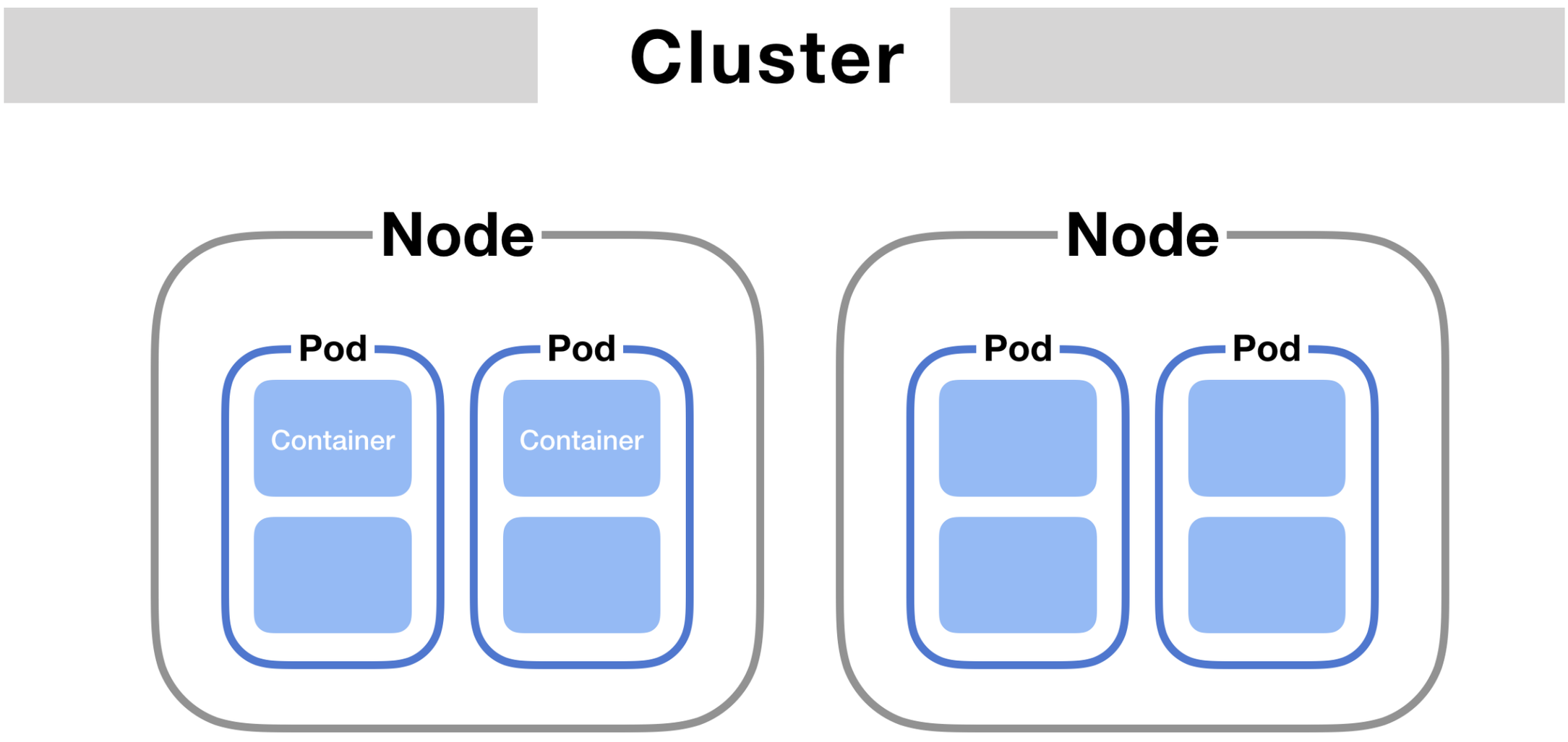
kubectl get nodes

Pod

Los pod son la unidad mínima de despliegue en k8s, en realidad son conjuntos de contenedores docker que funcionan de manera conjunta, también incluyen otras configuraciones, almacenamiento compartido

http://cursosdedesarrollo.com/

Pod



http://cursosdedesarrollo.com/

Pod

Por ejemplo para un pod podríamos disponer de nginx, php-fpm y mariadb en un mismo pod

http://cursosdedesarrollo.com/

Pod

El concepto es muy similar al docker compose

De hecho hay una herramienta llamada
Kompose

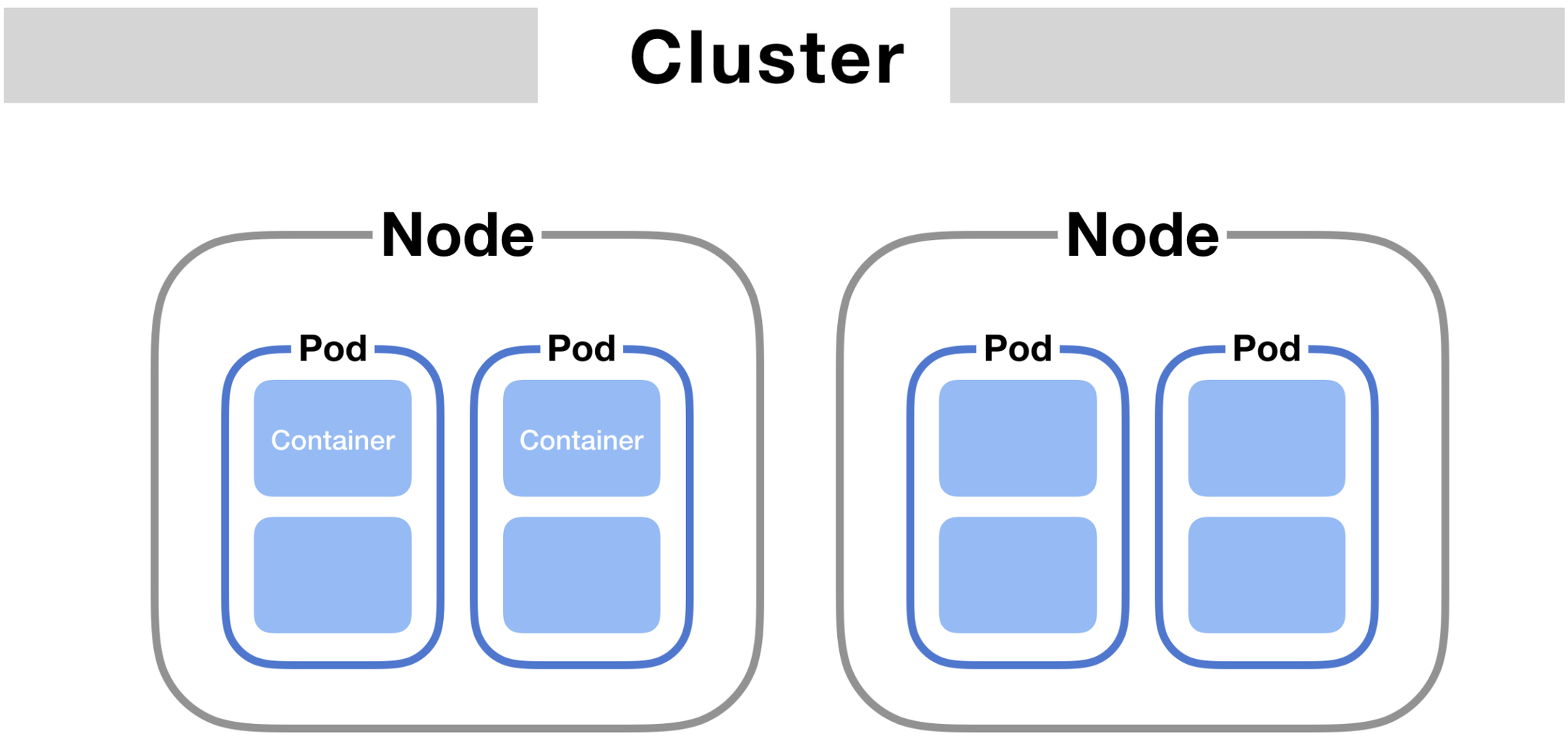
http://cursosdedesarrollo.com/

Pod

Lo interesante de los pods es que pueden ser replicados para dar más servicio

http://cursosdedesarrollo.com/

Pod



http://cursosdedesarrollo.com/

Volume

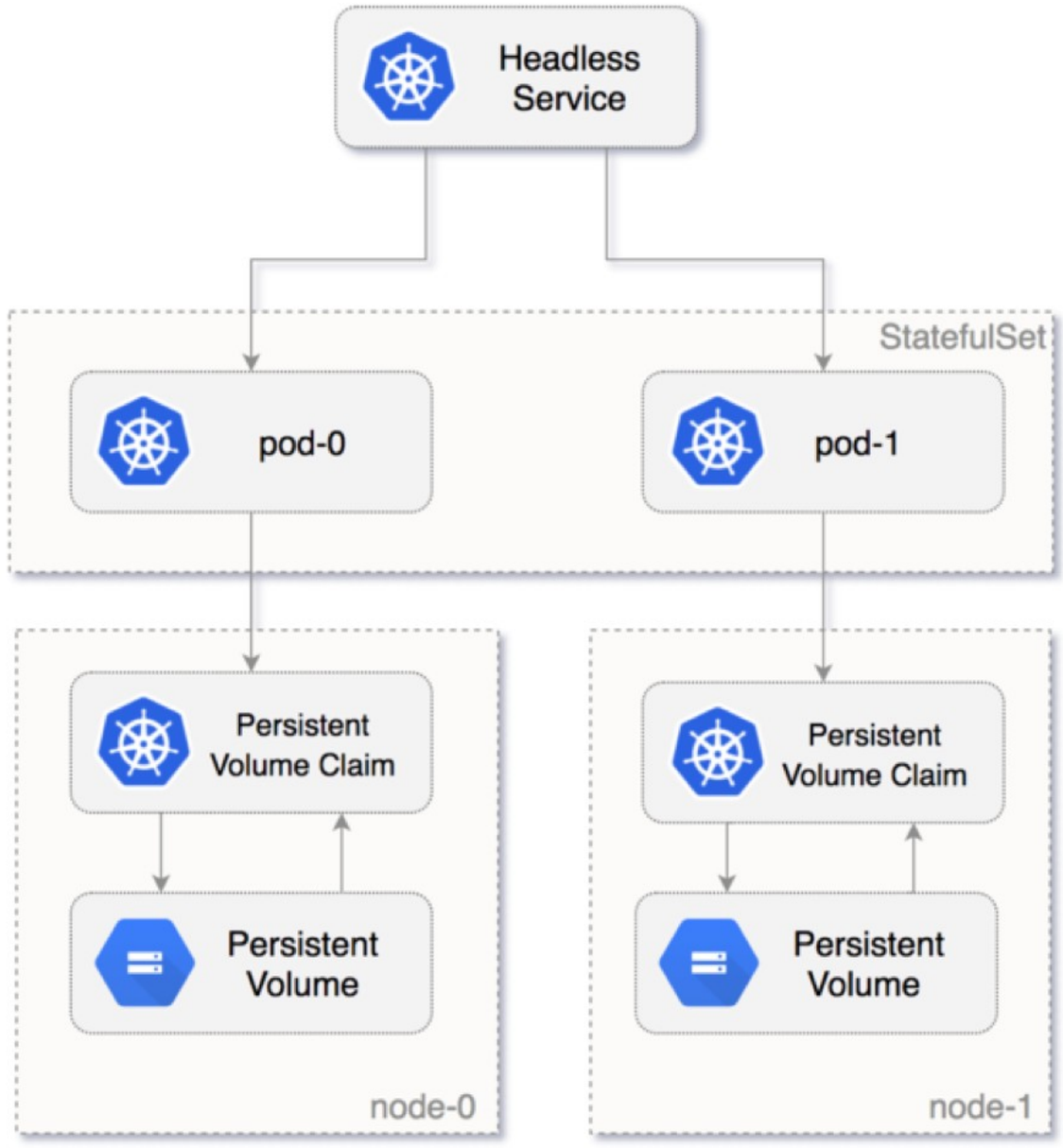
Los volúmenes son los directorios que almacenan información de los contenedores

http://cursosdedesarrollo.com/

Volume

Existen varios tipo de volúmenes dependiendo de los sistemas de almacenamiento en la nube integrados y directorios locales, como por ejemplo: AWS, Azure, GCE, Vsphere, emptyDir, hostPath y persistentVolumeClaim

Volume



http://cursosdedesarrollo.com/

Service

Un servicio permite exponer una aplicación que se ejecuta en un conjunto de pods

http://cursosdedesarrollo.com/

Service

Kubernetes ofrece una dirección IP interna y un nombre dns a cada set de pods y puede balancearlos

http://cursosdedesarrollo.com/

Service

Los pods afectados por ese servicio se seleccionarán por etiqueta (u otro tipo de selector)

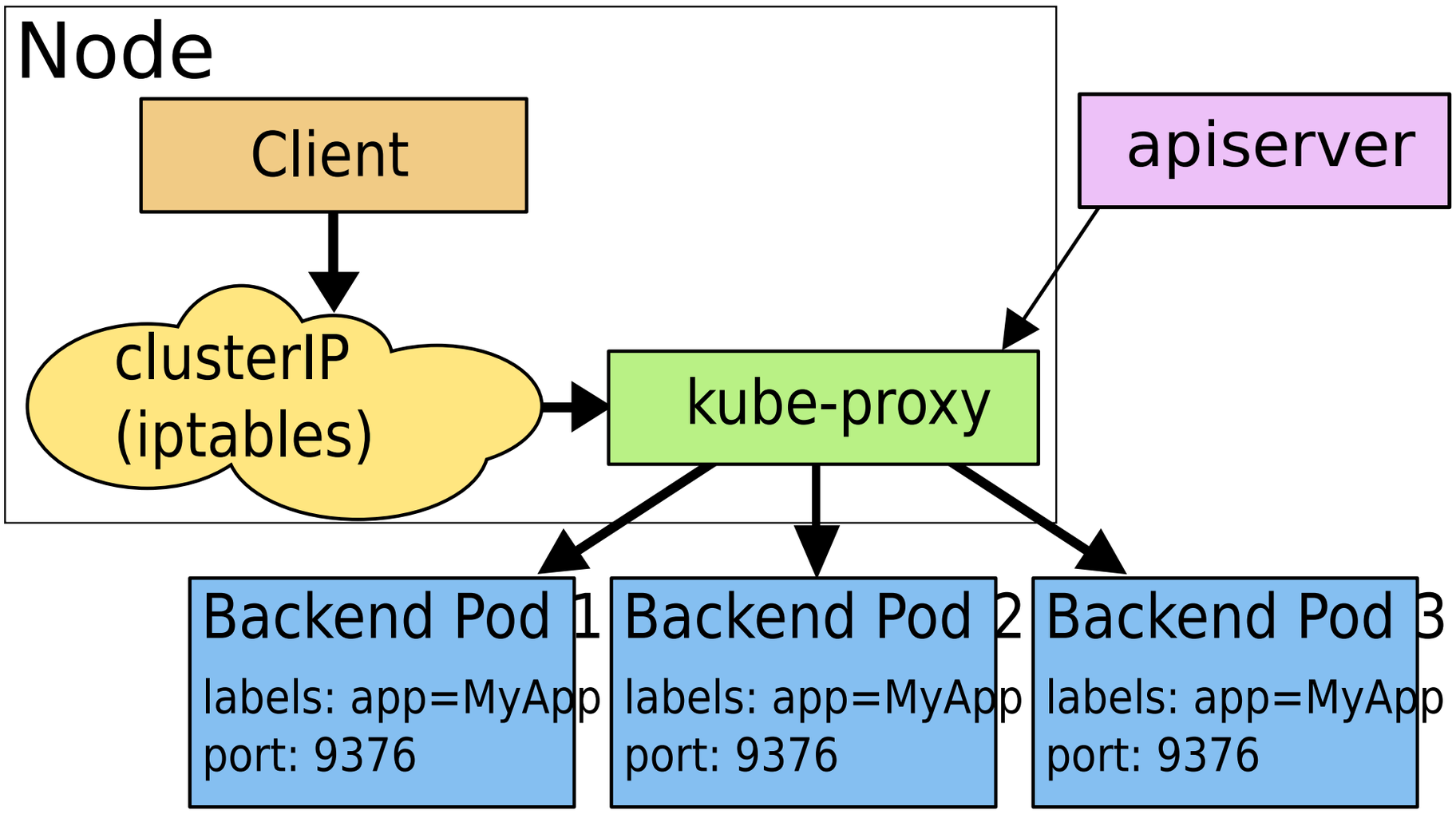
http://cursosdedesarrollo.com/

Service

Esto permite por ejemplo que los pods de frontend puedan encontrar a los de backend y el backend a la BBDD

http://cursosdedesarrollo.com/

Service



http://cursosdedesarrollo.com/

Service

Estos servicios tienen distintos tipos:

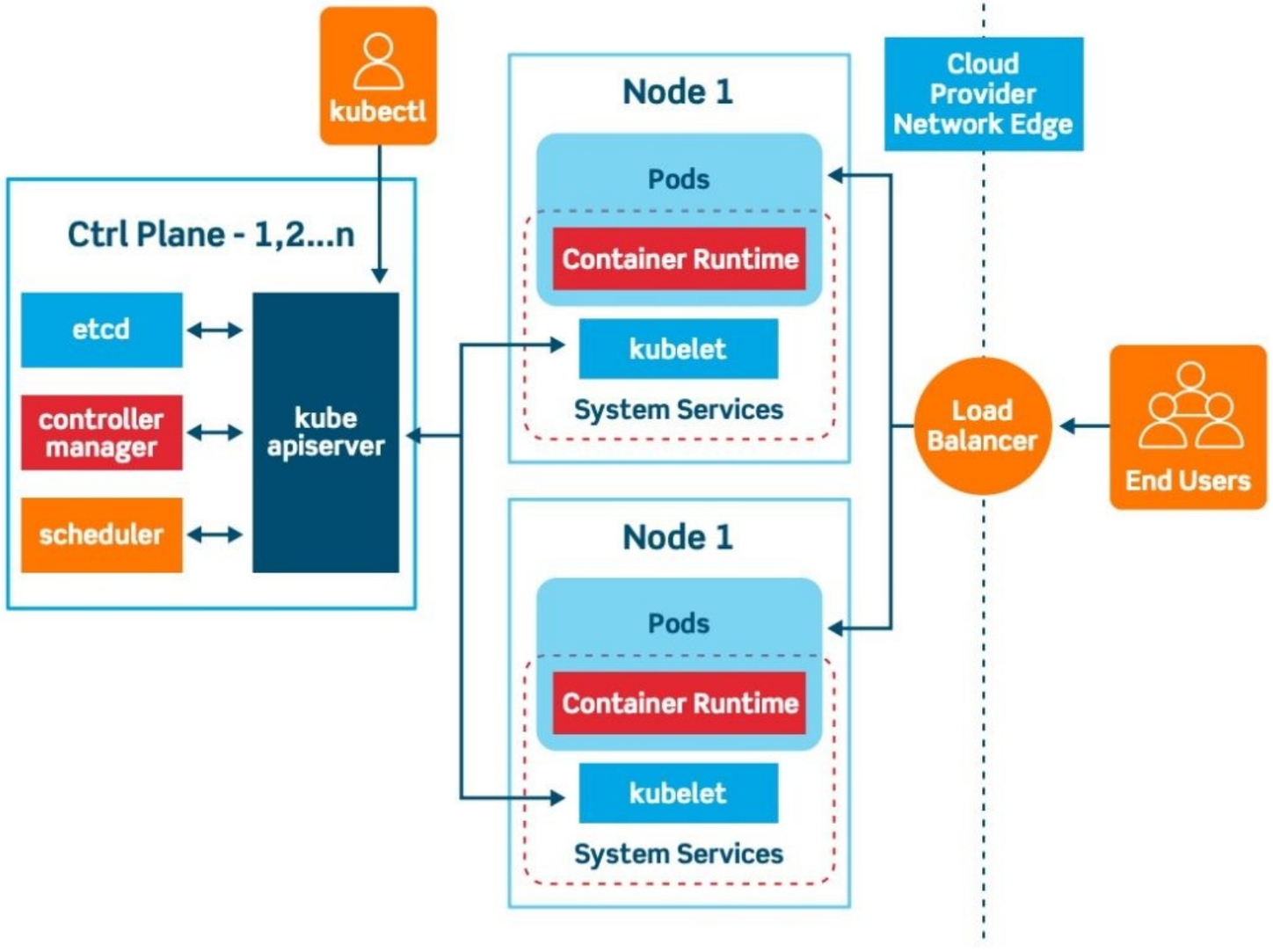
NodePort: expone el servicio en cada nodo en un puerto estático

LoadBalancer: expone el servicio externamente con la ayuda de algún Loadbalancer de una Cloud

ExternalName: Devuelve un CNAME

http://cursosdedesarrollo.com/

Service



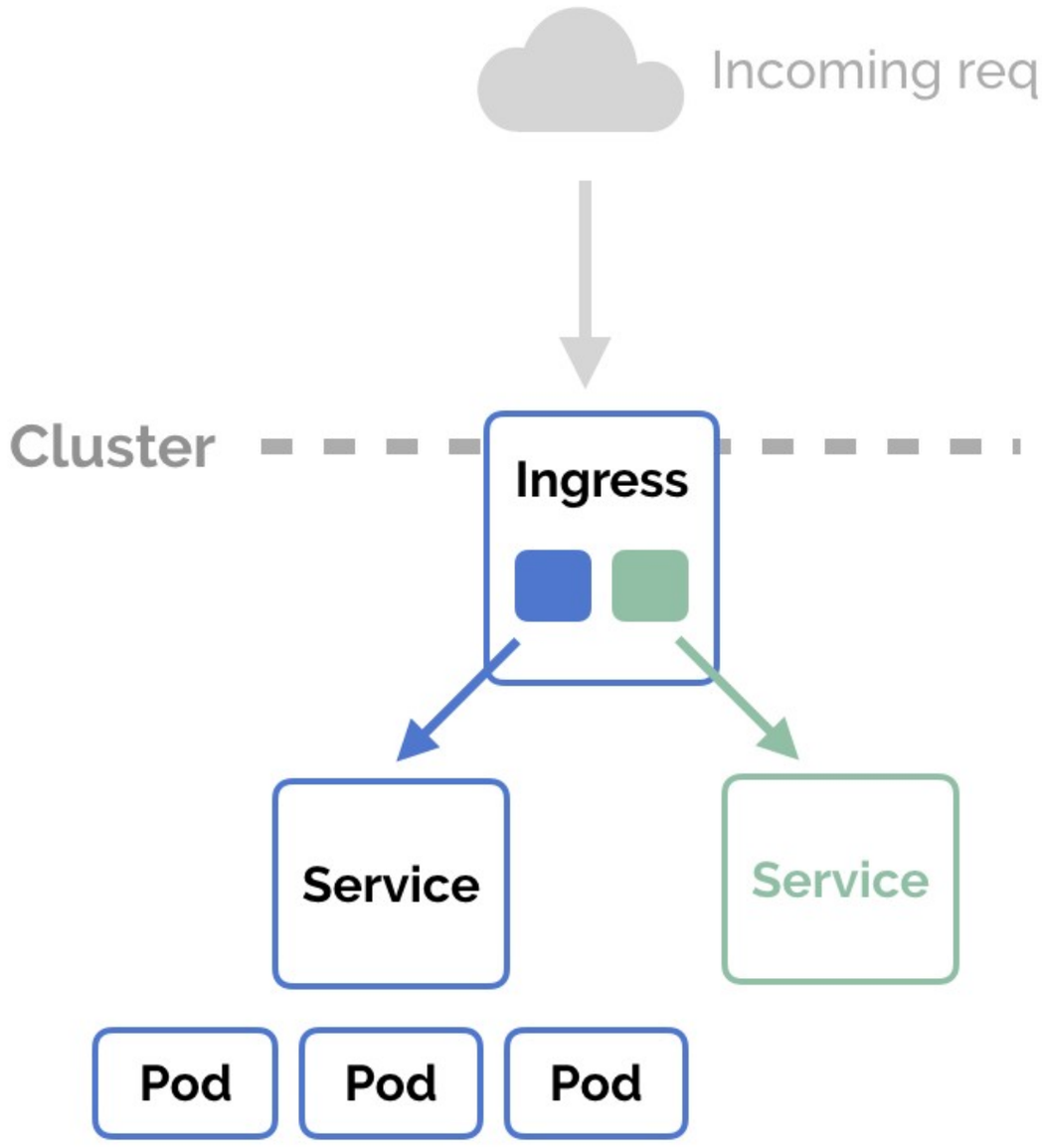
http://cursosdedesarrollo.com/

Service

Ingress es un servicio utilizable desde K8s para exponer los servicios de pods a internet

http://cursosdedesarrollo.com/

Service



http://cursosdedesarrollo.com/

NameSpace

Los espacios de nombres permiten definir divisiones entre los distintos pods (o resto de componentes) que tenemos en el cluster

http://cursosdedesarrollo.com/

NameSpace

Esto permite crear espacios de nombres para los distintos entornos que tenemos disponibles en el cluster

http://cursosdedesarrollo.com/

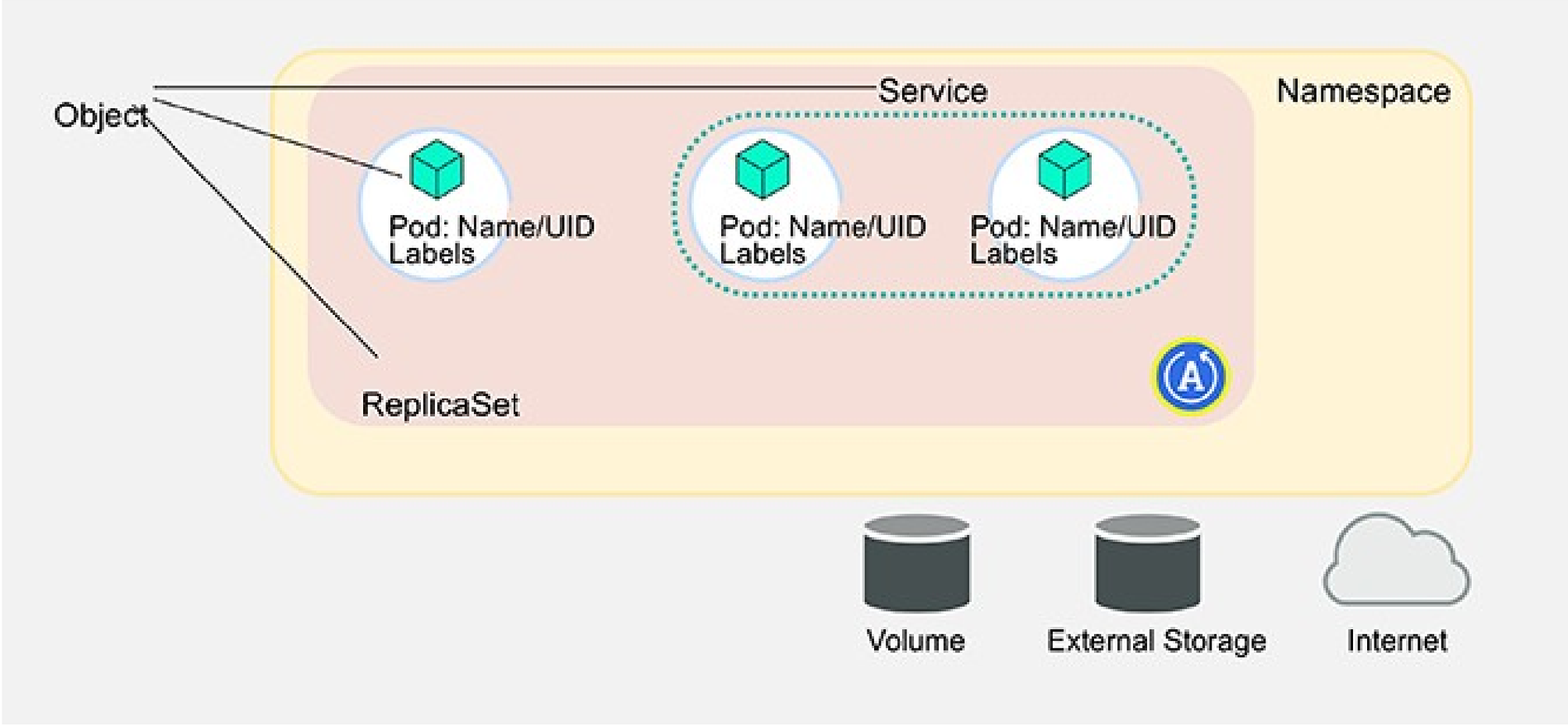
NameSpace

```
kubectl get namespace

NAME          STATUS    AGE
default       Active    1d
kube-system   Active    1d
kube-public   Active    1d
```

http://cursosdedesarrollo.com/

NameSpace



http://cursosdedesarrollo.com/

Replica Set

Un Replica Set en un Pods que escalan dentro del cluster, teniendo más de una instancia de ese Pod

http://cursosdedesarrollo.com/

Replica Set

K8s se encargará de que el número de réplicas del pod se mantengan activas

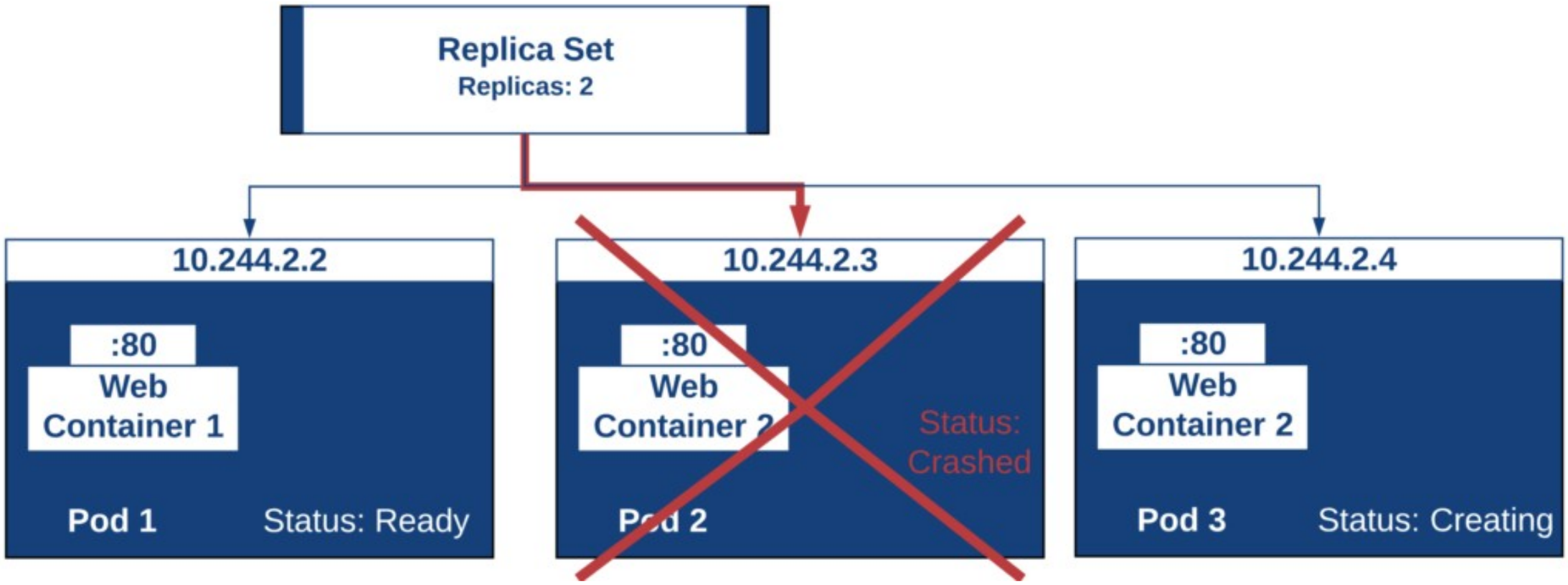
http://cursosdedesarrollo.com/

Replica Set

Si una réplica falla k8s arrancará una réplica en un nodo para compensar esa que ha fallado

http://cursosdedesarrollo.com/

Replica Set



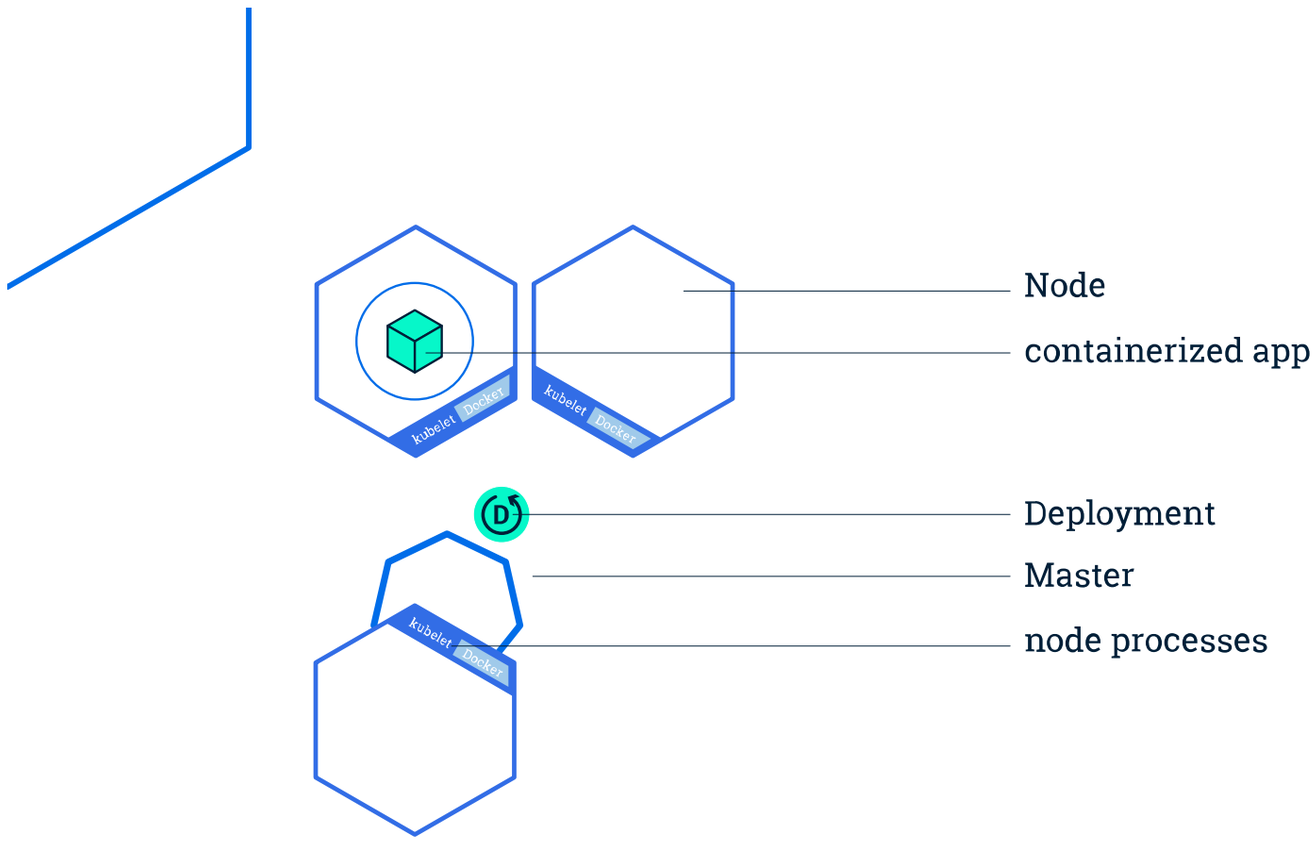
http://cursosdedesarrollo.com/

Deployment

Los despliegues son la manera de lanzar o cambiar un replica set en funcionamiento

http://cursosdedesarrollo.com/

Deployment



Kubernetes Cluster

http://cursosdedesarrollo.com/

Deployment

Estos despliegues suelen estar basados en fichero YAML que utilizamos para definir las replicas

http://cursosdedesarrollo.com/

Deployment

Tiene una coincidencia con los ficheros docker-compose.yaml de docker estandar

http://cursosdedesarrollo.com/

Deployment

El funcionamiento de Kompose en muy simple, una vez instalado con snap por ejemplo, ejecutaremos, en el directorio donde esté el docker-compose.yaml:

kompose convert

Deployment

Después nos bastaría con un:

kubectl create -f fichero.yaml

http://cursosdedesarrollo.com/

Deployment

O la opción vaga que es:

kompose up

Que hace el “convert” y el “create” juntos

http://cursosdedesarrollo.com/

Conclusiones

Hemos visto las bases del multiprocesado de datos en Python

http://cursosdedesarrollo.com/

Datos de Contacto

http://www.cursosdedesarrollo.com
david@cursosdedesarrollo.com

http://cursosdedesarrollo.com/

Licencia

David Vaquero Santiago

http://cursosdedesarrollo.com/