

# CONTENIDOS

## 1 DESPLIEGUE DE APLICACIONES EN CONTENEDORES (2 HORAS)

- Introducción a los contenedores
- Arquitectura de microservicios
- Tecnologías subyacentes y diferencias entre ellas: docker, cri-o, LXC, ...
- Ciclo de vida en el despliegue de aplicaciones con docker

## 2 INTRO A KUBERNETES (2 HORAS)

- Características, historia, estado actual del proyecto kubernetes (k8s)
- Arquitectura básica de k8s
- Alternativas para instalación simple de k8s: minikube, kubeadm, k3s
- Instalación con minikube
- Instalación y uso de kubectl
- Despliegue de aplicaciones con k8s

## 3 DESPLIEGUE DE APLICACIONES CON K8S (1:30 HORAS)

- Pods
- ReplicaSet: Tolerancia y escalabilidad
- Deployment: Actualizaciones y despliegues automáticos

## 4 COMUNICACIÓN ENTRE SERVICIOS Y ACCESO DESDE EL EXTERIOR (1:30 HORAS)

- Services
- DNS
- Ingress
- Ejemplos de uso y despliegues

## 5 CONFIGURACIÓN DE APLICACIONES (1 HORA)

- Variables de entorno
- ConfigMaps
- Secrets
- Ejemplo de despliegue parametrizado

## 6 ALMACENAMIENTO (1:30 HORAS)

- Consideraciones sobre el almacenamiento
- PersistentVolume
- PersistentVolumeClaim
- Ejemplo de despliegue con volúmenes

# CONTENIDOS

## 7 OTROS TIPOS DE DESPLIEGUES (1:30 HORAS)

- StatefulSet
- DaemonSet
- AutoScale
- Helm

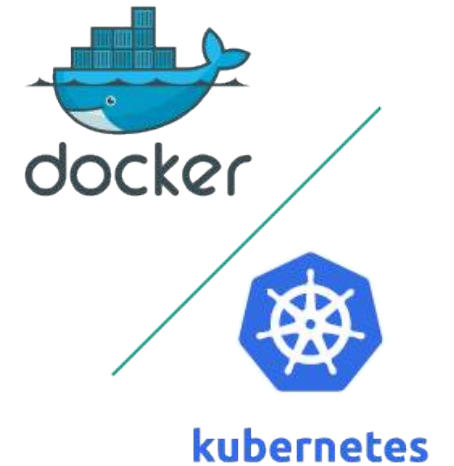
## 8 ADMINISTRACIÓN BÁSICA (1 HORA)

- Namespaces
- Usuarios
- RBAC
- Cuotas y límites

## 9 INSTALACIÓN PASO A PASO

### (4 HORAS)

- Consideraciones previas:  
Requisitos hardware, arquitectura física y lógica, entornos y herramientas para el despliegue
- Instalación completa componente a componente en múltiples nodos



## MÓDULO 9. INSTALACIÓN PASO A PASO

## Índice

- Consideraciones previas: Requisitos hardware, arquitectura física y lógica, entornos y herramientas para el despliegue
- Instalación completa componente a componente en múltiples nodos

## Consideraciones previas. Requisitos de hardware

- Kubernetes no es muy exigente en cuanto a recursos y el software propio tiene unos requisitos bastante
  - Master: 3GiB RAM y 1,5 cores
  - Nodo: 1 GiB RAM y 1 core
- El tamaño y los recursos de k8s están relacionados sobre todo con los requisitos de las aplicaciones que se ejecuten sobre él.
- En entornos en producción es obligado utilizar alta disponibilidad en los masters, por lo que la configuración habitual es de 3 nodos máster en estos casos
- Los nodos no necesitan estar en alta disponibilidad

## Consideraciones previas. Arquitectura física

- En la mayoría de las situaciones la arquitectura física es muy simple: Todos los nodos conectados en la misma red con una sola interfaz de red
  - Dirección IP externa. Puede ser una interfaz dedicada o típicamente una IP elástica/flotante del proveedor accesible desde el exterior mediante NAT
  - Dirección IP interna
- Segmentación de redes (seguridad)
- Almacenamiento
  - Típicamente proporcionado por la nube de infraestructura
  - Es posible utilizar cualquier sistema SAN/NAS en una instalación a medida
  - Ceph y Rook: <https://ceph.com/community/rook-automating-ceph-kubernetes/>  
<https://kubernetes.io/docs/concepts/storage/storage-classes/#provisioner>

## Consideraciones previas. Arquitectura lógica

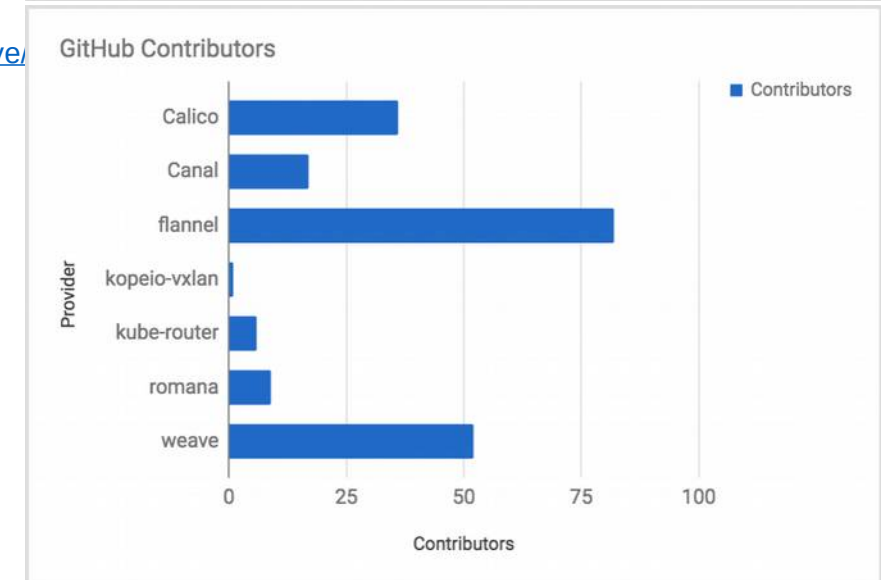
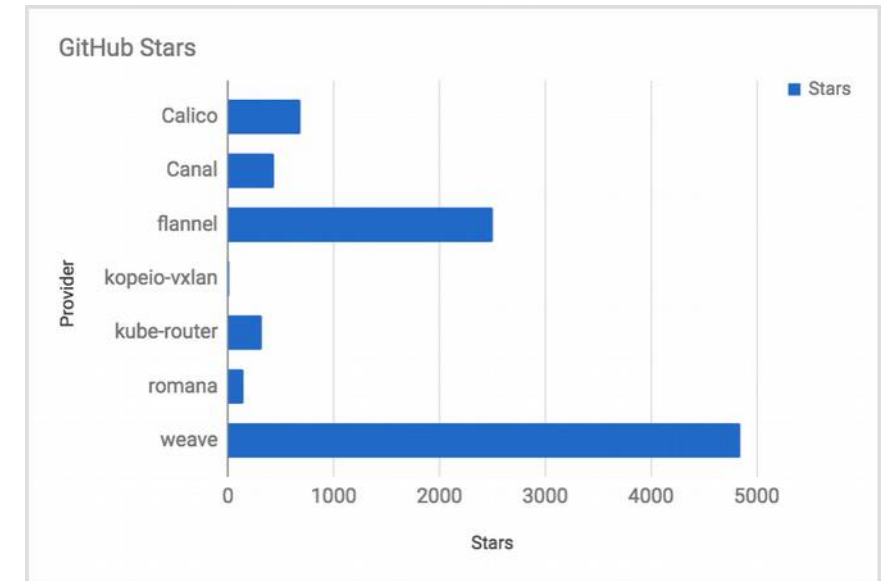
- Direcccionamiento
  - Red de nodos
  - Red de pods
  - Red de servicios
- Se puede acceder a otro contenedor dentro de un pod utilizando localhost, puesto que comparten la pila de red
- Comunicación entre pods sin NAT utilizando la IP asignada por k8s
- Comunicación Servicios a pods
- Comunicación con el exterior
- Redes Overlay
- Container Network Infrastructure (CNI)
- Consideraciones sobre almacenamiento

## Consideraciones previas. Proveedores de red CNI

- Flannel
- Calico
- Canal
- Weave
- ...

<https://rancher.com/blog/2019/2019-03-21-comparing-kubernetes-cni-providers-flannel-calico-canal-and-weave/>

<https://chrislovecnm.com/kubernetes/cni/choosing-a-cni-provider/>





## Consideraciones previas. Herramientas para el despliegue

- Instalación de k8s en entornos en producción: <https://kubernetes.io/docs/setup/#production-environment>
- ¿Un clúster para todas las aplicaciones o un clúster para cada aplicación? Ventajas e inconvenientes

## Instalación y uso en Google Kubernetes Engine

- Instalación de gcloud
- Configuración de gcloud (zona, region)
- Instalación de un clúster desde la consola web
- Configuración de kubectl:  
gcloud container clusters get-credentials "nombre-cluster"
- Características de un clúster gestionado
  - No hay nodo master, hay un servicio de pago por uso del proveedor
  - No tenemos acceso a los componentes del master, incluyendo etcd
  - Tenemos un entorno en el que lanzar pods y utilizar todas las características que proporciona
  - Disponibilidad, facilidad de instalación y actualización
  - Peero, perdiendo el control completamente sobre el cluster
  - En este caso tenemos acceso por ssh a los nodos en los que se ejecuta, pero no al master

## Instalación y uso en OpenStack con Magnum

- Suponemos un entorno OpenStack disponible con el componente Magnum
- Características de un clúster gestionado
  - Se utiliza una plantilla de heat para lanzar el escenario que crea un clúster con las características que deseemos
  - Sí hay nodo master y por tanto controlamos todos los componentes del master
  - La actualización de los clústeres no es trivial

## Instalación paso a paso desde cero

- Útil para entender cómo realmente funciona kubernetes
- No es la opción recomendada inicialmente, ya que es compleja y hay muchos conceptos que entender del propio funcionamiento de los objetos de k8s
- Recomendada para quienes tengan que realizar y mantener despliegues en entornos en producción y quieran un conocimiento en detalle
- En muchos casos una instalación real no se hará nunca paso a paso, se suelen utilizar instaladores

## Instalación paso a paso desde cero. Configuraciones iniciales

- sysctl:  

```
net.ipv4.ip_forward = 1  
net.ipv4.conf.all.forwarding = 1  
net.bridge.bridge-nf-call-iptables = 1
```
- Resolución estática de todos los nodos
- Instalamos docker en todos los nodos: `apt install docker.io`
- Seleccionamos un plugin para CNI: flannel
- Seleccionamos una red para flannel: 10.10.0.0/16
- Seleccionamos una red para los servicios: 10.20.0.0/24

## Instalación paso a paso desde cero. Autoridad certificadora

- Creamos la autoridad certificadora (no ponemos passphrase por simplificar):

```
openssl genrsa -out /etc/ssl/private/ca.key 4096  
chmod 0400 /etc/ssl/private/ca.key
```

- Generamos el certificado x509 autofirmado de la CA:

```
openssl req -x509 -new -nodes -key /etc/ssl/private/ca.key -sha256 -days 1024 -out /usr/local/share/ca-certificates/ca.crt -subj "/CN=ca"
```

- Copiamos este certificado a todos los nodos en la misma ubicación
- Ejecutamos update-ca-certificates en todos los nodos
- Si no existiera, creamos el grupo ssl-cert (gid 54) y asignamos /etc/ssl/private a este grupo (chmod 750 /etc/ssl/private)
- Utilizamos el openssl.cnf de github para crear alias
- Cada nodo de k8s debería tener su propia clave, CSR y firmarse por la CA:

```
openssl genrsa -out /etc/ssl/private/master.key 4096  
chmod 0440 /etc/ssl/private/master.key  
chgrp ssl-cert /etc/ssl/private/master.key  
openssl req -new -key /etc/ssl/private/master.key -out /tmp/master.csr -subj "/CN=master"  
openssl x509 -req -in /tmp/master.csr -CA /usr/local/share/ca-certificates/ca.crt -CAkey /etc/ssl/private/ca.key -out /usr/local/share/  
ca-certificates/master.crt -days 500 -extensions v3_req -extfile openssl.cnf
```

## Instalación paso a paso desde cero. etcd

- Instalamos etcd desde repositorios  
apt install etcd-server
- Almacena el estado del despliegue de kubernetes, siendo un componente esencial
- Desde el punto de vista de la seguridad es imprescindible controlar el acceso a etcd donde reside toda la información de k8s
- En un despliegue real se instalaría un cluster de servidores etcd
- Creamos el directorio /etc/etcd
- Configuración del certificado SSL:
  - chgrp -R ssl-cert /etc/ssl/private/

## Instalación paso a paso desde cero. etcd

- Configuración de etcd (/etc/default/etcd):
  - ETCD\_NAME="guadaltel"
  - ETCD\_DATA\_DIR="/var/lib/etcd/guadaltel"
  - ETCD\_ADVERTISE\_CLIENT\_URLS="https://localhost:2379,https://10.0.0.14:2379,https://master:2379"
  - ETCD\_LISTEN\_CLIENT\_URLS="https://localhost:2379,https://10.0.0.14:2379,https://master:2379"
  - ETCD\_TRUSTED\_CA\_FILE="/usr/local/share/ca-certificates/ca.crt"
  - ETCD\_CLIENT\_CERT\_AUTH
  - ETCD\_CERT\_FILE="/usr/local/share/ca-certificates/master.crt"
  - ETCD\_KEY\_FILE="/etc/ssl/private/master.key"
- Cambiamos propietario y grupo de /etc/etcd: `chown -R etc. /etc/etcd`
- Reiniciamos el servicio (existe una unidad de systemd lista)



## Instalación paso a paso desde cero

- Descarga de los binarios de servidores y clientes de k8s
- Creación de los certificados para los servicios de k8s
- Creación del usuario admin y configuración de kubectl
- Configuración de flannel
- Configuración de kube-apiserver, kube-scheduler y kube-controller-manager
- Configuración de kubelet y kube-proxy