

# Cómo montar tu propio clúster kubernetes

...

Daniel Aguado Araujo

# Índice

- Introducción a kubernetes
- Instalando kubernetes
- Desplegando aplicaciones en el cluster
  - Replication controllers & Pods
  - Volumes
  - Secrets
  - Services
  - Daemon sets

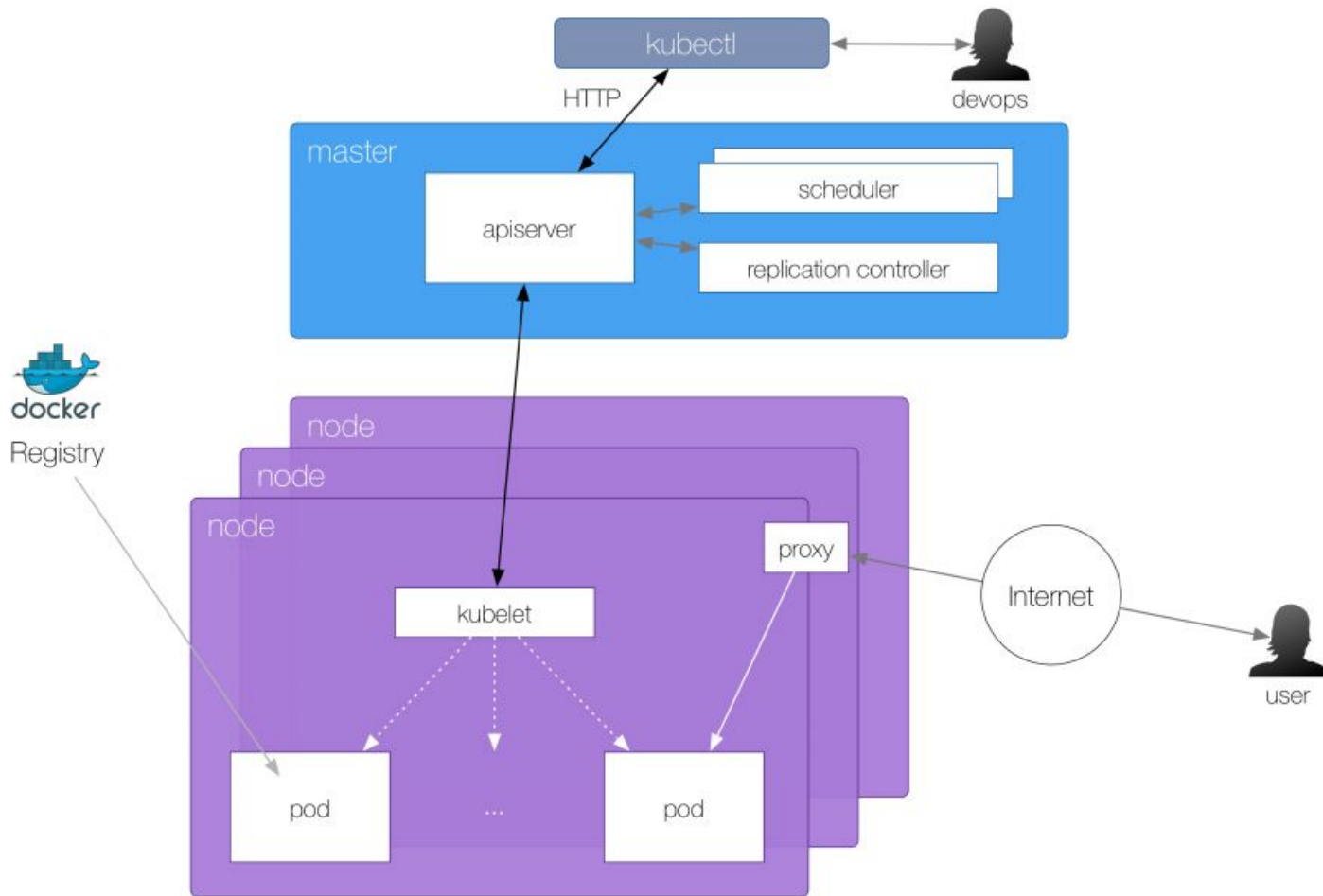
# Introducción a kubernetes

# Introducción a kubernetes

- ¿Qué es kubernetes?
  - Un orquestador de contenedores desarrollado por Google
- ¿Qué ofrece kubernetes?
  - Despliegue de aplicaciones sin Denegación de Servicio
  - Gestión automática de la red interna
  - DNS y balanceador interno para acceder a otros recursos
  - Escalado sencillo de aplicaciones
  - Panel de control
  - Recuperación ante errores en los nodos
  - Añadir de forma sencilla más nodos al clúster

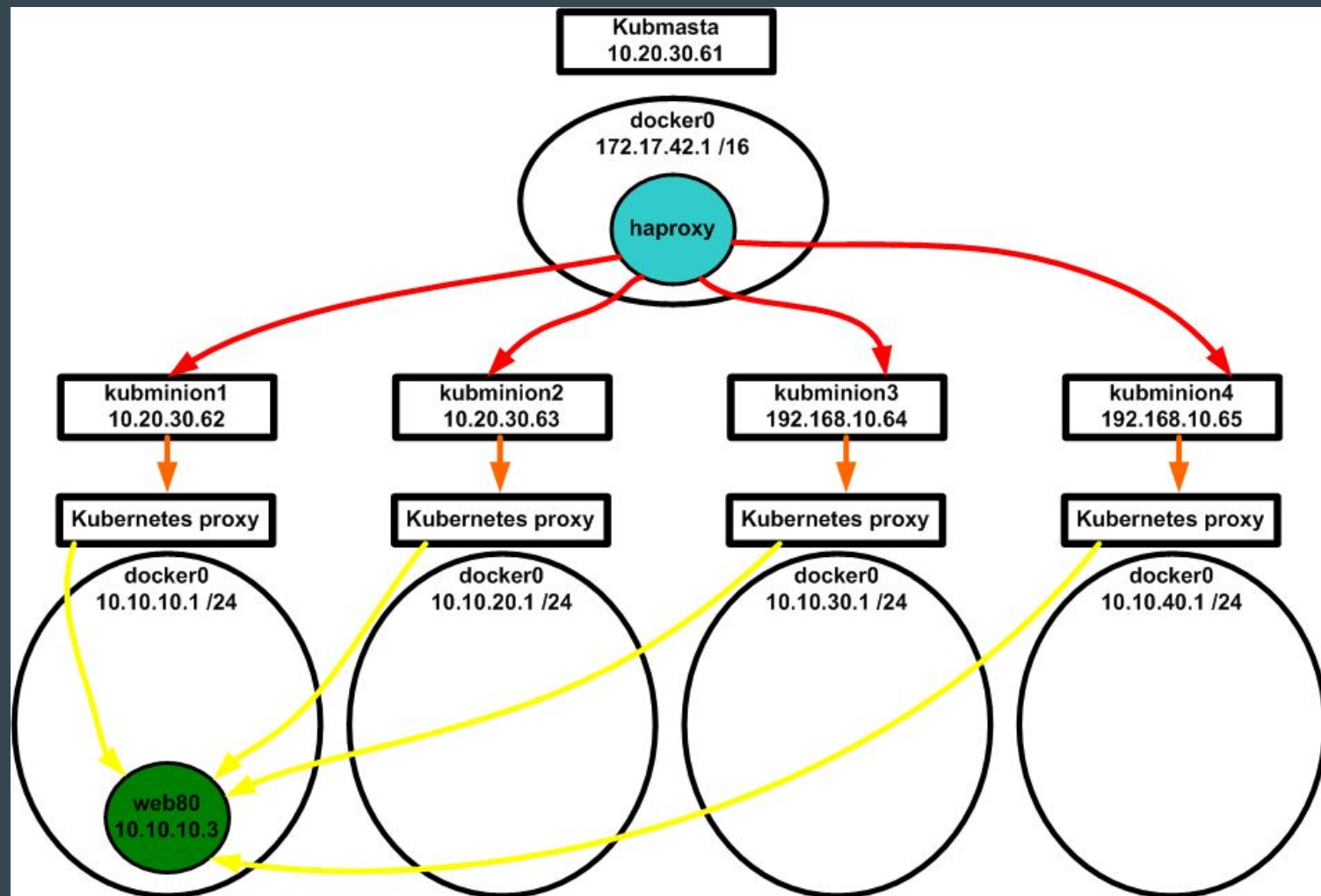
# Introducción a kubernetes. Vista de Infraestructura

- Componentes:
  - kubelet → agente de kubernetes, permite acceder a la api
  - kube-proxy → gestiona la red y actúa como balanceador de carga. Se encarga de dirigir el tráfico al nodo donde esté levantado el contenedor
  - kubernetes api, scheduler, replication controller
  - etcd → base de datos
  - flanneld → gestiona una red virtual entre todos los nodos del clúster
  - skydns → plugin para gestionar el DNS interno
  - dashboard → plugin para gestión visual del cluster
- Herramientas de gestión:
  - kubectl → aplicación de consola para gestionar de forma remota el clúster
  - dashboard → aplicación web para gestionar el clúster



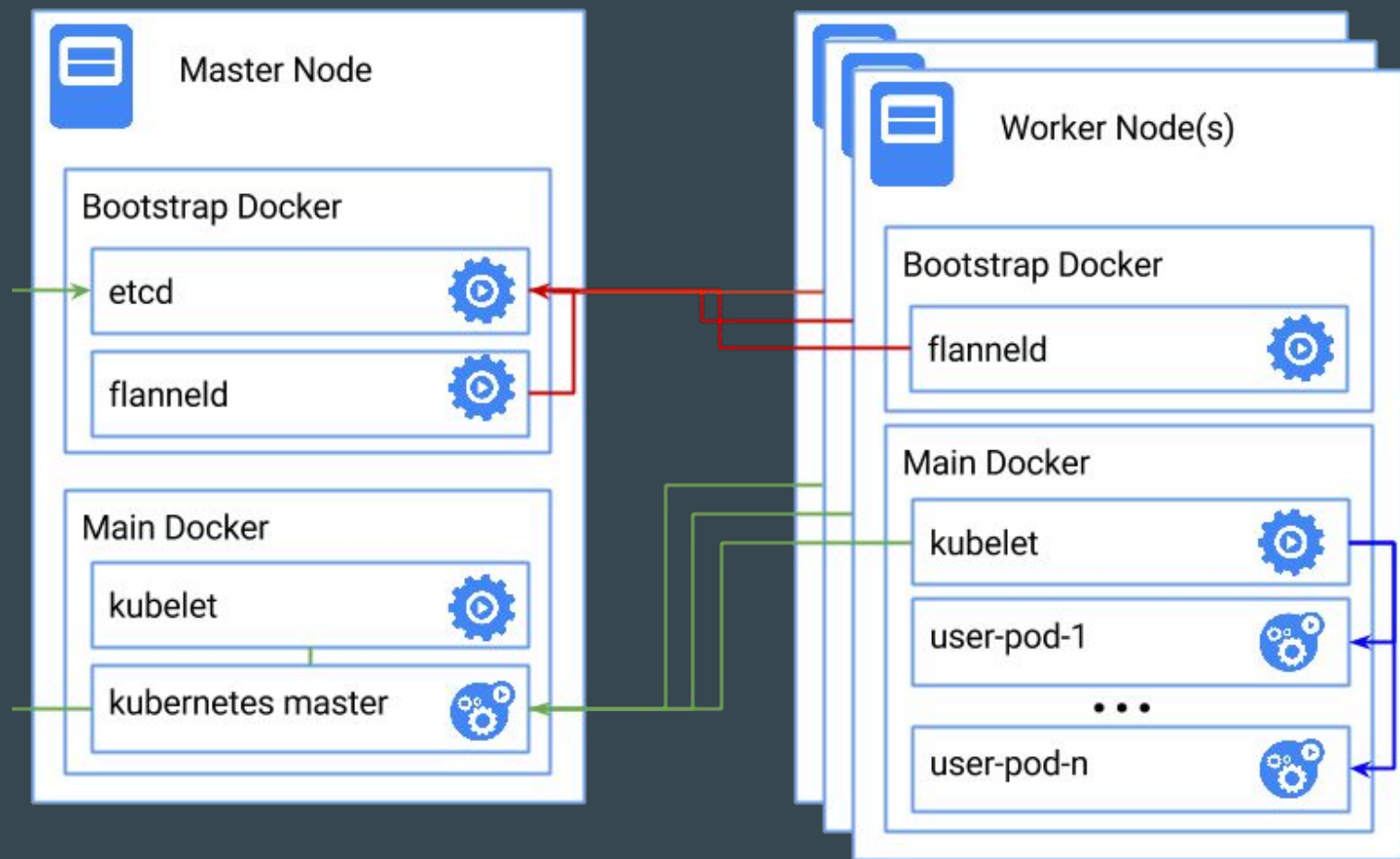
# Introducción a kubernetes. Vista de Aplicaciones

- POD
  - Contenedor funcionando en kubernetes
- Replication Controller
  - Componente que define uno o varios pods, sus componentes, su red, sus volúmenes y su número.
  - Le da nombre a los pods y configura el DNS y el balanceador de carga interno
- Service
  - Componente que conecta la red externa con los pods.
  - Configura el balanceador de carga externo





# Instalando kubernetes



# Instalando kubernetes

- Requisitos
  - Instalar Docker
  - Ejecutar

```
git clone https://github.com/kubernetes/kube-deploy
cd kube-deploy/docker-multinode
```

```
export IP_ADDRESS=my-private-ip
```

```
----- ON MASTER NODE -----
./master.sh
```

```
----- ON WORKER NODE ----
export MASTER_IP=master-private-ip
./worker.sh
```

# Instalando kubernetes

- Requisitos
  - Instalar Docker
  - Ejecutar

```
git clone https://github.com/kubernetes/kubernetes
cd kube-deploy/docker/multi-node
```

```
export KUBE_MASTER_IP=master-private-ip
```

```
----- ON MASTER NODE -----
```

```
./master.sh
```

```
----- ON WORKER NODE -----
```

```
export MASTER_IP=master-private-ip
./worker.sh
```

# Instalando kubernetes

- Descargar kubectl

```
curl -O https://storage.googleapis.com/kubernetes-release/release/v1.4.3/bin/linux/amd64/kubectl  
chmod +x kubectl  
mv kubectl /usr/bin/kubectl
```

- Acceder al dashboard
  - <http://ip-maestro:8080/ui>

# Desplegando aplicaciones en el clúster

# Desplegando aplicaciones en el clúster. Ejemplos

- Replication Controller
  - nginx
  - 2 versiones:
    - Carga una web desde un secret
    - Carga una web desde un directorio local del host
  - Rolling update de una versión a otra y test para probar el DoS sufrido
- Service
  - Configuraremos un servicio para el nginx
- DaemonSet
  - montaremos un cliente de Icinga2 en cada nodo del clúster

# Replication Controller & Pods. Volumes & Secrets

- Commands
  - `kubectl create secret generic nginx-website --from-file=./index.html`
  - `kubectl create -f nginx-replication-controller-v1.yaml`
  - `kubectl rolling-update nginx-replication-controller-v1 -f nginx-replication-controller-v2.yaml`



# Volumes

- Se definen en el POD
- Soportados:
  - gcePersistentDisk (google)
  - awsElasticBlockStore
  - azureFileVolume
  - azureDisk
  - vsphereVolume (VMware)
  - floker
  - rdb
  - cephfs
  - quobyte
  - hostPath
  - NFS
  - gitRepo
  - secret
  - iscsi
  - glusterfs
  - persistentVolumeClaim
  - downwardAPI

# Services

- Commands
  - `kubectl create service loadbalancer my-lbs --tcp=5678:8080`
  - `kubectl create -f nginx-service.yaml`

# DaemonSet

- Son un tipo de replication controller
- Se instala al menos uno por nodo
- Permite ejecutarse con privilegios
- Principalmente pensado para monitorización

