

Despliegue automático aplicaciones en la nube usando Contenedores

Foro tecnológico de empleo



Qué es **TELTEK** ?

- Empresa “EBT” nacida en Campus UVigo
- Centrada en Video-Online educativo:
 - Repositorios de video
 - Grabación automatizada de clases y conferencias
 - Plataformas MOOC, Videos para MOOCs
 - Soluciones masivas de videoconferencia web



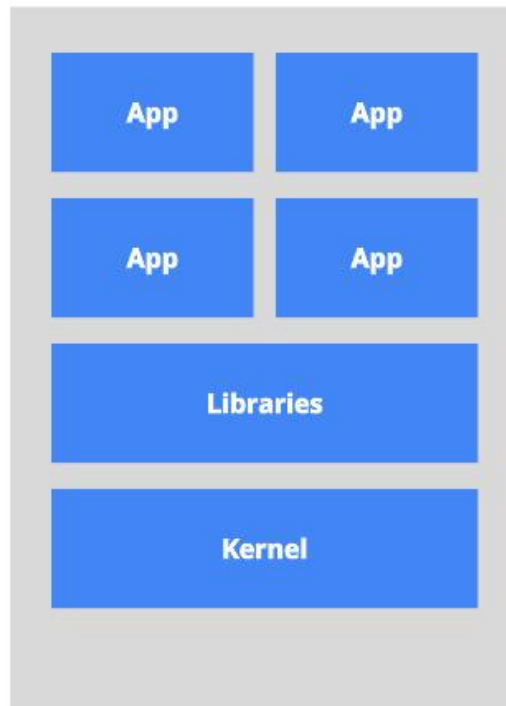
Contenido

- 1. Introducción a tecnología de contenedores**
- 2. Contenedores Docker**
- 3. Demo Docker**
- 4. Introducción a Kubernetes**
- 5. Demo Kubernetes**

¿Que es la tecnología de Contenedores?

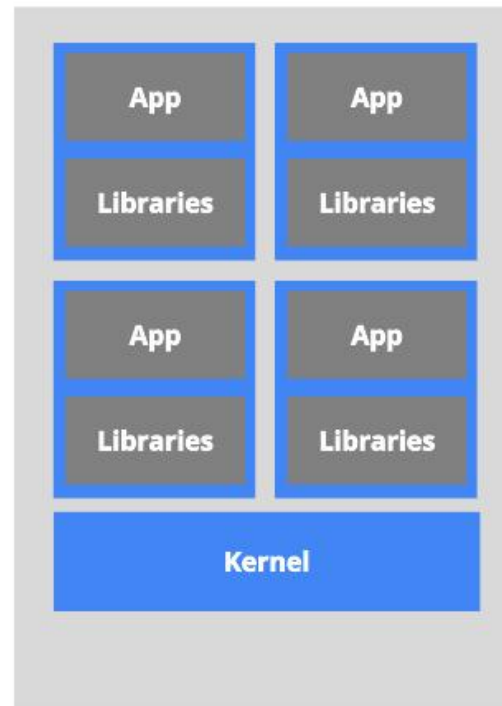
- Contenedores: Virtualización de nivel SO

The old way: Applications on host



*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers

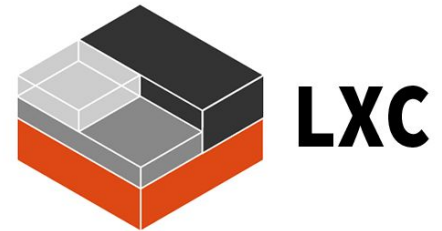


*Small and fast, portable
Uses OS-level virtualization*

Contenedores: LXC a Docker

Linux Containers (LXC)

- Múltiples sistemas Linux aislados (Contenedores)
- Host de control con único Linux kernel
- Kernel 2.6.24 (Agosto 2008)



Docker

- Facilita y automatiza la ejecución de contenedores
- Liberado en 2013
- Hasta versión 0.9 usa LXC
- libcontainer con cgroups + namespace
- Runtime actual: containerd



Componentes Docker

Servicio Dockerd

- Gestiona ciclo de vida los contenedores + información de estado
- Implementa Docker Engine API

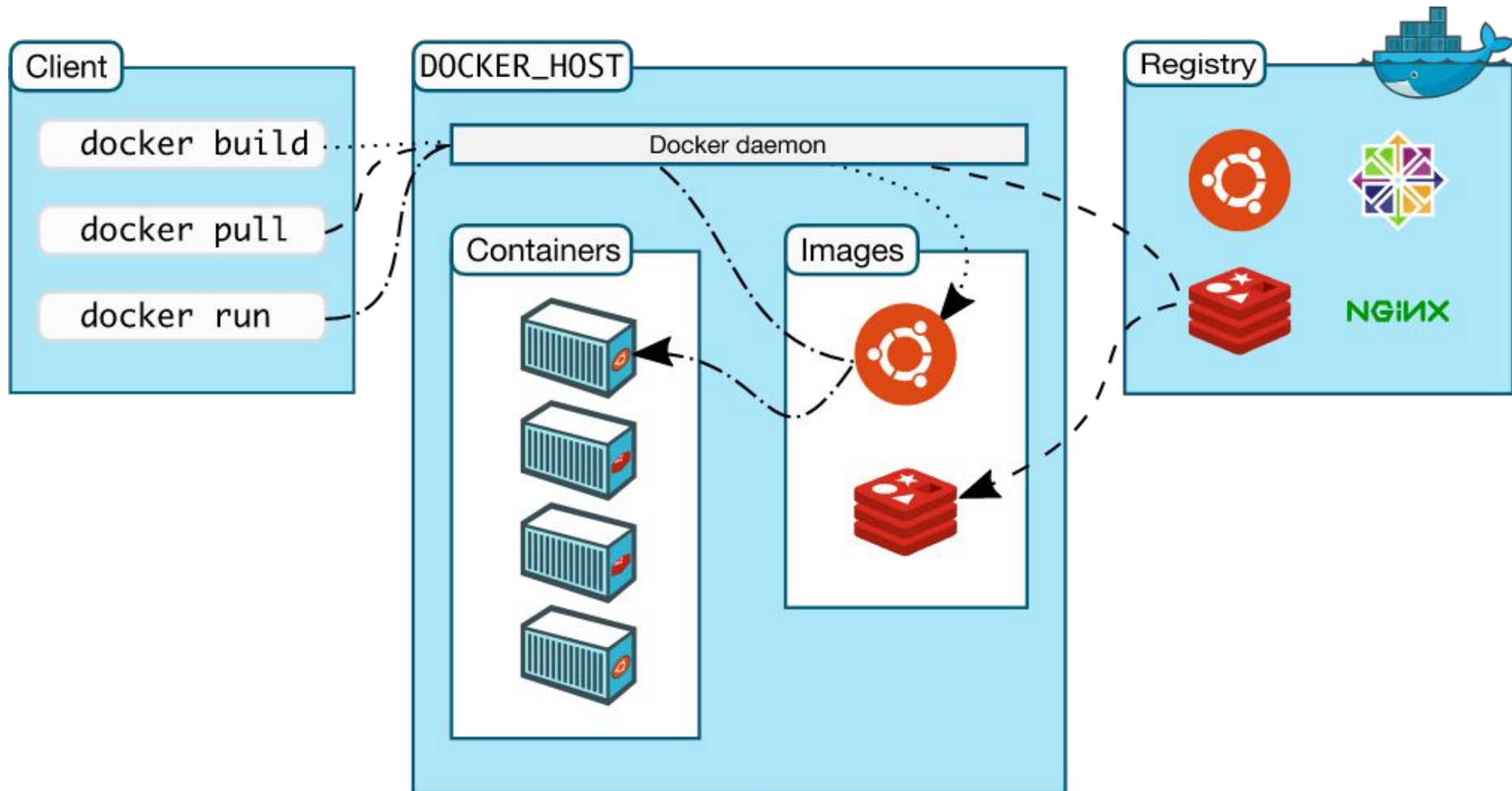
Cliente Docker

- Utilidad CLI permite interactuar con contenedores Docker
- “Habla” con Dockerd a través API

Docker Registry

- Repositorio de imágenes

Componentes Docker



Contenedores: Docker a OCI

OCI

- **Lanzada en 2015 Docker/CoreOS**
- **Propósito: Creación estándares abiertos p**
- **2 Especificaciones:**
 - **Runtime Specification (runtime-spec) (runc)**
 - **Image Specification (image-spec)**

Contenedores vs Virtualización

Virtualización:

- Instalación con gestor de paquetes. Repositorios de 3os!!
- Ciclo de vida: ejecutables, configuración, librerías
- Rollouts y rollbacks con imágenes muy pesadas (5-10G)
- SO soportado por sistema Virtualización
- Acceso al HW!!

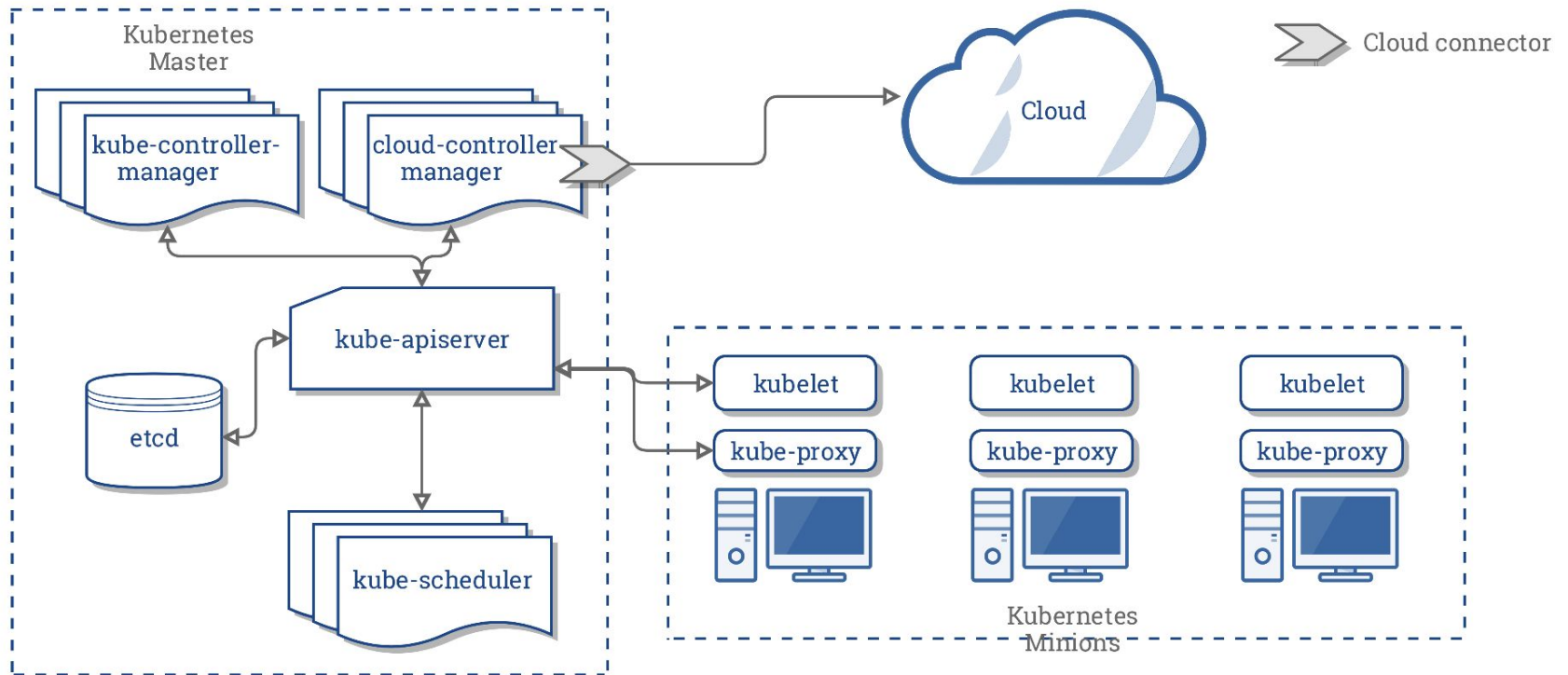
Contenedores:

- Aislado del anfitrión. Limitación
- Librerías independientes
- Independiente SO

Kubernetes: Introducción

- **Plataforma Open Source portable y extensible para manejar cargas de trabajo y servicios**
- **Configuración declarativa de contenedores**
- **Estándar para orquestación de contenedores**

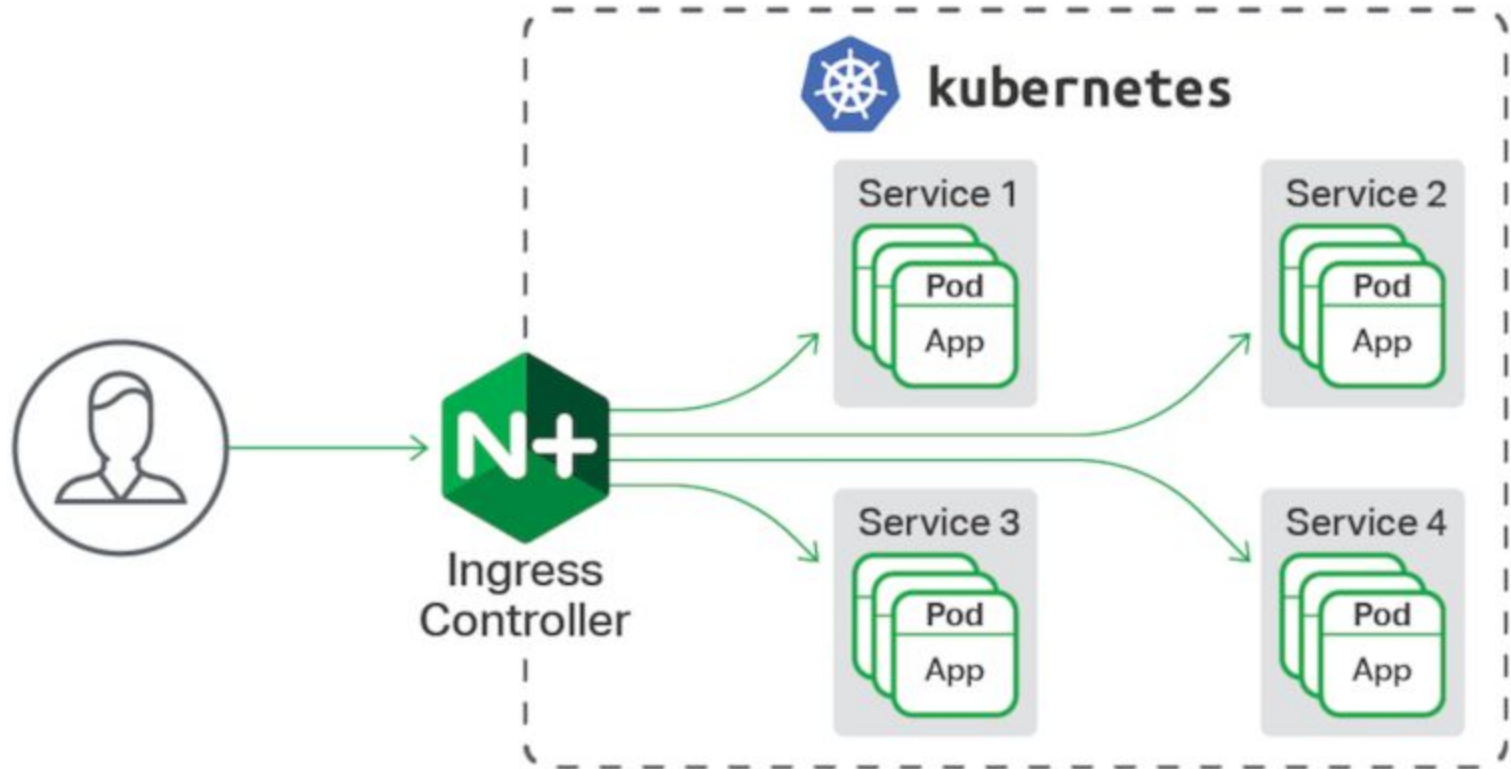
Kubernetes: Arquitectura



Kubernetes: Conceptos

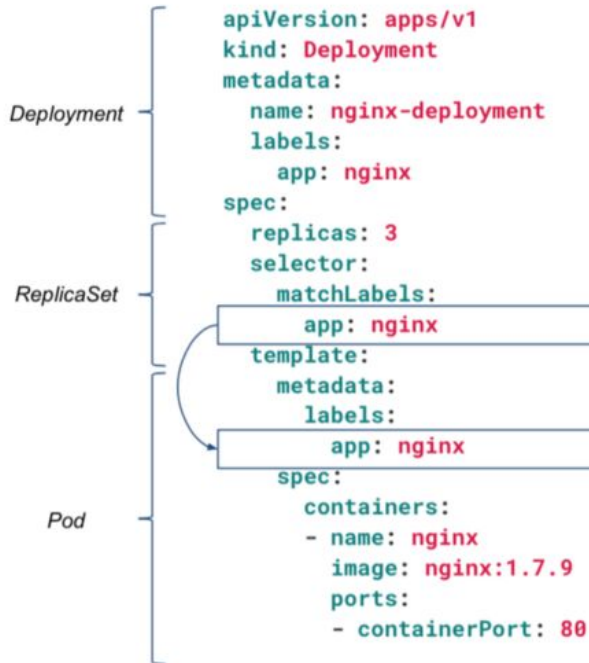
- **Pod:** grupo de uno o más contenedores con almacenamiento y red compartidos (“Host lógico”)
- **Volume:** Resuelve 2 problemas
 - Persistencia de datos de un pod
 - Compartición de datos entre pod
- **Service:** Exponer como servicio de red una aplicación que se ejecuta en grupo de pods
- **Deployment:** Abstracción que proporciona configuración declarativa sobre los Pod
- **Ingress Controller:** Expone rutas HTTP/HTTPs fuera del clúster

Kubernetes: Deployments, Services e Ingress



Kubernetes: Deployments, Services e Ingress

Deployment configuration:



Service configuration:

```
kind: Service
apiVersion: v1
metadata:
  name: my-service
spec:
  selector:
    app: nginx
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
```

Ingress configuration:

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: my-ingress
  annotations:
    kubernetes.io/ingress.class: nginx
spec:
  rules:
    - host: foo.bar.com
      http:
        paths:
          - path: /foo
            backend:
              serviceName: my-service
              servicePort: 80
          - path: /bar
            backend:
              serviceName: my-other-service
              servicePort: 80
```

Gracias!

Pablo Nieto
pnieto@teltek.es

www.TELTEK.es

