

# 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

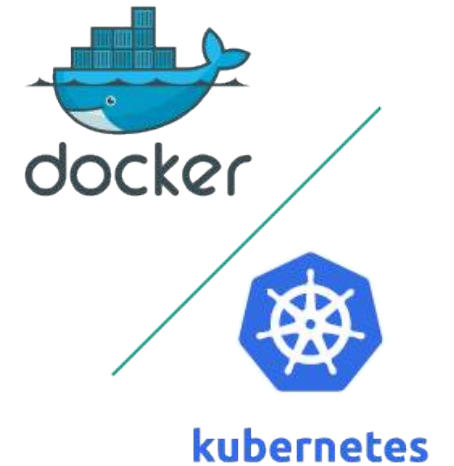
## 8 ADMINISTRACIÓN BÁSICA (1 HORA)

- 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 2. INTRODUCCIÓN A KUBERNETES

## Índice

- 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

## Introducción a Kubernetes. El proyecto kubernetes

- Gestiona el despliegue de aplicaciones sobre contenedores, automatizando el despliegue, con énfasis en la escalabilidad y controlando todo el ciclo de vida:
  - Despliega aplicaciones rápidamente
  - Escala las aplicaciones al vuelo
  - Integra cambios sin interrupciones
  - Permite limitar los recursos a utilizar
- Es un proyecto centrado en la puesta en PRODUCCIÓN de contenedores y por tanto indicada su gestión para administradores de sistemas y personal de equipos de operaciones.
- Afecta a los desarrolladores, ya que las aplicaciones deben adaptarse para poder desplegarse en kubernetes

## Introducción a Kubernetes. El proyecto kubernetes

- Para desplegarse en nube: pública, privada o híbrida. Puede instalarse también en servidores físicos, pero no es su entorno natural
- Extensible: Módulos, plugins, adaptable, ...
- Autoreparable



- Proyecto comenzado por Google en 2014, ahora gestionado por Cloud Native Computing Foundation
- Permite gestionar un clúster de nodos en los que desplegar aplicaciones sobre contenedores
- kubernetes es una palabra del griego antiguo que significa “timonel”
- Habitualmente se escribe k8s
- Es muy importante el ecosistema que se está desarrollando alrededor

## Introducción a Kubernetes. El proyecto kubernetes

- Primera versión: 7 junio 2014
- Última versión: 25 marzo 2019 (1.14). 1 cada 3 meses
- Desarrollado en Go
- Licencia Apache 2.0
- <https://github.com/kubernetes/kubernetes>
- Kubernetes está inspirado en Borg, software de Google encargado de la gestión de los servicios de Google
- Más de 2000 personas han contribuido
- 80000 commits en 5 años

## Introducción a Kubernetes. Arquitectura básica

- ~~minion~~ nodo. Componente de un clúster de k8s
- Puede ser una máquina virtual o física
- Tiene los servicios para ejecutar contenedores
- Es gestionado por los componentes master
- Cada nodo tiene:
  - Direcciones: Hostname, IP externa, IP interna
  - Condición: Campo que describe el estado (listo, sin red, sin disco, ...)
  - Capacidad: Describe los recursos disponibles
  - Info: Información general (kernel, sw instalado, versiones)
- Nodo controlador es el que contiene componentes para controlar otros nodos. Se suele denominar master



## Introducción a Kubernetes. Componentes del master

- **kube-apiserver** Gestiona la API de k8s
- **etcd** Almacén clave-valor que guarda la configuración del clúster
- **kube-scheduler** Selecciona el nodo donde ejecutar los pods
- **kube-controller-manager** Ejecuta los controladores
- **docker/rkt/containerd/...** Ejecuta los contenedores del master
- **cloud-controller-manager** Ejecuta los controladores que interactúan con el proveedor de nube:
  - nodos
  - enrutamiento
  - balanceadores
  - volúmenes

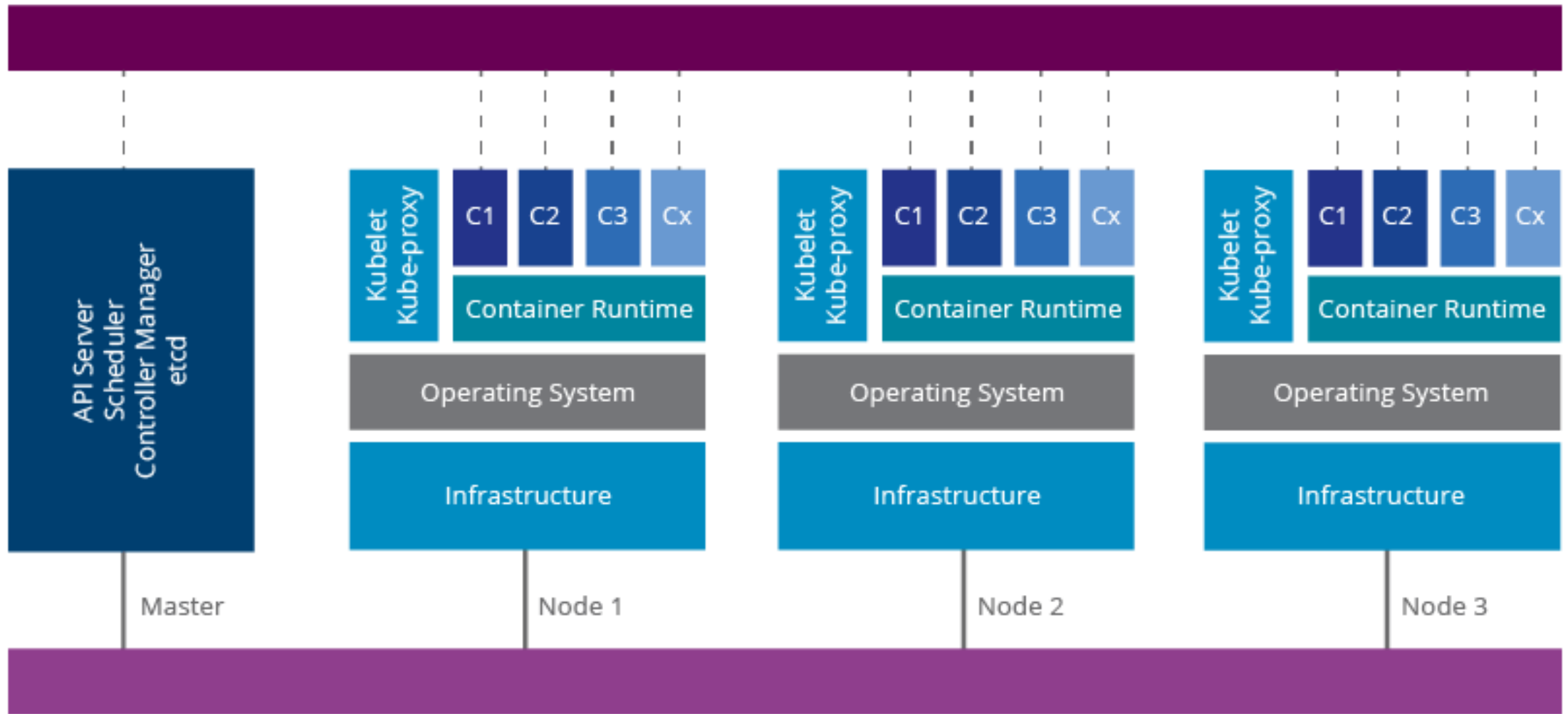
## Introducción a Kubernetes. Complementos (Addons)

- Pods y servicios que proporcionan funcionalidad al clúster
- **Cluster DNS** Proporciona registros DNS para los servicios de k8s (compatible con el DNS de la organización)
- **Web UI** Interfaz web para el manejo de k8s
- **Container Resource Monitoring** Recoge métricas de forma centralizada. Múltiples opciones: prometheus, sysdig
- **Cluster-level Logging** Almacena los logs de los contenedores

## Introducción a Kubernetes. Componentes de los nodos

- **kubelet** Vigila los pods asignados a su nodo
- **kube-proxy** Permite la conexión a través de la red
- **docker/rkt/containerd/...** Ejecuta los contenedores
- **supervisord** Monitoriza y controla kubelet y docker
- **fluentd** Proporciona logs de contenedores a cluster-level logging

## Overlay Network (Flannel/OpenVSwitch/Weave)



## Physical Network

## Introducción a Kubernetes. Alternativas para instalación simple

- Opciones de instalación y uso: <https://kubernetes.io/docs/setup/>
- ¿No se puede instalar desde paquetes de la distribución?

Proyectos propios de k8s	Ecosistema
<a href="#">Minikube</a>	<a href="#">Charmed Distribution of kubernetes (CDK on LXD)</a>
<a href="#">Kubeadm-dind</a>	<a href="#">Docker Desktop</a>
<a href="#">Kubernetes IN Docker</a>	<a href="#">Minishift</a>
	<a href="#">MicroK8s</a>
	<a href="#">k3s</a>

<https://kubernetes.io/docs/setup/>

## Introducción a Kubernetes. Instalación de kubectl

- kubectl es la herramienta principal para gestionar los clusters de k8s
- Desarrollada en go
- No existe como paquete oficial en las distribuciones
- Existe repositorios del proyecto k8s
- Instalación directa de binarios:

<https://kubernetes.io/docs/tasks/tools/install-kubectl/#install-kubectl-binary-with-curl-on-linux>

## Introducción a Kubernetes. Instalación con minikube

- <https://kubernetes.io/docs/tasks/tools/install-minikube/>
- Una vez finalizada la instalación: minikube start
  - Se crea una máquina virtual
  - Se instala kubernetes en dicha máquina
  - Se verifica el estado de los componentes
  - Se configura kubectl para acceder a ese clúster
- Verificación de instalación correcta:

```
kubectl get nodes
```

NAME	STATUS	ROLES	AGE	VERSION
minikube	Ready	master	2m2s	v1.14.2

- Se puede parar minikube con: minikube stop
- Se puede acceder al panel web con: minikube dashboard

## Introducción a Kubernetes. Instalación con minikube

```
kubectl get pods -A
```

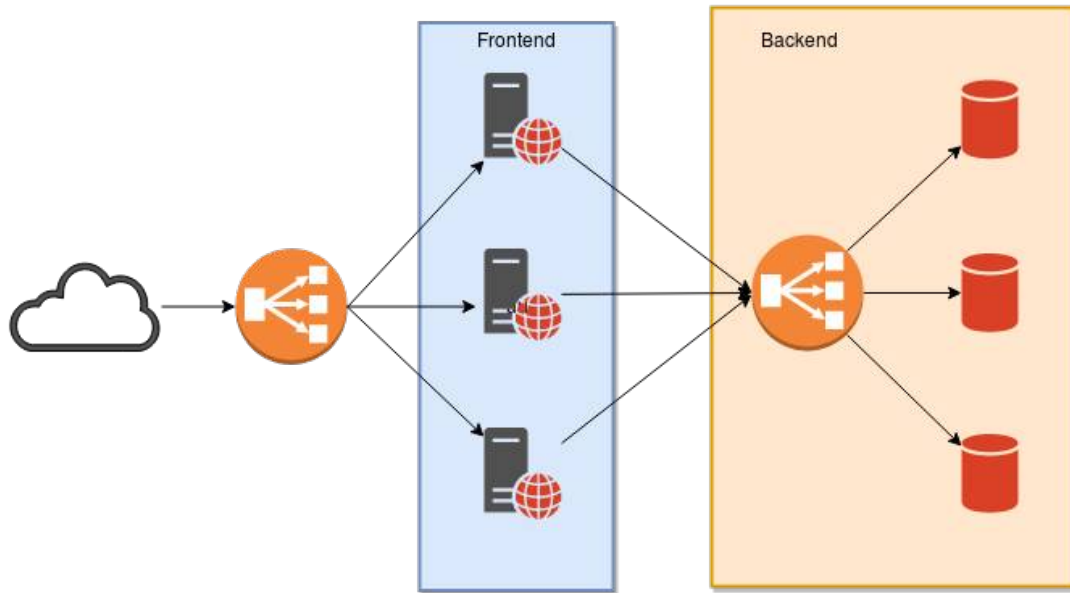
NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	coredns-fb8b8dccf-v5czs	1/1	Running	0	18m
kube-system	coredns-fb8b8dccf-xw7sr	1/1	Running	0	18m
kube-system	etcd-minikube	1/1	Running	0	17m
kube-system	kube-addon-manager-minikube	1/1	Running	0	16m
kube-system	kube-apiserver-minikube	1/1	Running	0	17m
kube-system	kube-controller-manager-minikube	1/1	Running	0	16m
kube-system	kube-proxy-lldbj	1/1	Running	0	18m
kube-system	kube-scheduler-minikube	1/1	Running	0	17m
kube-system	kubernetes-dashboard-79dd6bfc48-c95k4	1/1	Running	0	2m12s
kube-system	storage-provisioner	1/1	Running	0	18m



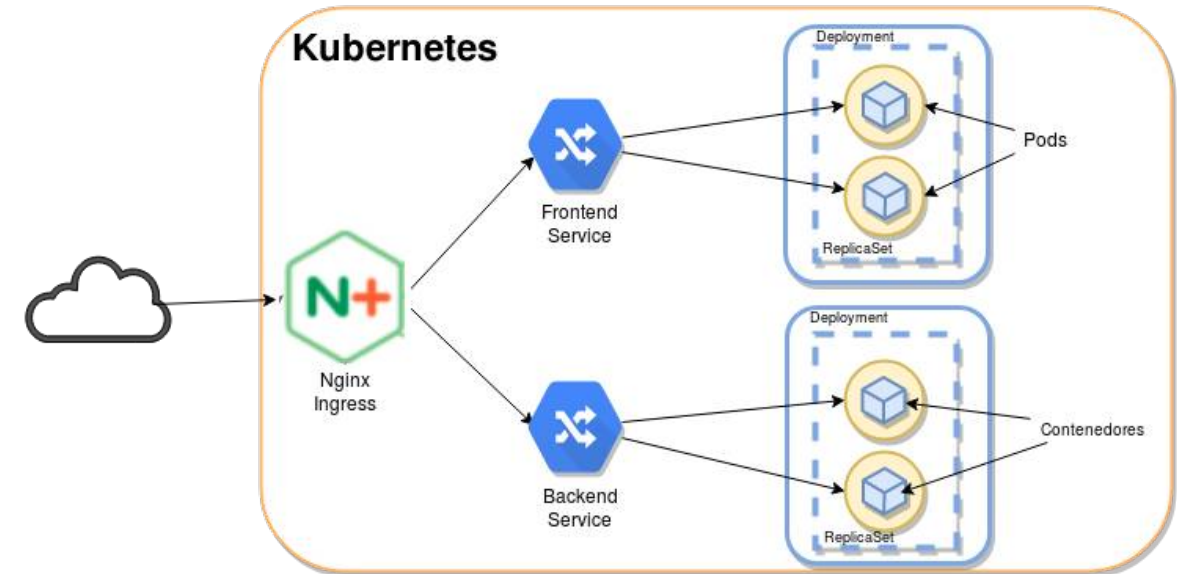
## Introducción a Kubernetes. Configuración y uso de kubectl

- kubectl puede configurarse para gestionar distintos clústers mediante el uso de contextos
- kubectl config
- La configuración se guarda en ~/.kube/config
- Uso de kubectl
- Con “kubectl proxy” puede interactuarse con la API directamente utilizando las credenciales del usuario

## Introducción a Kubernetes. Despliegue de aplicaciones



- Máquinas virtuales
- Balanceadores de carga



- Pods
- Deployments
- Services
- ReplicaSets
- Ingress

## Introducción a Kubernetes. Estrategias de despliegue en k8s

### DEPLOYMENT STRATEGIES

When it comes to production, a ramped or blue/green deployment is usually a good fit, but proper testing of the new platform is necessary.

Blue/green and shadow strategies have more impact on the budget as it requires double resource capacity. If the application lacks in tests or if there is little confidence about the impact/stability of the software, then a canary, a/b testing or shadow release can be used.

If your business requires testing of a new feature amongst a specific pool of users that can be filtered depending on some parameters like geolocation, language, operating system or browser features, then you may want to use the a/b testing technique.



Strategy	ZERO DOWNTIME	REAL TRAFFIC TESTING	TARGETED USERS	CLOUD COST	ROLLBACK DURATION	NEGATIVE IMPACT ON USER	COMPLEXITY OF SETUP
<b>RECREATE</b> version A is terminated then version B is rolled out	✗	✗	✗	■□□	■■■	■■■	□□□
<b>RAMPED</b> version B is slowly rolled out and replacing version A	✓	✗	✗	■□□	■■■	■□□	■□□
<b>BLUE/GREEN</b> version B is released alongside version A, then the traffic is switched to version B	✓	✗	✗	■■■	□□□	■■■	■■■
<b>CANARY</b> version B is released to a subset of users, then proceed to a full rollout	✓	✓	✗	■□□	■□□	■□□	■■■
<b>A/B TESTING</b> version B is released to a subset of users under specific condition	✓	✓	✓	■□□	■□□	■□□	■■■
<b>SHADOW</b> version B receives real world traffic alongside version A and doesn't impact the response	✓	✓	✗	■■■	□□□	□□□	■■■