Introducción a Kubernetes

KEEPCODING
Tech School

¿Quién soy?

- Manuel Cañete
- Senior SRE / Platform Engineer
- Ingeniero Informático
- +6 años trabajando con Kubernetes

Ahora, vuestro turno

- ¿Cómo te llamas?
- ¿A qué te dedicas?
- ¿Por qué estás estudiando este Bootcamp?
- ¿Qué esperas del módulo de Kubernetes?

Qué podemos esperar sobre el curso?

- Aprender las bases de Kubernetes.
- Ser capaces de desarrollar aplicaciones con Kubernetes.
- Ser capaces de operar un cluster.
- Aprender a resolver errores.
- … en definitiva, estar listos para empezar a trabajar con Kubernetes.

Evaluación

- Práctica final
  - ¿Cuándo debemos entregarla?
    - **29 de Septiembre a las 23:30 CET.**
  - ¿Cómo lo haremos?
    - **Mediante el formulario oficial.**
  - ¿Qué ocurre si el resultado es “No Apto”?
    - Habrá una segunda oportunidad hasta el **17 de Noviembre.**
  - ¿Y si no puedo entregarla en la primera convocatoria?
    - Tendréis **SÓLO** la segunda oportunidad para entregarlo.

Anteriormente, en Contenedores…

- Recapitulación docker:
  - Ventajas de Docker
    - Agilidad
    - Portabilidad
    - Comunidad

"Build, Ship and Run any App, Anywhere"

Anteriormente, en Contenedores…

- Recapitulación docker-compose:
  - Permite orquestar varios contenedores localmente
  - Permite la comunicación sencilla entre contenedores (networking, dns)
  - Ayuda con el storage a través de volúmenes
  - Muy útil para desarrolladores, integradores y testers para realizar pruebas en local.
  - Limitaciones (entre otras):
    - No es escalable (funciona a nivel de single-host)
    - No es resiliente ni proporciona alta disponibilidad (HA).

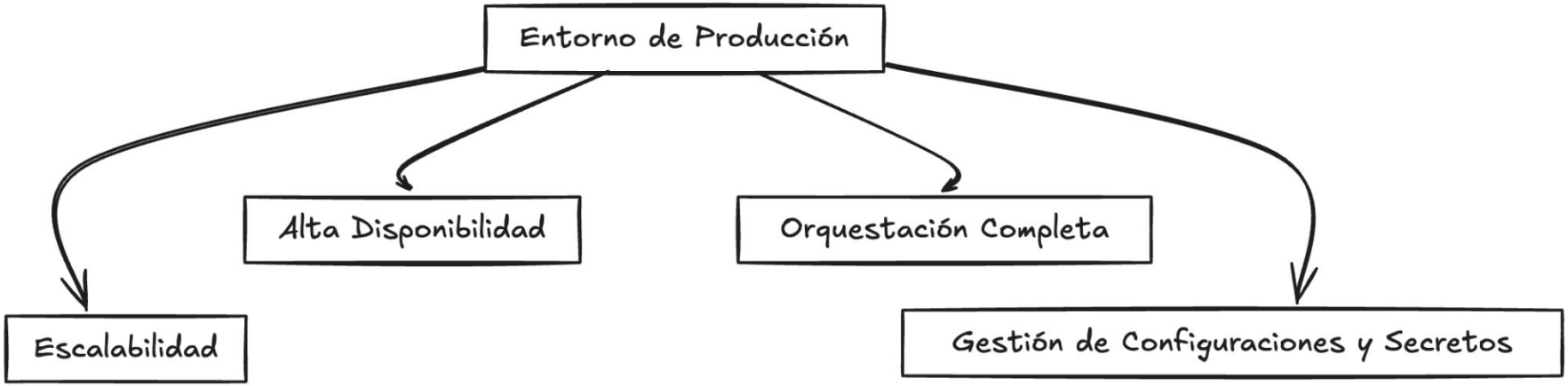
- ¿Podemos desplegar nuestras aplicaciones en producción así?
  - Necesitamos algo que nos proporcione escalabilidad y resiliencia. Una plataforma multi-nodo (cluster) que permita orquestar contenedores y proporcione mecanismos de control y alta disponibilidad.

Introducción a Kubernetes / Agenda

- ¿Qué necesitamos? ¿Qué buscamos?

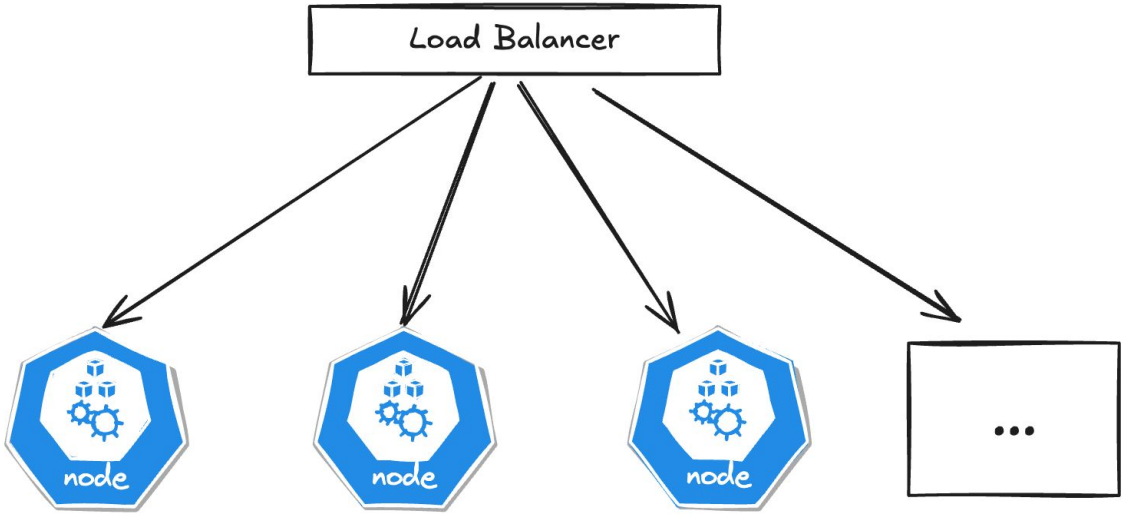
Introducción a Kubernetes / Agenda

- ¿Qué necesitamos? ¿Qué buscamos?



Introducción a Kubernetes / Agenda

- ¿Qué necesitamos? ¿Qué buscamos?



Introducción a Kubernetes / Agenda

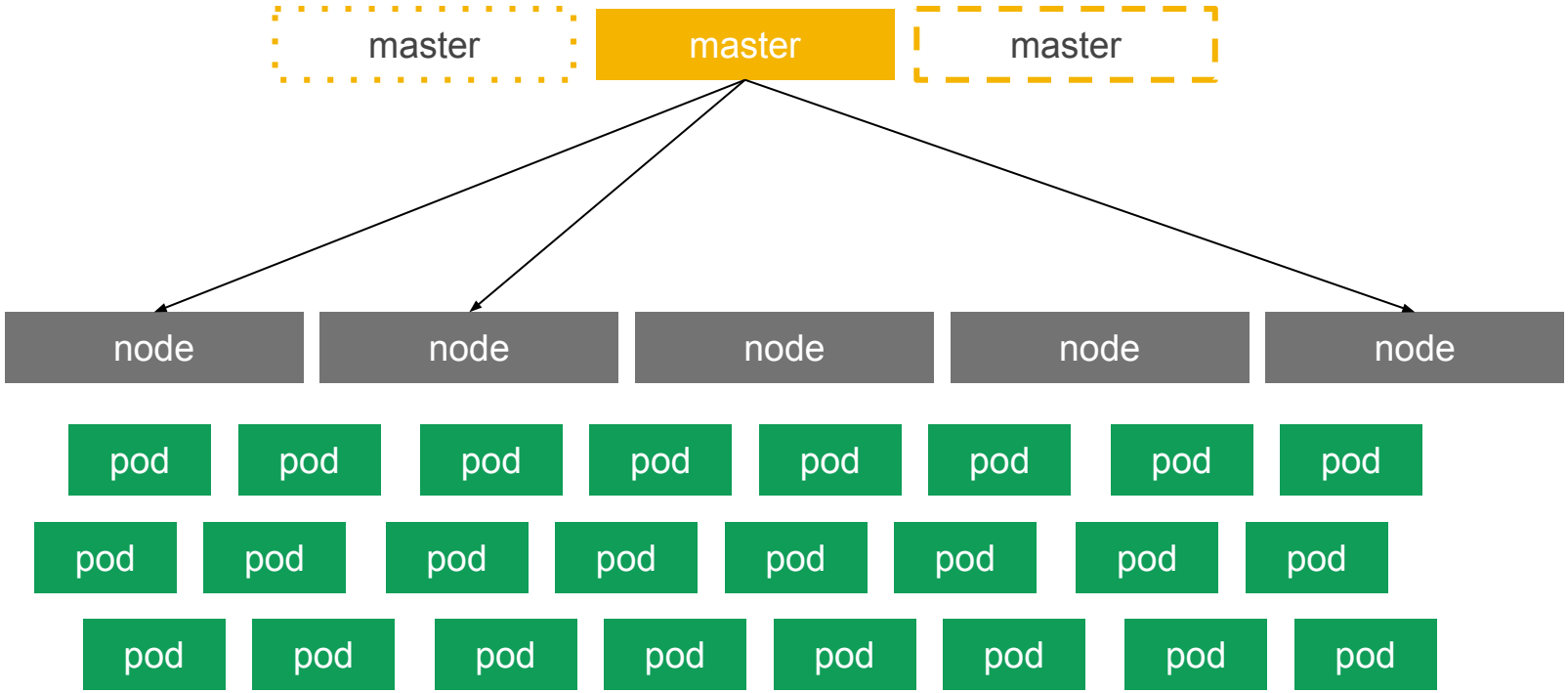
- ¿Qué es Kubernetes?
  - Definiciones oficiales:
    - A distributed computing system/platform that allows running containers.
    - Container-based platform for deploying, scaling and running applications.
    - A portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
  - Definiciones coloquiales:
    - "Orquestador de contenedores", "Plataforma open-source para manejar cargas de trabajo en forma de contenedores", "Cluster / Sistema distribuido diseñado para el despliegue de aplicaciones en contenedores".

Introducción a Kubernetes / Agenda

- ¿Qué es Kubernetes?
  - Palabras clave
    - PLATAFORMA → ¿Qué entendemos por "plataforma"?
    - WORKLOADS / Cargas de trabajo
    - COMPUTACIÓN DISTRIBUIDA / CLUSTER (múltiples nodos trabajando como una entidad lógica)
    - DECLARATIVE CONFIGURATION (yml)
    - OPEN-SOURCE

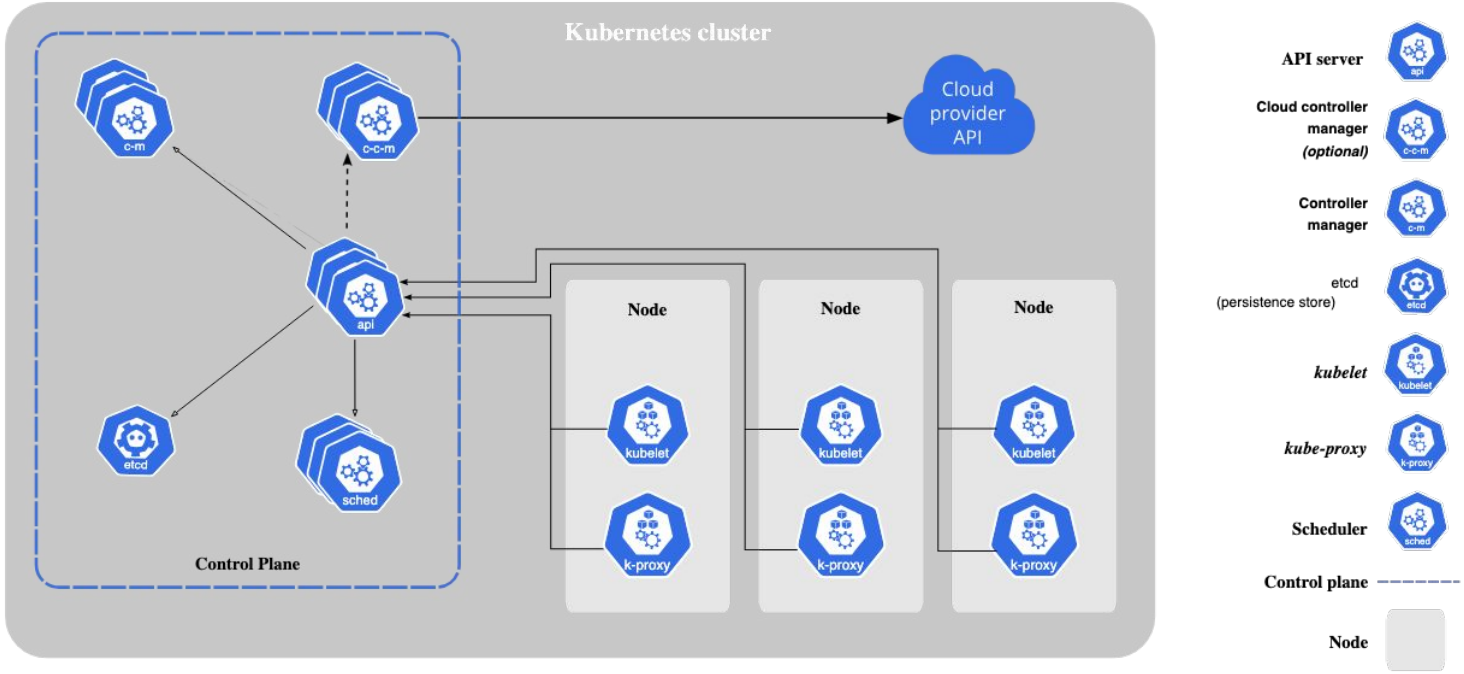
- Nota: ¿Qué es una "Plataforma"? Sistema con un propósito o utilidad concreto que proporciona alto nivel de abstracción sobre lo que hay detrás (ejemplo: plataformas cloud).

Introducción a Kubernetes / Agenda



Introducción a Kubernetes / Agenda

- Componentes Kubernetes

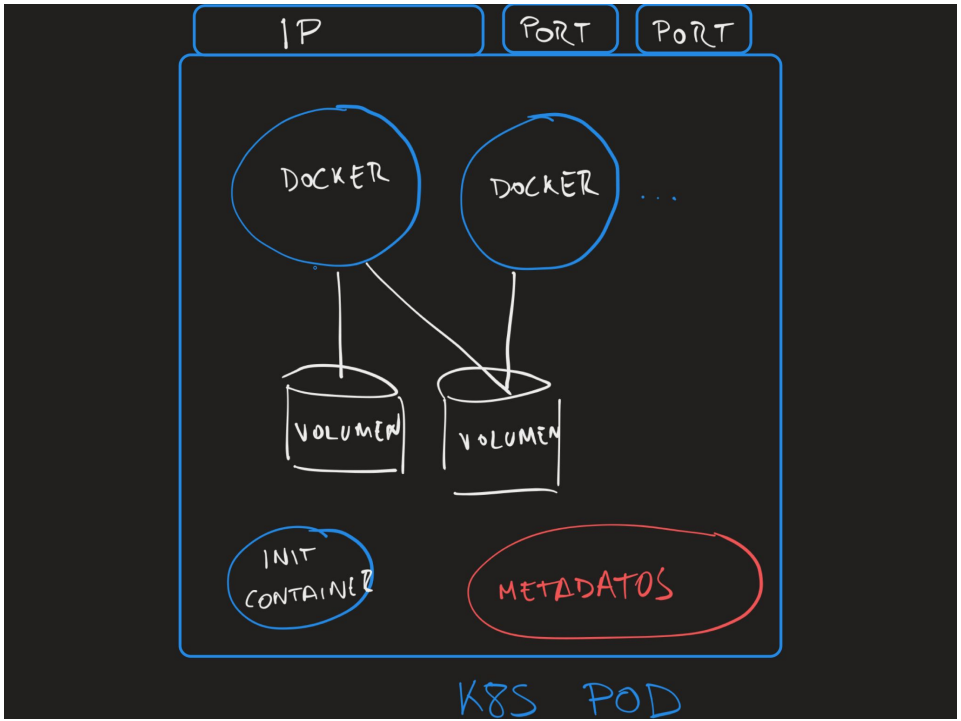


Introducción a Kubernetes / Agenda

- ¿Qué son esos PODs que aparecen en todos los diagramas?
  - Más o menos son los contenedores (dockers) que se ejecutan, pero lo veremos en detalle más adelante (realmente un POD puede estar formado por varios contenedores).
  - A nivel de diseño es el bloque básico de construcción en Kubernetes. Nuestro objetivo es crear, diseñar y planificar pods, ya que son las entidades que finalmente estarán en ejecución.
  - Podemos usar Docker para desplegar los contenedores, aunque también se soportan otros runtimes (indiferente para el desarrollador).

Introducción a Kubernetes / Agenda

- POD Overview



Introducción a Kubernetes / Agenda

- ¿Cómo se maneja?
  - A través de un API REST
  - **kubectl** → Aplicación cliente (comando) que nos va a permitir interactuar con distintos clusters.
  - Mediante ficheros "**yaml**" que definen los recursos deseados (desired state).

- Muchas similitudes con plataformas cloud:
  - Computing (VMs) → Pods
  - Disks, Volumes → Volumes
  - Load Balancers → Services

- Permite integración con servicios cloud (storage, load balancers)

Introducción a Kubernetes / Agenda

- ¿Cómo se instala?
  - As a service (GKE, AKS, …)
  - En nuestras máquinas (físicas o virtuales)
  - En nuestro PC (minikube, …)

Run Anywhere!!!

Introducción a Kubernetes / Agenda

- Tipos de usuario y áreas de conocimiento
  - **Administrador** de Kubernetes (CKA)
    - Instalación
    - Mantenimiento y Configuración de componentes internos
    - Arquitectura / Networking / Storage
    - Seguridad: RBAC, NetworkPolicies
    - Autoescalado (Cluster Autoscaling)
    - Monitorización
  - **Desarrolladores** / DevOps Engineers (CKAD)
    - Definir y diseñar cargas de trabajo (contenedores / pods)
    - Utilizar y sacar provecho de todas las características y funcionalidades de Kubernetes
    - Exponer estas cargas de trabajo al exterior o a otros pods.

Introducción a Kubernetes / Agenda

- Certificaciones oficiales (https://kubernetes.io/training/)
  - CKAD (Certified Kubernetes Application Developer)
  - CKA (Certified Kubernetes Administrator)
  - CKS (Certified Kubernetes Security Specialist)
  - KCNA (Kubernetes and Cloud Native Associate)
  - KCNSA (Kubernetes and Cloud Native Security Associate)

Introducción a Kubernetes / Agenda

- **Objects**: Recursos / Objetos a nuestra disposición:
  - Namespaces
  - Pods
  - Deployments / StatefulSets / DaemonSets / Jobs / CronJobs
  - ConfigMaps / Secrets
  - Volumes / PersistentVolumes / PersistentVolumeClaims

Introducción a Kubernetes / Agenda

- **Objects**: Cómo exponer el acceso a los pods
  - Servicios
  - Ingress & Ingress Controllers
  - Service Mesh

Introducción a Kubernetes / Agenda

- **Objects**: Otras funcionalidades
  - Controlar en qué nodos se ejecutarán los pods (Taints, Tolerations & AffinityRules)
  - Autoescalado de pods (Horizontal Pod Autoscaling, HPA).
  - Autoescalado de cluster (Cluster Autoscaling)
  - Extras (Limits, Policies, etc).

Introducción a Kubernetes / Agenda

- Kubernetes Observability: Logs y Métricas (Monitorización en Kubernetes)
  - Prometheus + Grafana
  - Elastic Stack
- Resolución de problemas / Troubleshooting

Introducción a Kubernetes / Agenda

- ¿Qué hacemos con tanto manifiesto YAML?
  - Helm Charts
  - Operadores

Introducción a Kubernetes / Agenda

- Objetivos del módulo de Kubernetes
  - Familiarizarnos con Kubernetes y `kubectl`
  - Sentirnos cómodos analizando entornos Kubernetes.
  - Conocer las funcionalidades y capacidades que ofrece en cuanto a orquestación de contenedores.
  - Diseñar cargas de trabajo para Kubernetes (Deployments, StatefulSets, DaemonSets, etc).
  - Exponer las aplicaciones tanto dentro como fuera del cluster
  - Uso de Helm y creación de charts (grupo de plantillas de manifiestos YAML).
  - Monitorización en Kubernetes

Características de Kubernetes

KEEPCODING
Tech School

¿Qué buscamos con Kubernetes?

- Resiliencia
- Alta disponibilidad
- Escalabilidad
- Seguridad
- Monitorización
- Logs centralizados
- Persistencia de datos
- Gestión de secretos
- Disaster recovery
- Integración con CI/CD
- Gestión de costes
- Integración con Cloud + Cloud Agnostic
- Multitenant

Integración con Cloud

Kubernetes es capaz de comunicarse con nuestra cloud y utilizar sus servicios.

Poder aprovechar todos los recursos del cloud y que estén integrados con nuestra plataforma, sin necesidad de conocer los detalles.

Puede funcionar en cualquier nube pública (GCP, AWS, Azure, Hetzner, …) o incluso en entornos on-premise.

Es un componente perfecto para entornos híbridos.

Run anywhere!

Resiliencia

Resiliencia de un sistema:

- *Capacidad que tiene para recuperarse automáticamente frente a comportamientos inesperados.*

"Nuestra aplicación y nuestra plataforma pueda recuperarse de disrupciones de servicio automáticamente"

- Service restart
- VM restart

Alta disponibilidad

“Asegurar grado absoluto de continuidad operacional”

- Redundancia
- Arquitecturas Complejas

Escalabilidad

“No perder calidad en servicios ofrecidos frente a crecimiento en la demanda”

- Escalabilidad vertical
- Escalabilidad horizontal

Persistencia de datos

Aplicaciones que guardan información (BBDD).
Trabajando con Docker utilizamos volúmenes.

- Rendimiento
- Backups

Gestión de secretos

Por seguridad es necesario gestionar ficheros de configuración y claves en nuestros contenedores de forma dinámica y segura.

Disaster recovery

Tenemos que ser capaces de replicar la infraestructura rápida y eficazmente.

La naturaleza de Kubernetes permitirá esto de forma más o menos sencilla.

Integración con CI/CD

Poder automatizar todos los flujos de integración y despliegue.

Capacidad para disponer de varios entornos.

Facilidad de creación de entornos efímeros.

Multitenant

Aislamiento de entornos.

Gestión de varios usuarios y proyectos.

RBAC para control de accesos.

Seguridad

Políticas de control de acceso (RBAC).

Seguridad a nivel de contenedor.

Redes seguras y cifradas.

Monitorización

Observabilidad completa

Integración con Prometheus

Alertas y métricas

Logs centralizados

Centralización de logs.

Integración con ELK stack.

Gestión de costes

Optimización de recursos.

Escalado eficiente.

Reducción de infraestructuras sobredimensionadas.

KEEPCODING

Tech School

Madrid | Barcelona | Bogotá