



Kubernetes

Tema 11

Grupo 13

Tomás Álvarez Escalante (60127)



Alejo Francisco Caeiro (60692)

Lucas Agustín Ferreiro (61595)

Román Gómez Kiss (61003)



¿Qué es Kubernetes?

- Plataforma de código abierto
 - Orquestador de contenedores
 - Automatiza la implementación, escalado y operaciones de aplicaciones en contenedores
 - Alta disponibilidad y escalabilidad
- 
- 



Funcionalidades de Kubernetes

**Rollout &
Rollback**

**Service
Discovery**

**Storage
Orchestration**

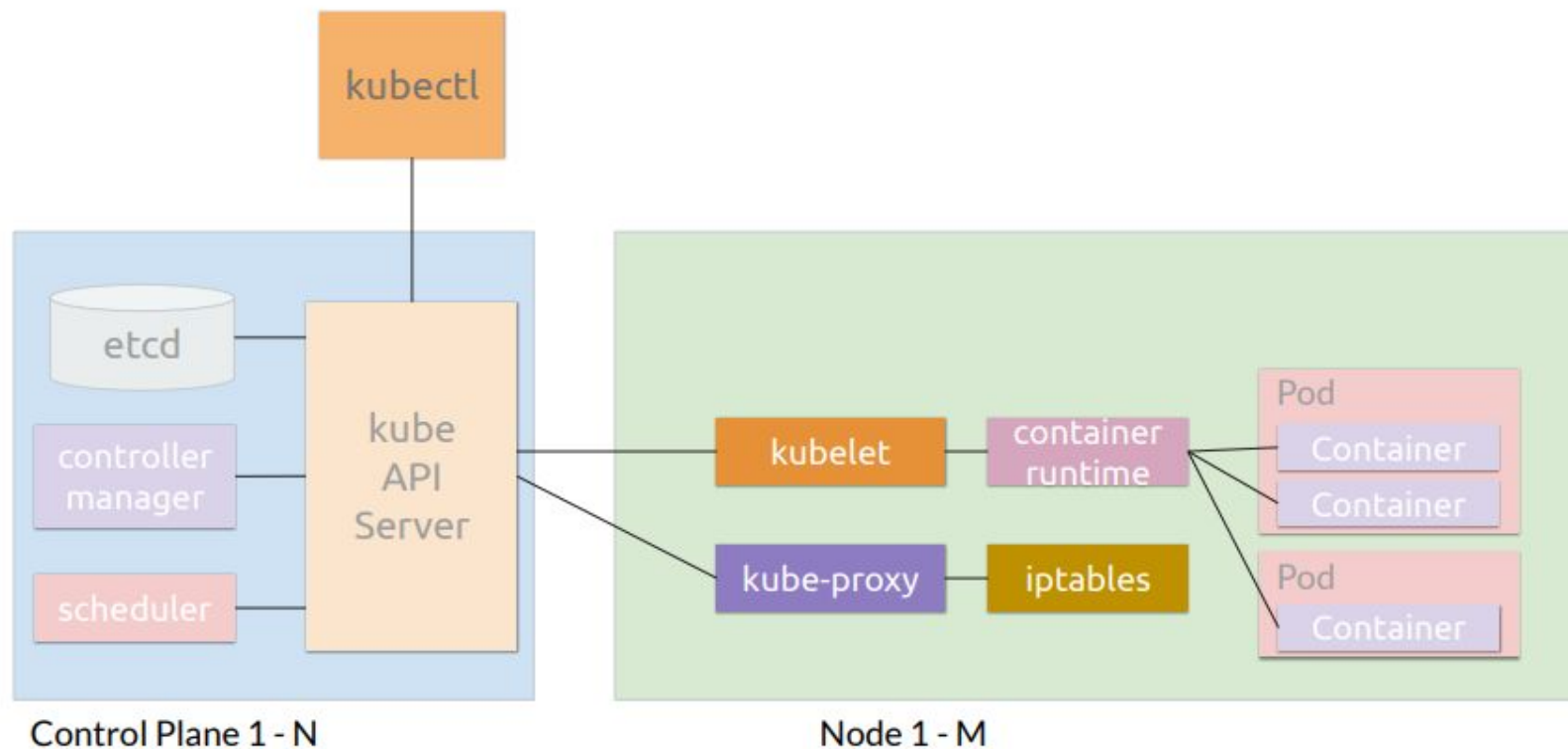
**Horizontal
Scaling**

Self Healing

**Service Load
Balancing**



Arquitectura de Kubernetes



Consigna del proyecto

- Crear un cluster de Kubernetes de un Master y al menos dos slave, que exponga una API en un puerto genérico (distinto a 80).
- Implementar una base de datos local en un servidor (por fuera del cluster) y exponer un servicio que redireccione el tráfico del cluster al servidor.
- Deployar un web server (nginx o Apache HTTPD escuchando en el 80) y hacer un proxy reverso a la API.
- Mostrar dos versiones de API distintas conviviendo.
- Integrar los servicios de Istio y Kiali al cluster.



Kind (Kubernetes IN Docker)

- Herramienta diseñada para ejecutar clusters de Kubernetes locales utilizando contenedores Docker como nodos.
- Ideal para pruebas y desarrollo sin necesidad de configurar una infraestructura compleja.
- Permite especificar la cantidad de nodos worker y master.

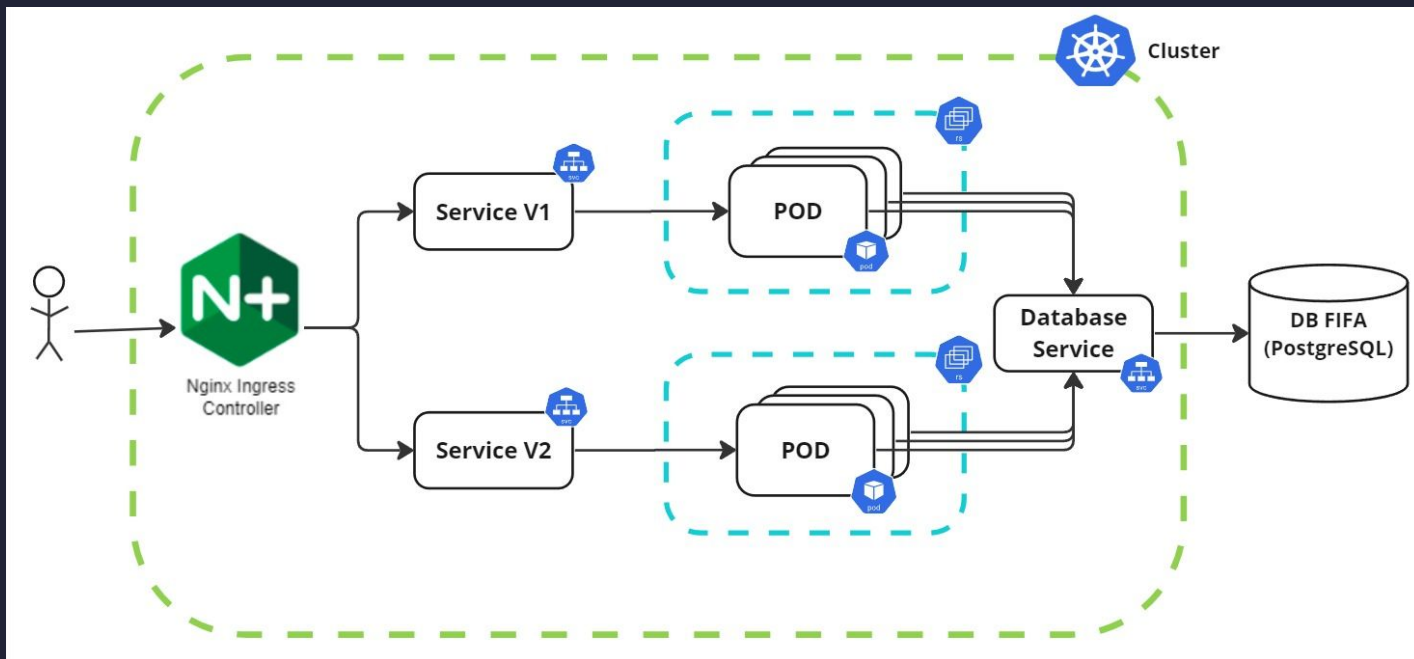


FIFA Players API

- Dataset: [Football Players Data](#)
- Dockerfile -> Imagen que levanta la API en el puerto 8080
- Desarrolladas con Flask
- 2 versiones de la API
- Headers con información del nodo y POD
- GET /players?name={name}&size=n}
- GET /players/{id}




Arquitectura del proyecto

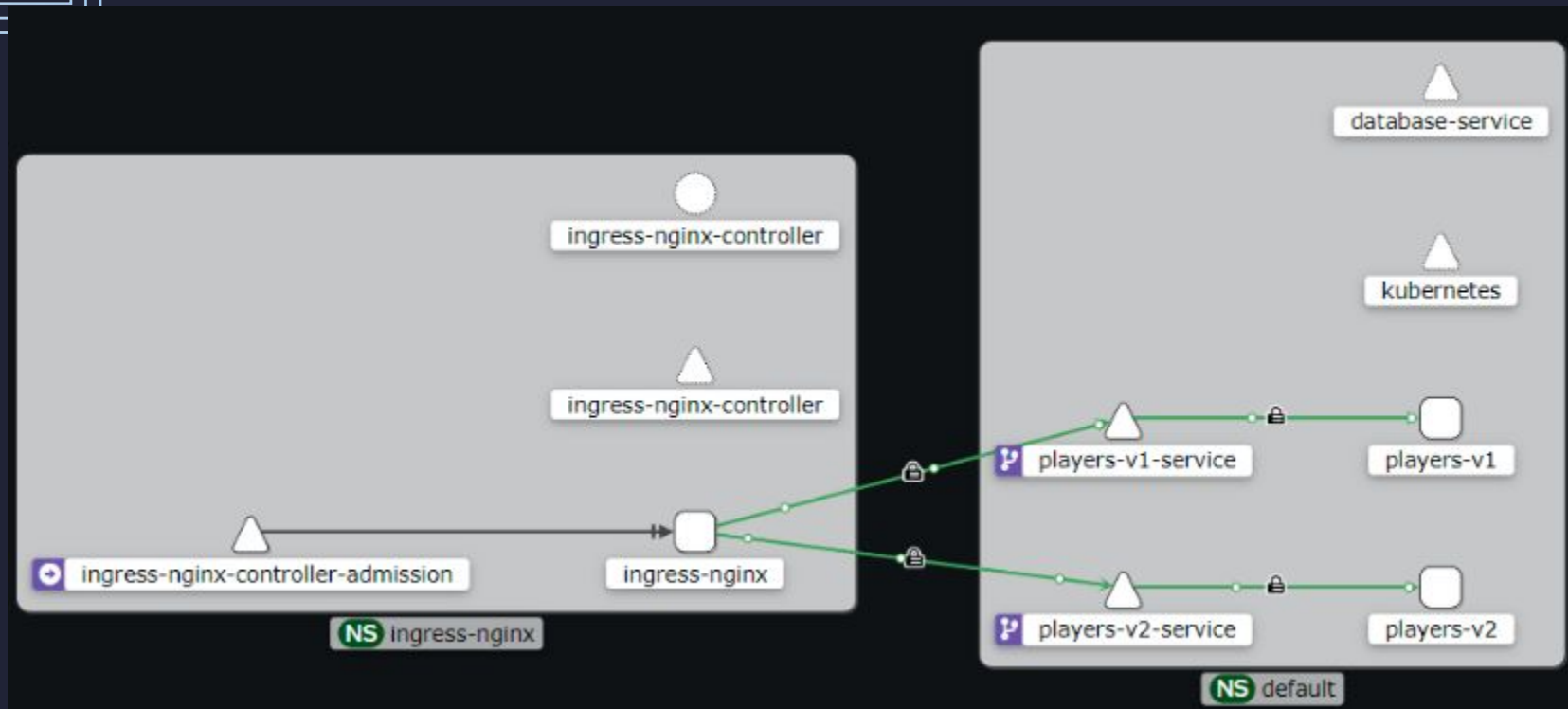




Especificaciones de la arquitectura del proyecto

- Deployment que especifica la imagen del contenedor y cantidad de réplicas.
 - Puerto del contenedor expuesto en el 8080
 - Replica Set con 3 réplicas de la API ejecutándose.
 - Servicios ClusterIP para exponer el acceso centralizado a los PODs.
 - Puerto donde escucha el servicio en el 9090.
 - Puerto a donde se redirige el tráfico (POD) en el 8080.
 - Ingress → Servicio tipo LoadBalancer que usa Ingress Controller de Nginx
 - Puertos: 80 - 443.
 - Host: api.players.com.
 - Path: /v1 → Redirige al puerto 9090 del servicio v1
 - Path: /v2 → Redirige al puerto 9090 del servicio v2
- 

Visualización en Kiali





Demostración práctica

[Link al repositorio](#)





¡Gracias!

