

2023

# Docker + Kubernetes 101



# Sponsors

**NTT DATA** **encamina**  
PIENSA EN COLORES

**plain  
concepts**

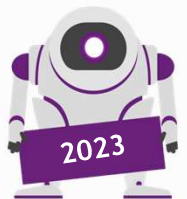
**intelequia**

**Verne**  
TECHNOLOGY GROUP

**TOKIOTA**

**ilitia**

#netcoreconf





¿Quién soy?

Leonardo Micheloni

**TOKIOTA**

@leomicheloni



Tricampeón del mundo



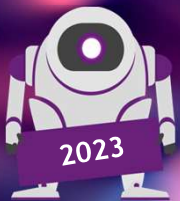
#netcoreconf



# Agenda

#netcoreconf

- 01 Introducción
- 02 Docker
- 03 Docker compose
- 04 Kubernetes
- 05 Preguntas

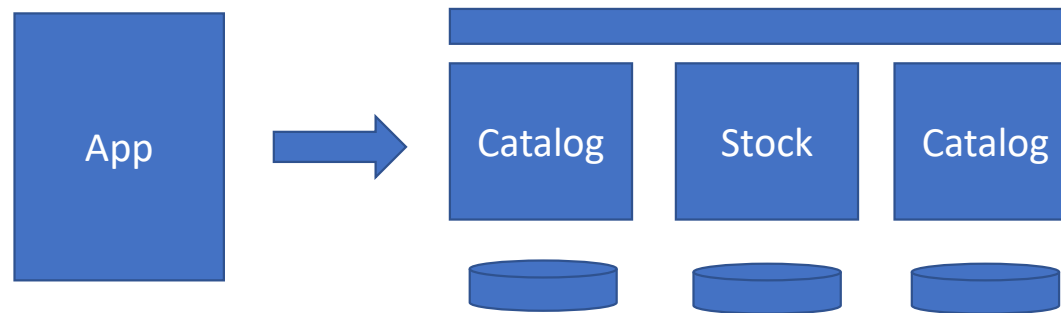


# Disclaimer

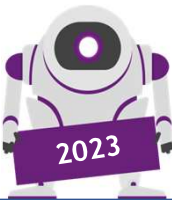
- No soy experto
- No vamos a profundizar
- Mi experiencia
- Nunca sigan un consejo mío



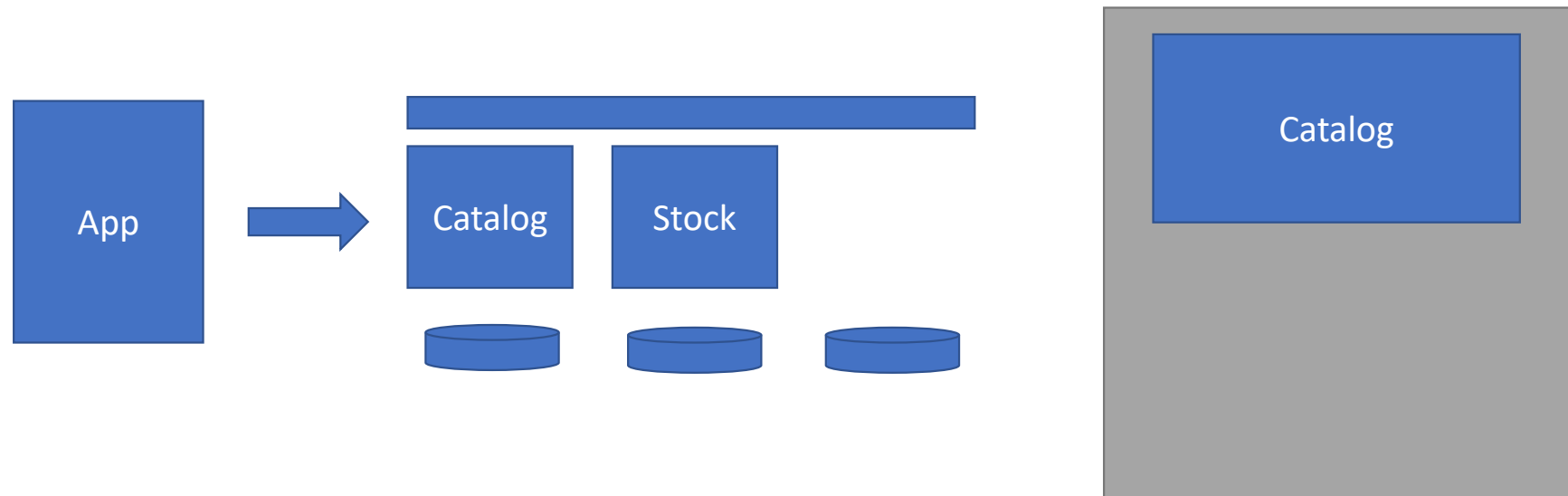
# Escenario



Separamos nuestra aplicación en “microservicios”



# Escenario

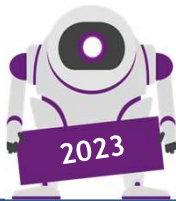
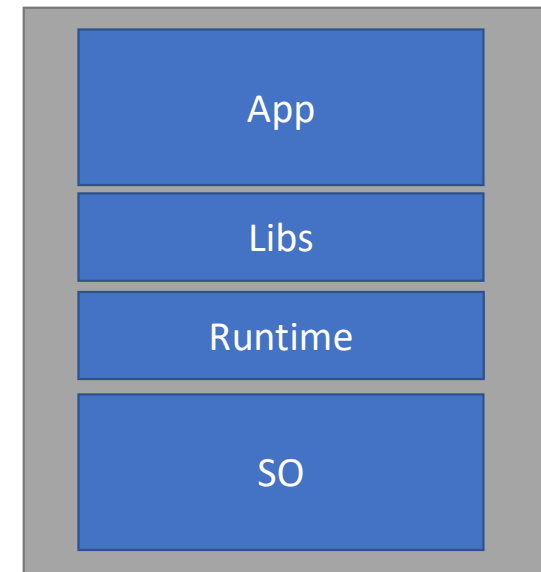


En lugar de que se ejecuten en el mismo entorno  
Creamos uno para cada app con runtime, código SO, etc.  
Cerrado al exterior



# Container

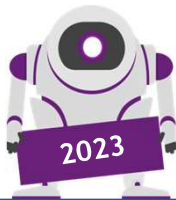
- Entorno aislado
- Propio SO
- No “ve” nada extra
- Lib / dependencias
- Runtime
- Cerrado al exterior
- Aprovecha ciertas features existentes en Linux





# Container

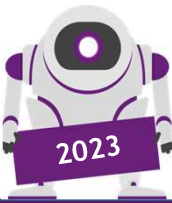
- OCI
  - <https://opencontainers.org/>
  - Container
  - Image
  - Distribution



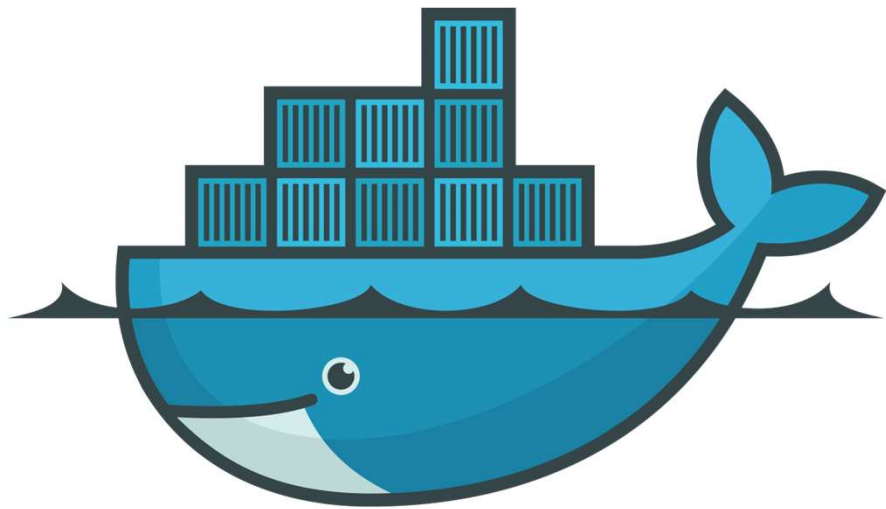
# Container

---

- Los container se crean a partir de imágenes
- Las imágenes se almacenan en repositorios
  - Públicos
  - Privados
  - Local
- Docker Hub



Docker



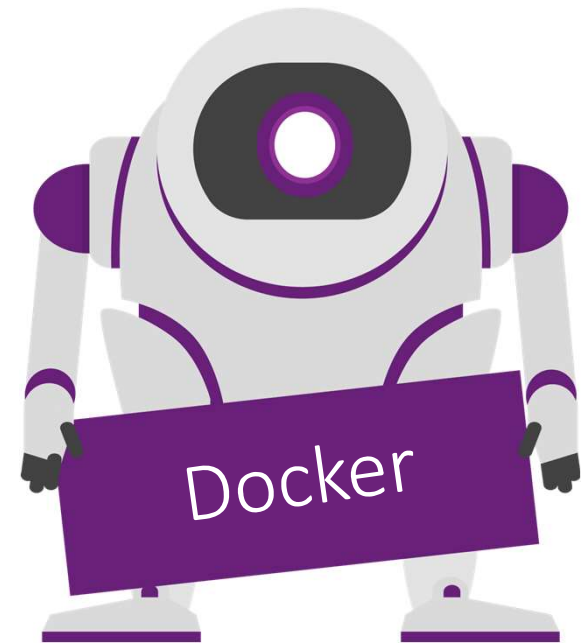
docker

#netcoreconf



# Docker

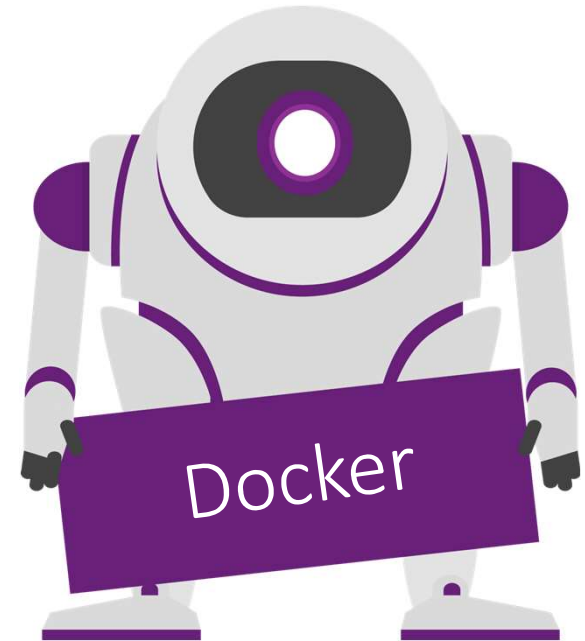
- Un sistema de administración de containers
- Permite crear, ejecutar, borrar, etc.
  - Containers
  - Images
- Existen otros
- Docker != microservicios
- Funciona en Windows / Linux / MacOS
- Contenedores Linux y Windows





# ¿Qué puedo hacer con Docker?

- Gestionar un sistema de microservicios
- Replicar entornos propios en local
- Hacer tests que sean repetitivos
- Probar en local
- Utilizar dependencias de terceros
- Facilitar los despliegues



DEMO

#netcoreconf



# Tags

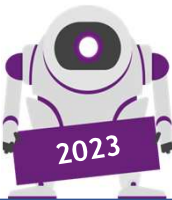
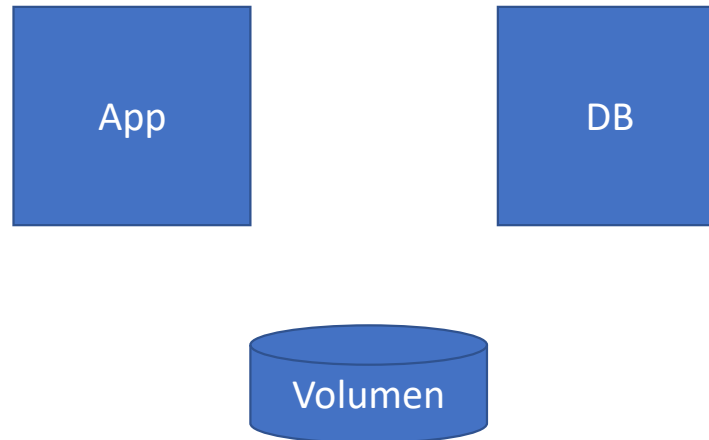
- Las imágenes se versionan con tags
- Los tags pueden significar muchas cosas
- No es recomendable utilizar “latest”
- Verificar siempre qué significa cada tag

## Supported tags and respective Dockerfile links

- `3.11.9`, `3.11`, `3`, `latest`
- `3.11.9-management`, `3.11-management`, `3-management`, `management`
- `3.11.9-alpine`, `3.11-alpine`, `3-alpine`, `alpine`
- `3.11.9-management-alpine`, `3.11-management-alpine`, `3-management-alpine`, `management-alpine`
- `3.10.18`, `3.10`
- `3.10.18-management`, `3.10-management`
- `3.10.18-alpine`, `3.10-alpine`
- `3.10.18-management-alpine`, `3.10-management-alpine`
- `3.9.28`, `3.9`
- `3.9.28-management`, `3.9-management`
- `3.9.28-alpine`, `3.9-alpine`
- `3.9.28-management-alpine`, `3.9-management-alpine`



# ¿Qué pasa con las dependencias?





# Docker compose

- Servicios
- Red interna (dns)
- Volúmenes
- Puertos
- Variables de entorno
- Dependencias
- Políticas de reinicio

```
version: '3.1'

services:
  wordpress:
    image: wordpress
    restart: always
    ports:
      - 8080:80
    environment:
      WORDPRESS_DB_HOST: db
      WORDPRESS_DB_USER: exampleuser
      WORDPRESS_DB_PASSWORD: examplepass
      WORDPRESS_DB_NAME: exampledb
    volumes:
      - wordpress:/var/www/html

  db:
    image: mysql:5.7
    restart: always
    environment:
      MYSQL_DATABASE: exampledb
      MYSQL_USER: exampleuser
      MYSQL_PASSWORD: examplepass
      MYSQL_RANDOM_ROOT_PASSWORD: '1'
    volumes:
      - db:/var/lib/mysql

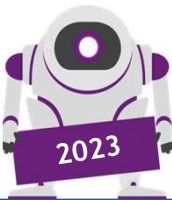
volumes:
  wordpress:
  db:
```



# Volumes

---

- Tienen su propio ciclo de vida
- Se pueden “attachar” a un directorio de cualquier container



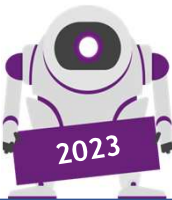
DEMO

#netcoreconf



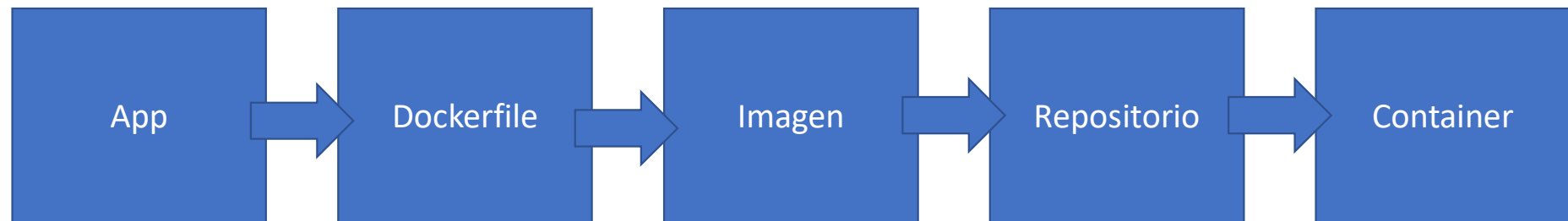
# Cómo meto mi app en un container

- Los container se crean a partir de imágenes
- Se pueden crear múltiples containers a partir de una imagen
- Las imágenes se crean utilizando **Dockerfile**
- Las imágenes se guardan en repositorios
- Existen imágenes de terceros que puedo usar





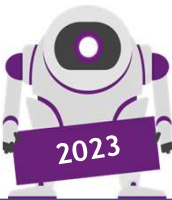
# ¿Cómo meto mi app en un container?



# Dockerfile

- Utiliza una imagen base
- Permite
  - Copiar
  - Ejecutar comandos

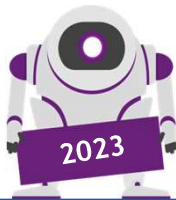
```
1 FROM alpine
2 CMD ["ping", "www.yahoo.com"]
```



# Dockerfile

```
FROM mcr.microsoft.com/dotnet/aspnet:6.0
COPY publish/ App/
WORKDIR /App
ENTRYPOINT ["dotnet", "WebApp.dll"]
```

```
1 ARG REPO=mcr.microsoft.com/dotnet/runtime
2 FROM $REPO:6.0.14-alpine3.16-amd64
3
4 # .NET globalization APIs will use invariant mode by default because DOTNET_SYSTEM_GLOBALIZATION_INVARIANT=true
5 # by the base runtime-deps image. See https://aka.ms/dotnet-globalization-alpine-containers for more information
6
7 # ASP.NET Core version
8 ENV ASPNET_VERSION=6.0.14
9
10 # Install ASP.NET Core
11 RUN wget -O aspnetcore.tar.gz https://dotnetcli.azureedge.net/dotnet/aspnetcore/Runtime/$ASPNET_VERSION/aspnetcore.tar.gz \
12     && aspnetcore_sha512='35fcfb6ba2048f1f942f667fffb333b0983347b8adf7b7d3db5152a222931f8b86fe5c62cb2fd45bf95a89' \
13     && echo "$aspnetcore_sha512 aspnetcore.tar.gz" | sha512sum -c - \
14     && tar -oxzf aspnetcore.tar.gz -C /usr/share/dotnet ./shared/Microsoft.AspNetCore.App \
15     && rm aspnetcore.tar.gz
```



#netcoreconf

DEMO

#netcoreconf





# Docker

---

- Es un software para gestionar contenedores OCI
- Podemos crear sistemas de microservicios
- Podemos ejecutar dependencias en nuestro entorno sin instalar
- El entorno es repetible

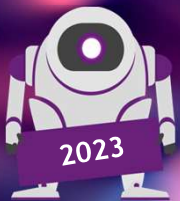


Kubernetes



kubernetes

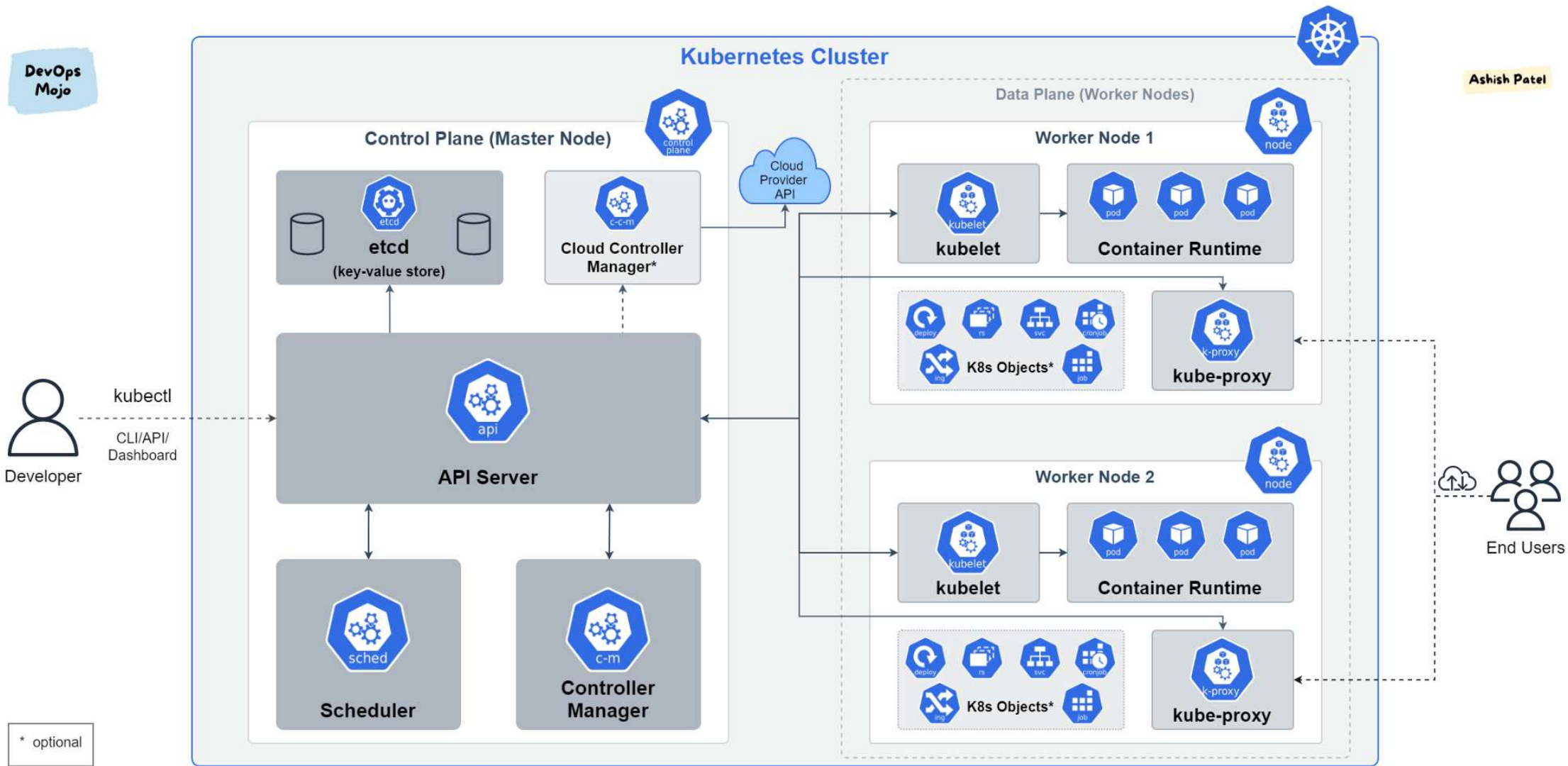
#netcoreconf



# Kubernetes

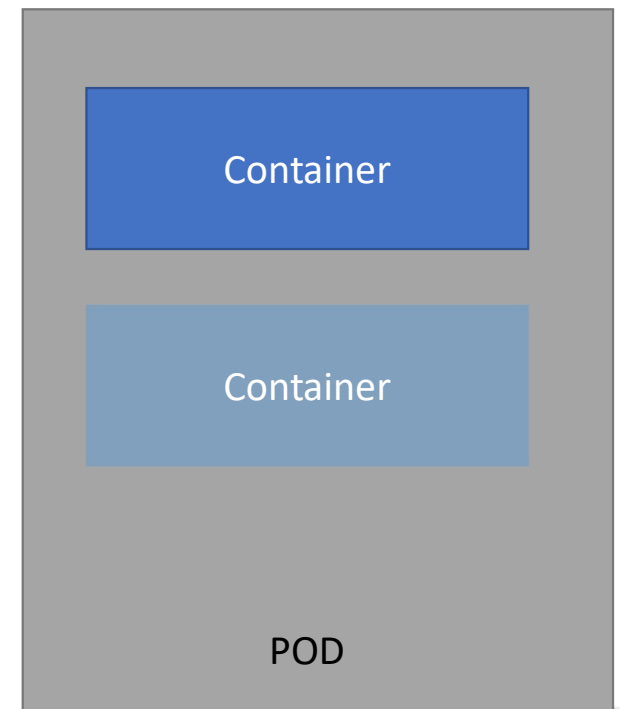
- Gestiona sistemas basados en contenedores
  - Salud de los contenedores
  - Escalabilidad
  - Disponibilidad
  - Versionado
  - Volúmenes
  - Configuración
- Orquestador
- Existen otros (Openshift, Docker Swarm, Nomad)

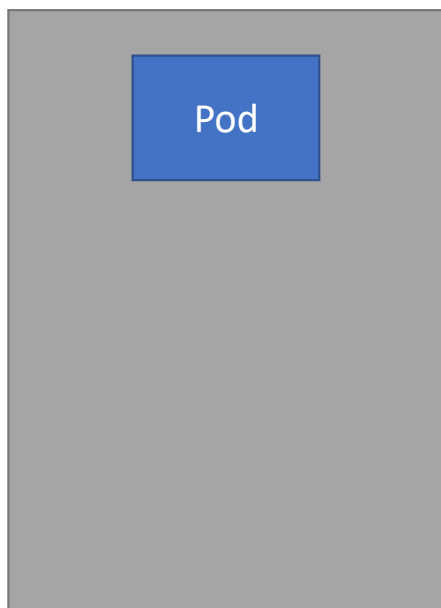




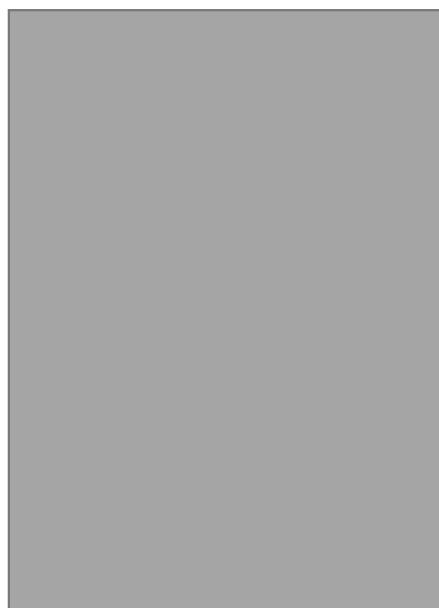
# POD

- Es la unidad mínima de ejecución de K8s
- Contiene uno o más contenedores
- Se configuran utilizando yaml
- Se pueden definir variables
- Volumes / mapeos
- Recursos
- No es accesible desde el exterior

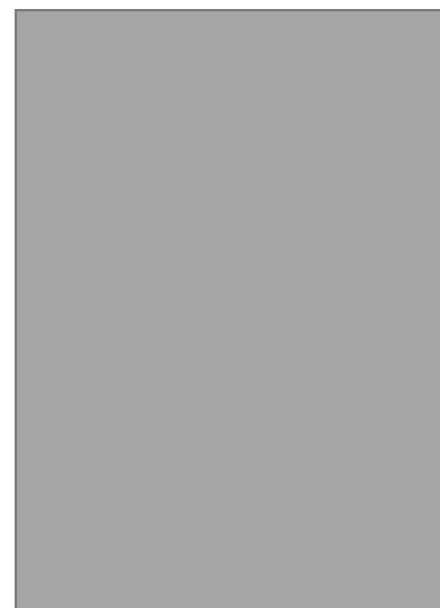




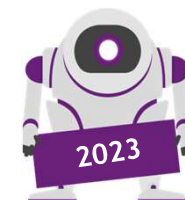
Node1



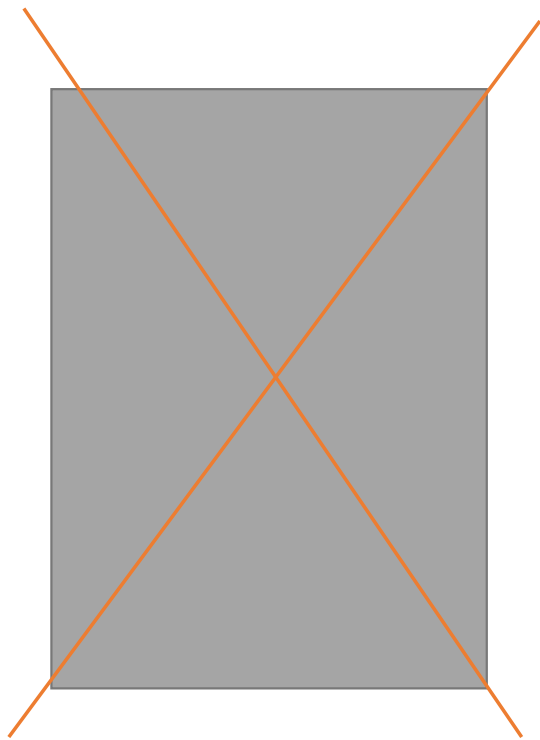
Node2



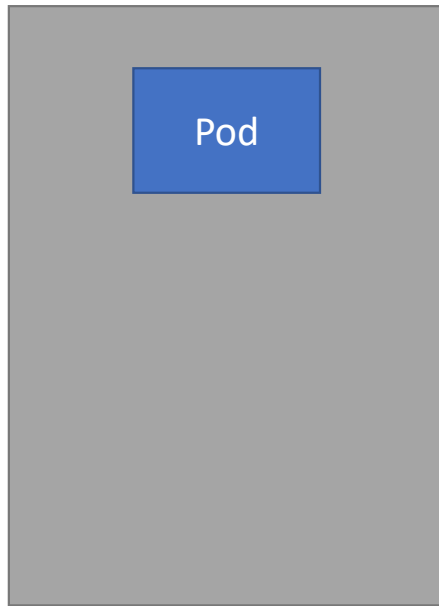
Node3



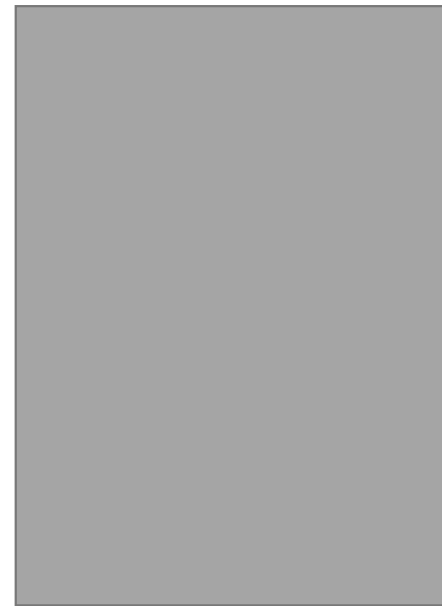




Node1

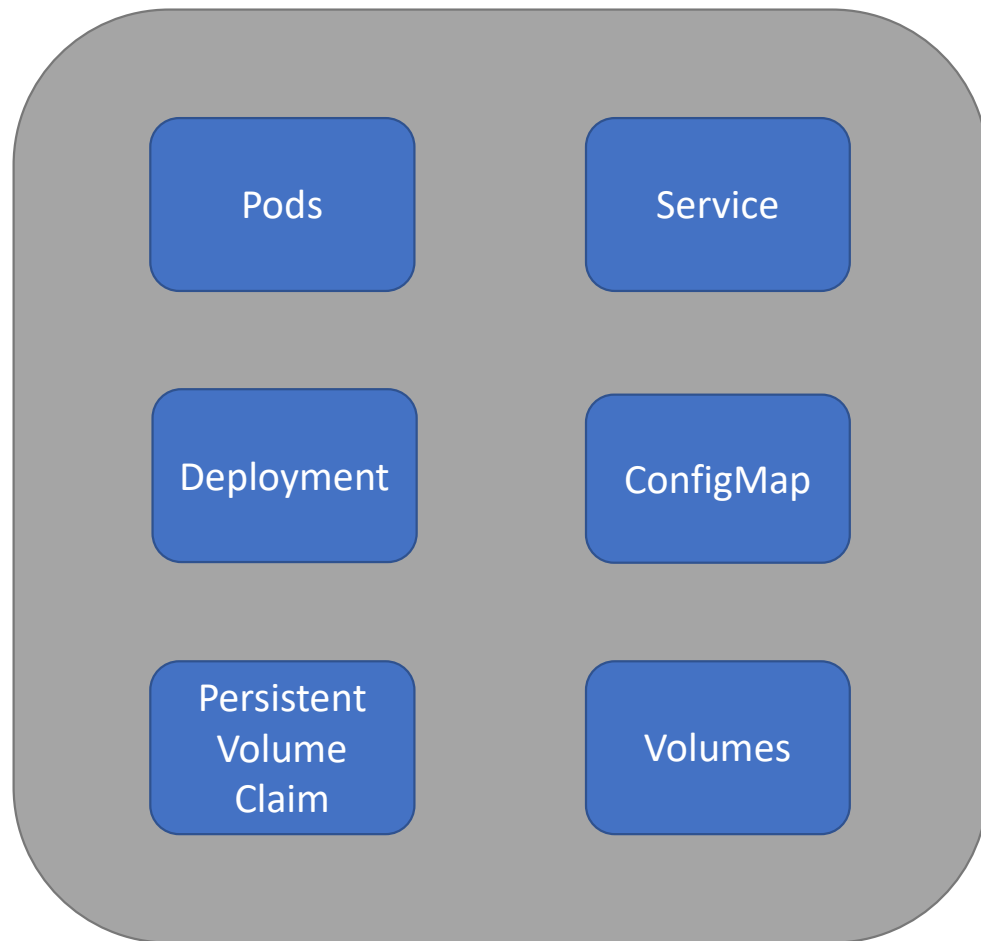


Node2



Node3





Network  
policies

Secrets

User  
account

Namespace

# Herramientas

---

- Docker for Windows
- Kubectl
- Visual Studio Code



DEMO

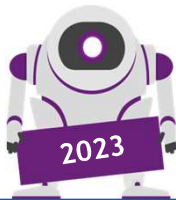
#netcoreconf



# Pod

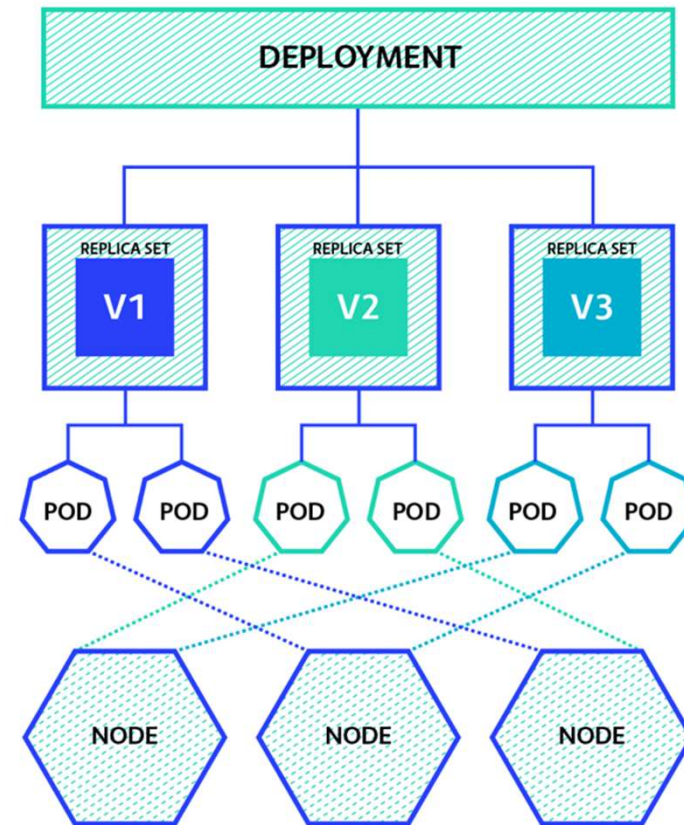
---

- Son la unidad mínima de K8s
- Pueden contener uno o más containers
- Permite establecer variables de entorno
- Permite mapear Paths con volúmenes
- Permite establecer parámetros de recursos
- Permite configurar un script inicial (sobre escribir el comando Docker)

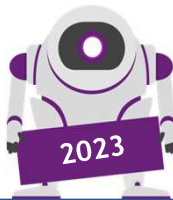


# Deployment

- Controlan los replica-sets de pods
- Permiten establecer escalamiento
- Intenta alcanzar el estado deseado
- Controla la estrategia de deploy / actualización



<https://godleon.github.io/blog/images/kubernetes/k8s-deployment.png>



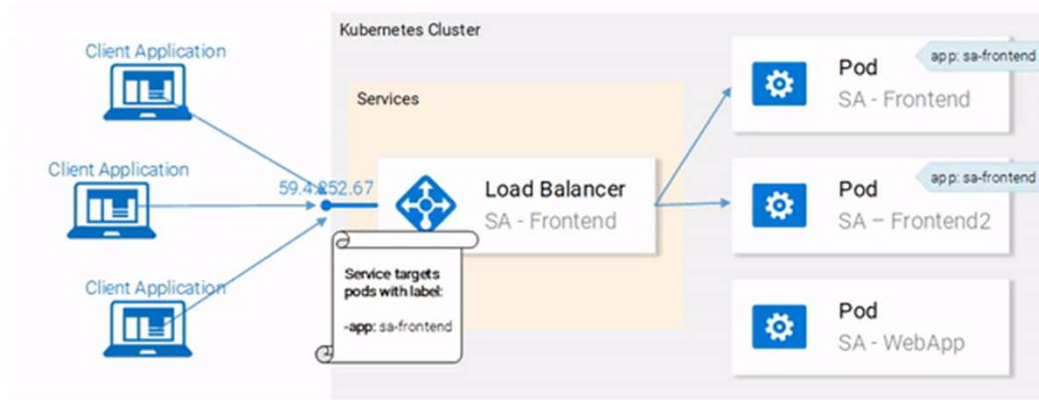


# Servicios

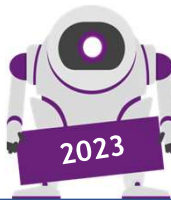
- Permiten el acceso a servicios
- Pueden ser load balancers
- Existen de diferentes tipos

```
samples > k8s > ! nginx-deployment.yaml
1  apiVersion: apps/v1
2  kind: Deployment
3  metadata:
4    name: my-nginx
5  spec:
6    selector:
7      matchLabels:
8        app: nginx
9    replicas: 1
10   template:
11     metadata:
12       labels:
13         app: nginx
14     spec:
15       containers:
16         - name: nginx
17           image: nginx:1.14.2
18           ports:
19             - containerPort: 80
```

```
samples > k8s > ! nginx-service.yaml
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: my-nginx-svc
5  labels:
6    app: nginx
7  spec:
8    type: LoadBalancer
9    ports:
10     - port: 80
11    selector:
12      app: nginx
```



<https://rinormaloku.com/wp-content/uploads/2018/02/loadbalancer.gif>

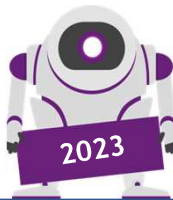


#netcoreconf

# Configmap

- Permiten guardar configuraciones
- Pueden ser de diferentes tipos
  - Clave / valor
  - Literal
- Se pueden mapear a variables de entorno
- Se pueden montar como volúmenes

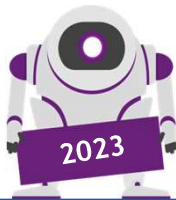
```
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app: my-wordpress
  name: mywordpress-config
data:
  MYSQL_RANDOM_ROOT_PASSWORD: '1'
  MYSQL_DATABASE: password
  MYSQL_USER: readWrite
  MYSQL_PASSWORD: password
  MYSQL_HOST: mysql
```



# Secrets

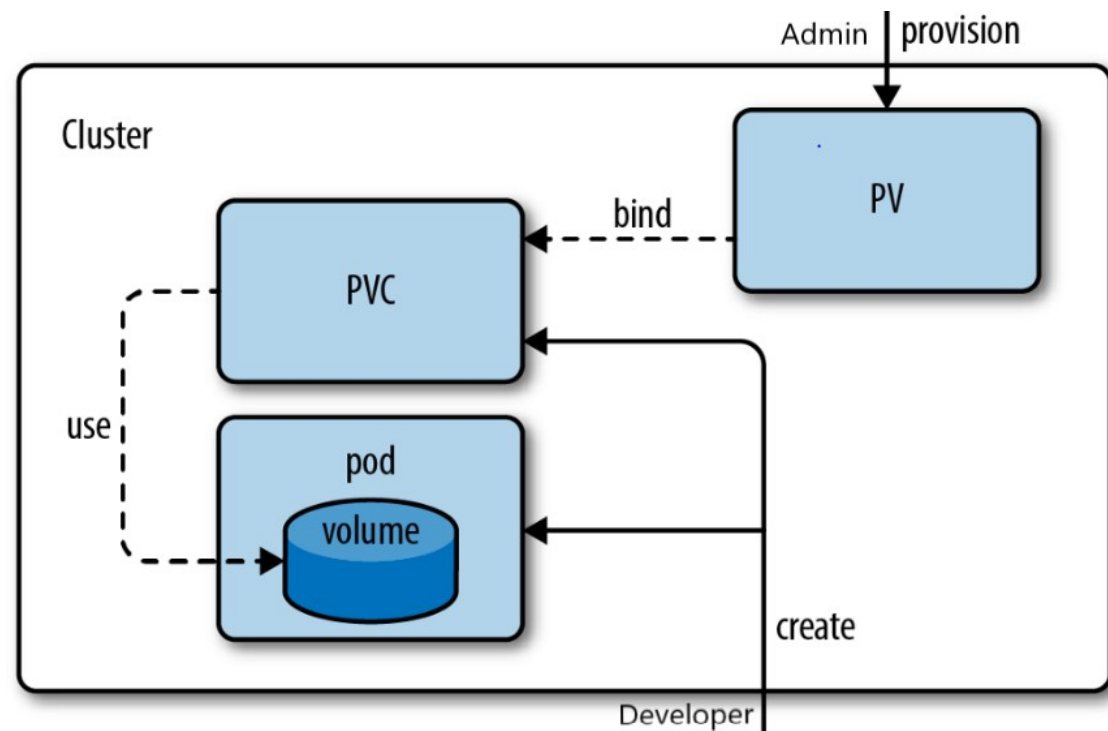
- Permiten guardar un valor sensible
- Se pueden mapear a variables de entorno
- Se pueden montar a directorios
- Es más interesante crearlo por comando

```
apiVersion: v1
kind: Secret
metadata:
  name: mysecrets
type: Opaque
data:
  MYSQL_PASSWORD: bXlwYXNzd29yZA==
```



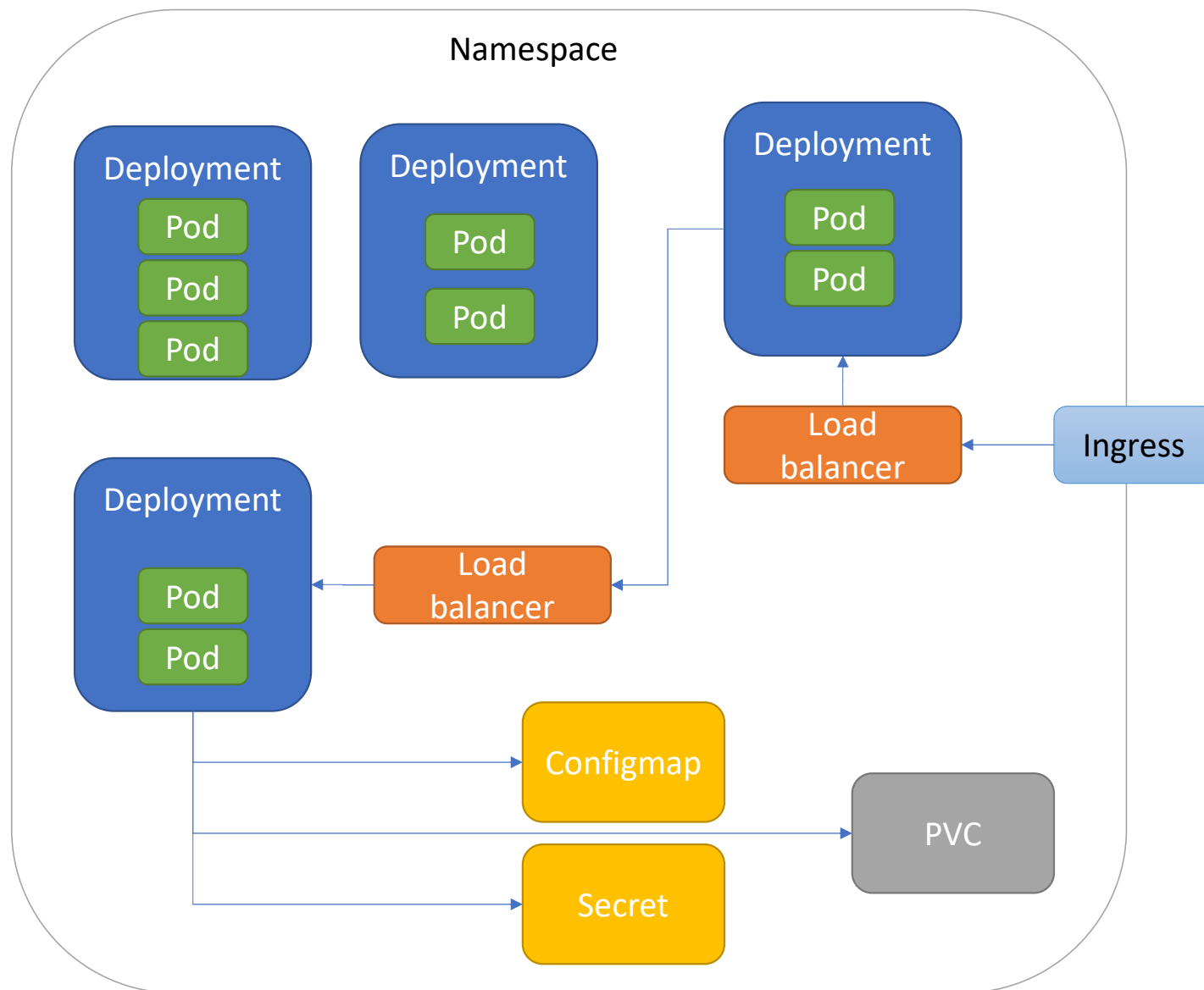
# Persistent Volumen Claim

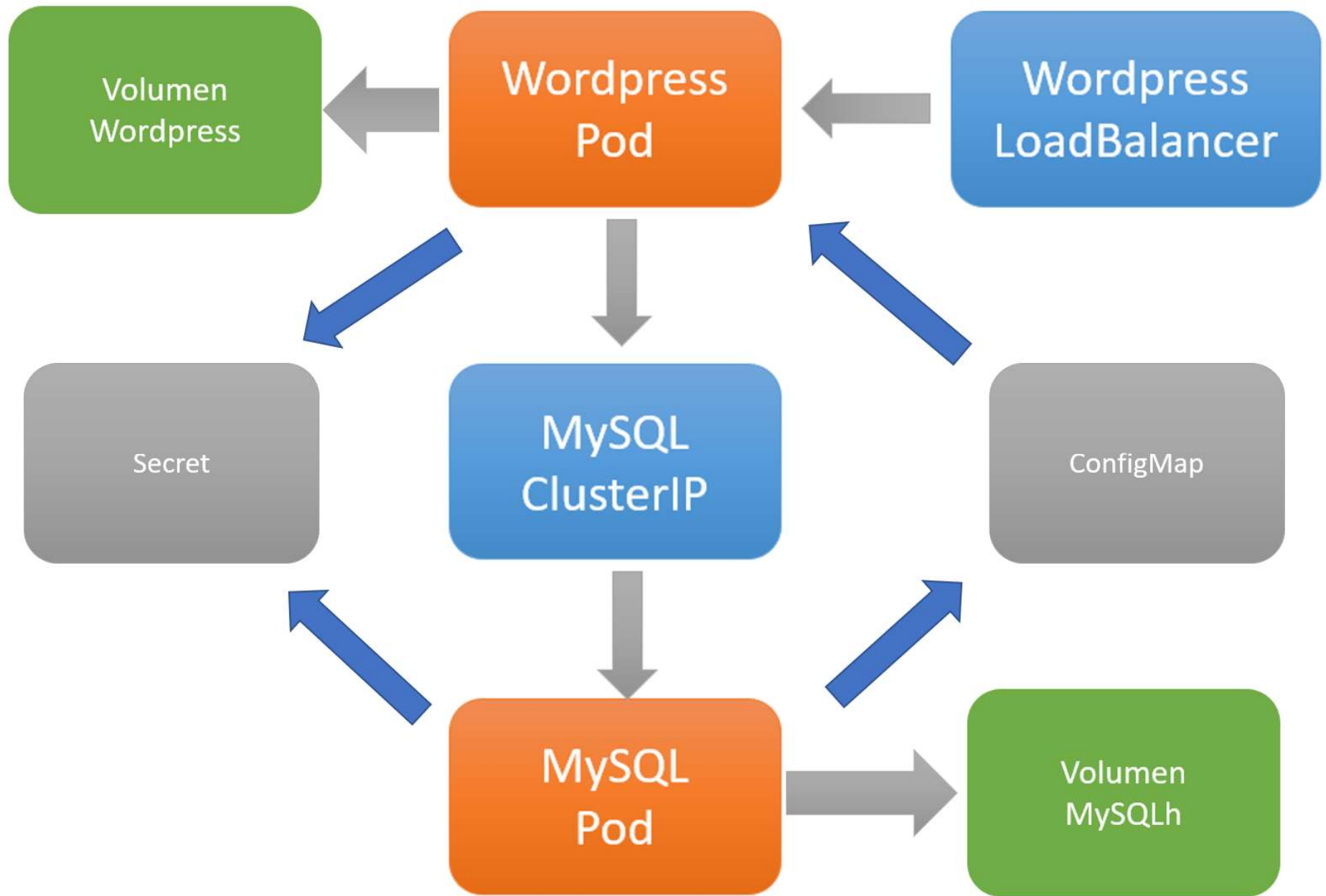
- Permiten acceder a un volumen
- Se pueden mapear a directorios



[https://i0.wp.com/blog.knoldus.com/wp-content/uploads/2021/05/1\\_keV2VBkCHb7cn\\_Rib0huYg.png?fit=868%2C553&ssl=1](https://i0.wp.com/blog.knoldus.com/wp-content/uploads/2021/05/1_keV2VBkCHb7cn_Rib0huYg.png?fit=868%2C553&ssl=1)









DEMO

#netcoreconf



# Preguntas



#netcoreconf



# Sponsors

**NTT DATA** **encamina**  
PIENSA EN COLORES

**plain  
concepts**

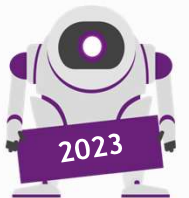
**intelequia**

**Verne**  
TECHNOLOGY GROUP

**TOKIOTA**

**ilitia**

#netcoreconf





More information:  
[info@netcoreconf.com](mailto:info@netcoreconf.com)  
@Netcoreconf

Visit on:  
[netcoreconf.com](http://netcoreconf.com)