

# Microservicios

The background is a dark navy blue. It features abstract, overlapping geometric shapes in various colors including bright green, cyan, magenta, orange, and light blue. These shapes are arranged in a way that creates a sense of depth and movement, with some shapes appearing to be layered on top of others. The overall aesthetic is modern and tech-oriented.

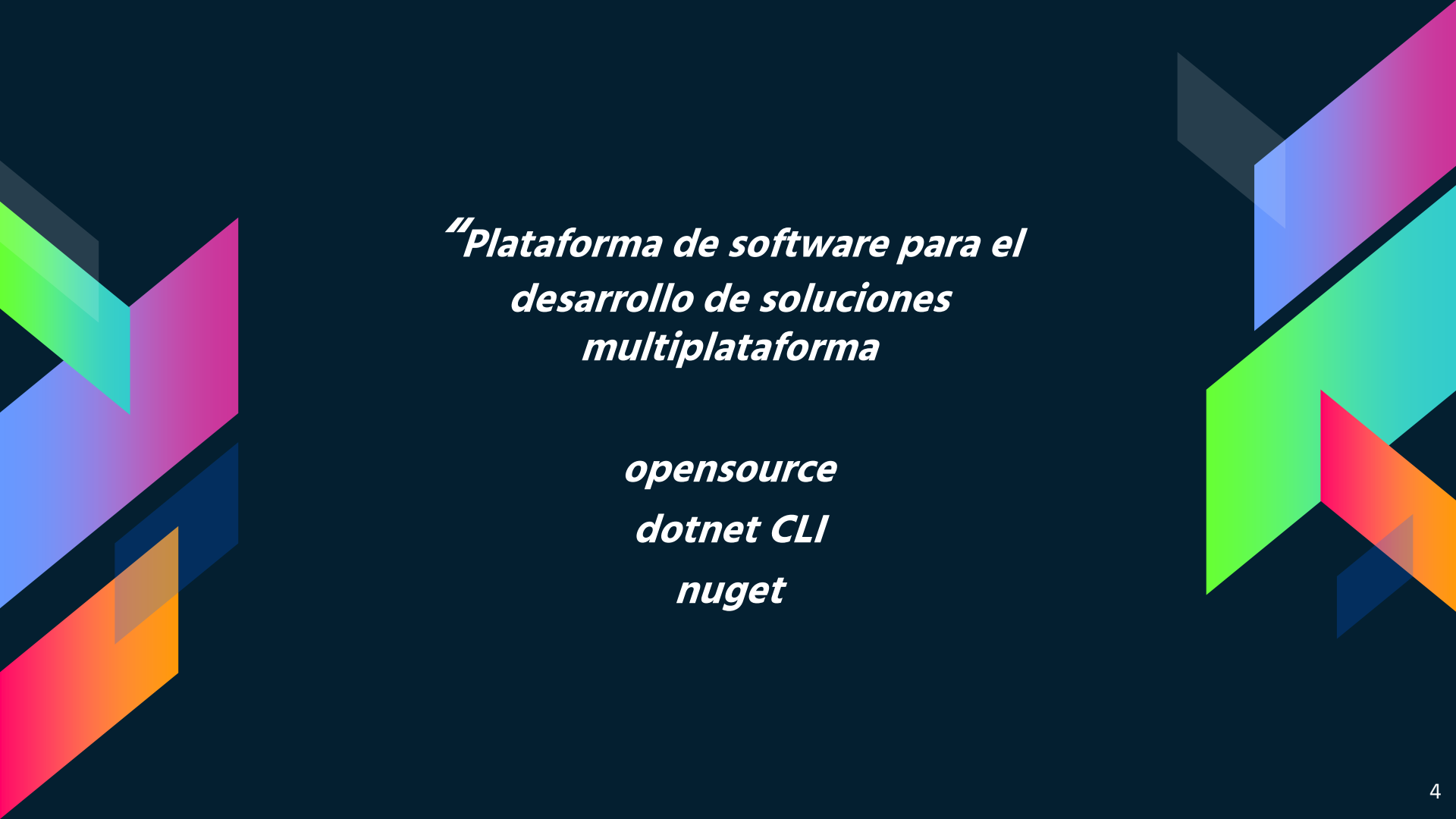
# Temario

- › .NET Core
- › Patrones de diseño
- › Api Design
- › Orquestación
- › Docker
- › Kubernetes
- › Azure AKS

The background is a dark navy blue. In the top-left and bottom-left corners, there are overlapping, semi-transparent geometric shapes in shades of cyan, lime green, orange, and magenta. In the top-right and bottom-right corners, there are similar overlapping shapes in shades of cyan, lime green, blue, and orange. The shapes are arranged in a way that they appear to be floating or layered, creating a modern, abstract aesthetic.

# **.NET Core**

Gestión y construcción de proyectos



***“Plataforma de software para el  
desarrollo de soluciones  
multiplataforma***

***opensource  
dotnet CLI  
nuget***

# Descargar .net core

<https://dotnet.microsoft.com/download/dotnet/5.0>

Verificar instalación con:

**dotnet --version**

# Lab 1 Generar proyecto .net Core

## Descargar visual studio code

<https://code.visualstudio.com/download>

## Generar una api y una prueba unitaria

### Ejecutar

```
dotnet new sln --name HelloWorld
```

```
dotnet new webapi --name Greetings --output Greetings
```

```
dotnet new classlib --name Serices --output Serices
```

```
dotnet new mstest --name ServiceTests --output ServiceTests
```

```
dotnet sln add .\Greetings\Greetings.csproj
```

```
dotnet sln add .\Serices\Services.csproj
```

```
dotnet sln add .\ServiceTests\ServiceTests.csproj
```

```
dotnet add .\Greetings\Greetings.csproj reference .\Serices\Services.csproj
```

```
dotnet add .\ServiceTests\ServiceTests.csproj reference .\Serices\Services.csproj
```



# Api Design

Mejores prácticas

# Api design





# Api design

Usar sustantivos y evitar los verbos

`/dogs` `/dogs/1234`

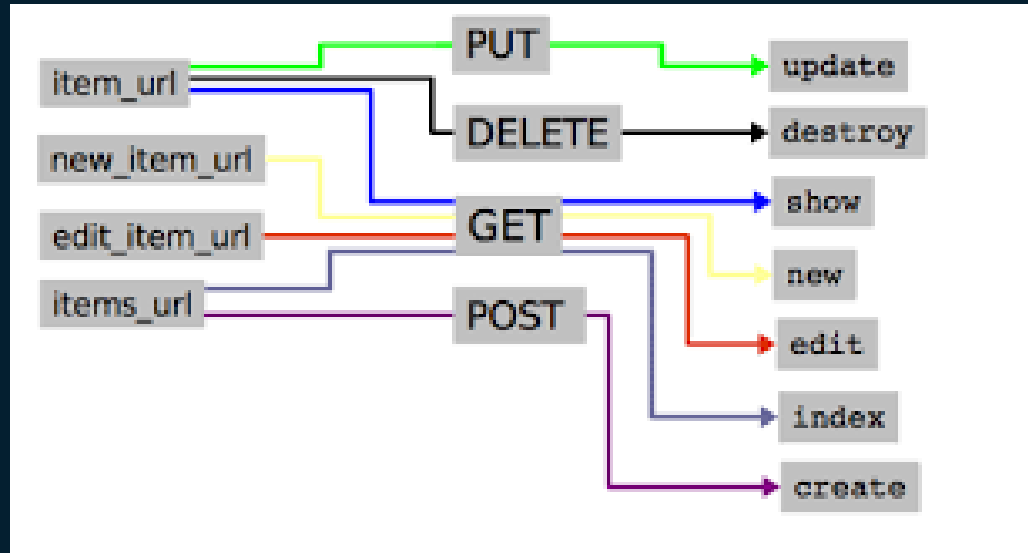
Evitar

`/getDogs` `/getDogs/1234`



# Api design

Utilizar verbos http correctos



# Api design

Recurso	POST	GET	PUT	DELETE
/dogs	Crea un nuevo perro	Lista de perros	Actualiza lista de perros	Borra todos los perros
/dogs/1234	Bad hombres	Muestra perro 1234	Si existe actualiza 1234	Borra 1234
			Si no existe error	



# Api design

## Plurales y nombres concretos

GET /dogs?color=red&state=running&location=park

POST /dog

GET /dogs

GET /dog

GET /getAllDogs



# Api design

## HTTP Status Codes

### Level 200 (Success)

200 : OK

201 : Created

203 : Non-Authoritative  
Information

204 : No Content

### Level 400

400 : Bad Request

401 : Unauthorized

403 : Forbidden

404 : Not Found

409 : Conflict

### Level 500

500 : Internal Server Error

503 : Service Unavailable

501 : Not Implemented

504 : Gateway Timeout

599 : Network timeout

502 : Bad Gateway

# Api design

## Versionamiento

Twilio /2010-04-01/Accounts/

Salesforce.com /services/data/v20.0/subjects/Account

Facebook ?v=1.0

## Best practice

/v1/accounts

/v2/accounts

## Content-Type

dogs/1 Content-Type: application/json

dogs/1 Content-Type: application/xml

dogs/1 Content-Type: application/png

# Api design

## Respuestas parciales

LinkedIn /people:(id,first-name,last-name,industry)

Facebook /joe.smith/friends?fields=id,name,picture

Google ?fields=title,media:group(media:thumbnail)

/dogs?fields=name,color,location

## Paginación

/dogs?limit=25&offset=50

Facebook - offset 50 and limit 25

Twitter - page 3 and rpp 25 (records per page)

LinkedIn - start 50 and count 25

# Api design

## Busqueda

Global search /search?q=fluffy+fur  
/owners/5678/dogs?q=fluffy+fur

## Seguridad

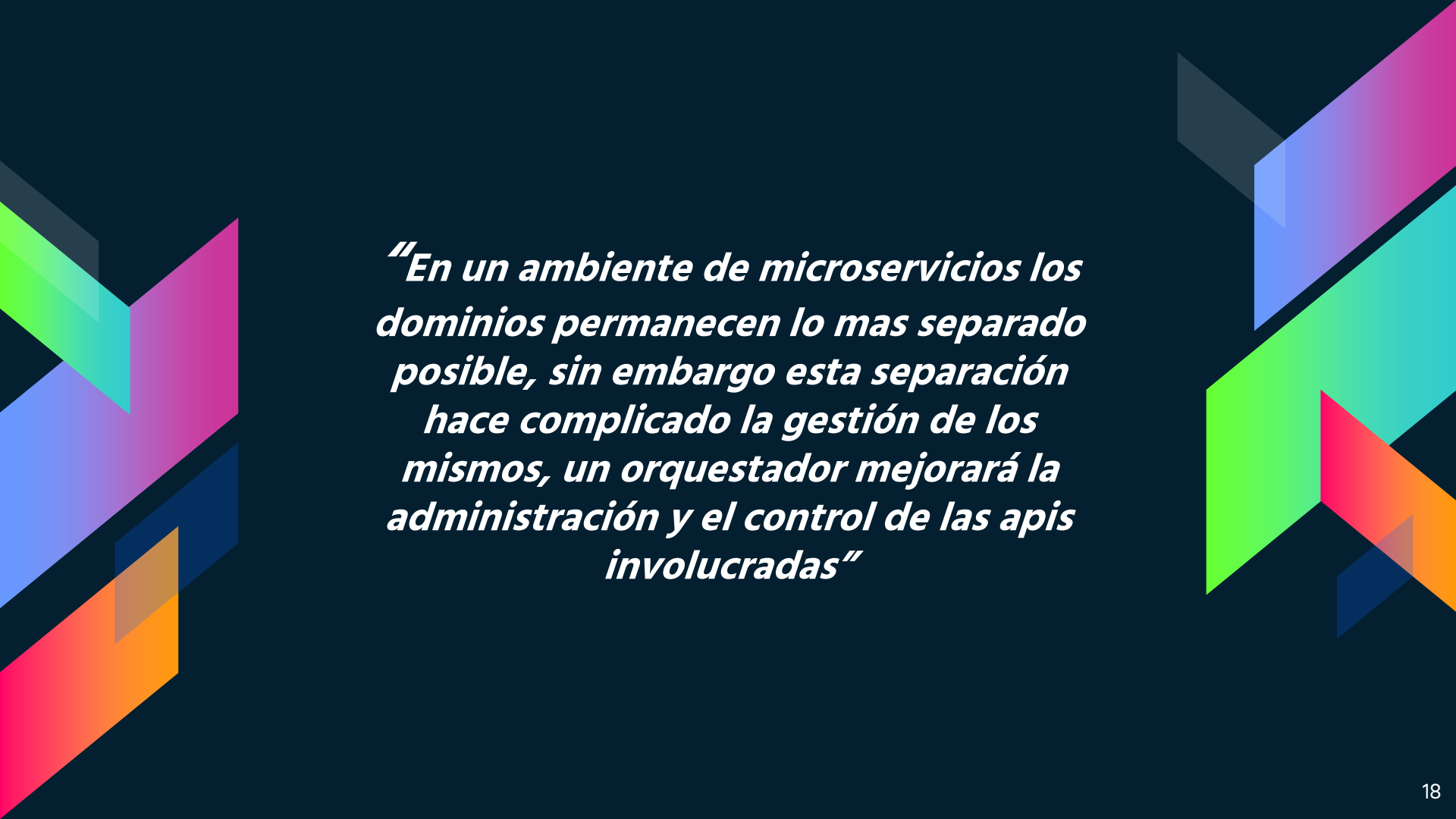
PayPal Permissions Service API  
Facebook OAuth 2.0  
Twitter OAuth 1.0a



The slide features a dark blue background with abstract geometric shapes in the corners. On the left, there are overlapping shapes in shades of green, blue, orange, and pink. On the right, there are similar shapes in shades of green, blue, pink, and orange. The central text is white and cyan.

# Orquestador

Gestión de microservicios



***“En un ambiente de microservicios los dominios permanecen lo mas separado posible, sin embargo esta separación hace complicado la gestión de los mismos, un orquestador mejorará la administración y el control de las apis involucradas”***

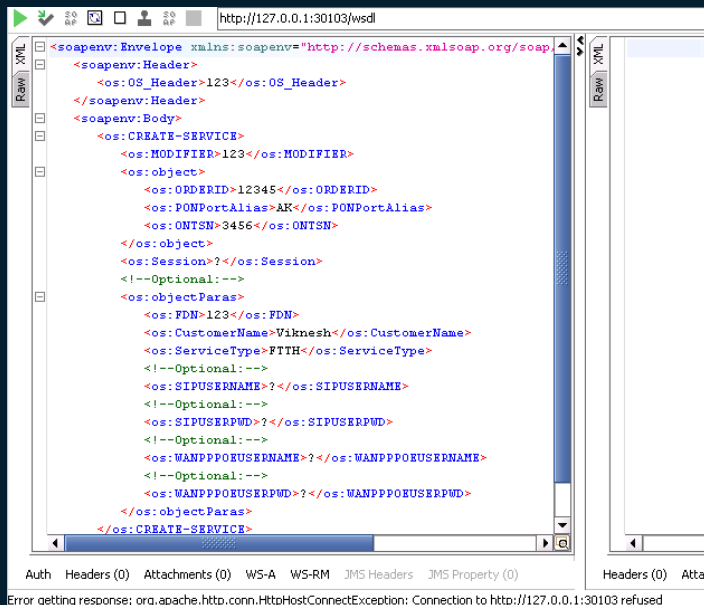
# Documentación



# Documentación

## Que es documentar un API?

Documento técnico que no que puede ser distribuido a los clientes que van a consumir nuestras APIs, la documentación debe ser capaz de transmitir claramente las entradas y salidas que nuestro manejará.



```
<?xml version='1.0' encoding='UTF-8'>
<soapenv:Envelope xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'>
  <soapenv:Header>
    <os:OS_Header>123</os:OS_Header>
  </soapenv:Header>
  <soapenv:Body>
    <os:CREATE-SERVICE>
      <os:MODIFIER>123</os:MODIFIER>
      <os:object>
        <os:ORDERID>12345</os:ORDERID>
        <os:PONPortAlias>AK</os:PONPortAlias>
        <os:ONTSN>3456</os:ONTSN>
      </os:object>
      <os:Session?</os:Session>
      <!--Optional:-->
      <os:objectParas>
        <os:FDN>123</os:FDN>
        <os:CustomerName>Wiknesh</os:CustomerName>
        <os:ServiceType>FITH</os:ServiceType>
        <!--Optional:-->
        <os:SIPUSERNAME?</os:SIPUSERNAME>
        <!--Optional:-->
        <os:SIPUSERPWD?</os:SIPUSERPWD>
        <!--Optional:-->
        <os:WANPPPORUSERNAME?</os:WANPPPORUSERNAME>
        <!--Optional:-->
        <os:WANPPPORUSERPWD?</os:WANPPPORUSERPWD>
      </os:objectParas>
    </os:CREATE-SERVICE>
  </soapenv:Body>
</soapenv:Envelope>
```

Auth Headers (0) Attachments (0) WS-A WS-RM JMS Headers JMS Property (0) Headers (0) Attac

Error getting response; org.apache.http.conn.HttpHostConnectException: Connection to http://127.0.0.1:30103 refused

# Documentación

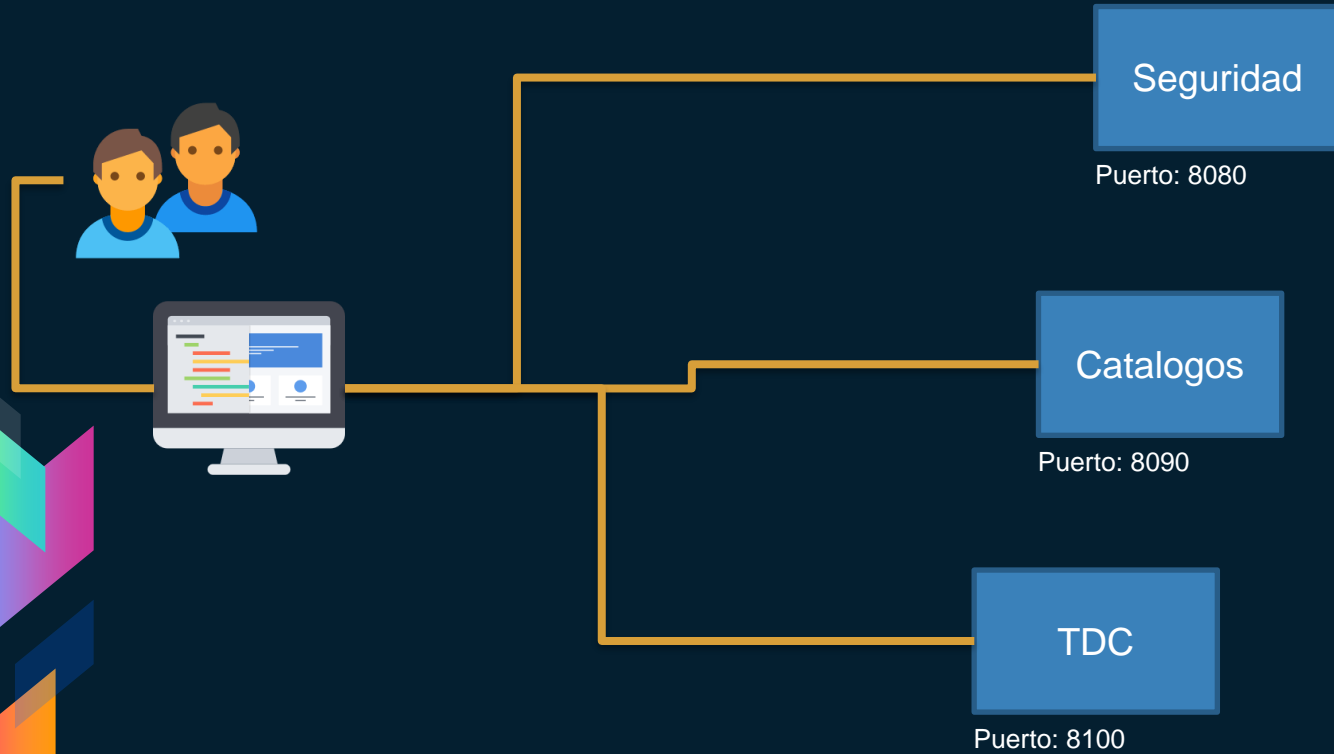
## Importa?

Los servicios rest no generan un contrato natural de entradas y salidas como lo hace un servicio SOAP con el wsdl.

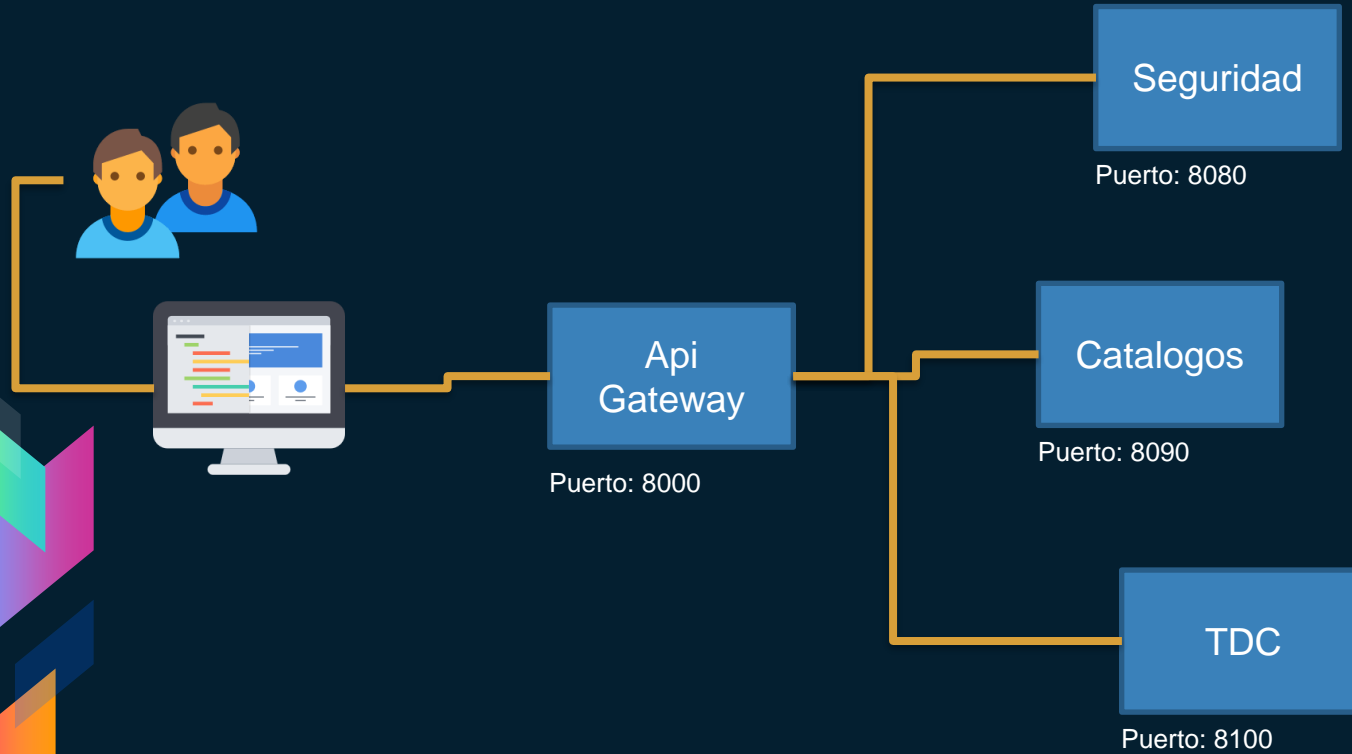


<http://localhost:8090/swagger-ui.html>

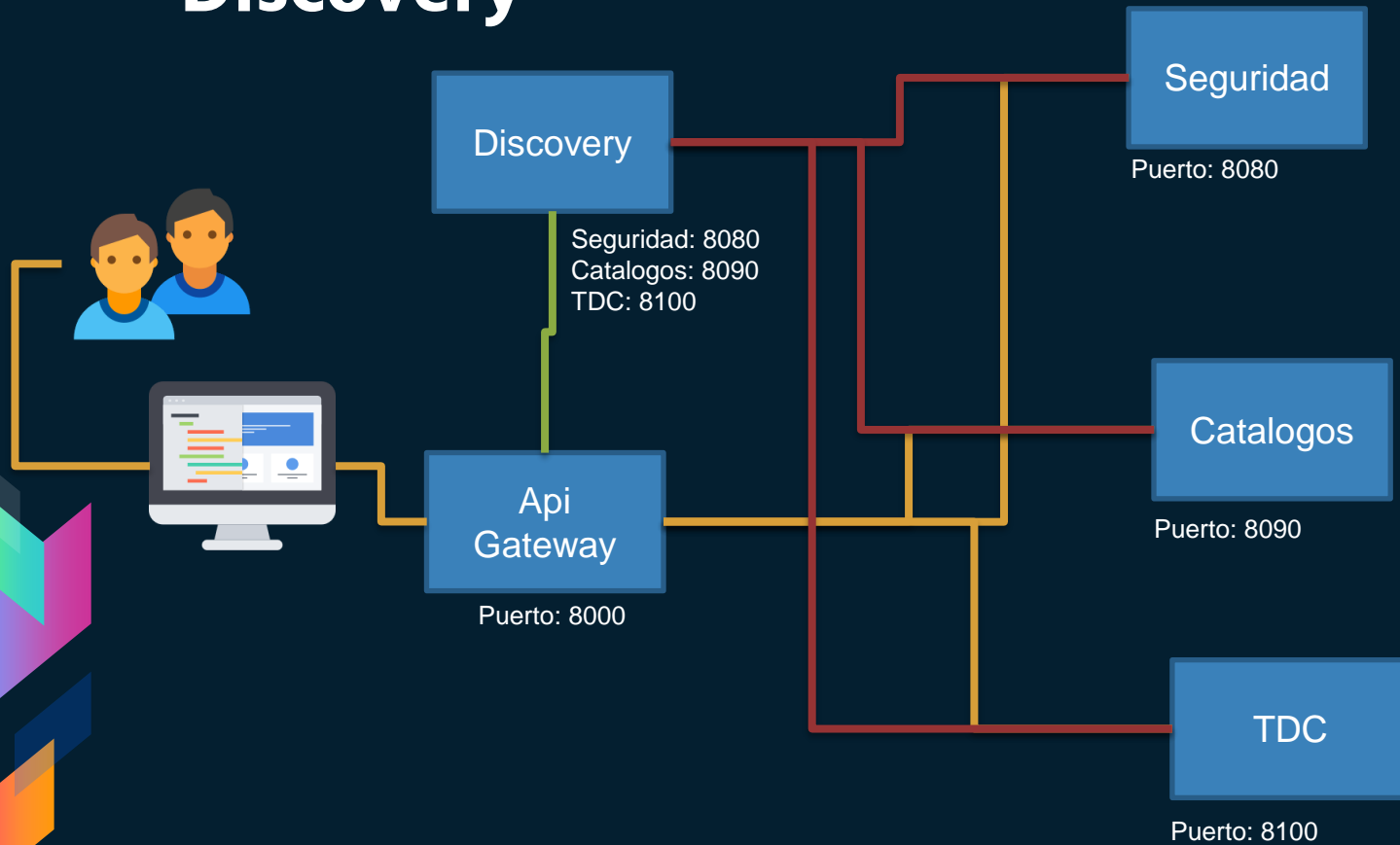
# Api Gateway



# Api Gateway

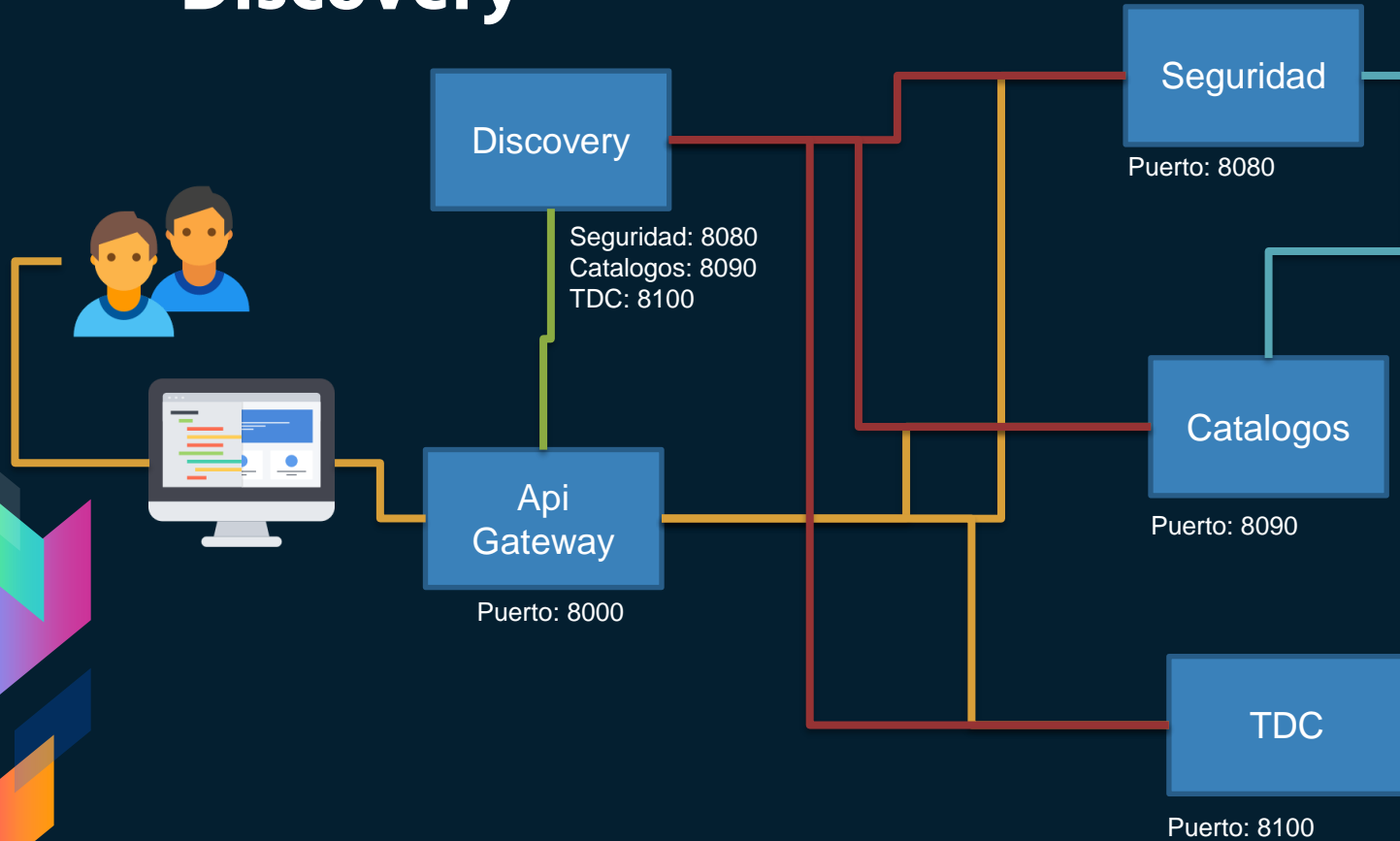


# Discovery





# Discovery



# Contenedores

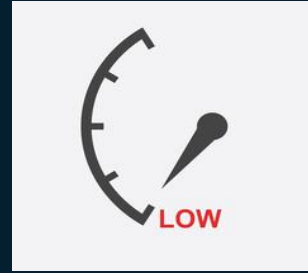
Automatizar despliegue de  
aplicaciones

# Traditional



# Tradicional

\$ \$ \$  
\$ \$ \$



**Qué tipo de  
servidor se  
requiere  
para esta  
aplicación?**

# Tradicional



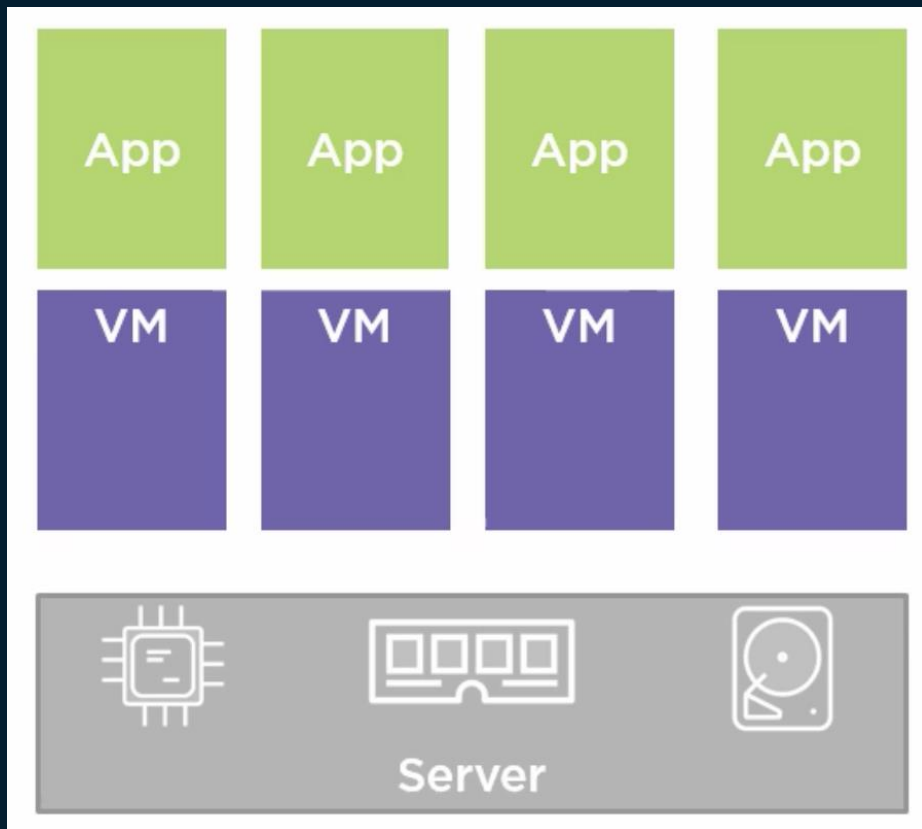
\$ \$ \$  
vmware®



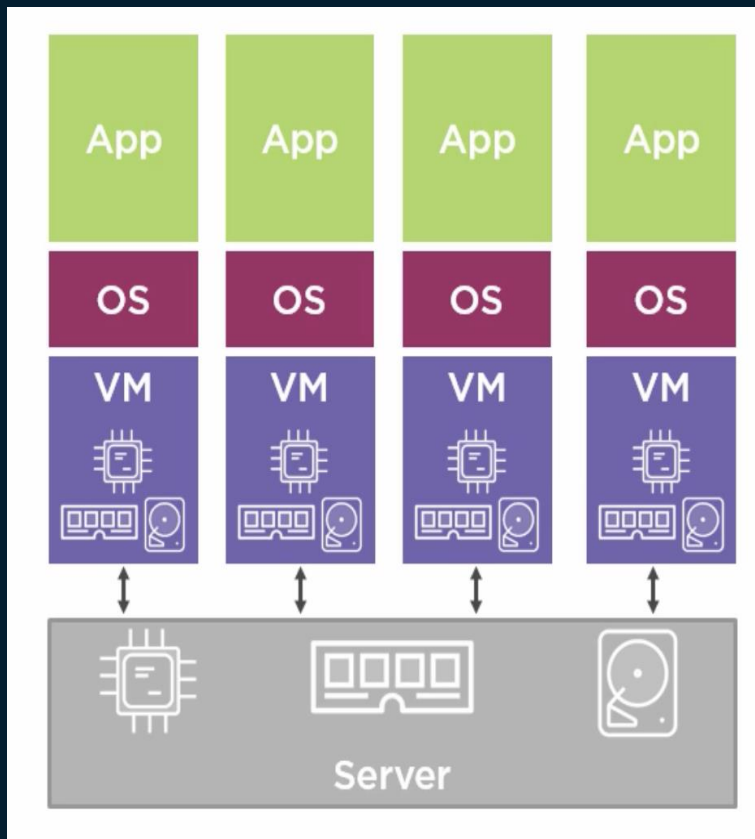
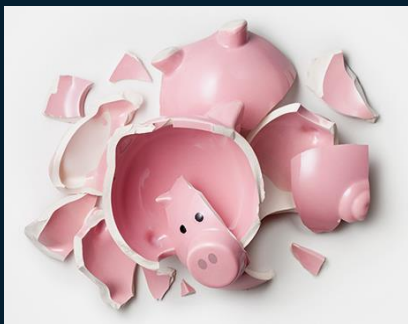
# Tradicional



# El problema



# El problema





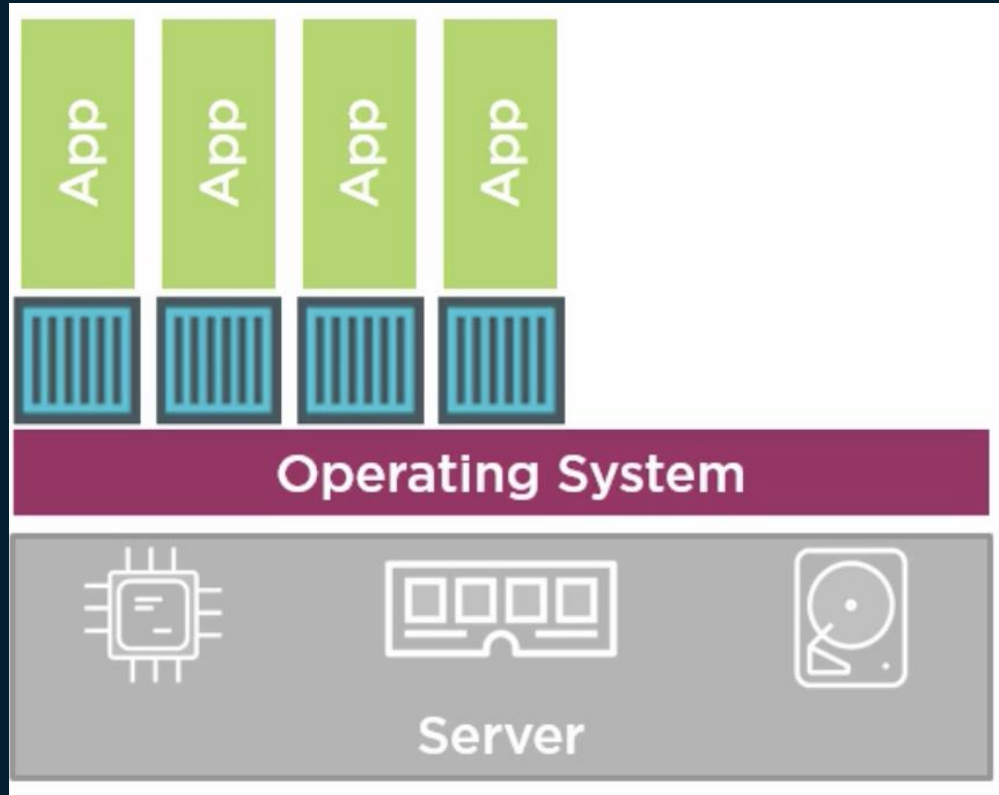
# Contenedores



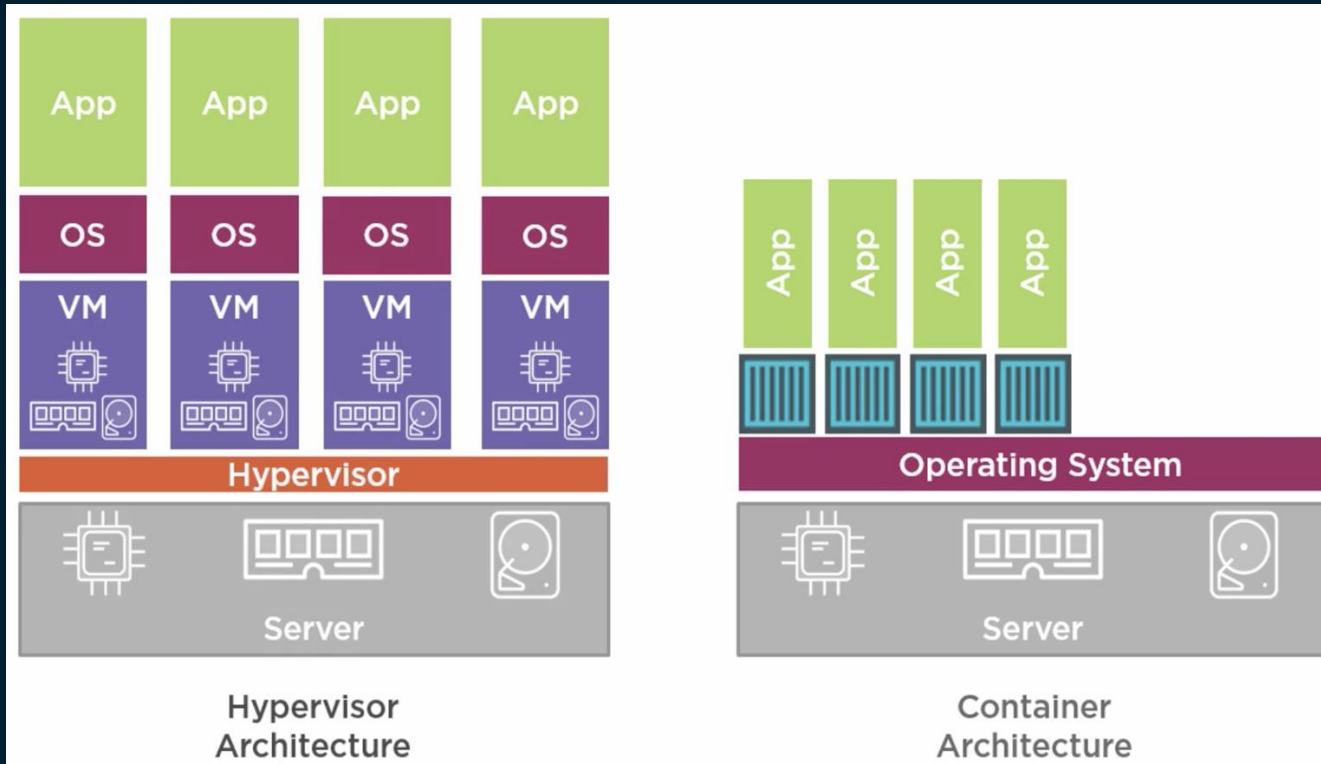
# Contendores



# Contendores

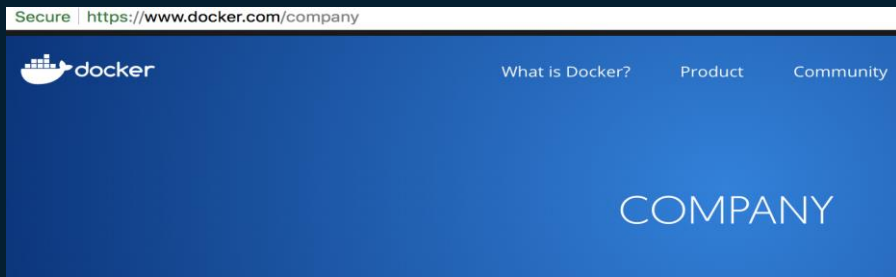


# Contenedores



# Qué es Docker?

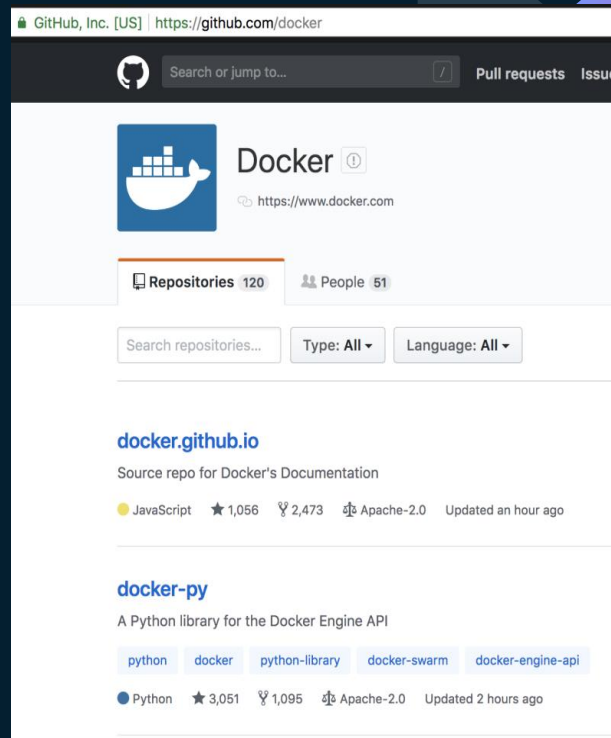
## Docker Company



## Open Container Initiative





## Docker Project



# Docker project

GitHub, Inc. [US] <https://github.com/docker>

Search or jump to... Pull requests Issues

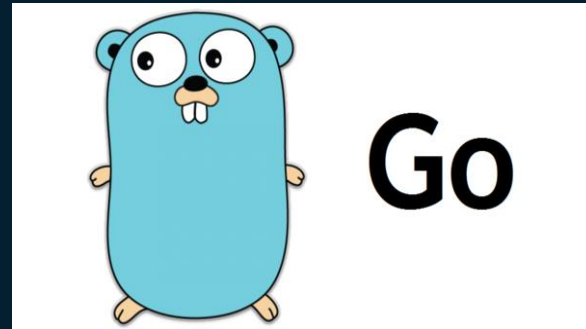
 **Docker**   
<https://www.docker.com>

**Repositories** 120 **People** 51

Search repositories... Type: All Language: All

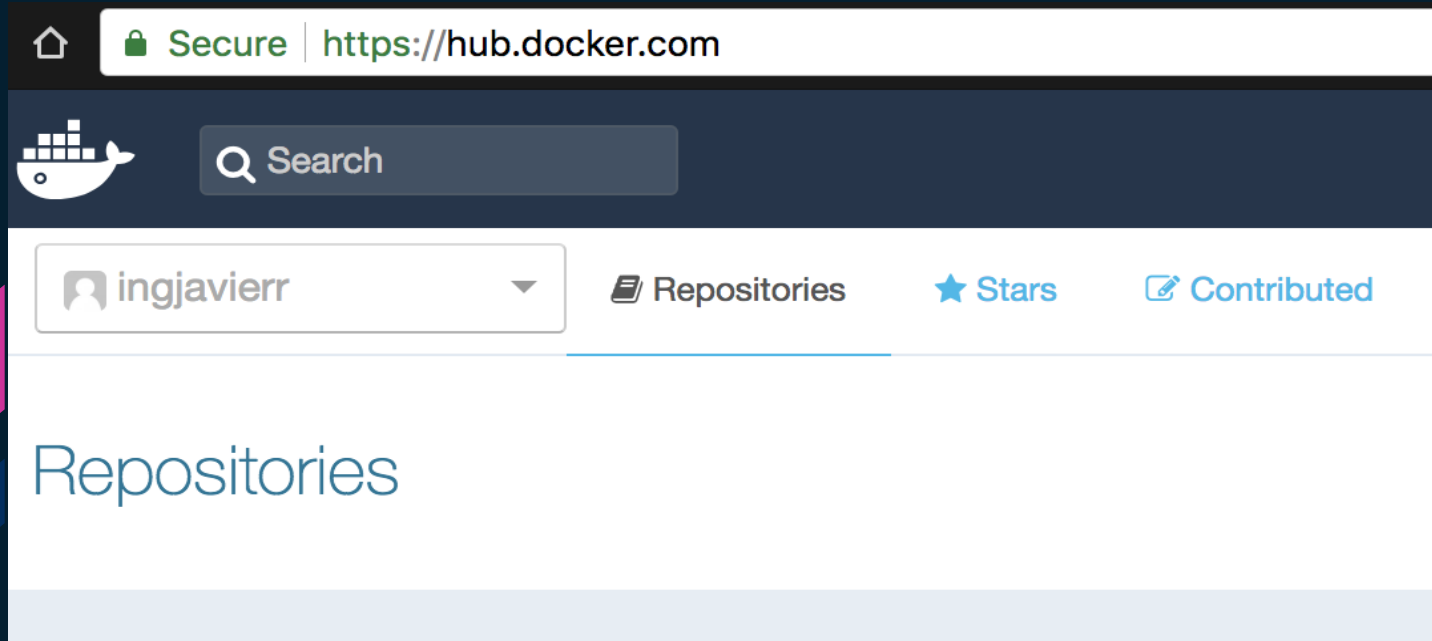
**docker.github.io**  
Source repo for Docker's Documentation  
JavaScript ★ 1,056 🍏 2,473 📄 Apache-2.0 Updated an hour ago

**docker-py**  
A Python library for the Docker Engine API  
python docker python-library docker-swarm docker-engine-api  
Python ★ 3,051 🍏 1,095 📄 Apache-2.0 Updated 2 hours ago



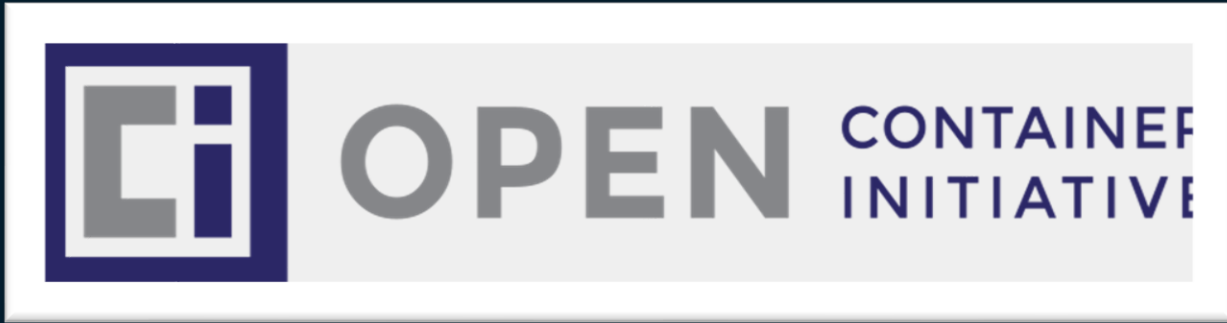
# Docker project

## Docker Hub





# Open container initiative



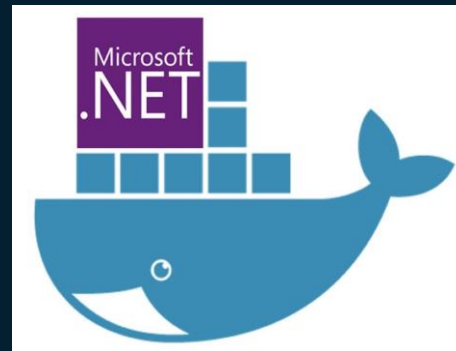


# Que es un contenedor?



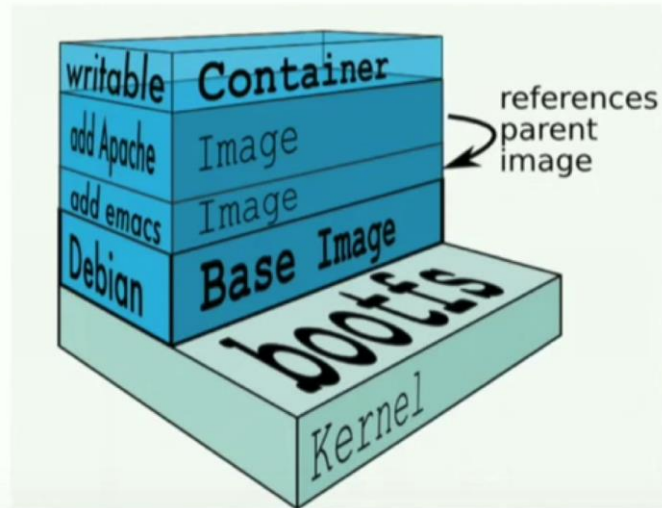
Es una instancia de un servidor en su mínima expresión.

# Qué puede vivir en contenedor?



# Docker image layers

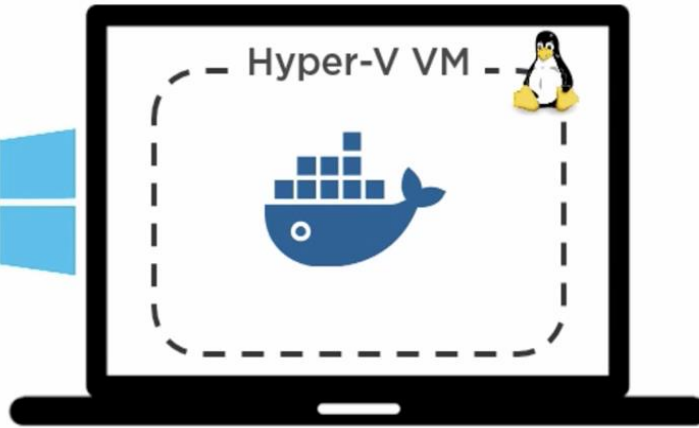
Image layers



# Docker en windows

## Pre-reqs

- Windows 10
- 64-bit
- Clean (ish) install



## Use cases

- Test
- Dev
- ~~Production~~

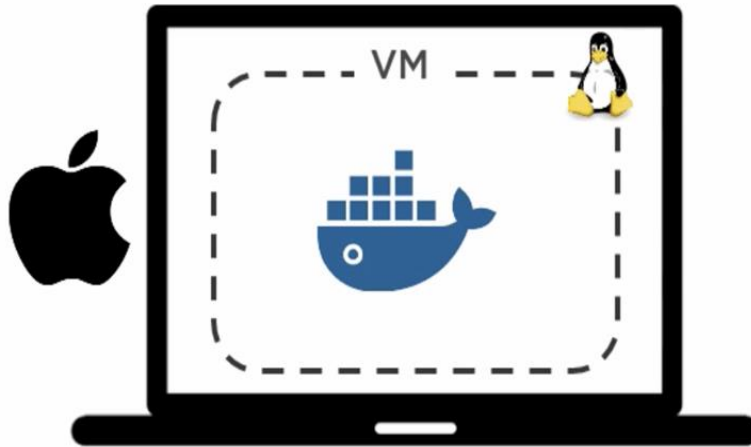
# Docker en MAC

## HyperKit

- <https://github.com/docker/hyperkit>

## DataKit

- <https://github.com/docker/datakit>



## Moby Linux

- Based on Alpine Linux

# Docker en linux (docker version)

```
Client:
 Version:      18.04.0-ce
 API version:  1.37
 Go version:   go1.9.4
 Git commit:   3d479c0
 Built: Tue Apr 10 18:21:36 2018
 OS/Arch:     linux/amd64
 Experimental: false
 Orchestrator: swarm

Server:
 Engine:
  Version:      18.04.0-ce
  API version:  1.37 (minimum version 1.12)
  Go version:   go1.9.4
  Git commit:   3d479c0
  Built:       Tue Apr 10 18:25:25 2018
  OS/Arch:     linux/amd64
  Experimental: false
```

# Docker en windows server 2016+

```
PS C:\Windows\system32> docker version
Client:
 Version:      1.12.0-dev
 API version:  1.24
 Go version:   go1.5.3
 Git commit:   8e92415
 Built:        Thu May 26 17:08:34 2016
 OS/Arch:     windows/amd64

Server:
 Version:      1.12.0-dev
 API version:  1.24
 Go version:   go1.5.3
 Git commit:   8e92415
 Built:        Thu May 26 17:08:34 2016
 OS/Arch:     windows/amd64
```

# Dockerfile

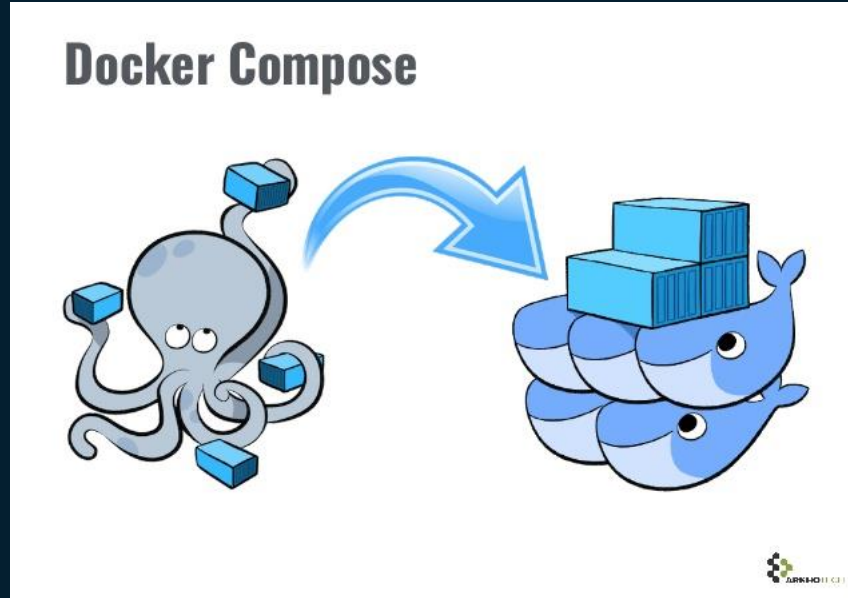
Docker puede crear imágenes tomando instrucciones de un archivo de configuración llamado Dockerfile

```
FROM ubuntu  
RUN apt-get update -y && apt-get install -y apache2 apache2-utils  
COPY hola.html /var/www/html/hola.html  
CMD /usr/sbin/apache2ctl -D FOREGROUND
```

```
docker build -t hola .  
docker run -d -p "8090:80" hola
```



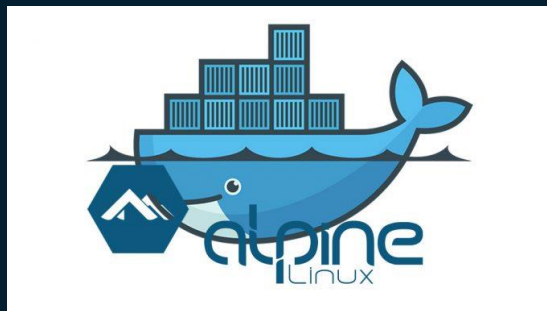
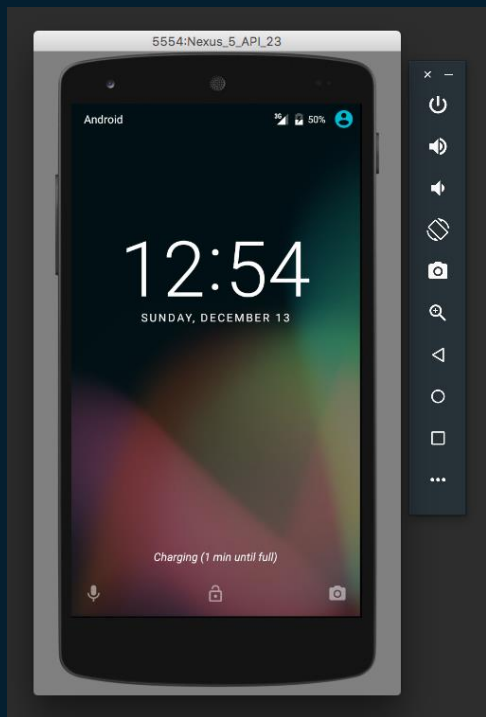
# Docker compose



# Y también...

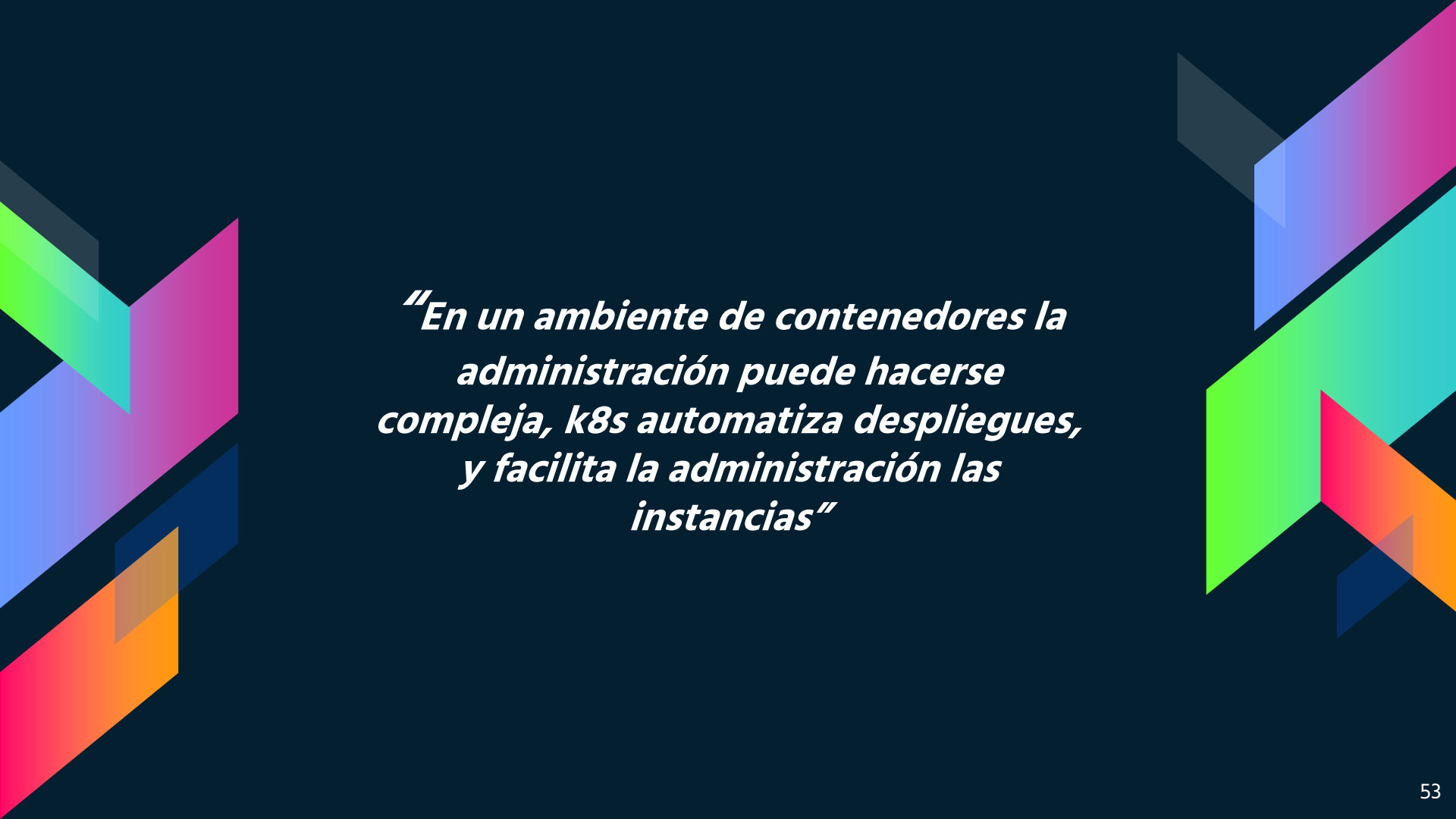


# Docker y android



# Kubernetes

Administrar contenedores



***“En un ambiente de contenedores la administración puede hacerse compleja, k8s automatiza despliegues, y facilita la administración las instancias”***

# Historia



**Donado a la Linux Foundation en 2014**

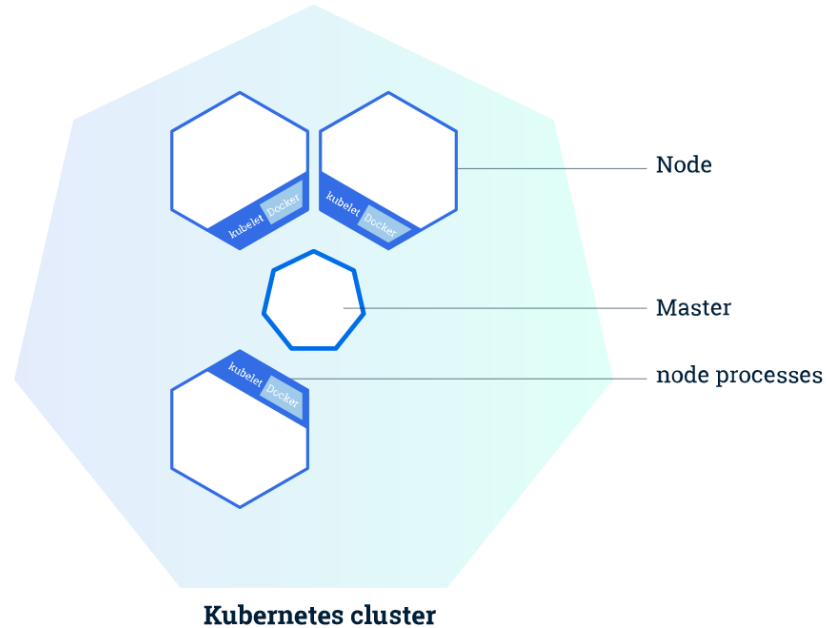
**Desarrollado en Go**

**<https://github.com/kubernetes>**

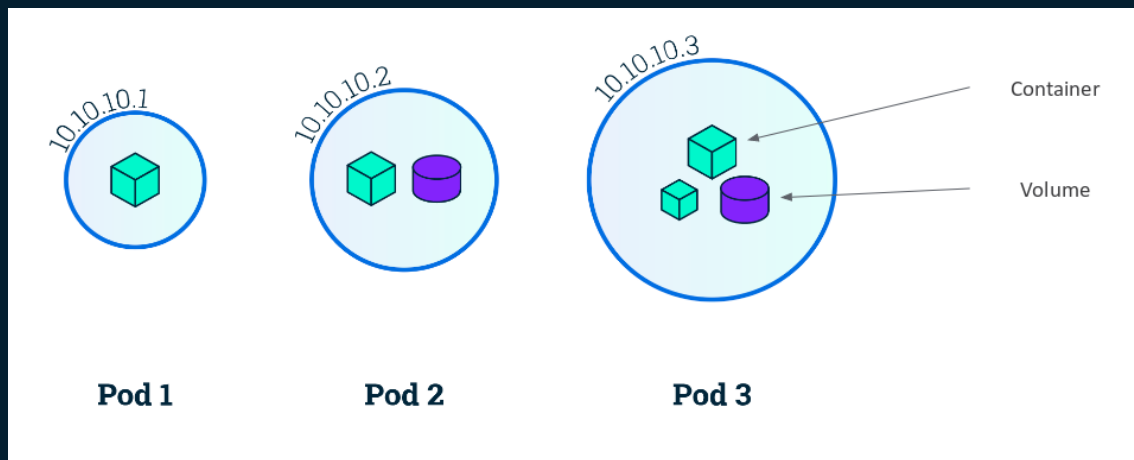
**Kubernetes (k8s)**

# Arquitectura

## Cluster Diagram

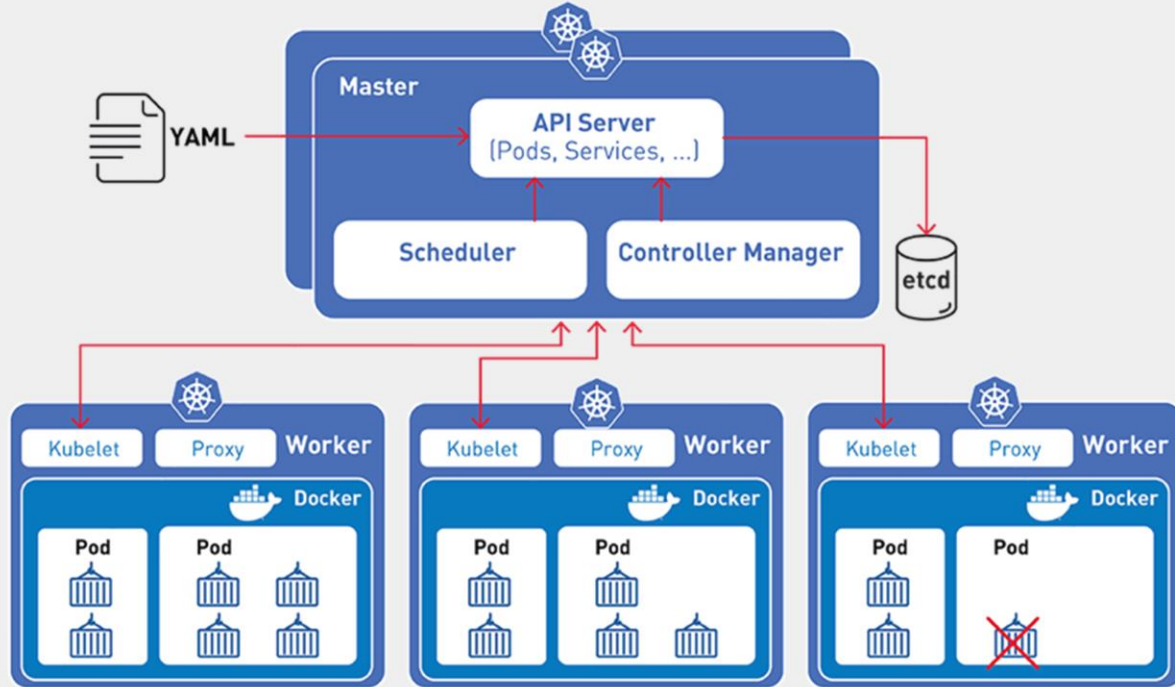


# Arquitectura

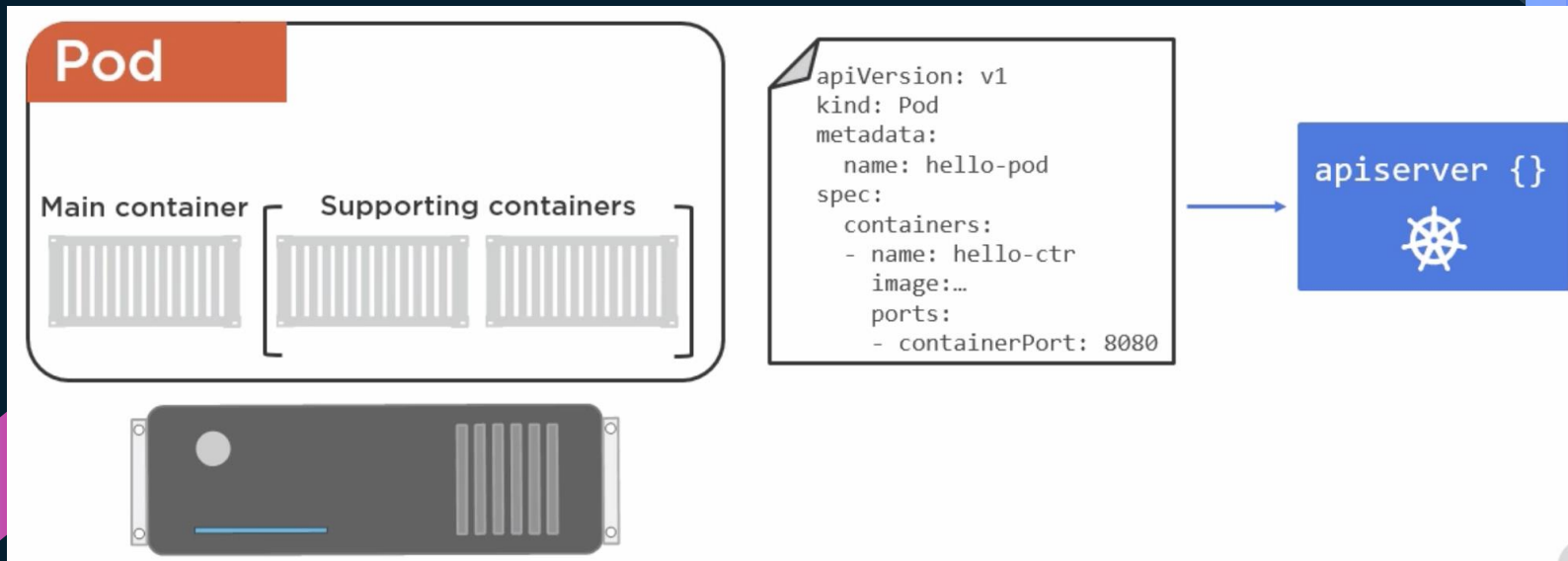




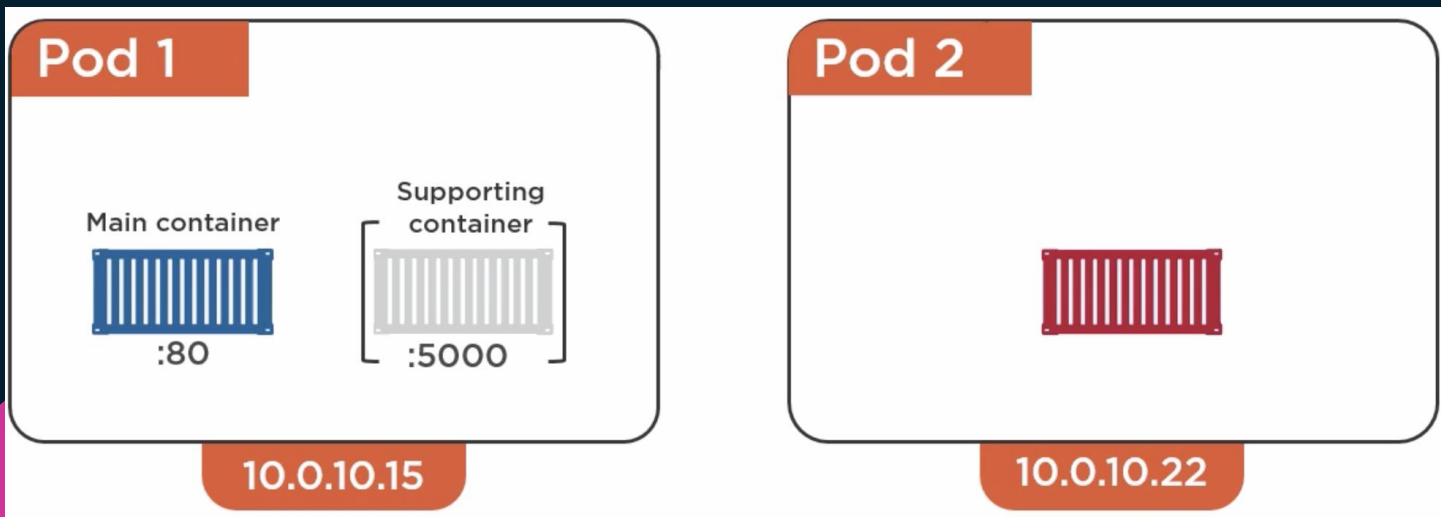
# Arquitectura



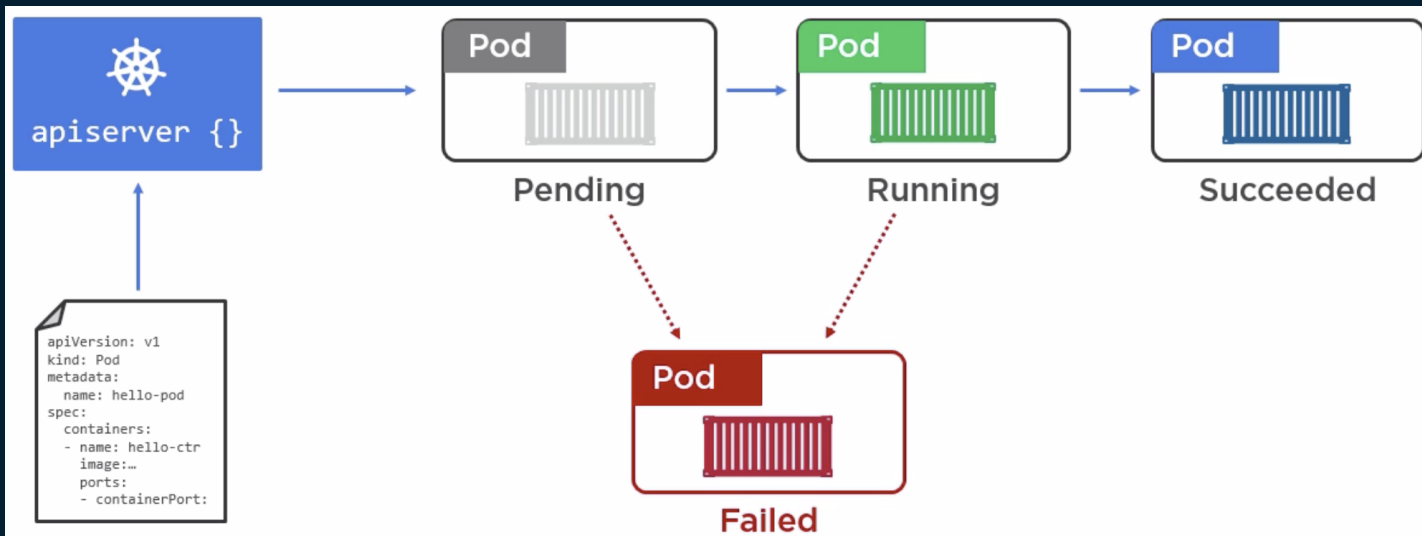
# Pods



# Pods



# Pod lifecycle



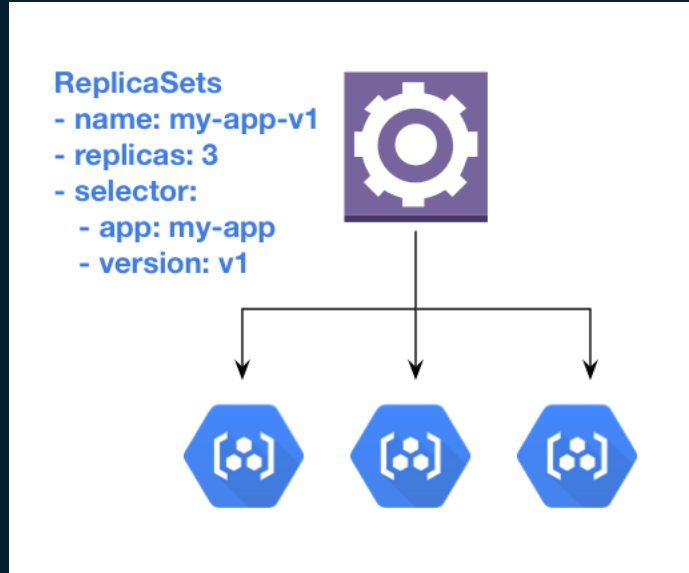
# Pods

- > `$ kubectl get pods`
- > `$ kubectl get pods --all-namespaces`

Manifiesto pod.yml

- > `$ kubectl create -f pod.yml`
- > `$ kubectl get pods`
- > `$ kubectl describe pods`

# Pods Replication controller (desired state)



# Pods (Replication controller)

- > `$ kubectl get pods`
- > `$ kubectl delete pods hello-pod`

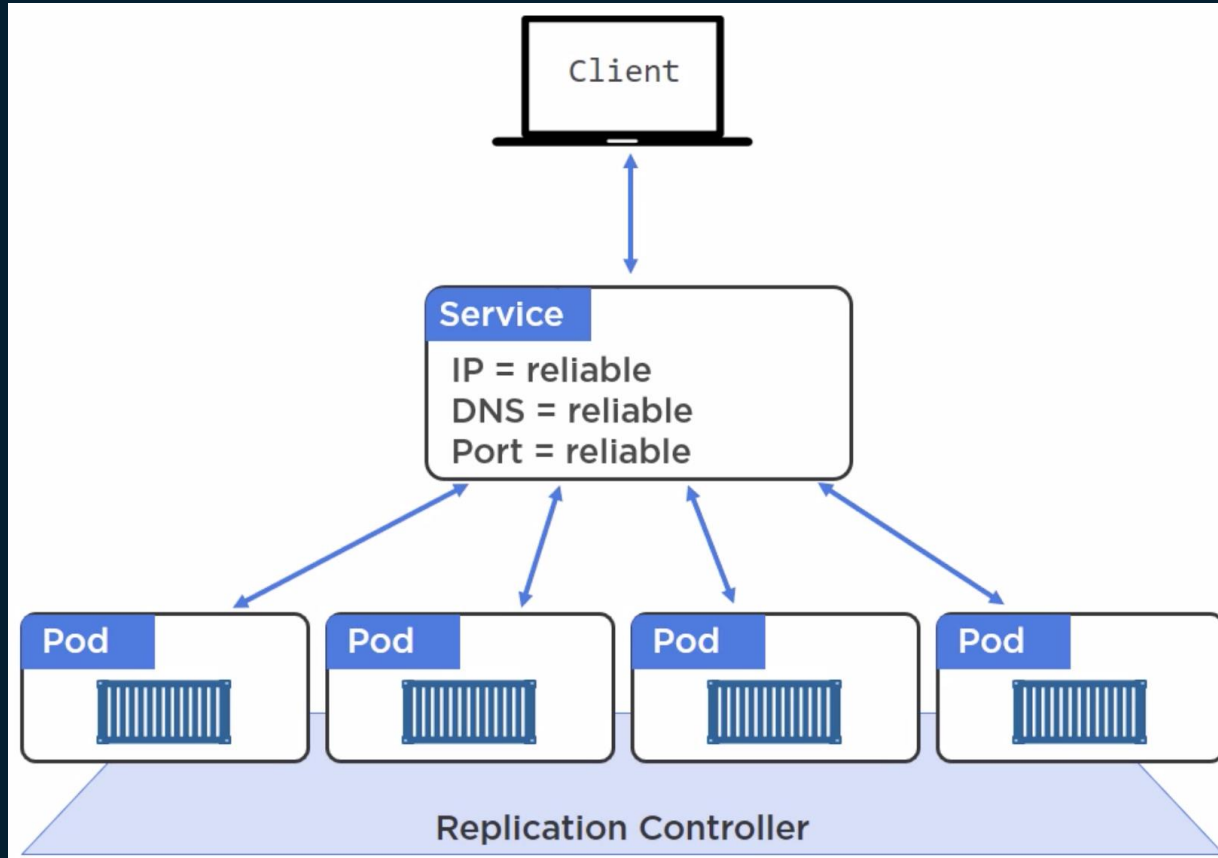
Manifiesto rc.yml

- > `$ kubectl create -f rc.yml`
- > `$ kubectl get rc`
- > `$ kubectl describe rc`

Modificar rc.yml

- > `$ kubectl apply -f rc.yml`
- > `$ kubectl get rc -o wide`
- > `$ kubectl get pods`

# Services





# Services

- > `$ kubectl get rc`
- > `$ kubectl expose rc hello-rc --name=hello-svc --target-port=80 --type=NodePort`
- > `$ kubectl describe svc hello-svc`
- > `$ minikube ip`
- > `$ kubectl delete svc hello-svc`

# Services (declarative yml)

Generar manifiesto svc.yml

- > \$ kubectl describe pods | grep app
- > \$ kubectl create -f svc.yml
- > \$ minikube ip
- > \$ kubectl describe svc hello-svc
- > \$ kubectl get ep
- > \$ kubectl describe ep hello-svc