Microservicios

Temario

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

.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 Services --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


App User
App Store
App
App Developer
World of APIs
API
API Team
Internal Systems

Api design

Usar sustantivos y evitar los verbos

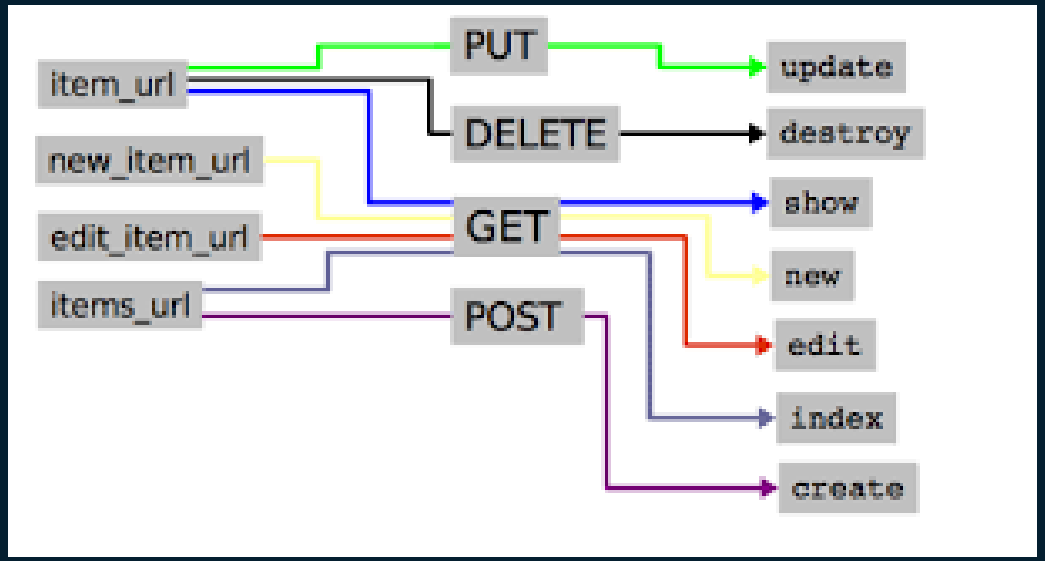
/dogs /dogs/1234

Evitar

/getDogs /getDogs/1234

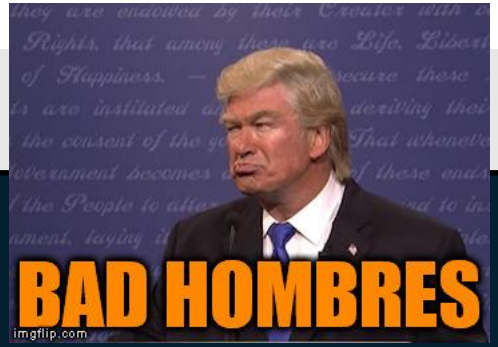
Api design

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

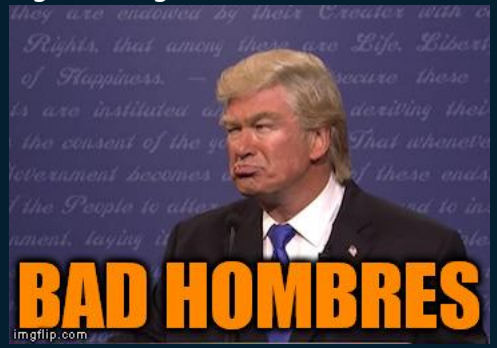
GET /dogs?color=red&state=running&location=park

POST /dog

GET /dogs

GET /dog

GET /getAllDogs



Api design

HTTP Status Codes

Level 200 (Success)	Level 400	Level 500
200 : OK	400 : Bad Request	500 : Internal Server Error
201 : Created	401 : Unauthorized	503 : Service Unavailable
203 : Non-Authoritative Information	403 : Forbidden	501 : Not Implemented
204 : No Content	404 : Not Found	504 : Gateway Timeout
	409 : Conflict	599 : Network timeout
		502 : Bad Gateway

Api design

Versionamiento

Twilio /2010-04-01/Accounts/

Salesforce.com /services/data/v20.0/sobjects/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

Orquestador

Gestión de microservicios

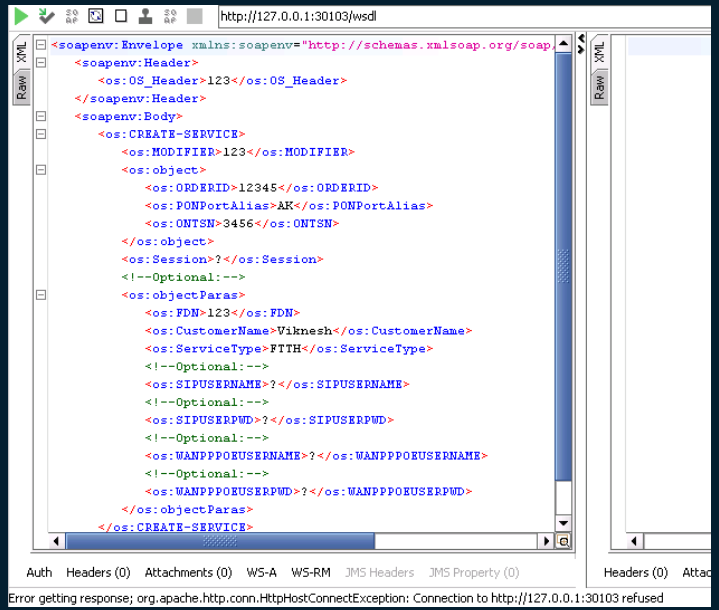
“En un ambiente de microservicios los dominios permanecen lo mas separado posible, sin embargo esta separación hace complicado la gestión de los mismos, un orquestador mejorará la administración y el control de las apis involucradas”

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.



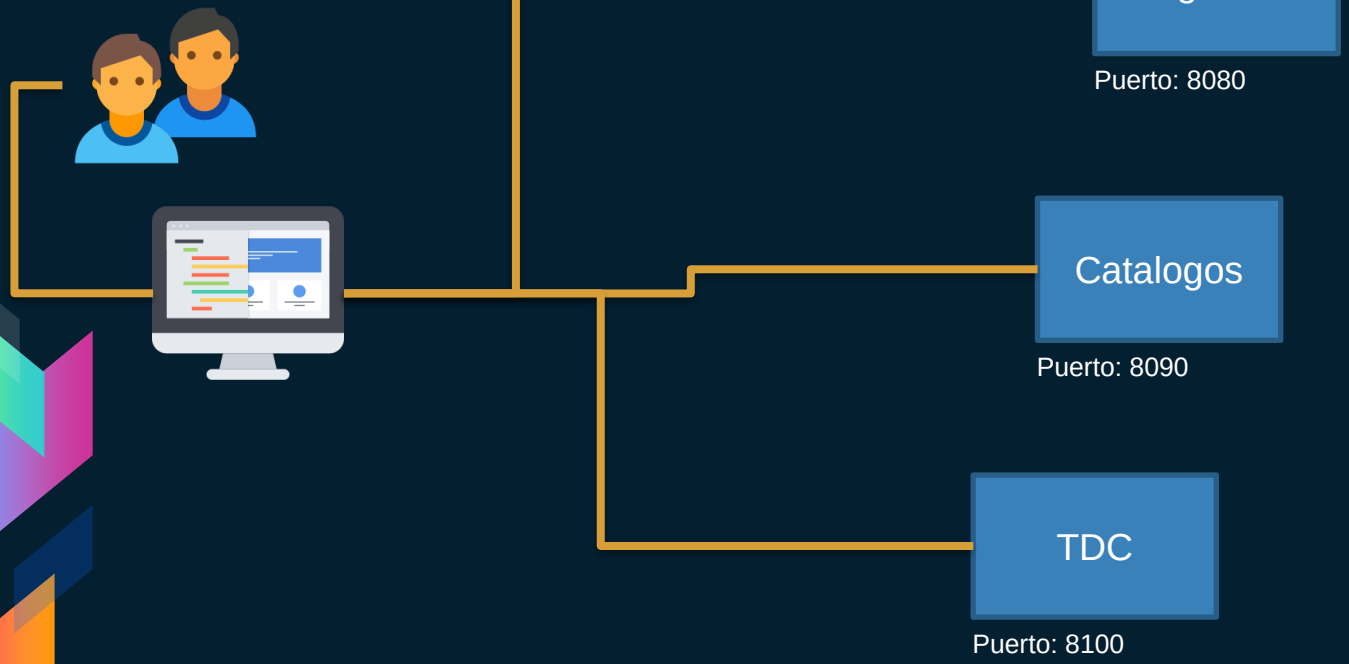
Documentación

Importa?

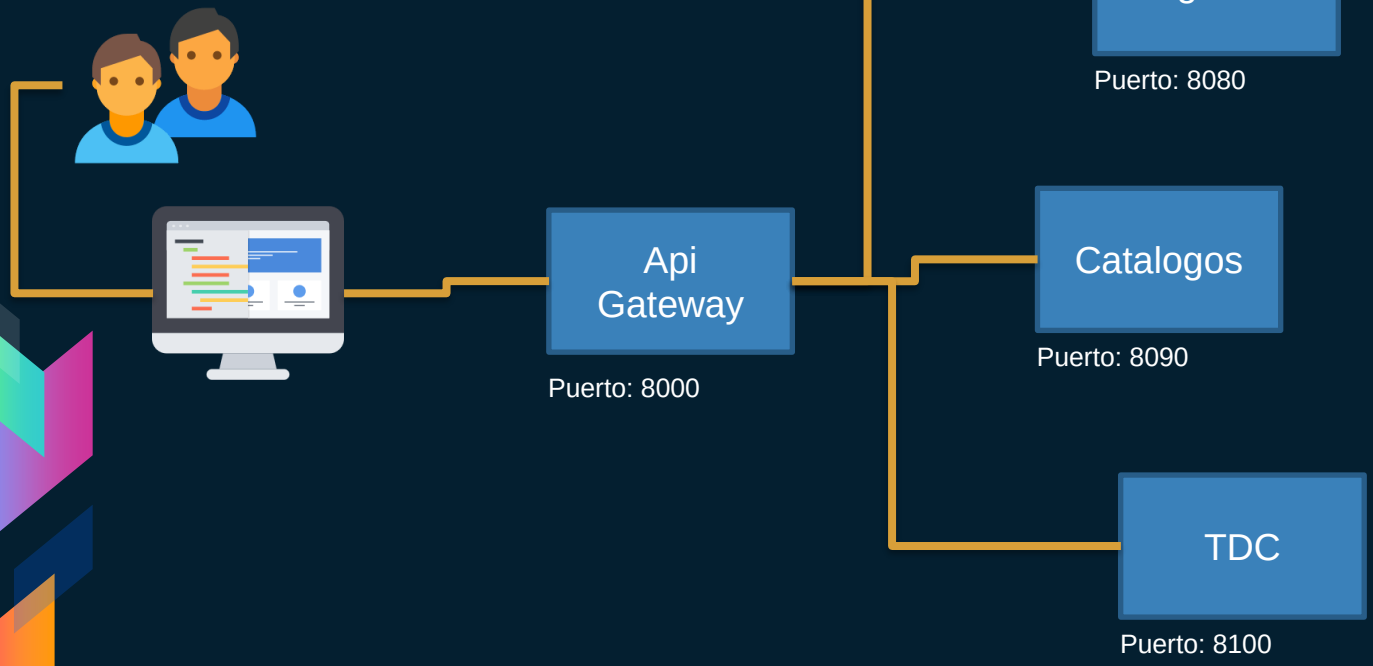
Los servicios rest no generan un contrato natural de entradas y salidas como lo hace un servicio SOAP con el wsdl.

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

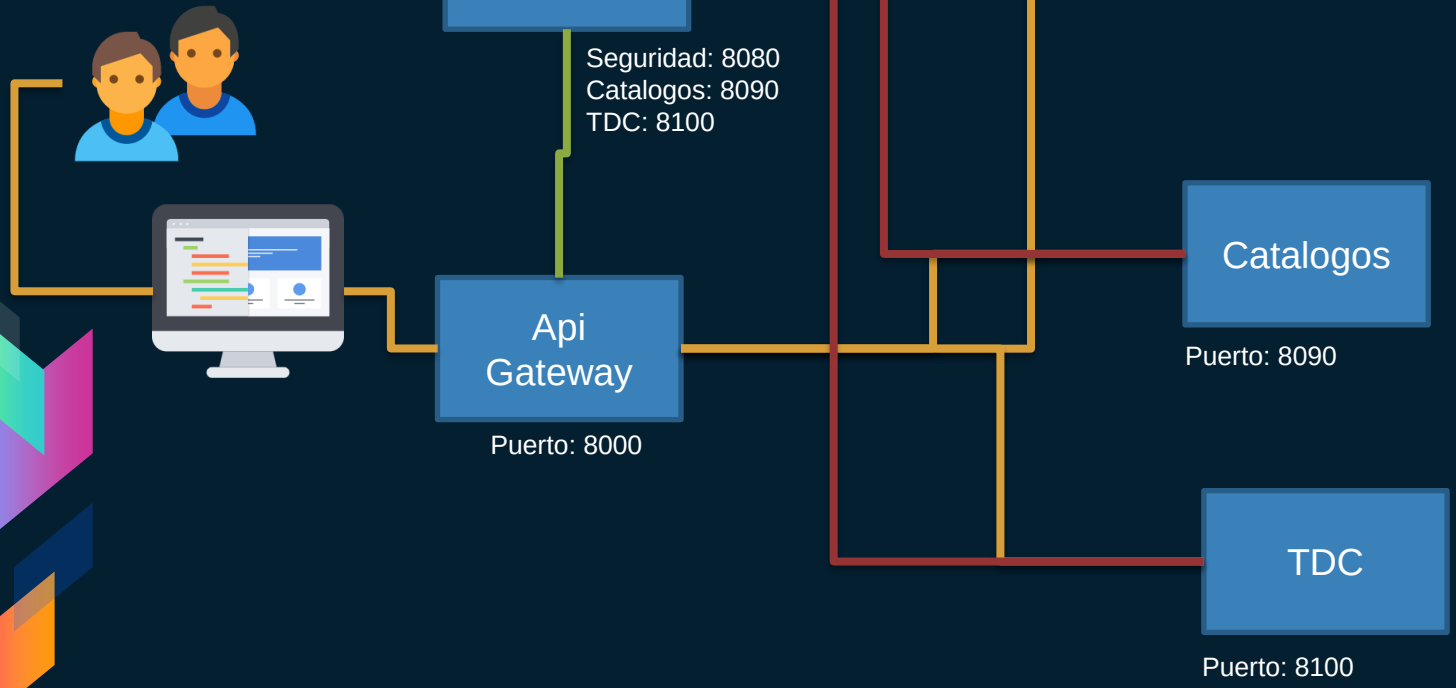
Api Gateway


Seguridad
Puerto: 8080
Catalogos
Puerto: 8090
TDC
Puerto: 8100

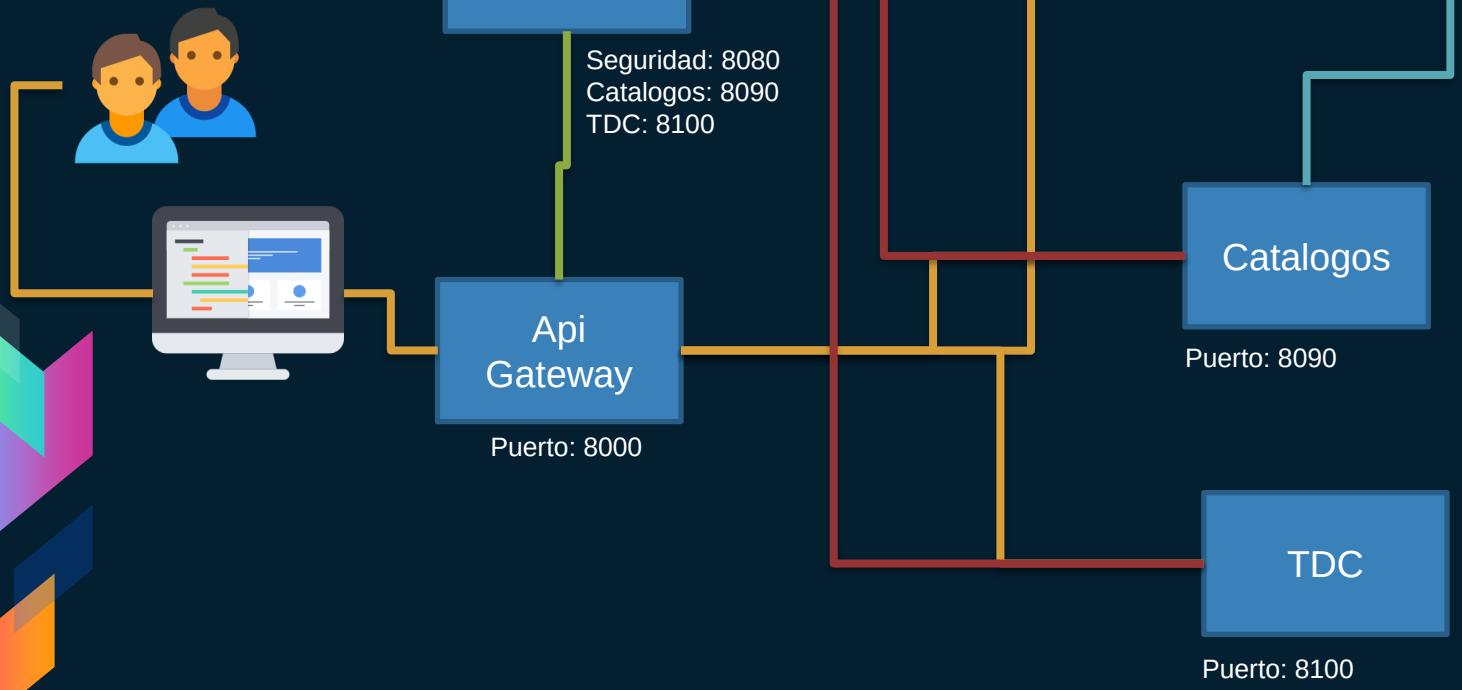
Api Gateway


Seguridad
Puerto: 8080
Api
Gateway
Puerto: 8000
Catalogos
Puerto: 8090
TDC
Puerto: 8100

Discovery


Discovery
Seguridad
Puerto: 8080
Seguridad: 8080
Catalogos: 8090
TDC: 8100
Catalogos
Puerto: 8090
Api Gateway
Puerto: 8000
TDC
Puerto: 8100

Discovery


Discovery
Seguridad: 8080
Catalogos: 8090
TDC: 8100
Seguridad
Puerto: 8080
Catalogos
Puerto: 8090
Api Gateway
Puerto: 8000
TDC
Puerto: 8100

Contenedores

Automatizar despliegue de aplicaciones

Tradicional

Tradicional

$ $ $

$ $ $

Qué tipo de servidor se requiere para esta aplicación?

Tradicional

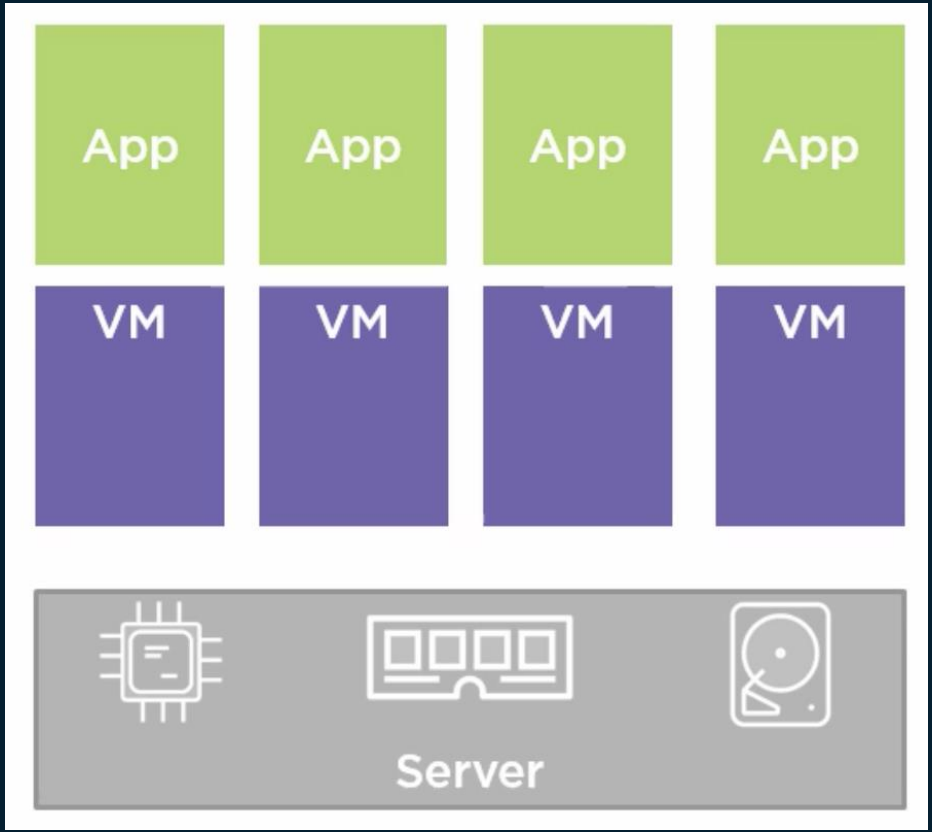
$ $ $

vmware®

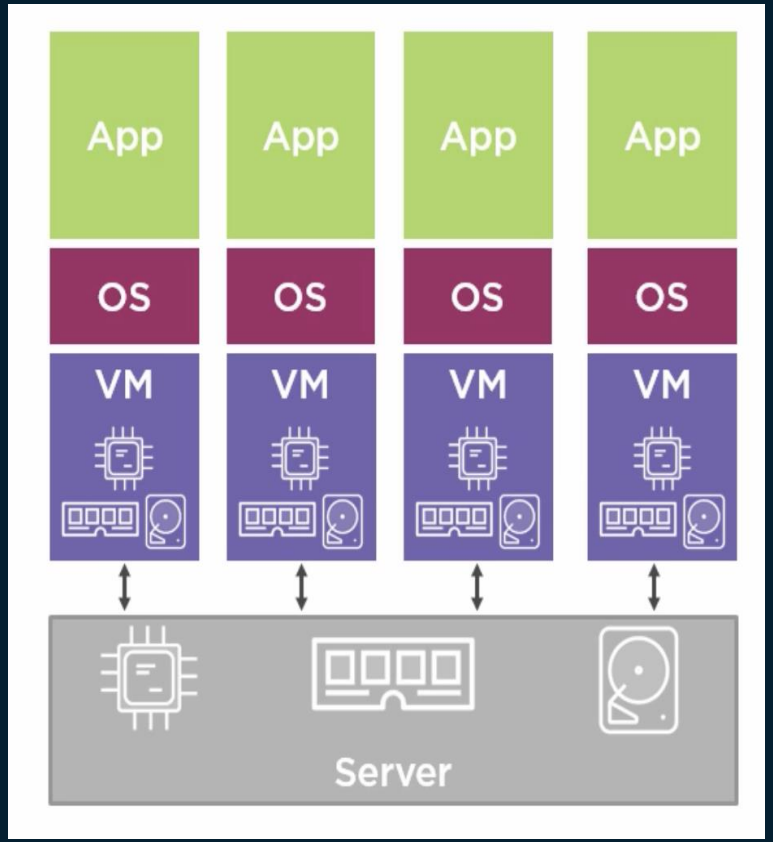
Microsoft
Hyper-V

Tradicional

El problema



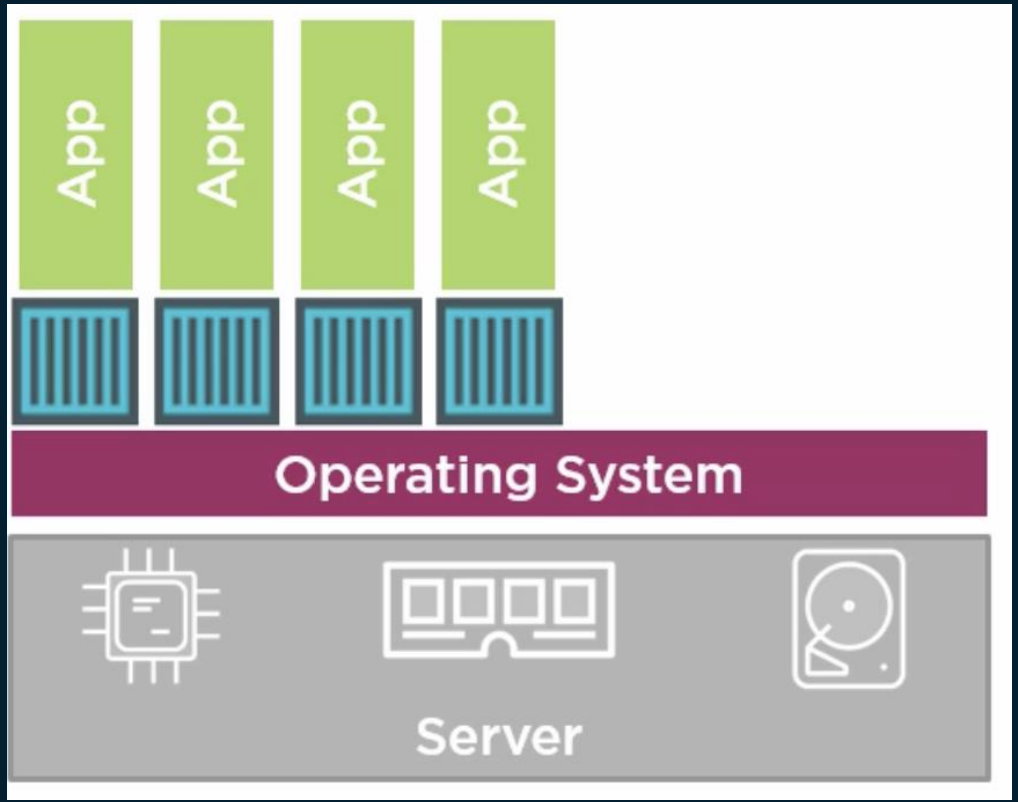
El problema



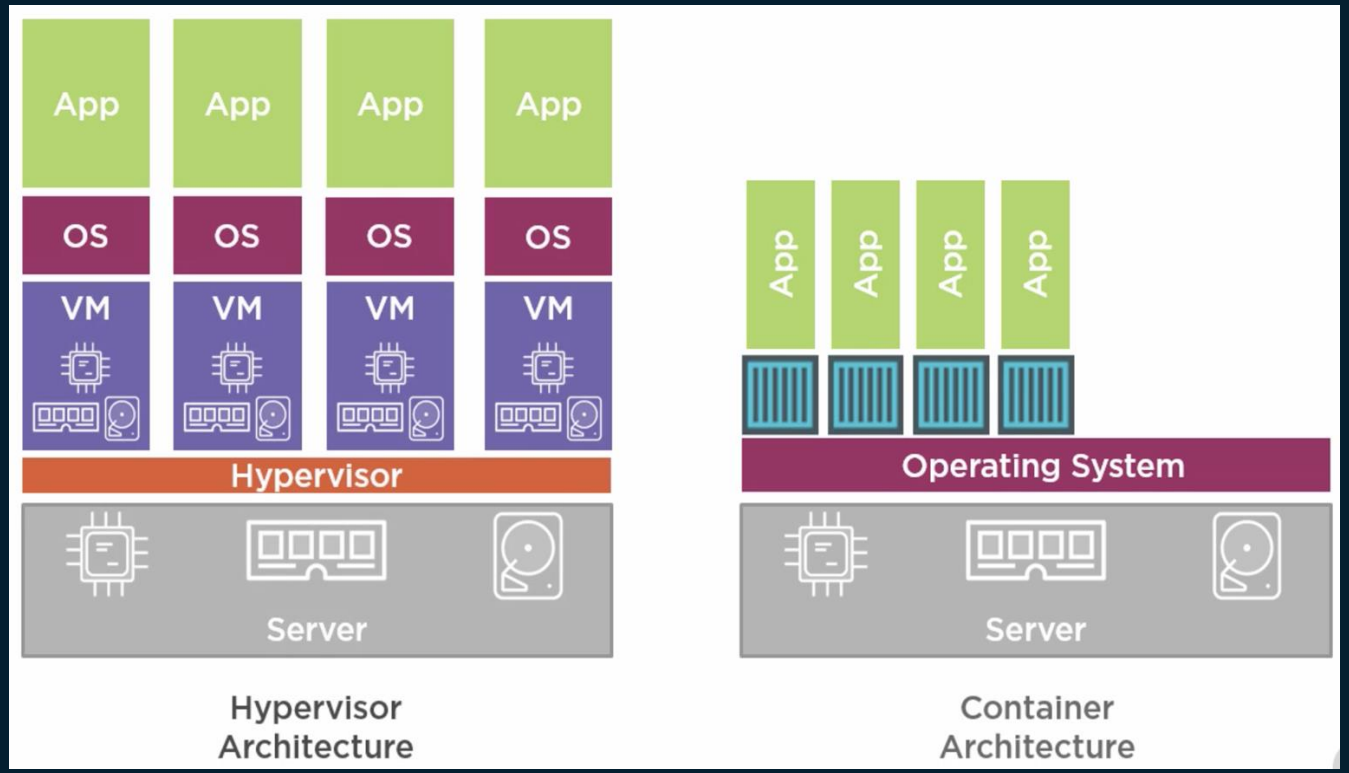
Contenedores

Contendores

Contendores

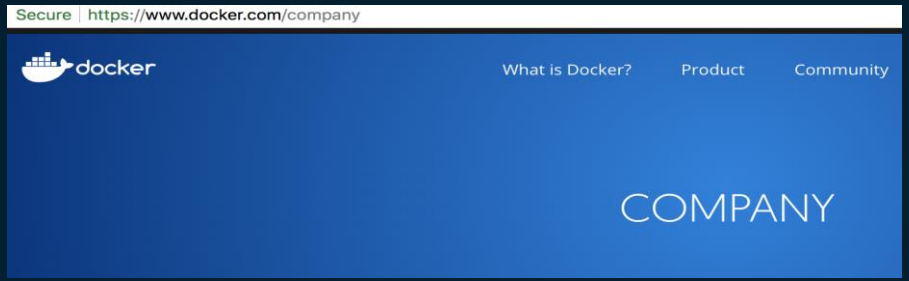


Contenedores



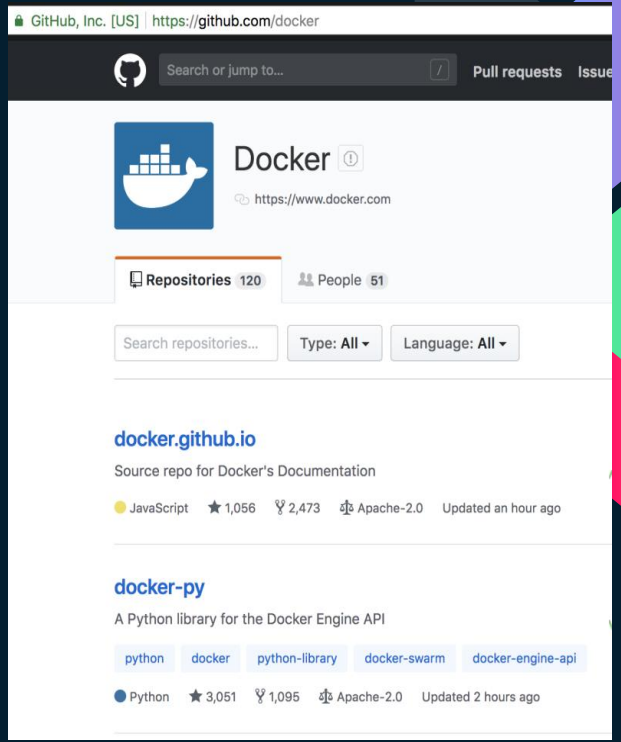
Qué es Docker?

Docker Company

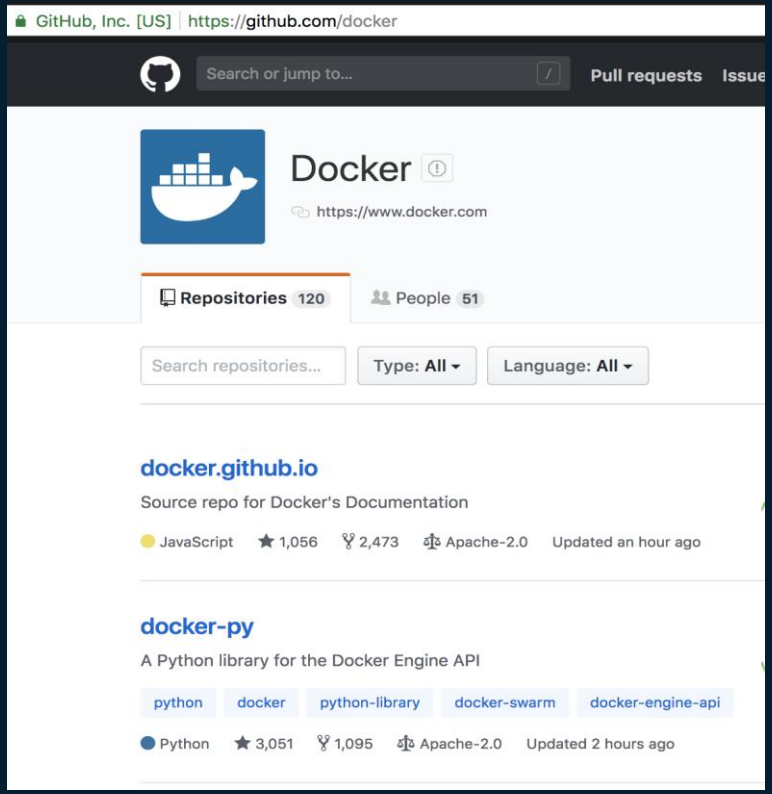


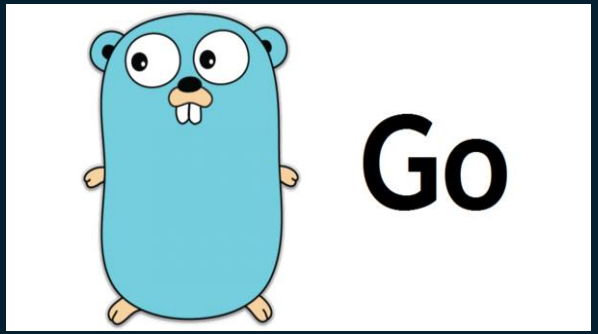
Open Container Initiative

Docker Project



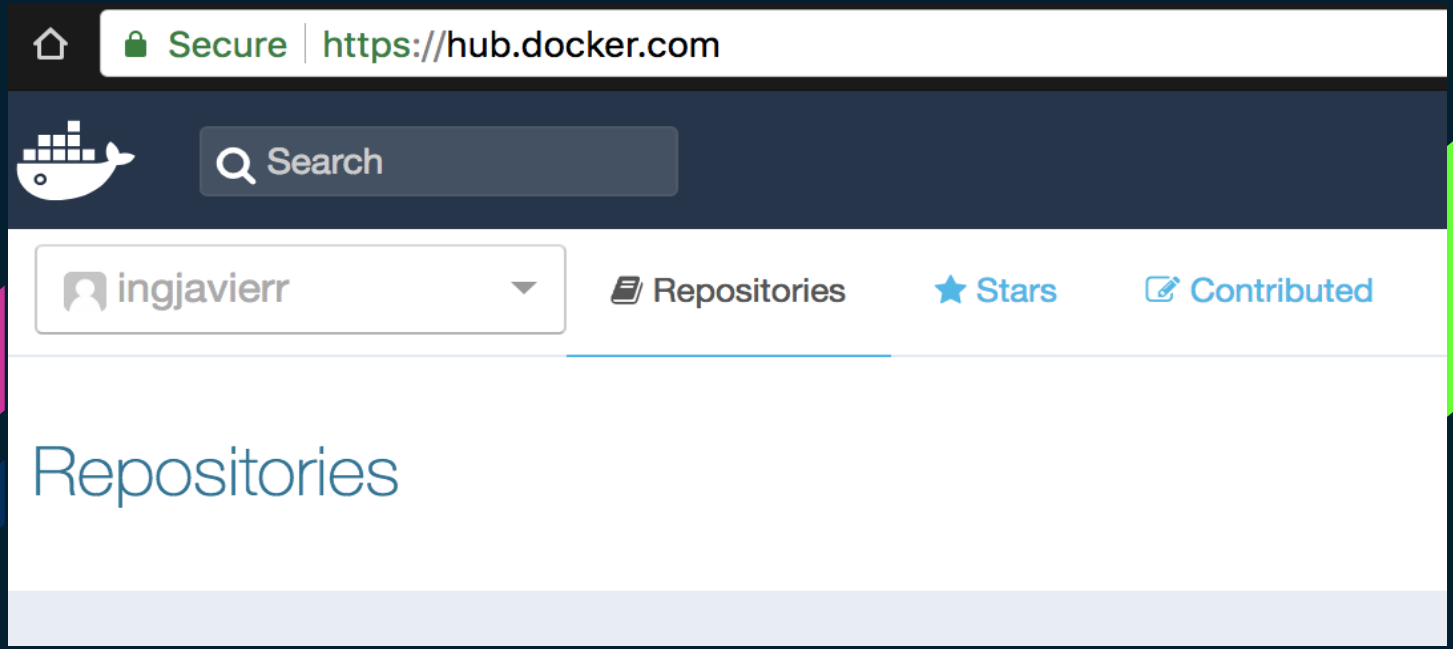
Docker project





Docker project

Docker Hub

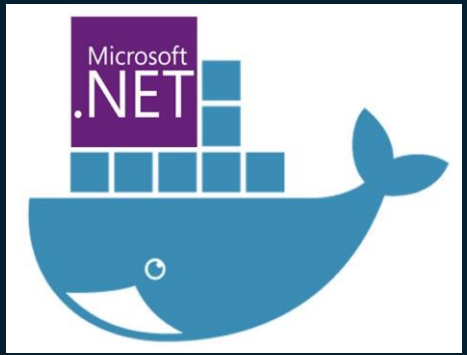


Open container initiative

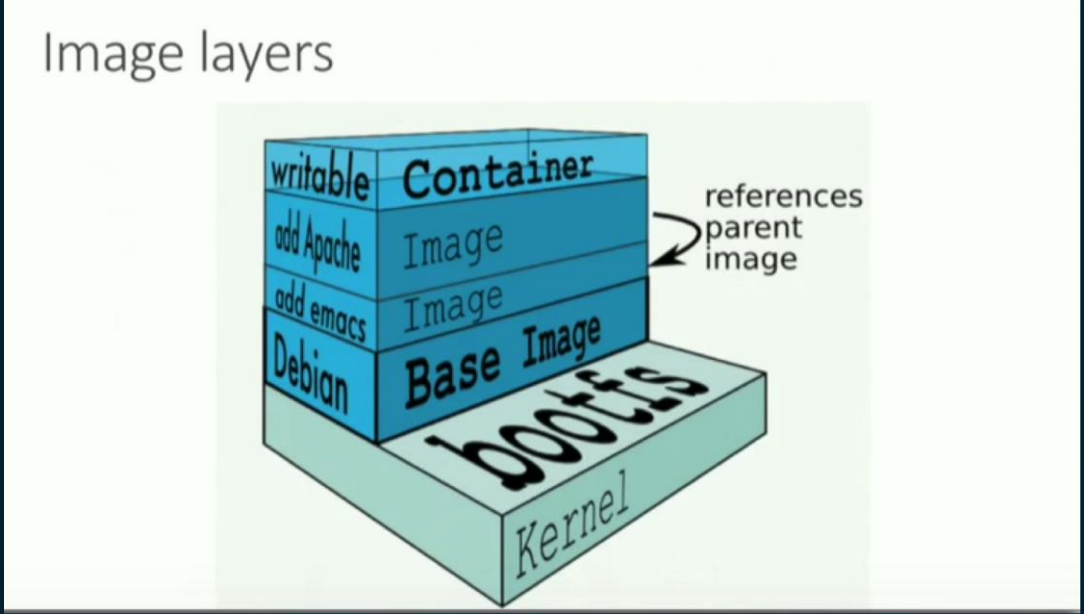
Que es un contenedor?

Es una instancia de un servidor en su mínima expresión.

Qué puede vivir en contenedor?



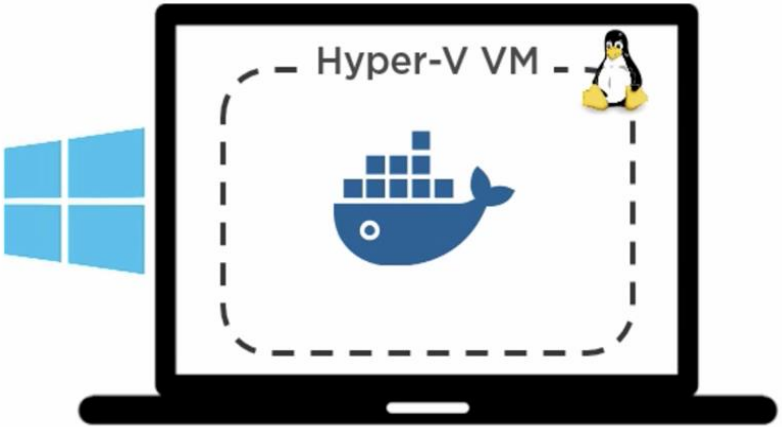
Docker image layers



Docker en windows

Pre-reqs

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



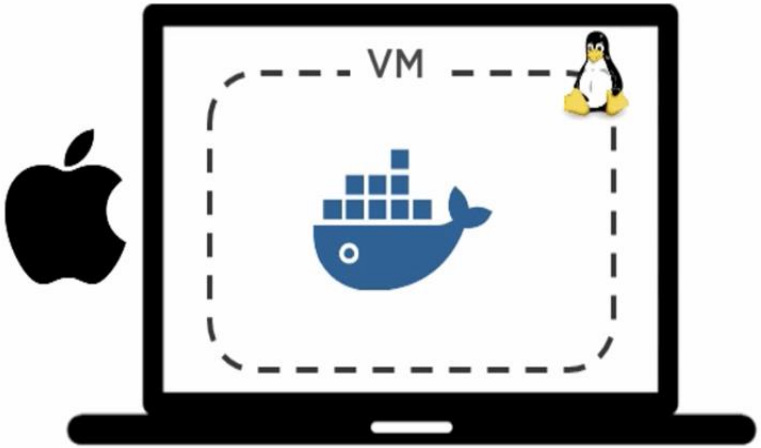
Use cases

- Test
- Dev
- ~~Production~~

Docker en MAC

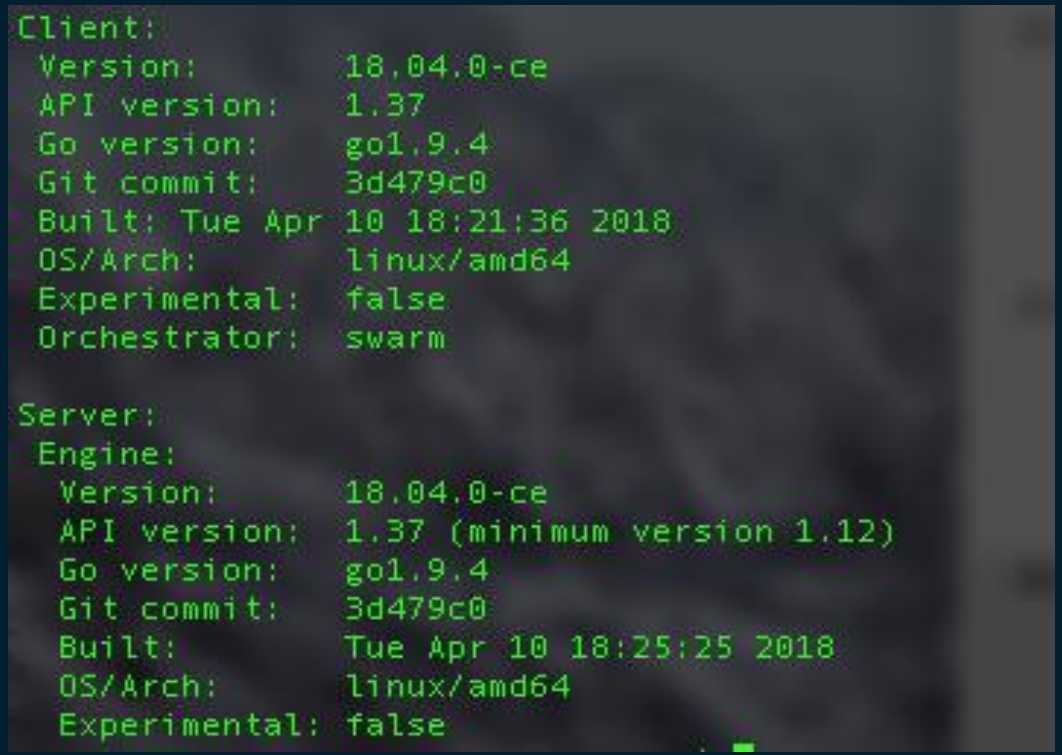
HyperKit
- https://github.com/docker/hyperkit

DataKit
- https://github.com/docker/datakit



Moby Linux
- Based on Alpine Linux

Docker en linux (docker version)



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
```

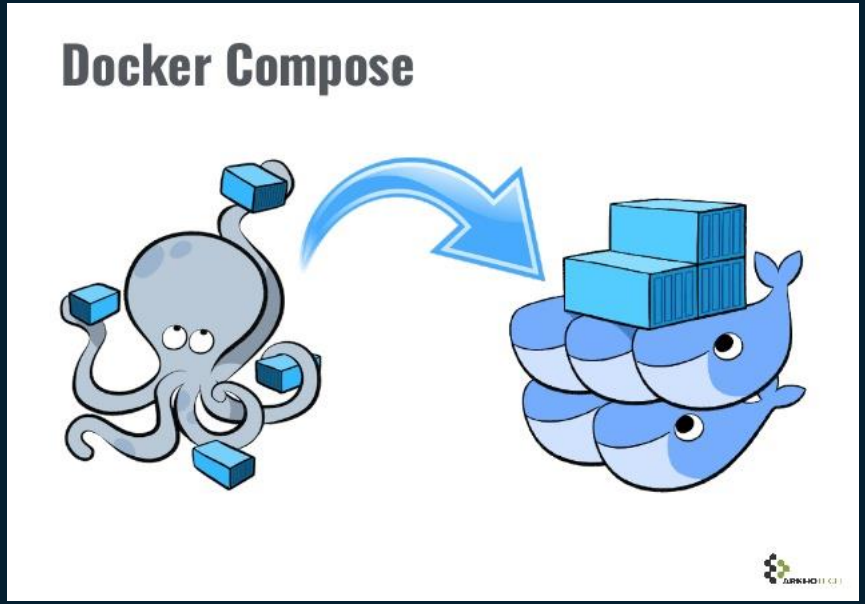
Dockerfile

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
```

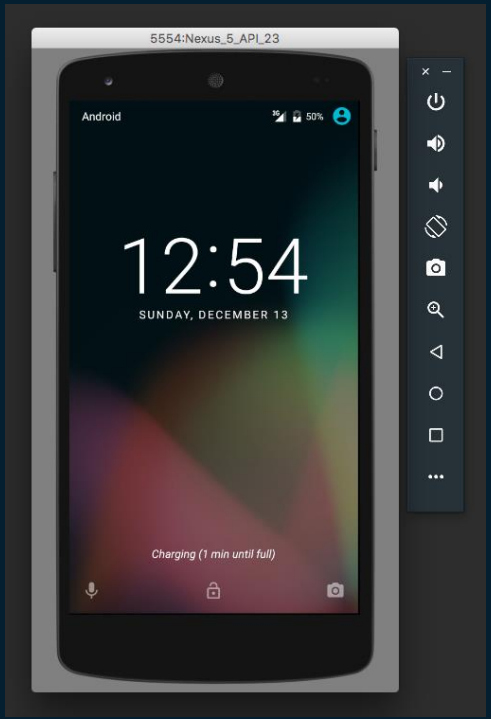
```
docker build -t hola .
docker run -d -p “8090:80” hola
```

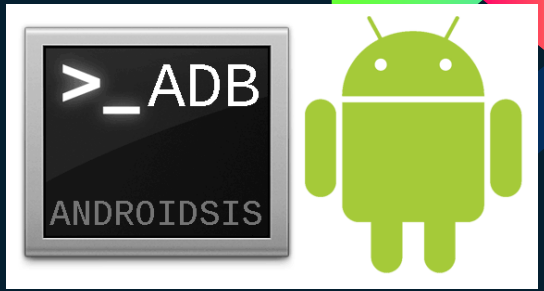
Docker compose



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

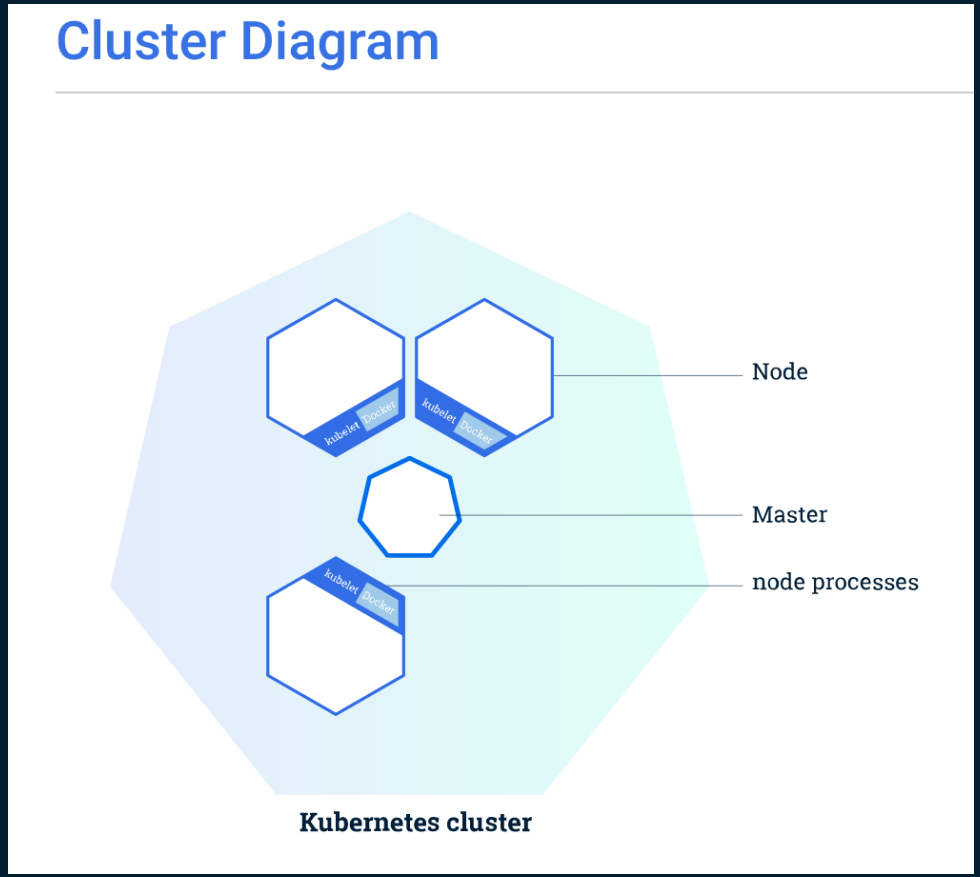
Donado a la Linux Fundation en 2014

Desarrollado en Go

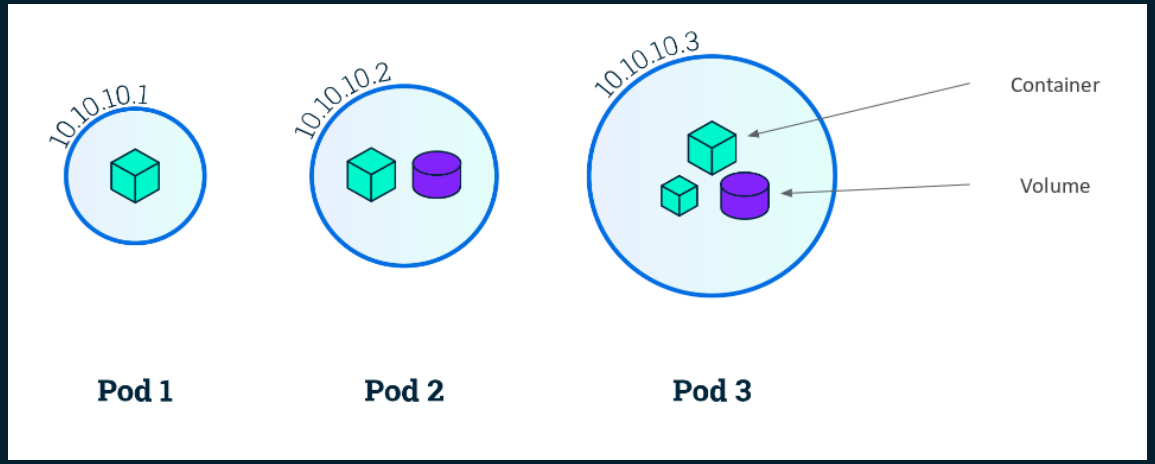
https://github.com/kubernetes

Kubernetes (k8s)

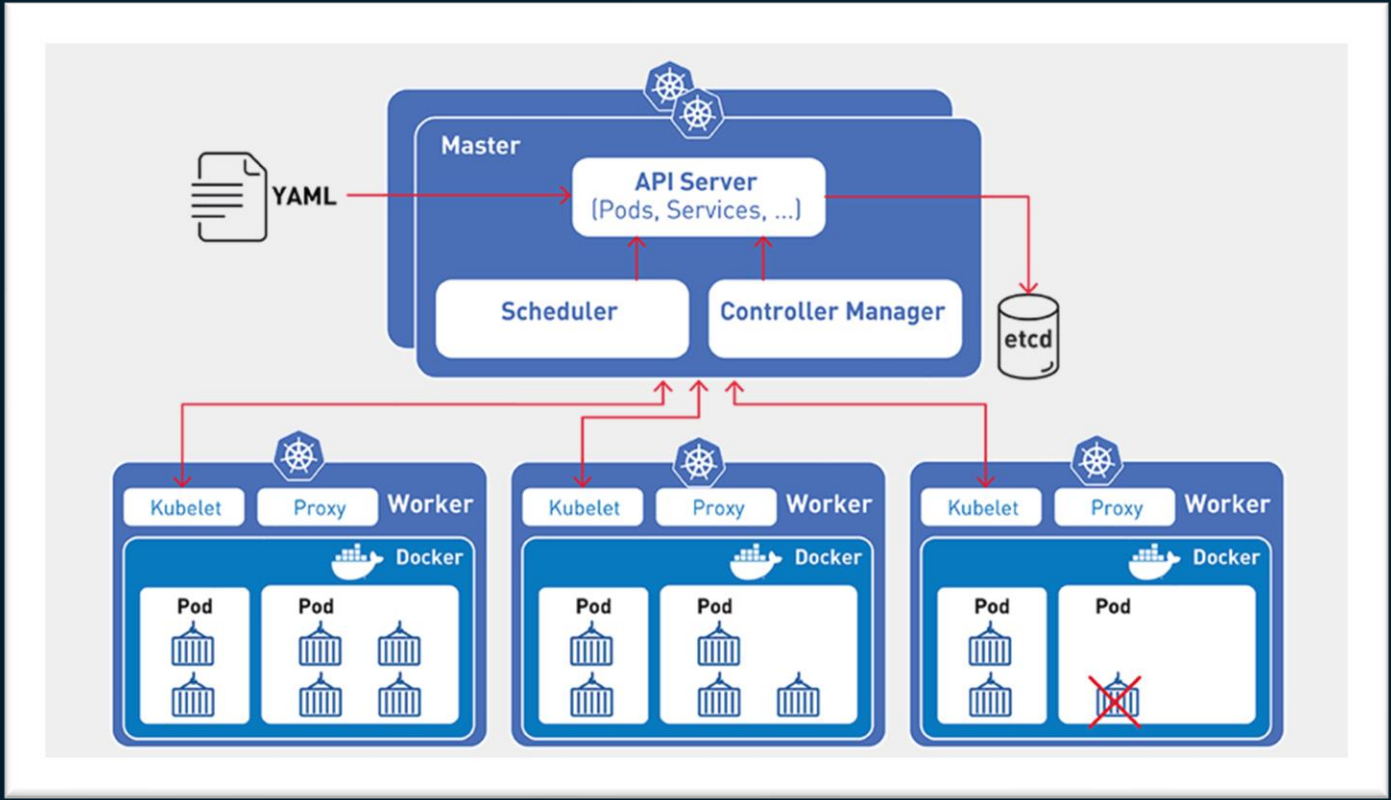
Arquitectura



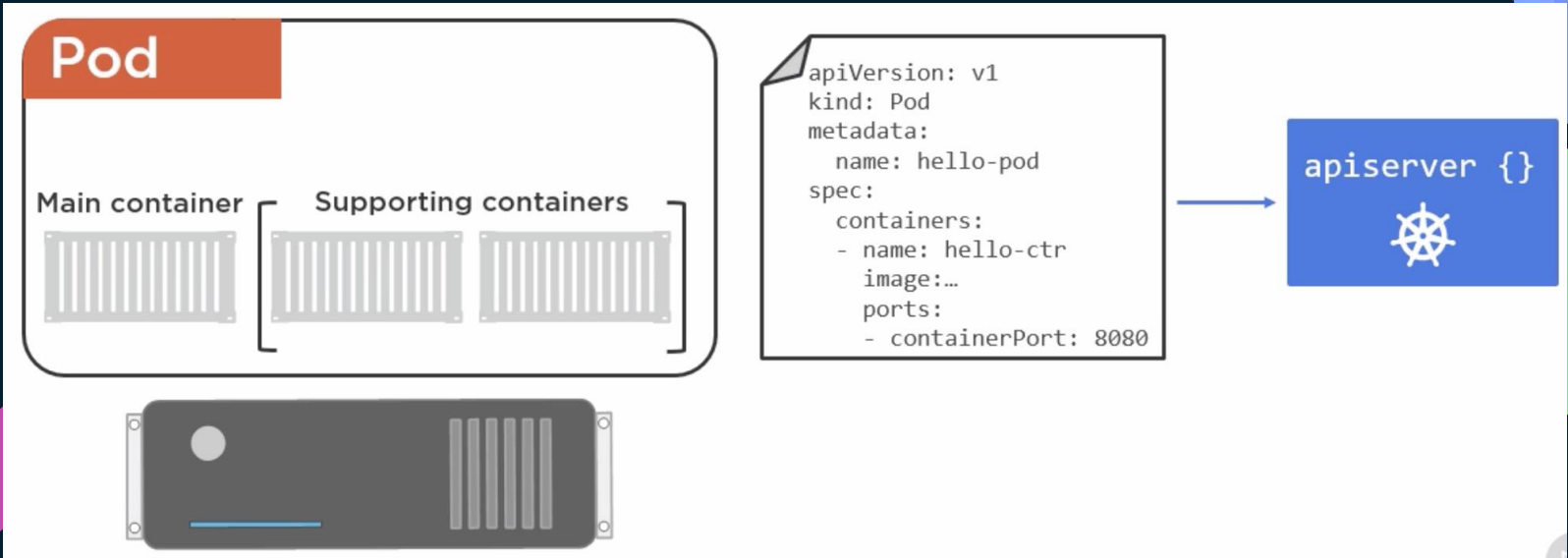
Arquitectura



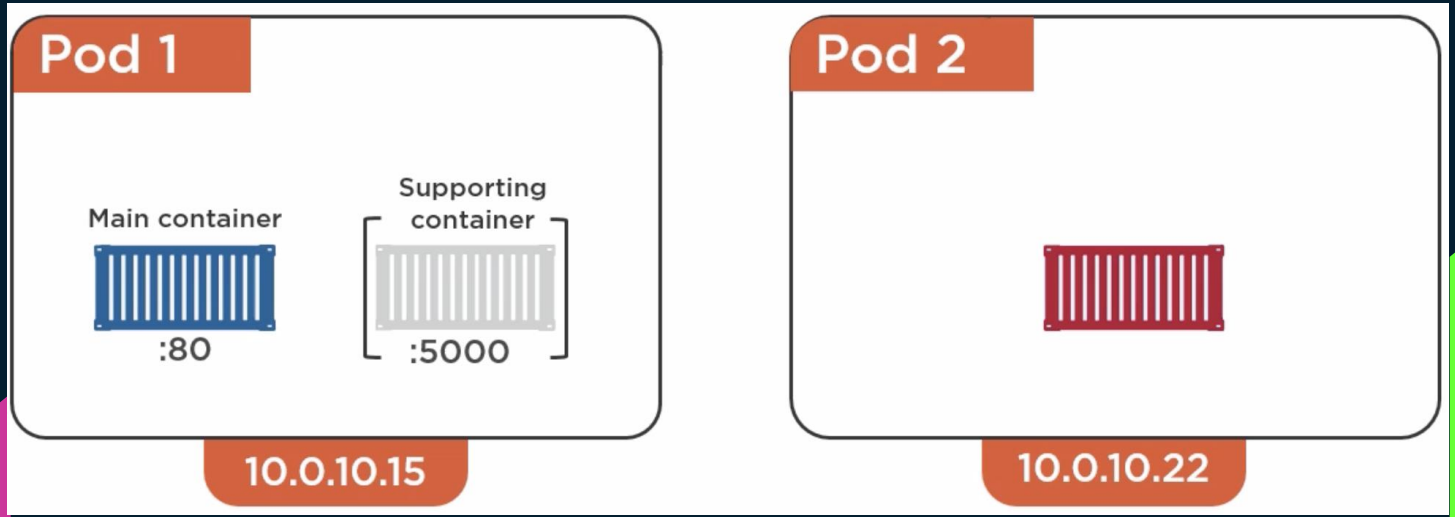
Arquitectura



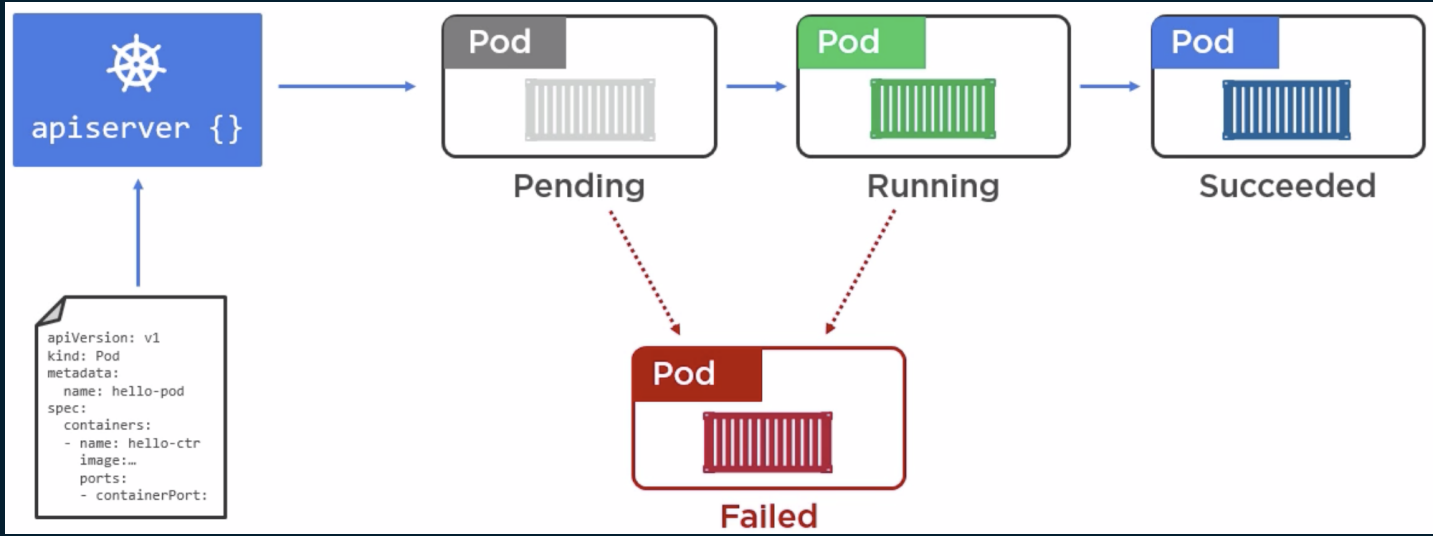
Pods


Pod
Main container
Supporting containers
apiVersion: v1
kind: Pod
metadata:
name: hello-pod
spec:
containers:
- name: hello-ctr
image:…
ports:
- containerPort: 8080
apiserver {}

Pods


Pod 1
Main container
:80
Supporting container
:5000
10.0.10.15
Pod 2
10.0.10.22

Pod lifecycle



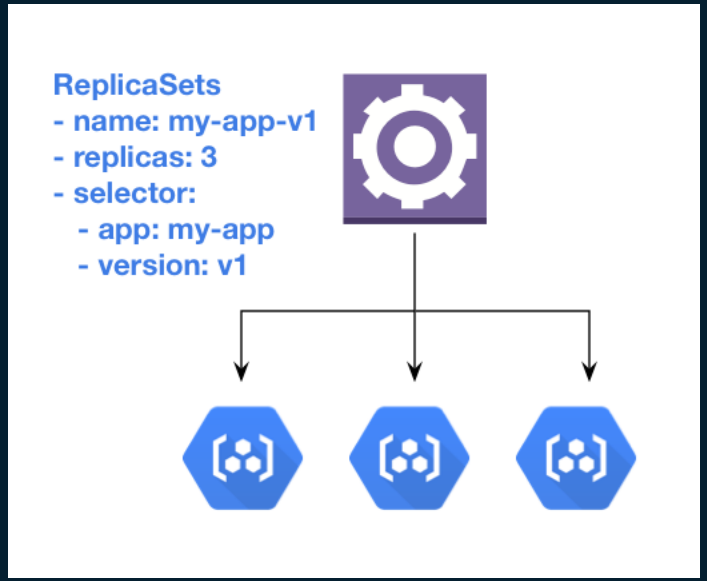
Pods

- $ 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

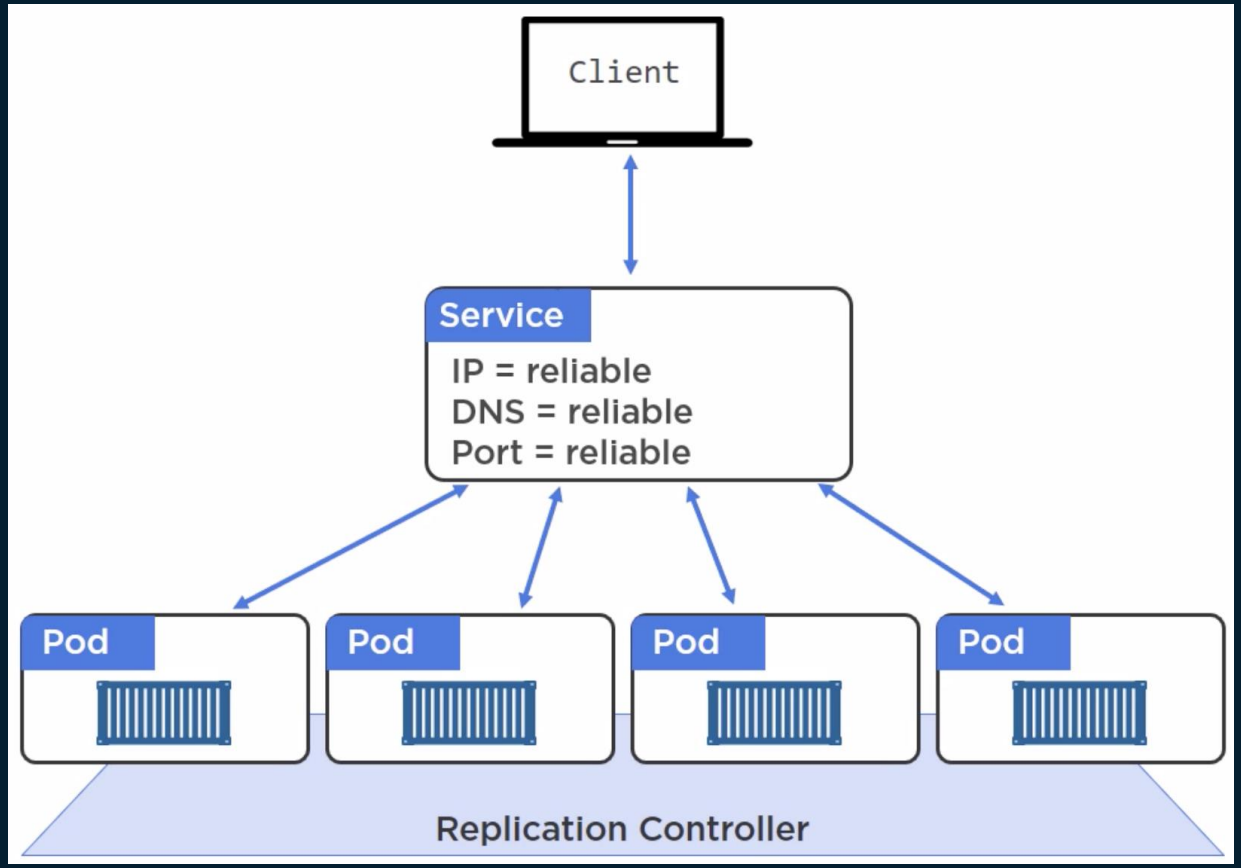
Manifiesto rc.yml

- $ 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


Client
Service
IP = reliable
DNS = reliable
Port = reliable
Pod
Pod
Pod
Pod
Replication Controller

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