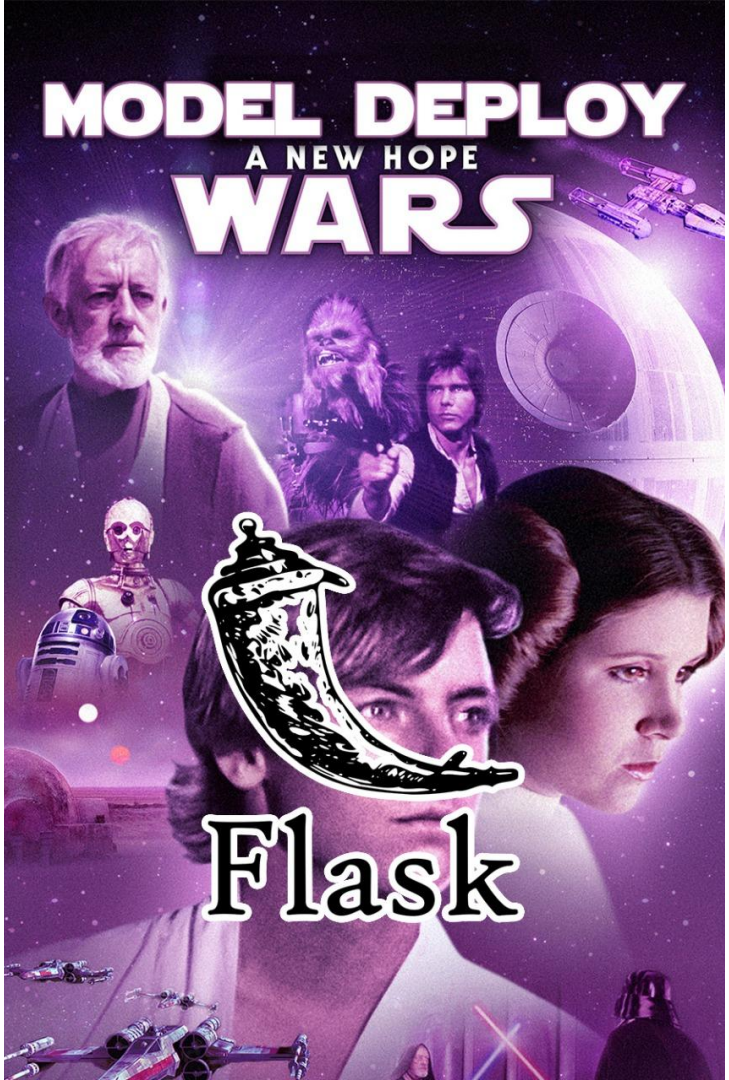
Deploy de Modelos

Parte 3: Tensorflow Serving en Kubernetes


MODEL DEPLOY
A NEW HOPE
WARS
Flask


MODEL DEPLOY
THE FRONT STRIKES BACK
WARS
JS
TensorFlow
.JS


MODEL DEPLOY
RETURN OF THE DOCKER
WARS

Temario

- Tensorflow Serving
  - Qué es?
- Docker
  - Motivación
  - Qué es un container?
- Ejemplo práctico docker + TF Serving
- Kubernetes
  - Introducción Teórica
  - Cómo montar en GKE?
- Ejemplo Práctico GKE Cluster + TF Serving

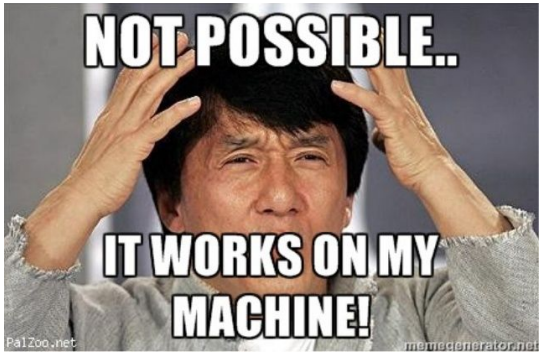
Tensorflow Serving

Qué es Tensorflow Serving?

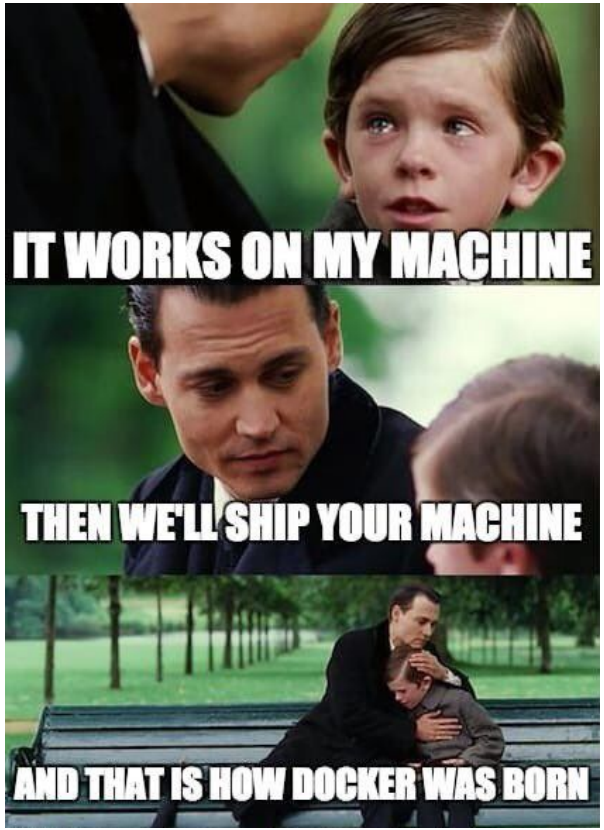
- Sistema de publicación flexible y de alto rendimiento para modelos de aprendizaje automático, diseñado para entornos de producción.
- Admite modelos de tensorflow y algunas otras plataformas.
- Client API: gRPC / REST
- Docker image de TF Serving.

Docker

Docker: Motivation



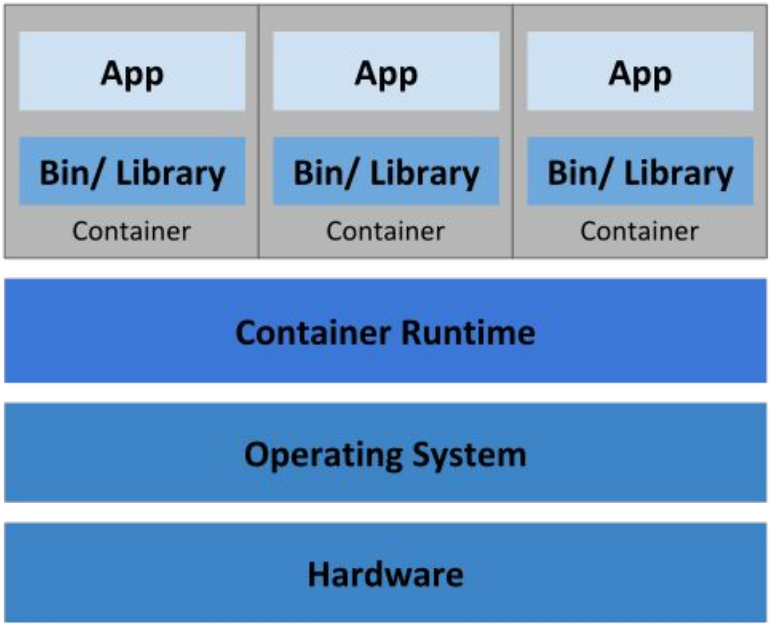
- En mi computadora si funciona
- Si pero no le vamos a dar tu computadora al cliente



Docker

- Virtualización a nivel de Sistema Operativo.
- Aplicaciones autocontenidas: Incluyen software / librerías / dependencias.
- Se pueden levantar varios contenedores con la misma aplicación.
- Se puede correr en mi PC, la de un compañero o en producción.

Docker


App
App
App
Bin/ Library
Bin/ Library
Bin/ Library
Container
Container
Container
Container Runtime
Operating System
Hardware

Container

- Paquete de software que contiene los servicios esenciales para que la aplicación contenida en este paquete pueda funcionar de manera adecuada.
- Sin necesidad de tener una dependencia directa en el sistema operativo (muy versátiles y livianos).
- Contiene exclusivamente los servicios necesarios para que la aplicación que se empaqueta pueda correr sin ningún problema.
- Se puede replicar de acuerdo a las necesidades del usuario.

Docker Example

Dockerfile

```
FROM node:12-alpine

WORKDIR /app

COPY . .

RUN yarn install --production

CMD ["node", "src/index.js"]
```

Terminal

```
docker build --tag getting-started .
```

```
docker run --name mydockerapp -p 8080:8080
getting-started
```

Ejemplo Práctico

TF Serving con docker

Kubernetes

Qué es Kubernetes?

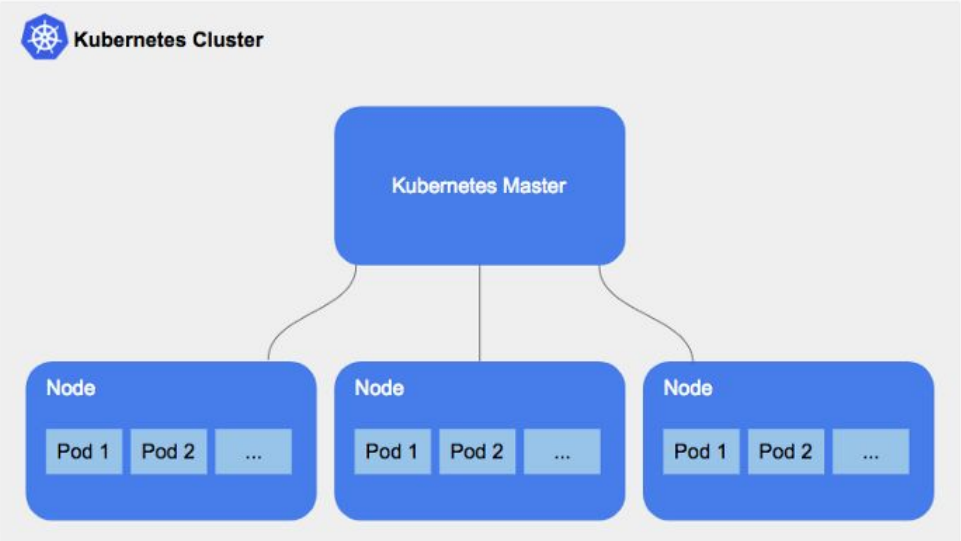
- Timonel en griego.
- Orquestador de contenedores.
- Manejar varios / muchos a la vez.
- Optimización de recursos.
- Múltiples nodos de hardware puedan mantener múltiples aplicaciones basadas en containers en tiempo real.
- Creado por google y donado a la Linux Foundation (open source).

Conceptos básicos de k8s

Cluster: Conjunto de VMs. Cada VM es un nodo. Existe siempre un nodo master.

Node Pool: Cada cluster puede tener varias pools de nodos, con distintas máquinas virtuales.

Pod: Mínima unidad en k8s. Conjunto de 1 o más contenedores (almacenamiento y recursos de red compartidos).



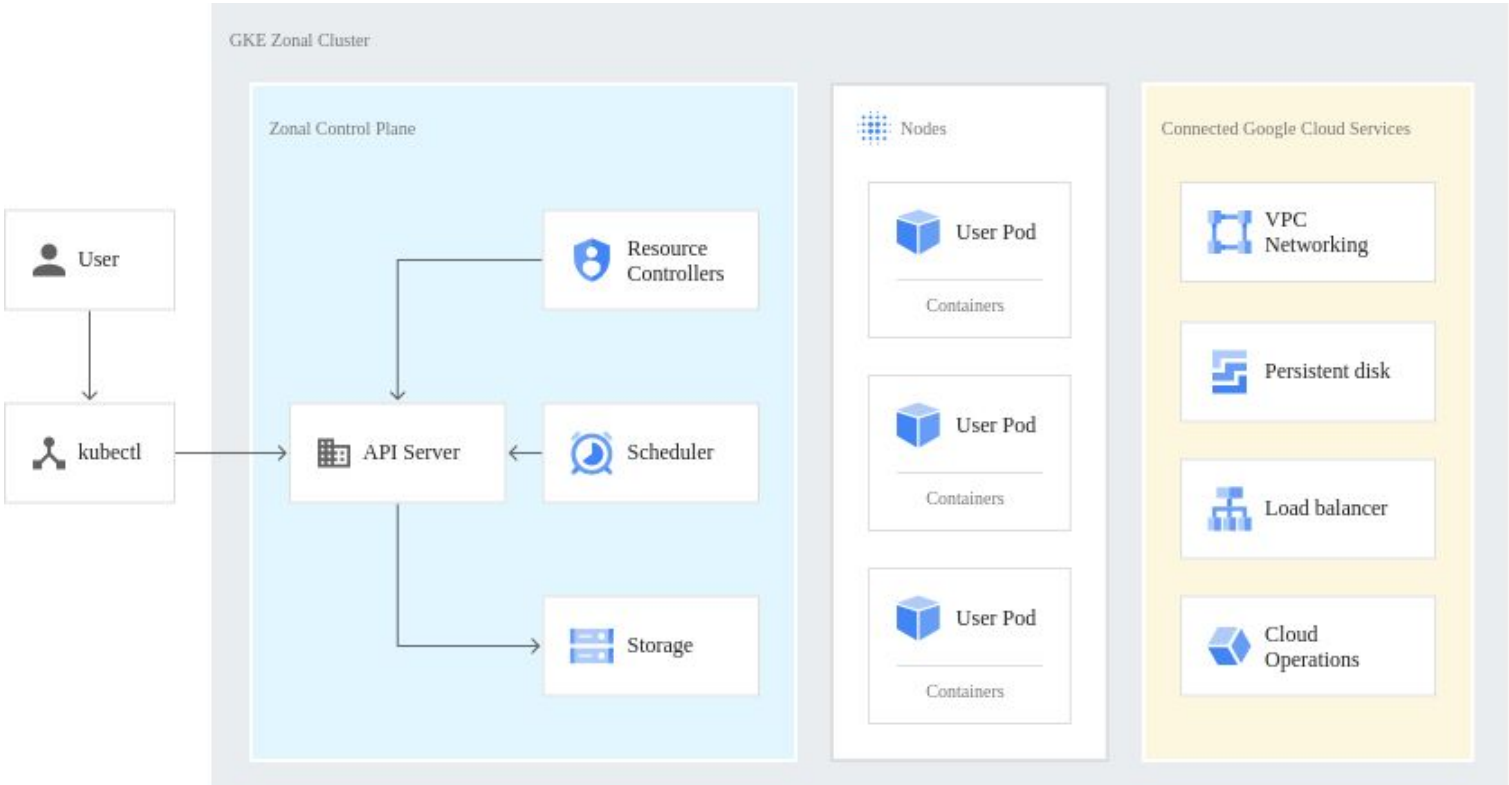
Conceptos Básicos

Deployment: conjunto de órdenes declaradas que se le dan a nuestro cluster de Kubernetes. Todos los ReplicaSets y los Pods se alinean a lo que diga un Deployment.

ReplicaSet: Reglas de cuántos pods debe haber corriendo.

Service: Forma abstracta de exponer pods a un servicio de red (maneja direcciones IP, nombre DNS y load balancer entre los pods)

Arquitectura en Google (GKE)



Ejemplo Práctico

TF Serving con Kubernetes

Algunos Tips

- Levantar model config desde un bucket de S3 (AWS) o GCS (GCP).
- Node pools con GPU (para modelos que necesiten).
- Autoscaling de Nodepool y de réplicas.
- Rolling Update

Preguntas?


GRACIAS
VUELVA PRONTO
GENERADORMEMES.COM