DESPLIEGUE DE MODELOS DE MACHINE LEARNING EN ENTORNOS DE PRODUCCIÓN

Python, Flask, Gunicorn, Docker y Amazon Web Services

GABRIEL VILLACIS

SOCIEDAD ECUATORIANA ESTADÍSTICA

Agenda

Introducción al Despliegue de Modelos de Machine Learning

Implementación vía REST API

Contenerización con Docker

Despliegue en Amazon Web Services

GABRIEL VILLACIS

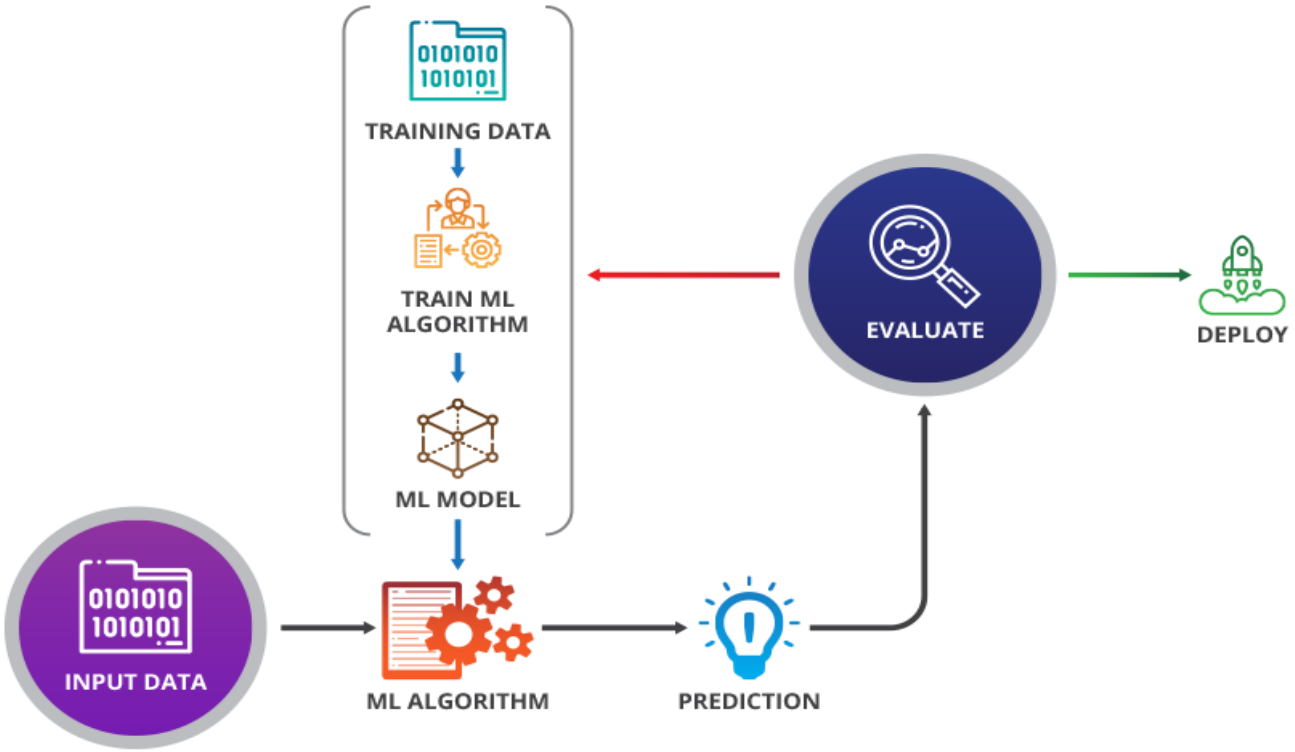
- Ingeniero en Sistemas Computacionales

- Arquitecto, Desarrollador, Capacitador y Conferencista de Software

@gab_villacis

gabriel-villacis

ML: PROCESO DE DESARROLLO/EXPERIMENTACIÓN



Trabajo de semanas con un científico de datos

DE QUÉ SE TRATA EL DESPLIEGUE DE MODELOS DE ML?

```python
def predict(data):
  data['is_good'] = data['rating'] > 3
  return model.predict(data)
```

```r
predict <- function(data):
  data$is_good = data$rating > 3
  return predict(model, data)
```

Machine Learning

APP 1 **APP 2** **APP 3** **ML APPS**

Microsoft IIS · Apache Tomcat · node JS · N

.NET Core · Java · JS · Python

INFRASTRUCTURE

QUÉ DEBEMOS CONSIDERAR PARA DESPLEGAR MODELOS DE ML?

1. **Facilidad de integración**
   Cualquier lenguaje, cualquier motor de ejecución.

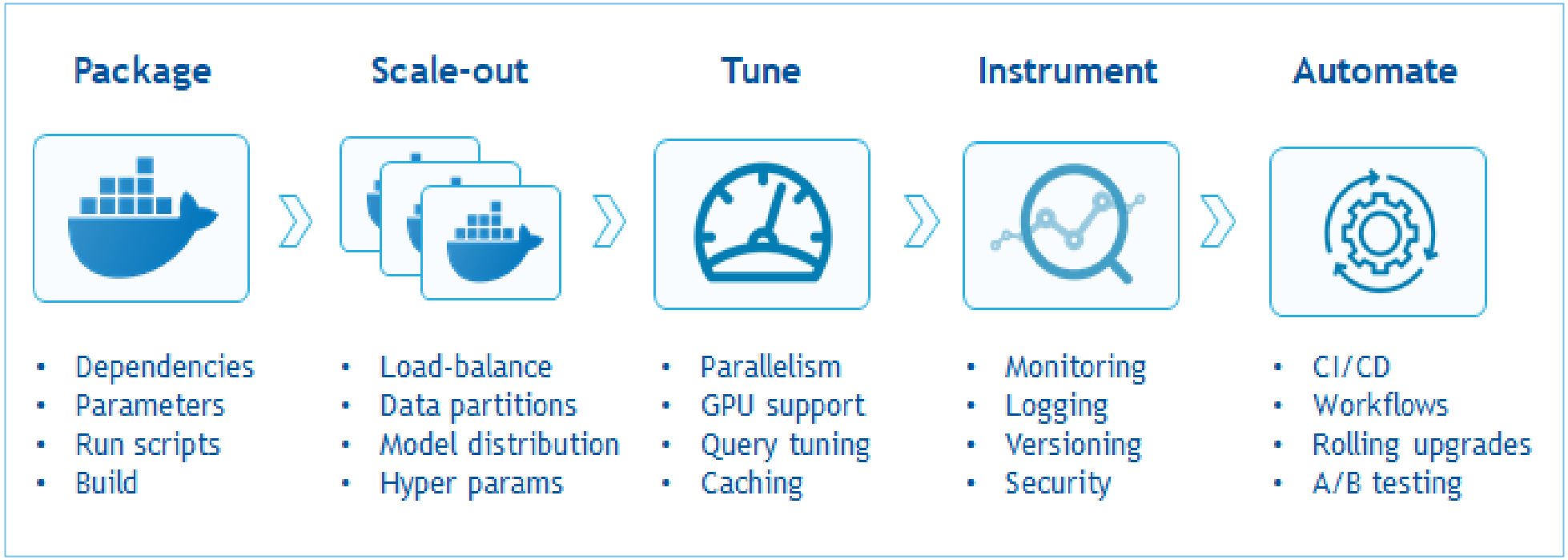
2. **Predicciones de baja latencia**
   Poder computacional, caché de predicciones frecuentes.

3. **Tolerancia a fallos**
   Replicar modelos, ejecución en clúster.

4. **Escalable**
   Subir o bajar nodos elásticamente.

5. **Facilidad en el mantenimiento**
   Actualizar fácilmente las nuevas versiones de los modelos

PROCESO DE DESPLIEGUE DE MODELOS DE ML

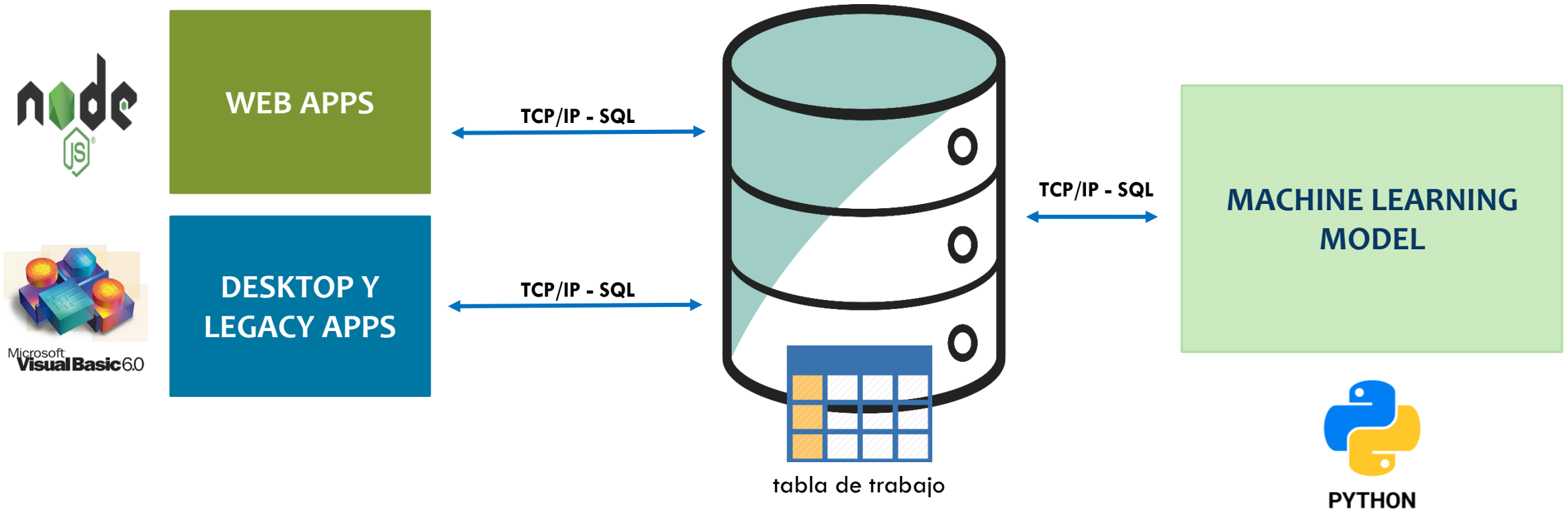


Trabajo de meses con un equipo de ingenieros de software, científicos e ingenieros de datos y DevOps

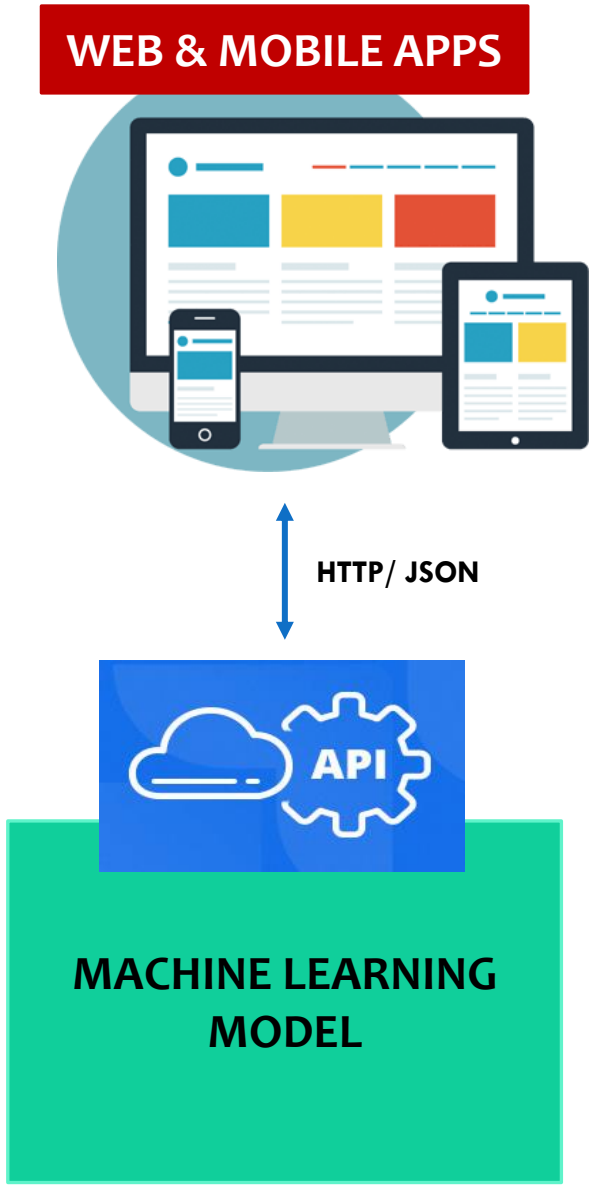
ESQUEMAS DE DISEÑO ARQUITECTÓNICO PARA INTEGRAR MODELOS DE ML

	(Esquema 1) BASE DE DATOS COMPARTIDA	(Esquema 2) REST API	(Esquema 3) STREAMING
Predicción	Batch	Sobre la marcha	Streaming
Entrega del resultado de la predicción	Batch	Vía REST API	Streaming vía cola de mensajes
Latencia en la predicción	**Alta**	Media	**Baja**
Dificultad en la gestión del sistema	Fácil	Media	Muy difícil

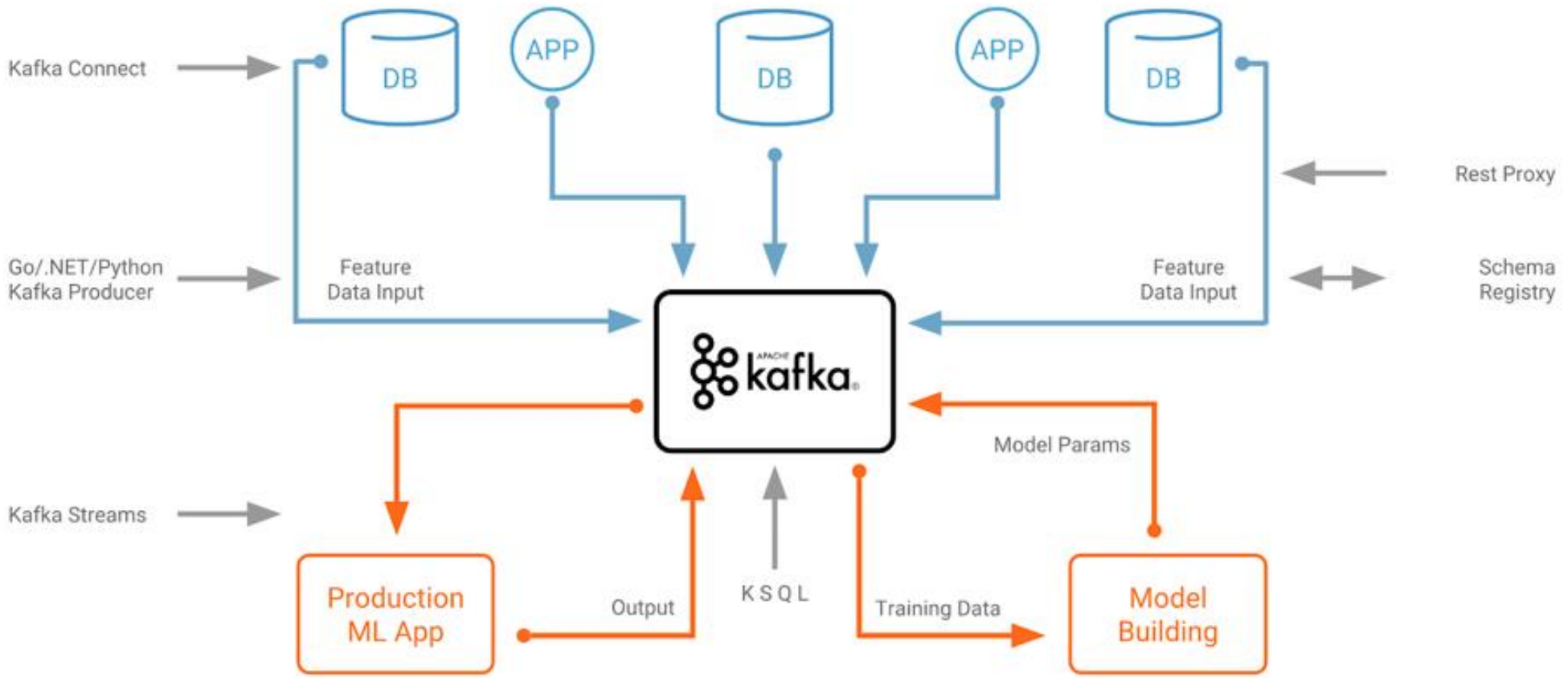
ESQUEMA: BASE DE DATOS COMPARTIDA



ESQUEMA: REST API



ESQUEMA: STREAMING



HTTP
Request/Response

gunicorn
WSGI HTTP SERVER

api/diabetes-predictions

Flask
web development,
one drop at a time

HTTP REST API

ML Model
.pkl

scikit learn

DOCKER CONTAINER

DEMO (INTRODUCCIÓN):

ARQUITECTURA DE DESPLIEGUE CON REST API

aws | Amazon Lightsail

QUÉ ES UNA REST API?

API: Application Programming Interface

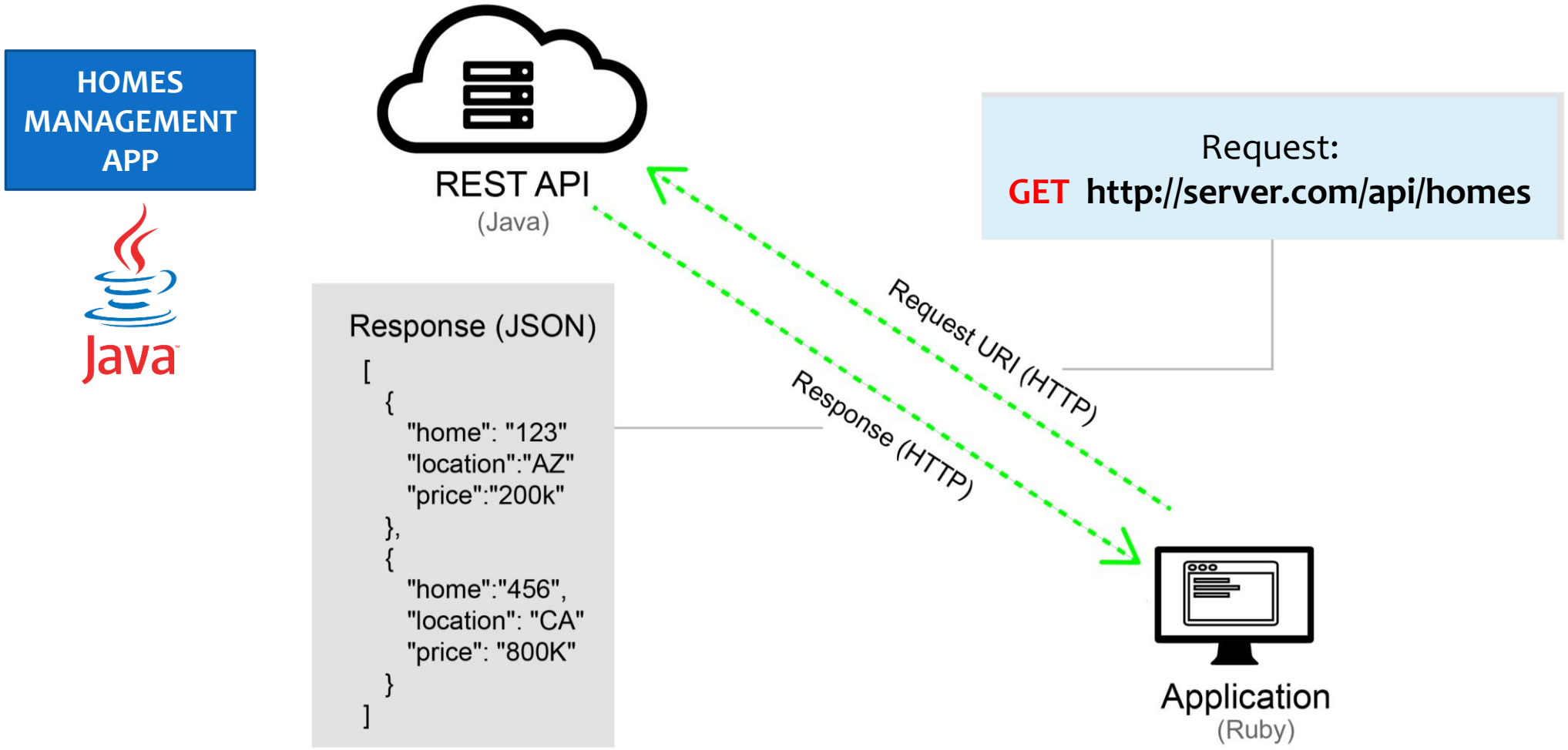
"Una **API** especifica cómo dos componentes de software deben interactuar (comunicarse entre sí)"

Componente A

Componente B

"Una REST API define un **conjunto de funciones (capacidades)** a las que otras aplicaciones pueden **ejecutar solicitudes(requests) y recibir respuestas(responses)** a través del protocolo **HTTP**.

REST API: MODELO DE COMUNICACIÓN



Algunas convenciones REST:

Usar HTTP VERBS: GET, POST, PUT, DELETE

Usar HTTP STATUS CODE: 200, 4XX, 5XX

Definir mecanismo de serialización de datos: JSON, XML.

QUÉ ES FLASK?

Es un micro framework de Python para desarrollar aplicaciones web.

Micro framework: mantiene el core simple pero extensible.

“Micro” no significa que toda su aplicación web tenga que caber en un archivo de Python.

FLASK ES…

- Fácil de programar.
- Fácil de configurar.
- Flask no te liga a decisiones de diseño prestablecidas.
- Tiene una excelente documentación.
- RESTful.

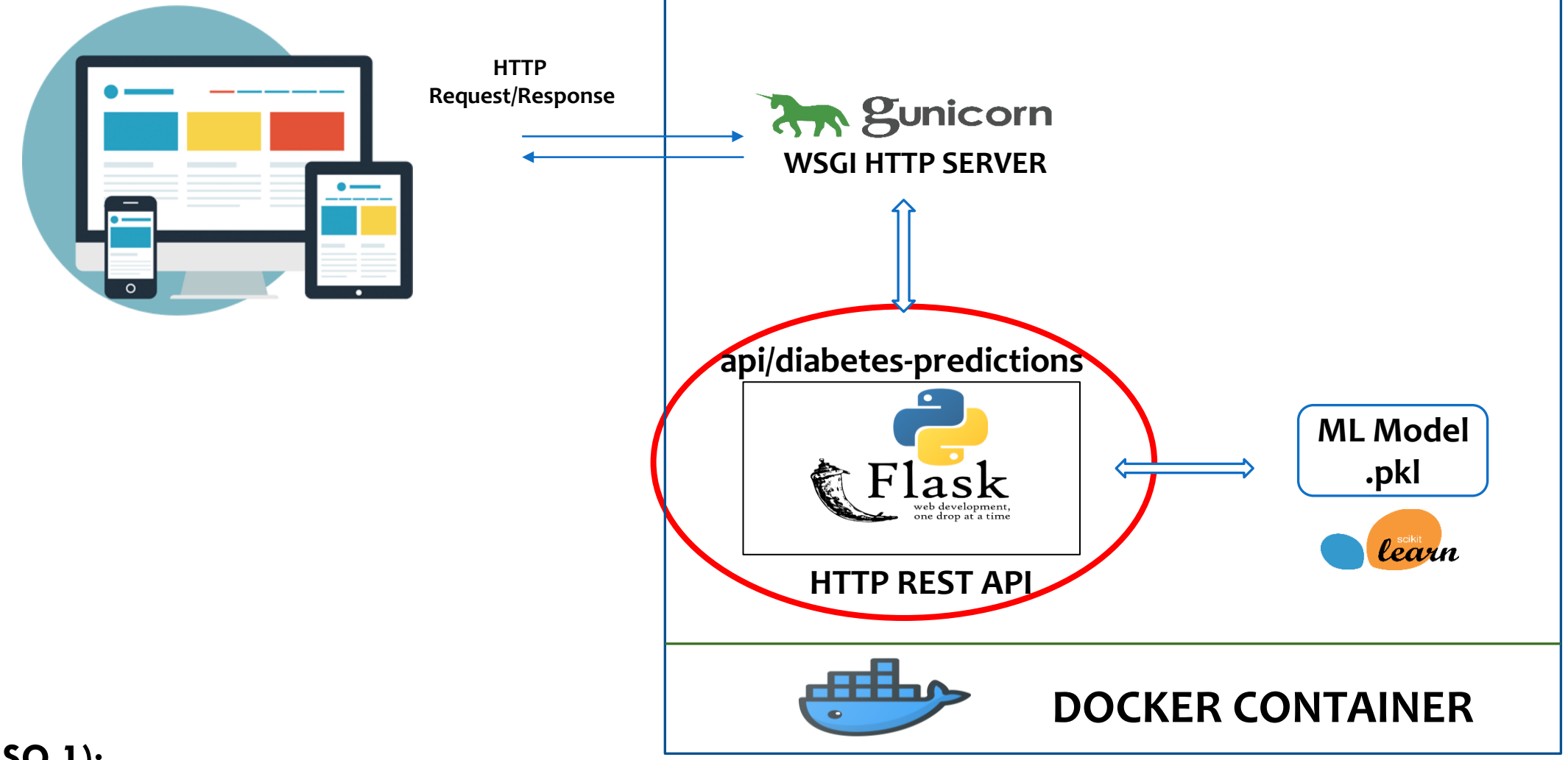
FLASK: Hello World

```python
from flask import Flask, escape, request

app = Flask(__name__)

@app.route('/')
def hello():
    name = request.args.get("name", "World")
    return f'Hello, {escape(name)}!'
```

```
$ env FLASK_APP=hello.py flask run
 * Serving Flask app "hello"
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```



DEMO (PASO 1):

PROGRAMAR REST API USANDO FLASK PARA EXPONER MODELO DE ML.

aws | Amazon Lightsail

QUÉ ES GUNICORN?

Gunicorn, también conocido como Green Unicorn (Unicornio Verde), es un servidor HTTP para Python.

- Compatible con varios frameworks de Python (incluído Flask).
- Implementado de forma simple.
- Consumo de recursos ligero.
- Bastante rápido.

QUÉ ES DOCKER?

Docker es una plataforma abierta que automatiza el desarrollo, implementación y ejecución de aplicaciones dentro de ***contenedores aislados***.

Los contenedores permiten a los desarrolladores ***agrupar una aplicación con todas las partes que necesita***, como bibliotecas y otras dependencias, y ***transferirla como un paquete.***

MÁQUINAS VIRTUALES VS CONTENEDORES DOCKER (I)

Máquinas virtuales

Pesadas

- En el orden de los GB.
- Muchas VMs en el mismo host suelen repetirse en lo que contienen.

Administración costosa

- Una VM tiene que ser administrada como cualquier otra computadora: patching, updates, etc.
- Hay que administrar la seguridad interna entre apps

"Lentas"

- Correr nuestro código en una VM implica no sólo arrancar nuestras aplicaciones, sino también esperar el boot de la VM en sí.

Contenedores Docker

Versátiles

- En el orden de los MB
- Tienen todas las dependencias que necesitan para funcionar correctamente
- Funcionan igual en cualquier lado

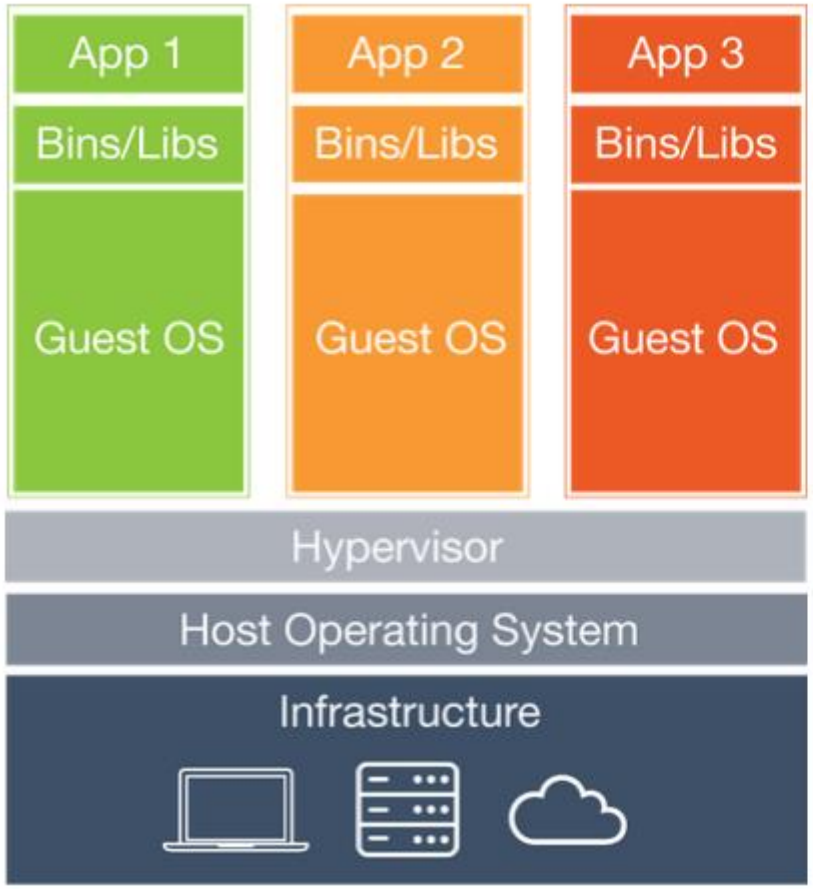
Eficientes

- Comparten archivos inmutables con otros contenedores
- Sólo se ejecutan procesos, no un SS.OO. completo

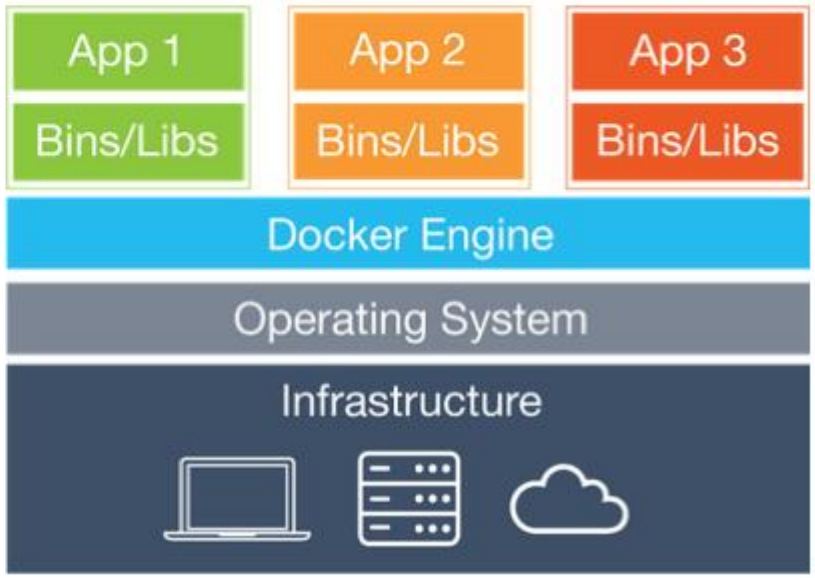
Aislados

- Lo que pasa en el container queda en el container
- No pueden alterar su entorno de ejecución(a menos que explícitamente se indique lo contrario)

MÁQUINAS VIRTUALES VS CONTENEDORES DOCKER (II)



Máquinas virtuales



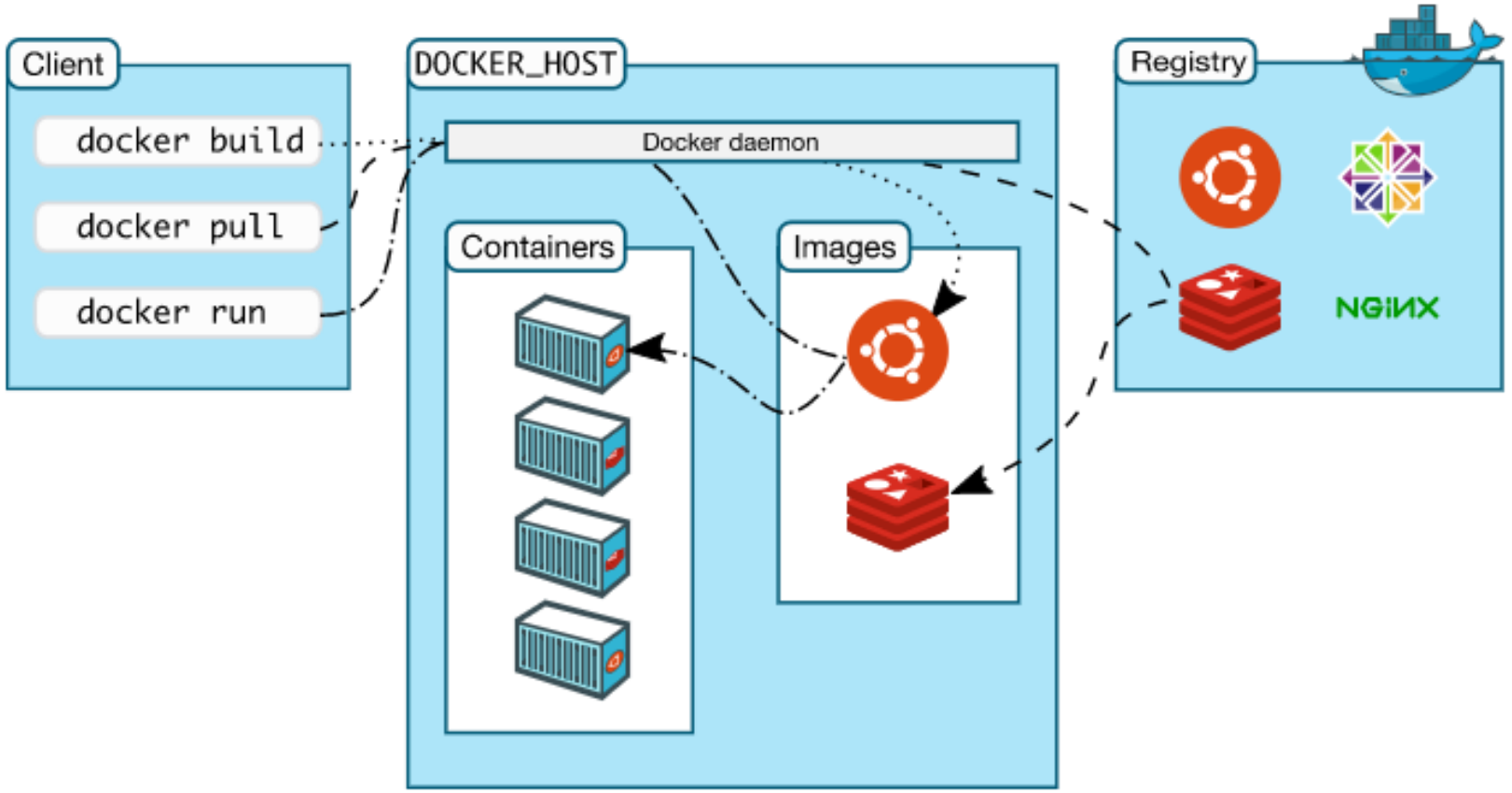
Contenedores Docker

BENEFICIOS DE USAR DOCKER

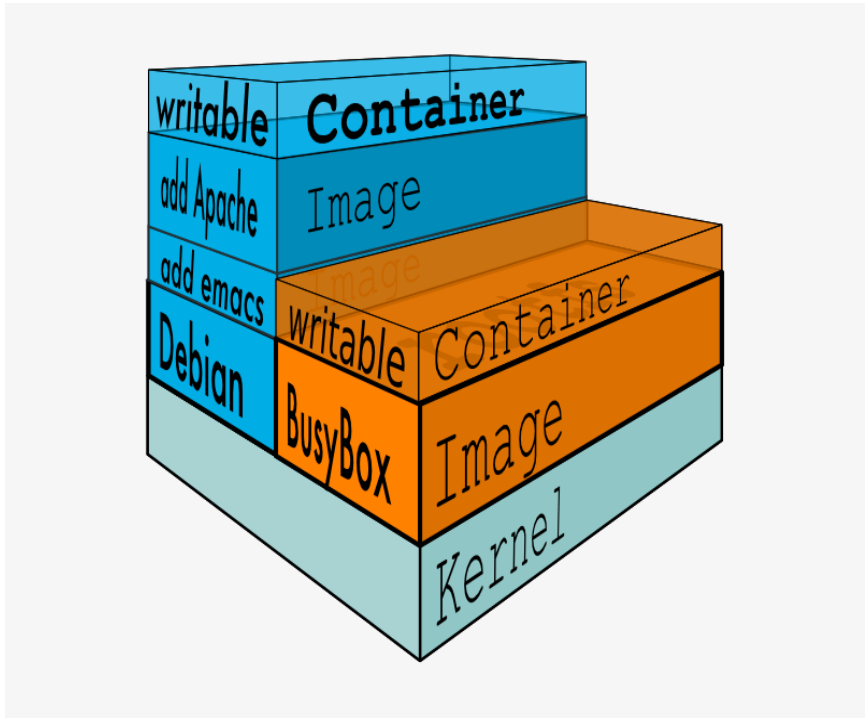
Algunos beneficios de utilizar Docker:

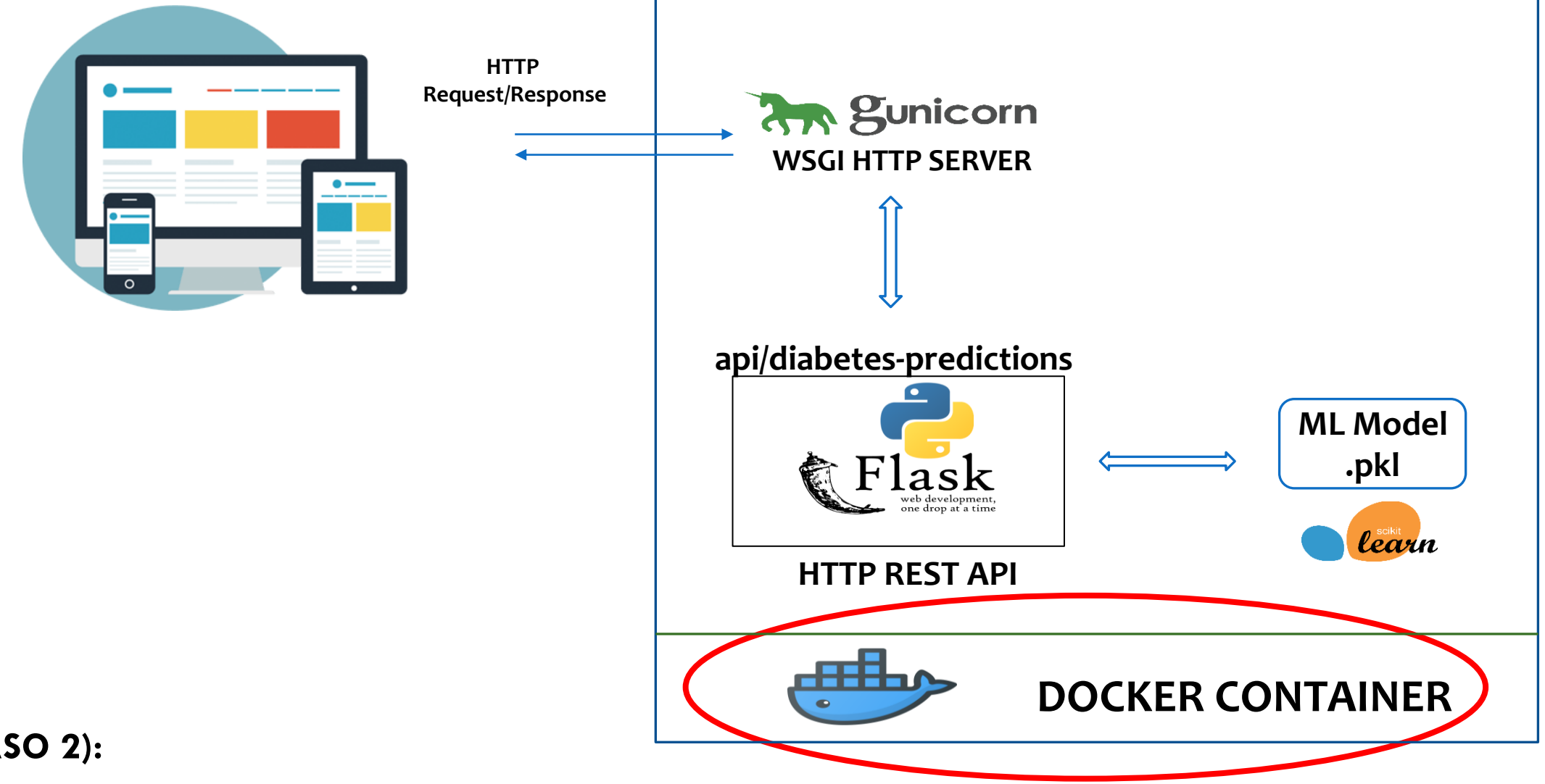
- **Estandarización y productividad**
  - Consistencia garantizada a lo largo del proceso de desarrollo y despliegue de una aplicación, estandarizando el entorno en el que se ejecuta.
- **Compatibilidad y mantenimiento más fácil**
  - Erradica el problema de "en mi máquina funciona" de una vez por todas.
- **Aislamiento**
  - Cada contenedor tiene sus propios recursos y están aislados de otros contenedores.
- **Simplicidad y configuraciones más rápidas**
  - Se puede usar en una amplia variedad de entornos (**agnóstico de insfraestructura:** Laptop, PC, Server – Cloud: AWS, GCP, Azure, entre otros.), configuraciones personalizadas.
- **Despliegue y escalabilidad instántaneo**
  - Reducir el despliegue a cuestión de segundos, escalamiento horizontal (réplicas de contenedores).

ARQUITECTURA DE DOCKER



DOCKER: DE IMAGEN A CONTENEDOR DOCKER: PASO A PASO





DEMO (PASO 2):

GENERAR IMAGEN DOCKER Y EJECUTAR CONTENEDOR LOCALMENTE.

aws | Amazon Lightsail

AWS LIGHTSAIL

Amazon Lightsail proporciona diferentes características diseñadas para ayudar a hacer realidad los proyectos con rapidez. Lightsail, diseñada como un VPS fácil de usar, ofrece una solución completa para todas las necesidades relacionadas con la nube: Servidores, Almacenamiento, Networking, Balanceo de Carga, Bases de datos y Contenedores.

HTTP
Request/Response

gunicorn
WSGI HTTP SERVER

api/diabetes-predictions

Flask
web development,
one drop at a time

HTTP REST API

ML Model
.pkl

scikit learn

DOCKER CONTAINER

DEMO (PASO 3):

DESPLEGAR CONTENEDOR DOCKER EN AWS LIGHTSAIL CONTAINERS

aws | Amazon Lightsail

RESULTADO FINAL: DIABETES PREDICTION MOBILE APP


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
BOLZÏN
MERCI
BÏYAN
SHUKRIA
TINGKI
DANKSCHEEN
SPASSIBO
NUHUN
SNACHALHUYA
CHALTU
TASHAKKUR ATU
YAQHANYELAY
YUSPAGARÀTAM
HUI
WABEEJA
MAITEKA
SUKSAMA
EKHMET
DHANYABAAD
AJIHA
ATTO
MERSI
SPASIBO
DENKAUJA
NENACHALHYA
UNALCHÉESH
GUI
HATUR
EKOJU
SIKOMO
MAKETAI
MINMONCHAR
MEHRBANI
GRAZIE
PALDIES
MAAKE
SUMNIDA
KOMAPSANCO
MERASTAWHY
GAEJTHO
LAH
GOZAIMASHITA
EFCHARISTO
AGUYJE
FAKAAUE
TAVTAPUCH
MEDAWAGSE
JUSPAXAR
BAJKA