



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA



Machine Learning Operations (MLOps) Clase 3

Leticia Rodríguez

Septiembre 2024 - 2do Cuatrimestre - 4to. Bimestre

Universidad de Buenos Aires - FCEyN - Departamento de Computación

Encuesta y Asistencia

Kahoot de respaso

MLOps - Principios y Componentes

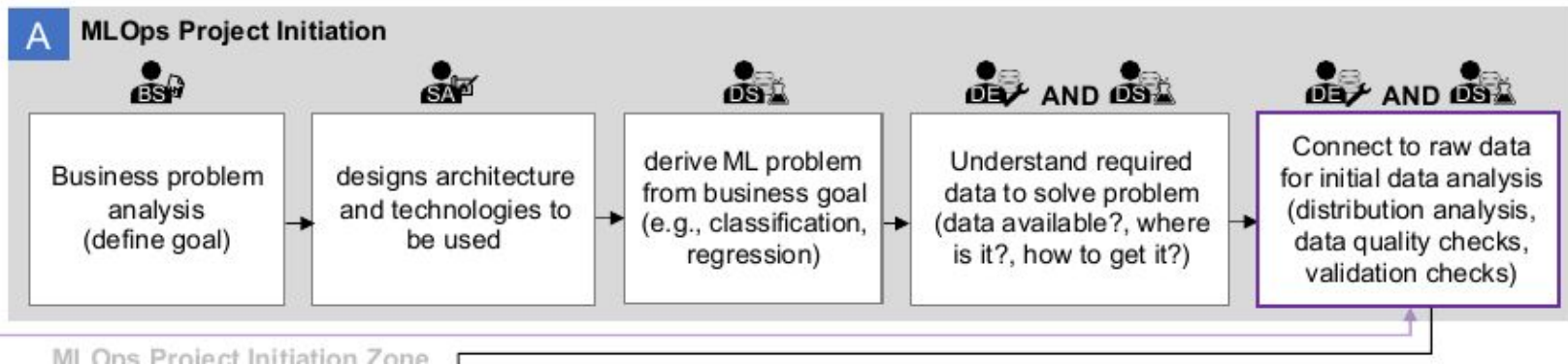
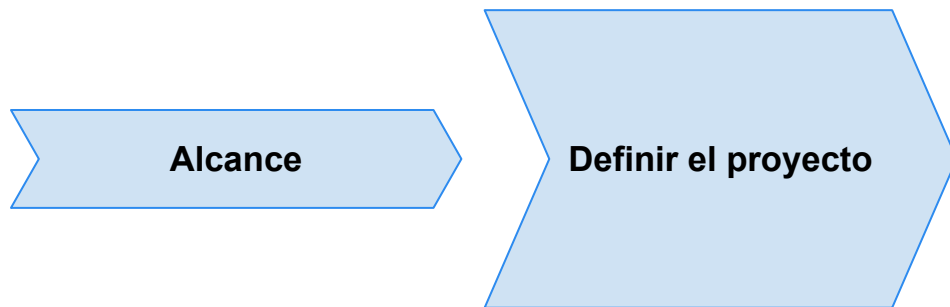


PRINCIPLES

- P1** CI/CD automation
- P2** Workflow orchestration
- P3** Reproducibility
- P4** Versioning of data, code, model
- P5** Collaboration
- P6** Continuous ML training & evaluation
- P7** ML metadata tracking
- P8** Continuous monitoring
- P9** Feedback loops

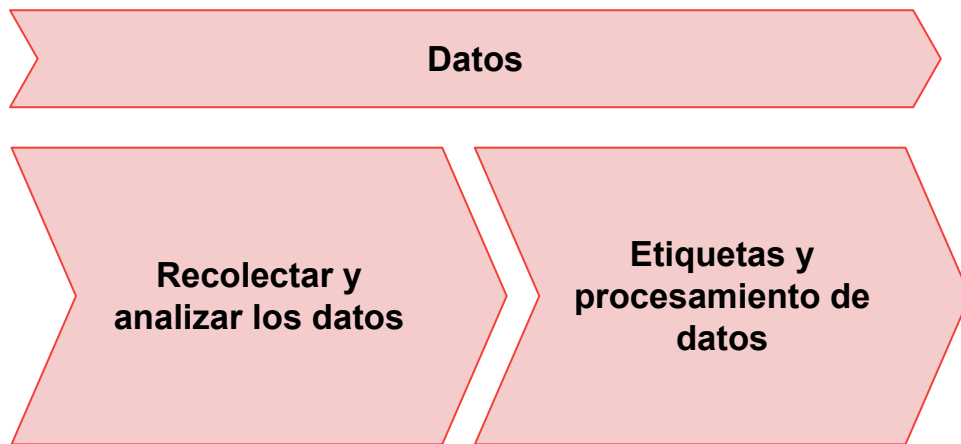
COMPONENT

MLOps - La clase pasada



MLOps - La clase pasada y continuamos

Procesamiento de Datos



Feature Store

- Constituyen una fuente de datos única para proyectos de Análisis de Datos y Aprendizaje Automático

Un sistema de AI puede tener distintas features de distinta procedencia

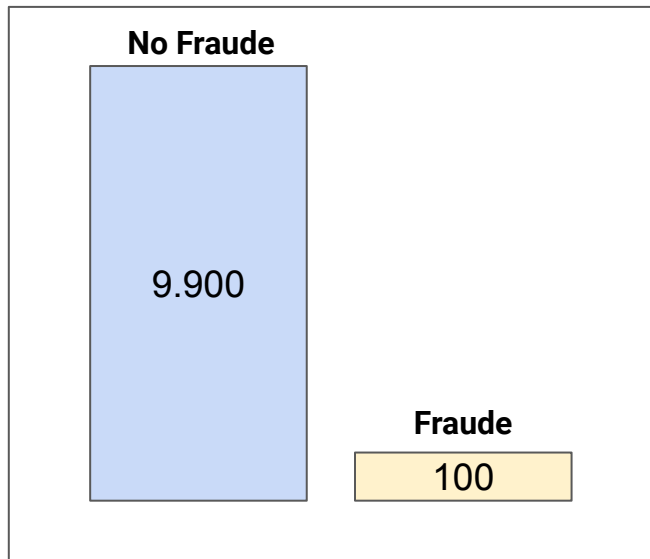


Actividad: Desbalanceo de Clases

En grupos analicen el paper, “Approaches to handle Data Imbalance Problem in Predictive Machine Learning Models: A Comprehensive Review” de Govind M. Poddar et al. siguiendo las consignas:

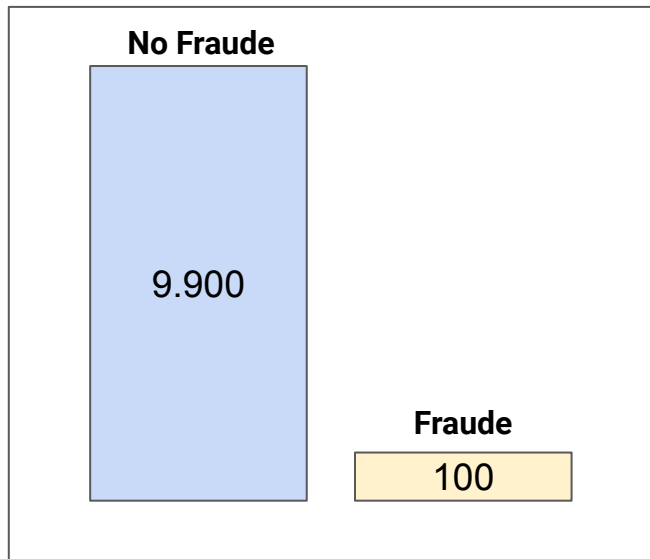
1. Lean y discutan la sección “3. Even Distribution Versus Uneven Distribution” que trata sobre el concepto de desbalanceo de clases. Compartan sus experiencias.
2. Lean la sección “6. Different approaches to tackle Class Imbalance Problem” sólo la introducción y luego, elijan al menos 3 métodos entre los que figuran en la tabla 1 de undersample, tabla 2 de oversample o algorithmic approaches. Los métodos de Ensamble los dejaremos para más adelante.
3. Por último, lean la sección 7: “7. Performance Evaluation Metrics for Class Imbalanced Datasets”
4. Compartan su trabajo con el resto de la clase.

Desbalanceo de Clases



- Si tenemos un modelo con **accuracy 99%**, podemos pensar que es muy bueno.
- Pero ese 99% de exactitud podría estar ubicado exclusivamente en detectar la clase No Fraude por lo cual, para datasets desbalanceados no es una buena métrica.
- Tenemos que considerar además el recall.

Desbalanceo de Clases



- $\text{Recall} = \text{TP} / (\text{TP} + \text{FN})$
- Si tenemos un modelo con recall 99%, podemos pensar que es muy bueno.
- Pero ese 99% de recall no nos dice nada de los falsos positivos
- Entonces podría estar detectando los No Fraude como Fraude

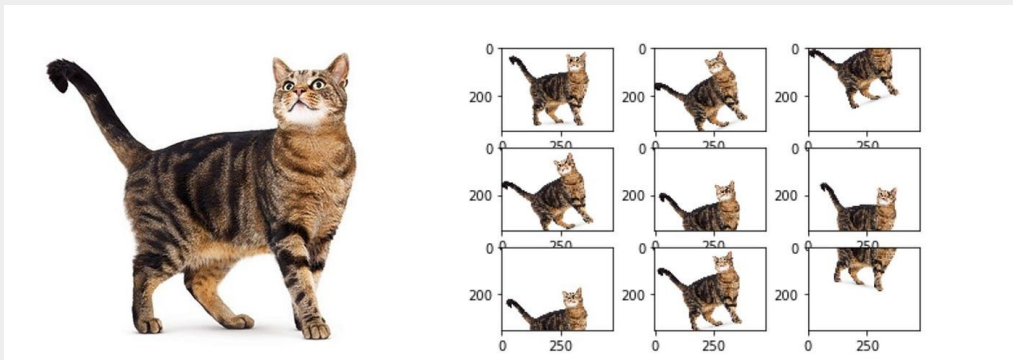
Desbalanceo de Clases: Posibles Soluciones

- **Elección de las métricas correctas**
 - Evaluar el tradeoff recall-precision y decidir el threshold acorde el negocio
 - **Precisión:** $TP / (TP+FP)$ Y **Recall:** $TP / (TP+FN)$
 - Curva AUC PR
 - Matrices de confusión
- **Usar métodos a nivel de datos: Resamplero de Datos**
 - SMOTE - Undersampling - Oversampling
- **Usar métodos a nivel de algoritmos**
 - Cost-sensitive learning
 - Class-balanced loss
 - Focal loss

		Predicción	
		Positivo	Negativo
Real	Positivo	TP	FN
	Negativo	FP	TN

Data Augmentation

- Distintas técnicas que permiten incrementar la cantidad de datos de entrenamiento y mejoran el aprendizaje con bajo costo. A veces, este incremento se hace desde los mismos datos de entrenamiento actuales.
- Se volvió fundamental para la creación de modelos en Computer Vision, y recientemente en NLP.
- Depende altamente del formato de los datos, es distinta para texto e imágenes.



Simple Label-Preserving Transformations

Perturbation:

Agregar noise (ruido) como ejemplo adversario

Data Synthesis:

Generar data sintética

Simple Label-Preserving Transformations:

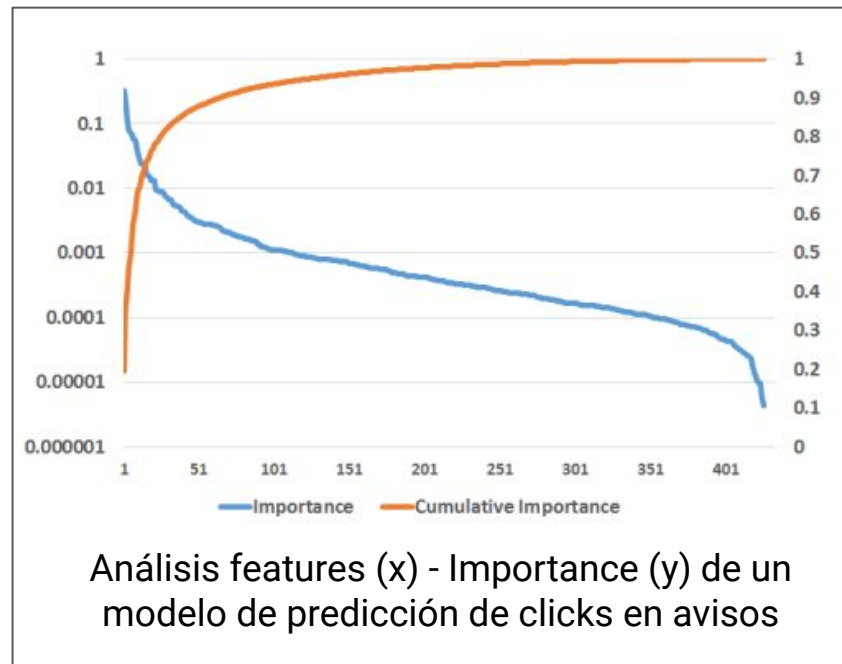
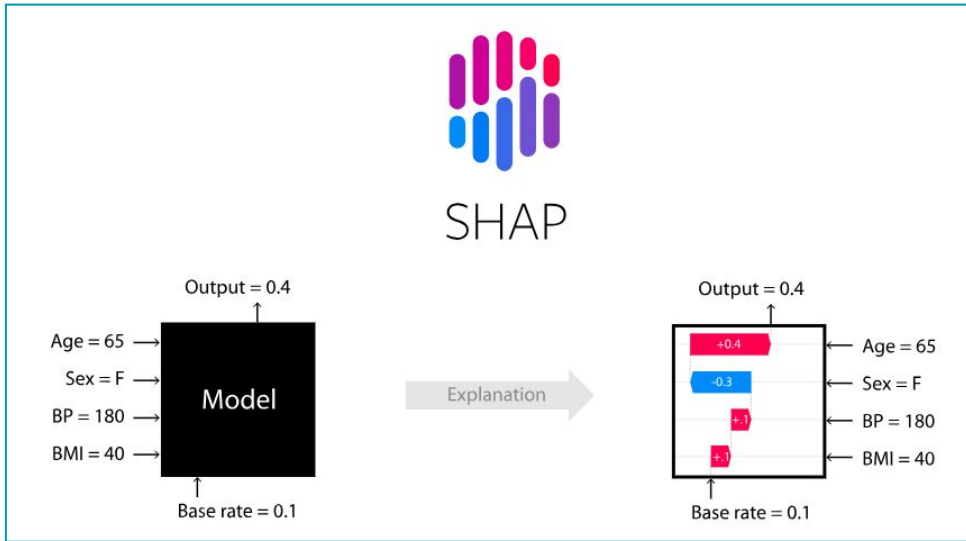
Aplicar transformaciones (rotaciones, mover, flip, crop) a las imágenes del training set.

Data Leakage

- Se llama Data Leakage al fenómeno en el cual los labels (etiquetas) de un aprendizaje supervisado se cuelan de alguna forma dentro de los features haciendo que el modelo a entrenar tenga el sesgo de ese dato para predecir.
- Consecuencias:
 - Performance demasiado optimista
 - Generalización pobre
 - Conocimiento engañoso
- Causas:
 - **Distribuir los datos de forma aleatorio y no por tiempo**
 - **Scalar antes de dividir los sets:** error común es usar toda la data para sacar las estadísticas (media, varianza) para escalar. Solución: tomar la estadística en training y usarla para escalar todos los splits.
 - **Llenar datos faltantes con estadísticas del split de testeo.**
 - **Manejo pobre de los datos duplicados antes de dividir los sets:** sacarlos antes de dividir los sets
 - **Group leakage:** datos similares del mismo grupo con diferencia de tiempo o minima cae uno en testeo y otro en training.
 - **Leakeage derivado de la generación de datos**
- Formas de detectarlo:
 - Estudiar la correlación entre label y features
 - Hacer pruebas sacando algún feature y viendo si degrada demasiado la performance del modelo
 - Estar atento a los features que se van agregando al modelo

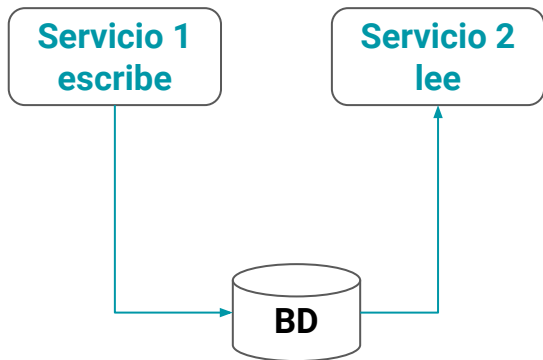
IA Explicable

- Feature Importance: es una técnica que nos dice mediante un score que tan importante es un feature en relación a otros features para el modelo.
- Algunas implementaciones de árboles como XGBoost traen la funcionalidad implementada
- SHAP (SHarply Additive exPlanations) es agnostico al modelo y no solo mide la importancia, sino también la contribución en una predicción

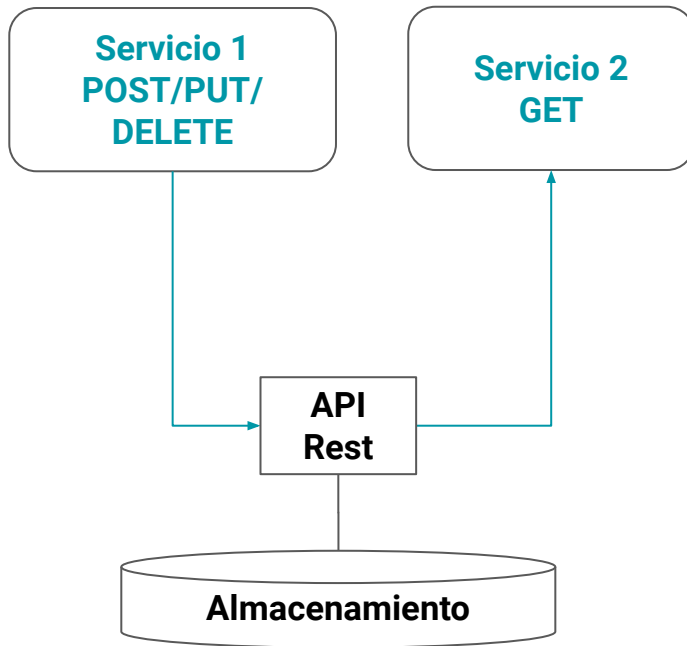


Flujo de los Datos (DataFlow)

Pasaje a través de Base de Datos

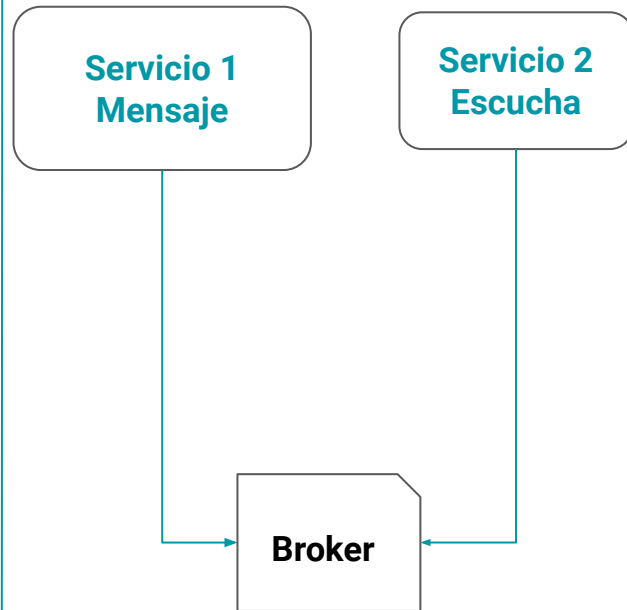


Pasaje a través de Servicios



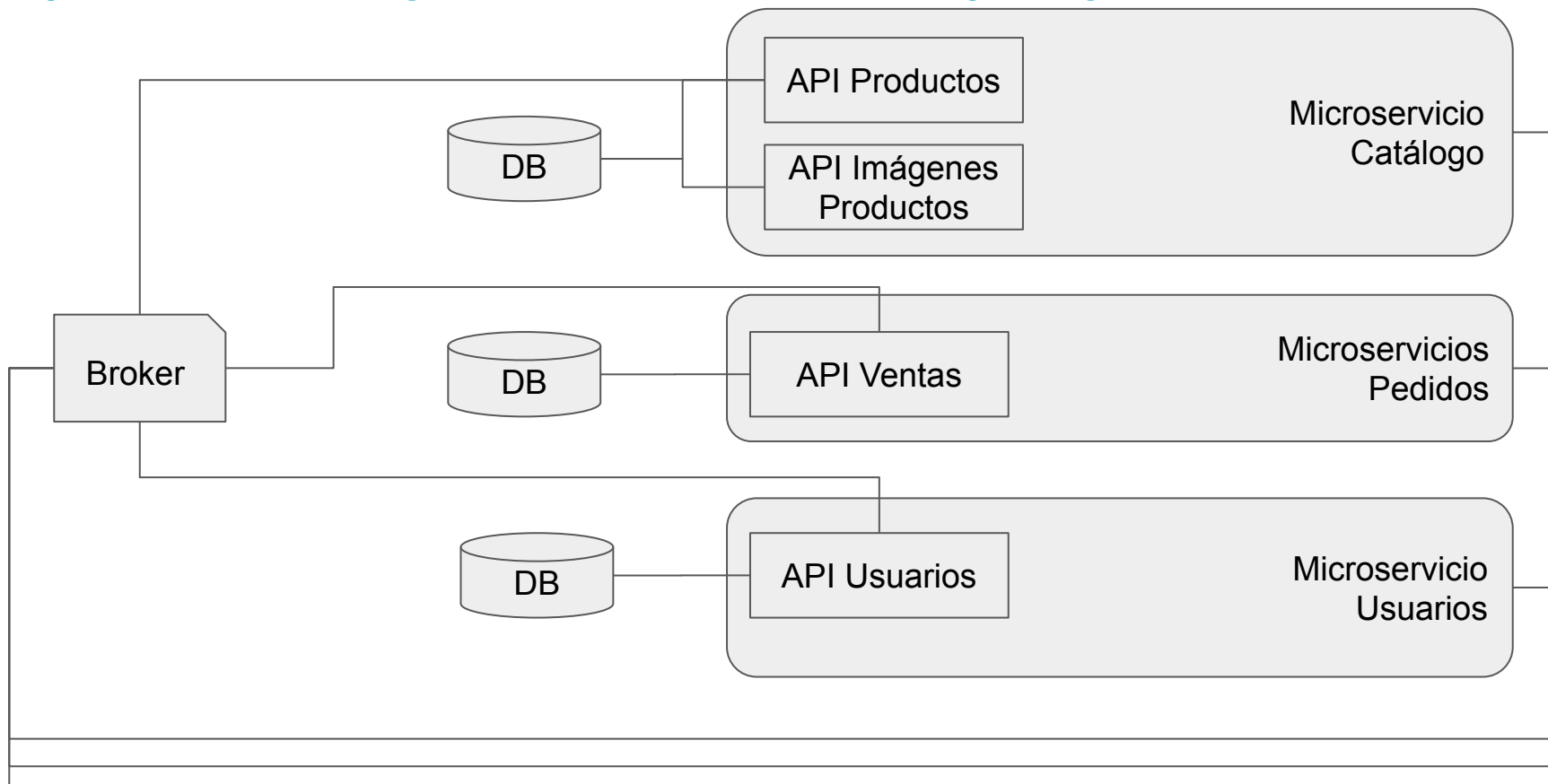
El servicio puede ser RPC u otro

Pasaje a través de un transporte de tiempo real



Kafka, RabbitMQ ejemplos

Flujo de Datos y Microservicios - Ejemplo



Arquitectura simplificada que combina dos tipos de pasajes de datos

Data Pipelines

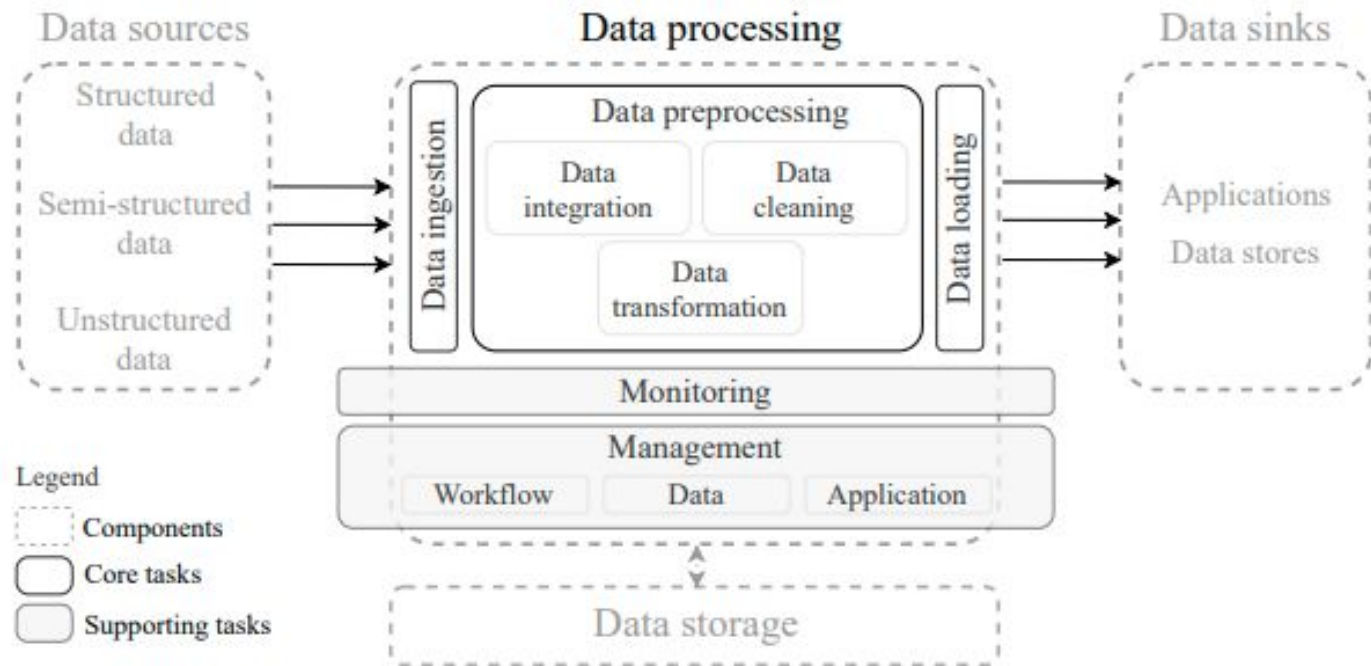


Figure 1: High-level data pipeline architecture

Actividad: Lectura del paper de Data Pipelines

En grupos, leamos la sección 1. Introduction. Luego, responder:

1. ¿Qué es un data pipeline?
2. ¿Cuáles son los desafíos de estos pipelines en producción?
3. De la sección 4 Use Cases, elijan 2 de los pipelines en alguna empresa A,B,C y coméntenlos en el grupo.
4. De la sección 5 Challenges to Data Pipeline Management lean 1 de las 3 secciones: Infrastructure, Organizational or Data Quality y coméntenla entre uds.
5. De la sección 6 Opportunities selección 3 oportunidades y coméntenlas entre uds.

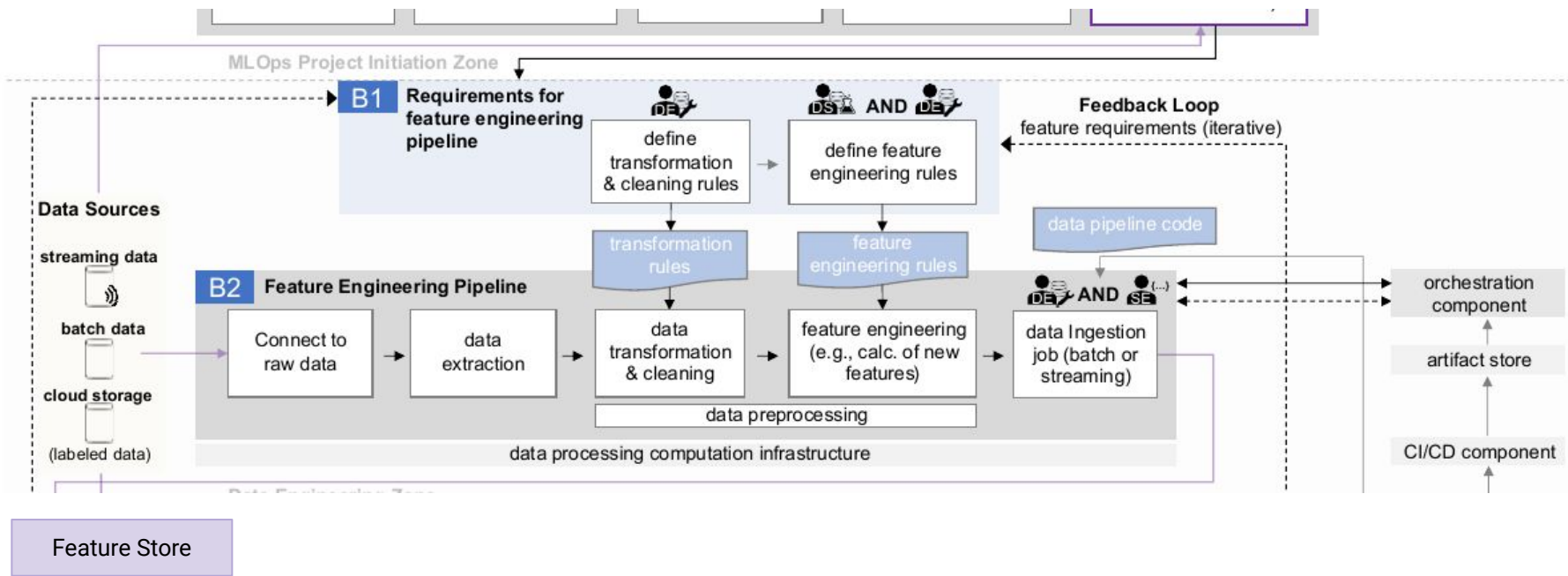
Todos juntos revisemos lo más interesante que leyó cada grupo

Batch vs. Online

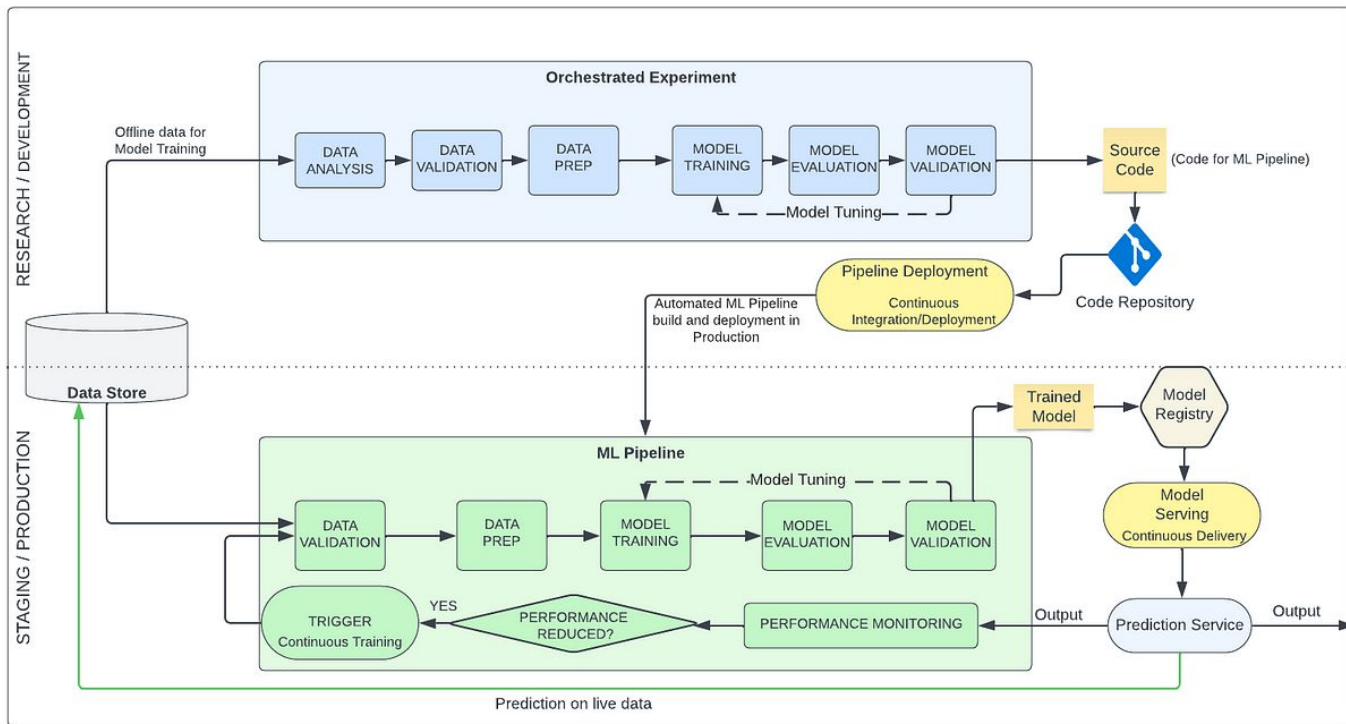
Existen dos formas de procesamiento de datos en forma Batch u Online.

- Online: procesar los datos al momento que llegan
 - requerimientos de infraestructura para la variabilidad de la cantidad los pedidos
 - servicios de mensajería: Kafka, RabbitMQ, servicios cloud para este fin.
- Batch: procesar los datos en lote
 - procesos programados que corren en horarios o frecuencia determinada
 - procesan grandes cantidades de datos
 - tener en cuenta temas lockeo de datos
 - infraestructura escalable acorde la cantidad de datos a procesar

Pipeline de Datos en el paper de MLOps



Creando un pipeline de datos para ML



Metadatos y trazo de los datos

- Data Lineage: es la técnica de rastrear el origen de los datos y sus etiquetas.
- Los datos de un dataset pueden tener diverso origen y haber pasado por diversos procesos de etiquetado
- El etiquetado puede requerir contratar equipos externos que manualmente lo realizan o mediante diferentes algoritmos y técnicas automáticas
- Llevar registro del origen de los datos y qué datos se usaron para entrenar a que modelos es fundamental para cuando haya que buscar evidencia sobre errores o problemas e incluso para entender comportamientos del modelo.
- También la data no estructurada puede ir acompañada de metadata estructurada que de más información sobre el origen de los datos y simplifique la construcción de modelos (ej. audios, con metadata que indique duración, interlocutores u otros tags)
- Esto lleva también almacenamiento estructurado adicional

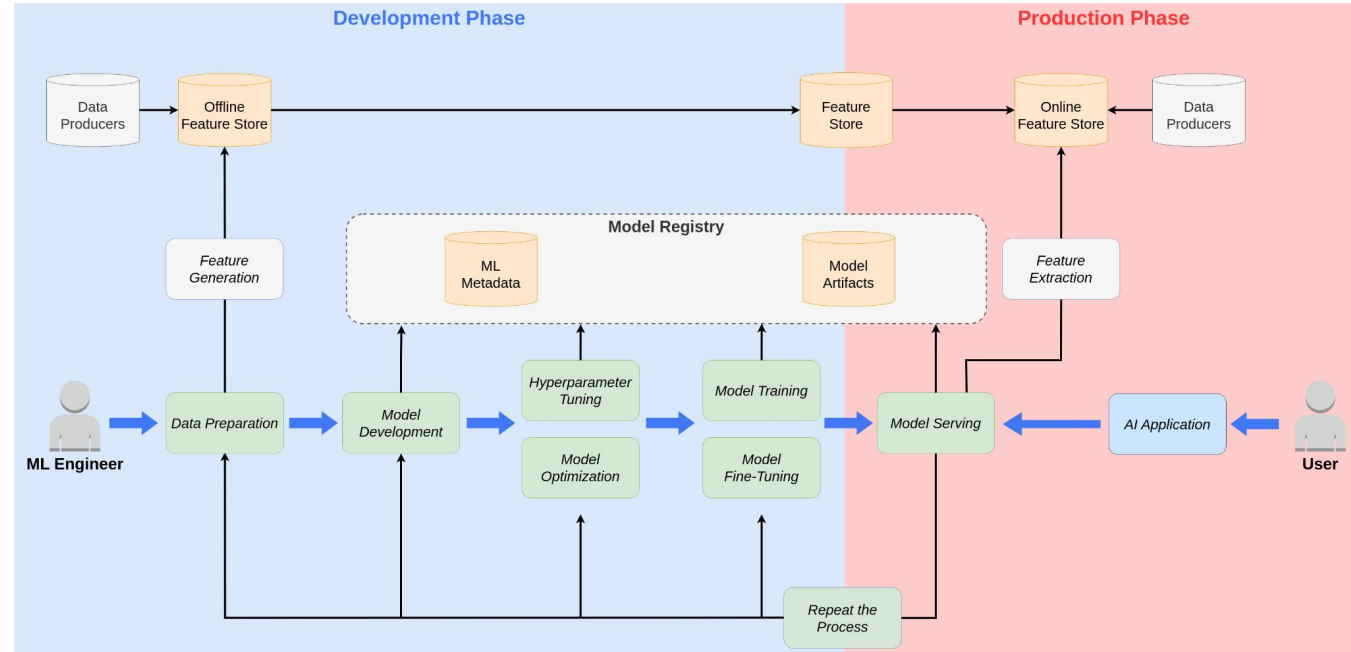
Los cambios en los datos - Drifts

- Imaginemos un sistema de ecommerce que funciona hace 10 años... sus datos cambiaron en dos dimensiones
 - Contenido. Por ejemplo, podría haber más usuarios
 - Schema - Esquema . Por ejemplo, recientemente se agregó un campo para especificar un link a redes sociales
- A su vez, un mismo campo de datos puede tomar valores distintos o nuevo, por ejemplo, medio de pago tradicionalmente era contado, tarjeta y agrega billetera electrónica
- También, podría haber cambios en la distribución de los datos. Anteriormente, las transacciones eran en su mayoría al contado, hoy la mayor parte son con billetera electrónica.
- Podría haber cambio en la significancia del dato (esto sería más raro)
- Incluso nuevos requerimientos del lado del software puedan agregar datos o features nuevas que serían de utilidad
- **Los datos y su significado cambian a través del tiempo**

Herramientas opensource para Pipelines de ML

Existen distintas herramientas para armar flujos de datos y de Machine Learning
Las más populares son:

- Apache Airflow
- Kubeflow



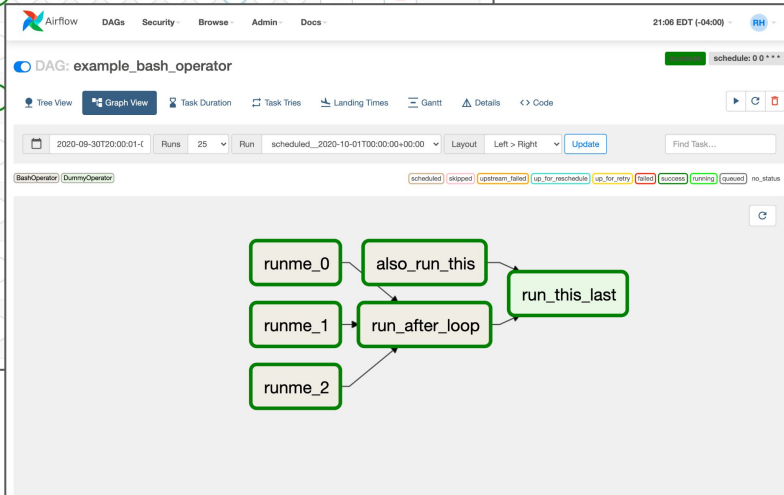
De la documentación
de Kubeflow

Apache Airflow

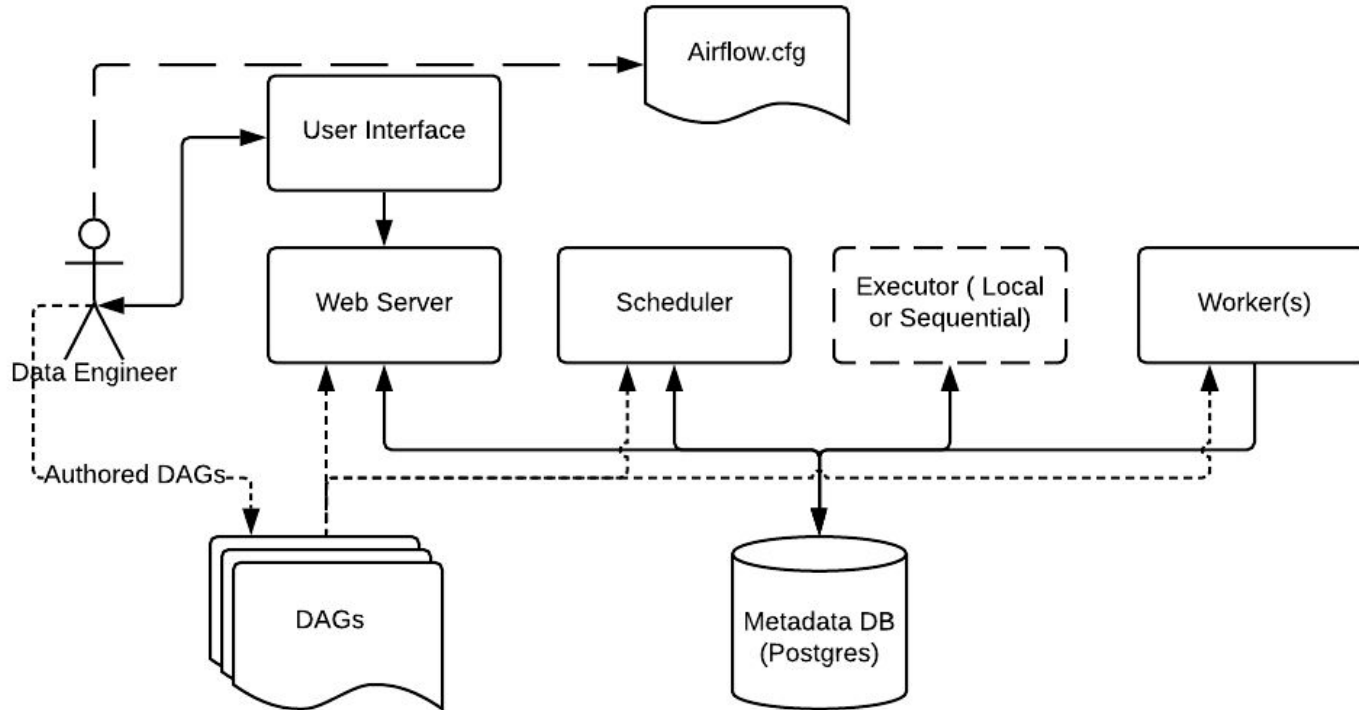
The screenshot shows the Apache Airflow web interface. At the top, there's a navigation bar with links for DAGs, Security, Browse, Admin, and Docs. The current time is 21:11 UTC. Below the navigation bar, the 'DAGs' section is active. It features a filter bar with 'All' (26), 'Active' (10), and 'Paused' (16) buttons. A search bar is also present. The main table lists several DAGs:

DAG	Owner	Runs	Schedule	Last Run	Recent Tasks	Actions	Links
<input checked="" type="checkbox"/> example_bash_operator example example2	airflow	2	0 0 *	2020-10-26, 21:08:11	6	[Play] [Refresh] [Delete]	...
<input checked="" type="checkbox"/> example_branch_dop_operator_v3 example	airflow	0	* / 1 *			[Play] [Refresh] [Delete]	...
<input type="checkbox"/> example_branch_operator example example2	airflow	1	@daily	2020-10-23, 14:09:17	11	[Play] [Refresh] [Delete]	...
<input checked="" type="checkbox"/> example_complex example example2 example3	airflow	1	None	2020-10-26, 21:08:04	37	[Play] [Refresh] [Delete]	...
<input checked="" type="checkbox"/> example_external_task_marker_child	airflow	1	None	2020-10-26, 21:07:33	1	[Play] [Refresh] [Delete]	...

```
19 """Example DAG demonstrating the usage of the BashOperator."""
20
21 from datetime import timedelta
22
23 from airflow import DAG
24 from airflow.operators.bash import BashOperator
25 from airflow.operators.dummy_operator import DummyOperator
26 from airflow.utils.dates import days_ago
27
28 args = {
29     'owner': 'airflow',
30 }
31
32 dag = DAG(
33     dag_id='example_bash_operator',
34     default_args=args,
35     schedule_interval='0 0 * * *'.
```



Apache Airflow



<https://airflow.apache.org/docs/apache-airflow/2.0.1/concepts.html>

Kubeflow

Kubeflow Pipelines (KFP) es una plataforma para crear y deployar flujos de Machine Learning

Un pipeline es la definición de un flujo de datos con uno o más componentes formando un grafo.

Esta definición se ejecuta en un **contenedor** y puede preprocesar datos hasta crear modelo de Machine Learning.

Integrations

JupyterLab

VSCode

RStudio

PyTorch

HuggingFace

TensorFlow

DeepSpeed

XGBoost

Megatron-LM

Horovod

Scikit-Learn

MPI

Optuna

Hyperopt

Kubeflow Components and External Add-Ons

Kubeflow Components

Kubeflow Pipelines

Kubeflow Notebooks

Central Dashboard

Training Operator

Katib

MPI Operator

KServe

Model Registry

Spark Operator

External Add-Ons

Feast

Elyra

BentoML

Infrastructure



kubernetes



Istio



dex



Google Cloud



aws



Azure



Local



Self Hosted



Public Cloud

Hardware



NVIDIA



intel



AMD

Kubeflow - Pipelines

Getting Started

Pipelines

Experiments

Runs

Recurring Runs

Artifacts

Executions

Documentation

Github Repo

<

Pipelines

+ Upload pipeline

Refresh

Delete

Filter pipelines

<input type="checkbox"/>	Pipeline name	Description	Uploaded on ↓
<input type="checkbox"/>	▶ [Tutorial] V2 lightweight Pyth...	source code Shows different component input and output options for KFP v2 components.	8/25/2021, 4:36:05 PM
<input type="checkbox"/>	▶ [Tutorial] DSL - Control struct...	source code Shows how to use conditional execution and exit handlers. This pipeline will ra...	8/25/2021, 4:36:04 PM
<input type="checkbox"/>	▶ [Tutorial] Data passing in pyth...	source code Shows how to pass data between python components.	8/25/2021, 4:36:03 PM
<input type="checkbox"/>	▶ [Demo] TFX - Taxi tip predicti...	source code GCP Permission requirements . Example pipeline that does classification with ...	8/25/2021, 4:36:02 PM
<input type="checkbox"/>	▶ [Demo] XGBoost - Iterative m...	source code This sample demonstrates iterative training using a train-eval-check	

```
from kfp import dsl
```

```
@dsl.component
```

```
def say_hello(name: str) -> str:
    hello_text = f'Hello, {name}!'
    print(hello_text)
    return hello_text
```

```
@dsl.pipeline
```

```
def hello_pipeline(recipient: str) -> str:
    hello_task = say_hello(name=recipient)
    return hello_task.output
```

You can [compile the pipeline](#) to YAML with the KFP SDK DSL [Compiler](#) :

```
from kfp import compiler
```

Kubeflow - Experiments



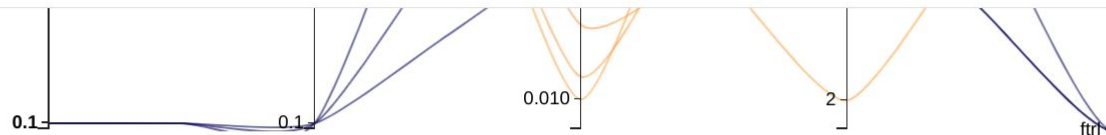
- Home
- Notebooks
- Tensorboards
- Models
- Volumes
- Experiments (AutoML)**
- Experiments (KFP)
- Pipelines
- Runs
- Recurring Runs
- Artifacts
- Executions

kubeflow-user (Owner)



← Experiment details

DELETE



OVERVIEW

TRIALS

DETAILS

YAML

Trial name	Status	Validation accuracy	Train accuracy	Lr	Num layers	Optimizer
random-example-26305f40	Succeeded	0.96437	0.9617	1.05842e-2	4	adam
random-example-33243bc0	Succeeded	0.979	0.99283	1.32217e-2	5	sgd
random-example-4d4417ce	Succeeded	0.96188	0.96062	1.18011e-2	3	adam
random-example-5f5785ff	Succeeded	0.9789	0.9925	1.6367e-2	2	sgd
random-example-6ae3641d	Succeeded	0.94556	0.93611	2.51815e-2	5	adam
random-example-89efe504	Succeeded	0.11385	0.11242	2.49797e-2	4	ftrl
random-example-8d3bfac9	Succeeded	0.94655	0.94289	2.14048e-2	3	adam

Kubeflow - Runs



Kubeflow



Home



Notebooks



TensorBoards



Volumes



Katib Experiments



KServe Endpoints



Pipelines

Pipelines

Experiments

Runs

Recurring Runs

Artifacts

Executions

Manage Contributors

build version - dev_local

team-1 (Owner)



Runs

+ Create run

Compare runs

Clone run

Archive

Refresh

Active

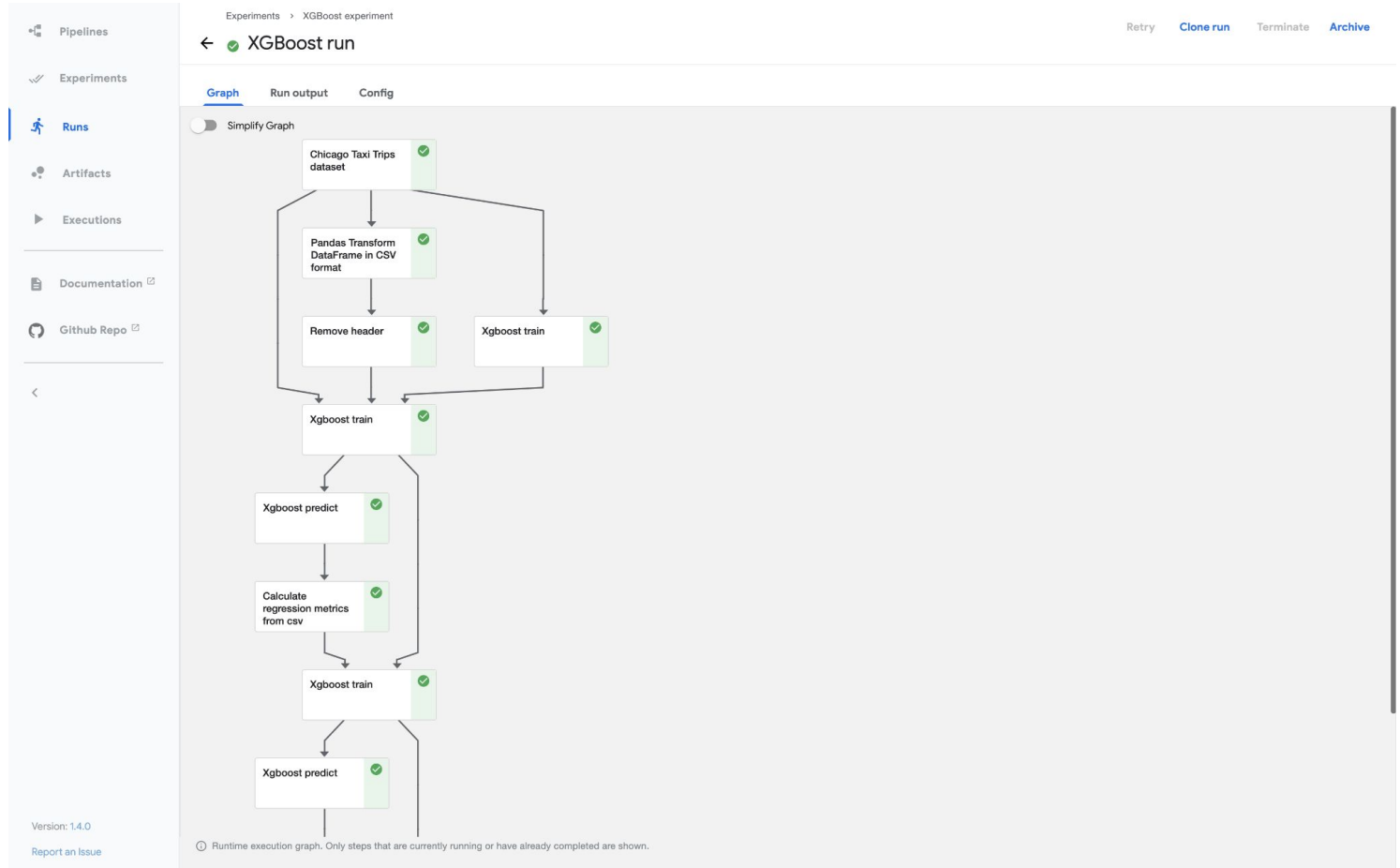
Archived

Filter runs

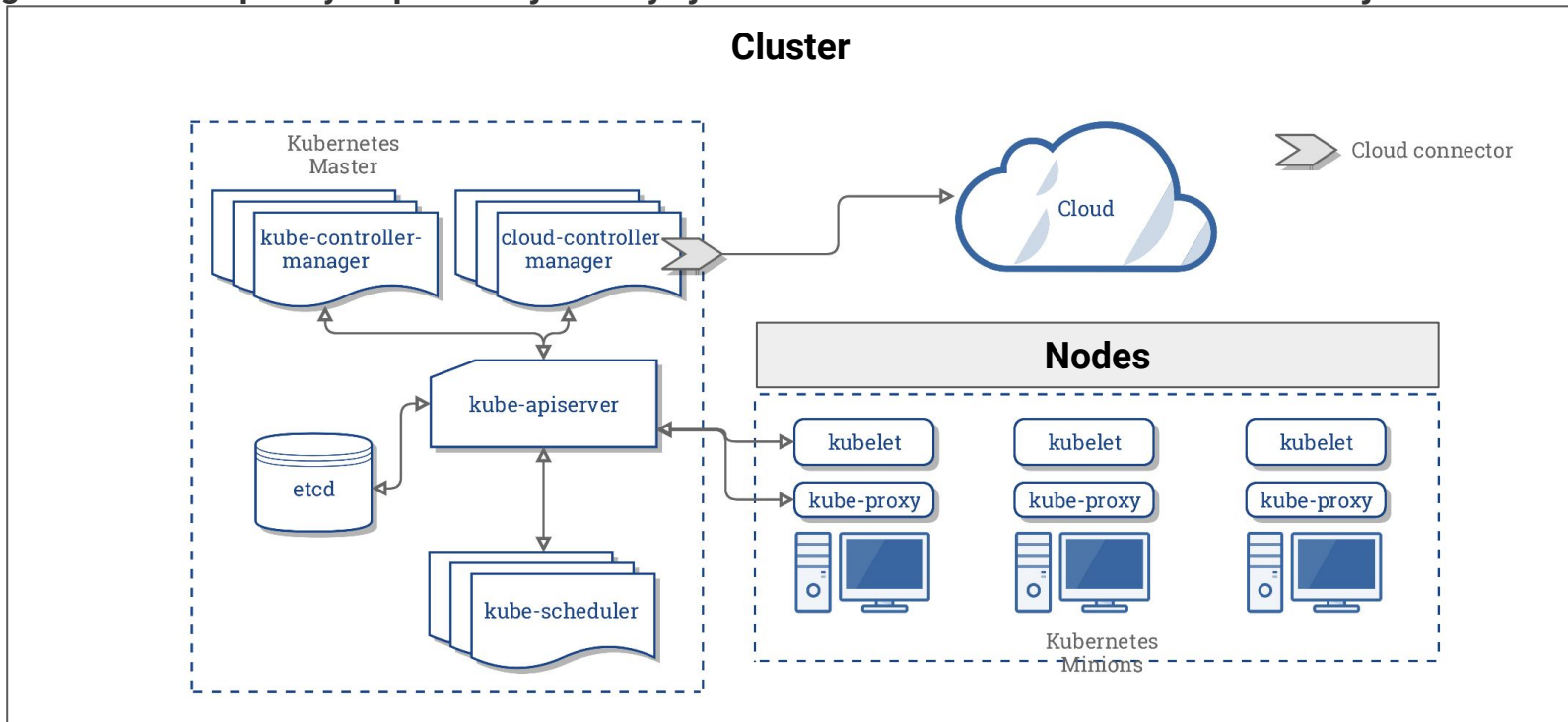
<input type="checkbox"/>	Run name	Status	Duration	Experiment	Pipeline Version	Recurring R...	Start time ↓
<input type="checkbox"/>	pipeline_v2.yaml 2024-05-10 21-45-03	✓	0:00:34	test-v2	[View pipeline]	-	5/10/2024, 9:45:03 PM
<input type="checkbox"/>	pipeline_v1.yaml 2024-05-10 21-42-31	✓	0:00:11	test-v1	[View pipeline]	-	5/10/2024, 9:42:31 PM
<input type="checkbox"/>	pipeline_v2_compatible.yaml 2024-0...	✓	0:00:30	test-v2-compatible	[View pipeline]	-	5/10/2024, 9:34:49 PM
<input type="checkbox"/>	pipeline_v2.yaml 2024-05-10 21-33-29	✓	0:00:34	test-v2	[View pipeline]	-	5/10/2024, 9:33:29 PM
<input type="checkbox"/>	viz_pipeline_v1.yaml 2024-05-10 21-...	✓	0:00:11	test-v1	[View pipeline]	-	5/10/2024, 9:31:56 PM
<input type="checkbox"/>	viz_pipeline_v1.yaml 2024-05-10 21-...	✓	0:00:21	test-v1	[View pipeline]	-	5/10/2024, 9:17:50 PM
<input type="checkbox"/>	Run of [Tutorial] DSL - Control struct...	✓	0:02:58	test-v2	[Tutorial] DSL - Control ...	-	5/10/2024, 9:11:10 PM

Rows per page: 10 < >

Kubeflow - Runs



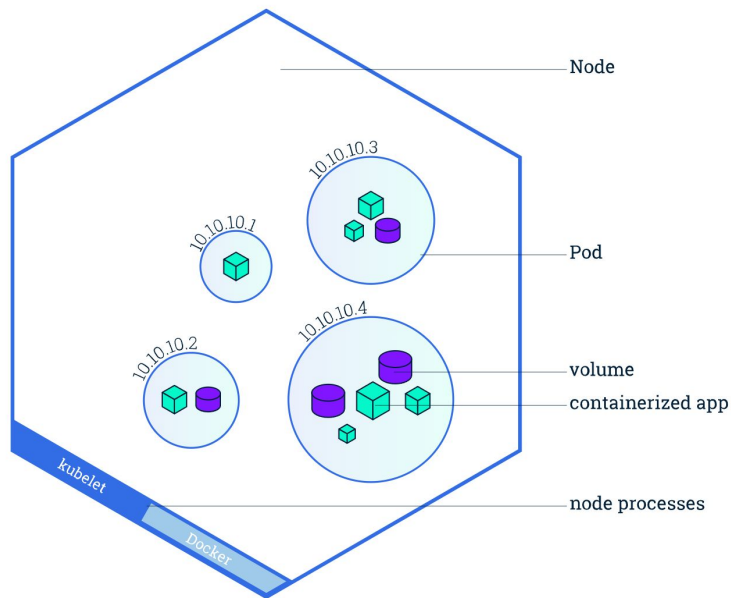
Kubernetes es un software de código abierto que le permite implementar y administrar aplicaciones en contenedores a escala. Kubernetes administra un **clúster** y **programa contenedores** para que se ejecuten en el clúster en función de los recursos informáticos disponibles y de los requisitos de recursos de cada contenedor. Los **contenedores se ejecutan en agrupaciones lógicas llamadas pods** y es posible ejecutar y ajustar la escala de uno o más contenedores juntos como un pod.

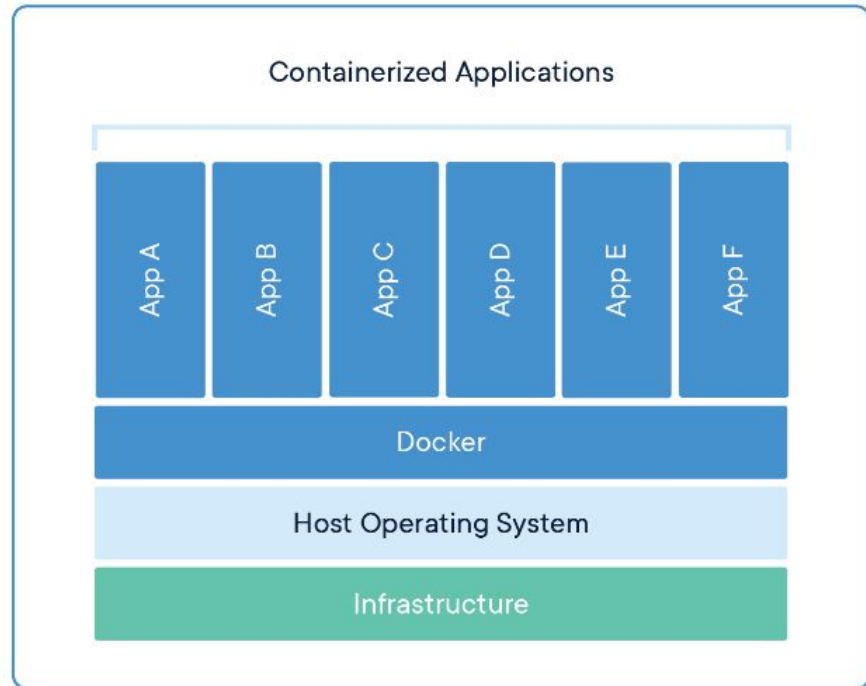


Kubernetes - Pods

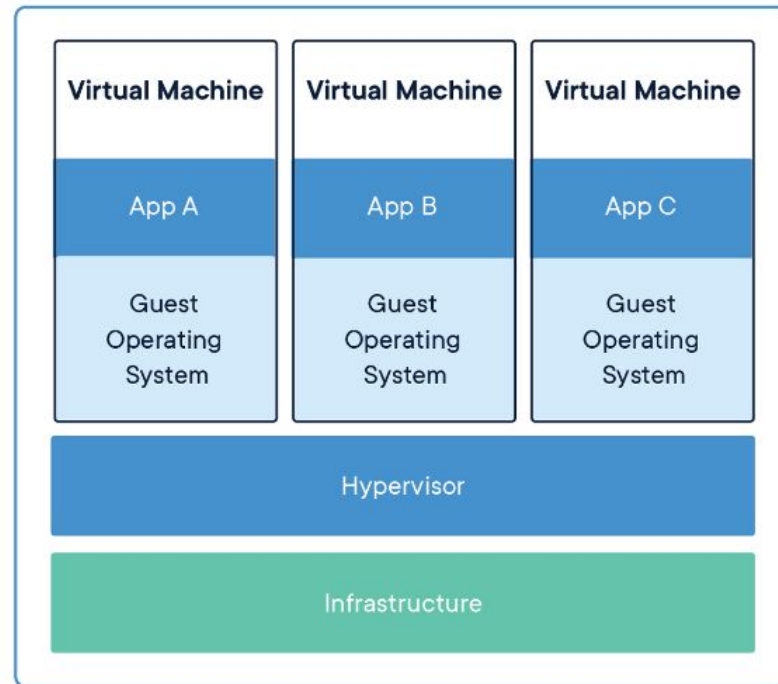
Un Pod es un grupo de uno o más contenedores, con recursos de red y almacenamiento compartidos, y una especificación sobre cómo ejecutar los contenedores.

El contenido de un Pod siempre está ubicado y programado en el mismo lugar, y se ejecuta en un contexto compartido. En contextos que no son de nube, las aplicaciones ejecutadas en la misma máquina física o virtual son análogas a las aplicaciones de nube ejecutadas en el mismo host lógico.





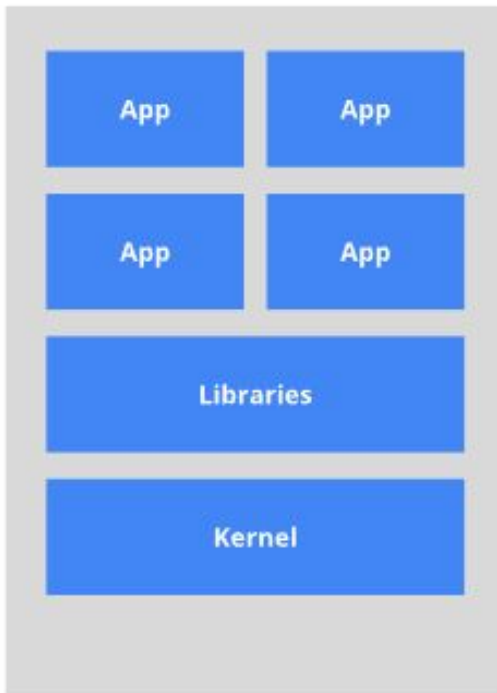
El contenedor es una abstracción en la capa de aplicación. Varias aplicaciones corren en una misma máquina compartiendo el kernel del sistema operativo. La imagen de un contenedor es más pequeña, MBs.



Máquina Virtual es una abstracción que se crea sobre una misma infraestructura que contiene el sistema operativo y diferentes aplicaciones.

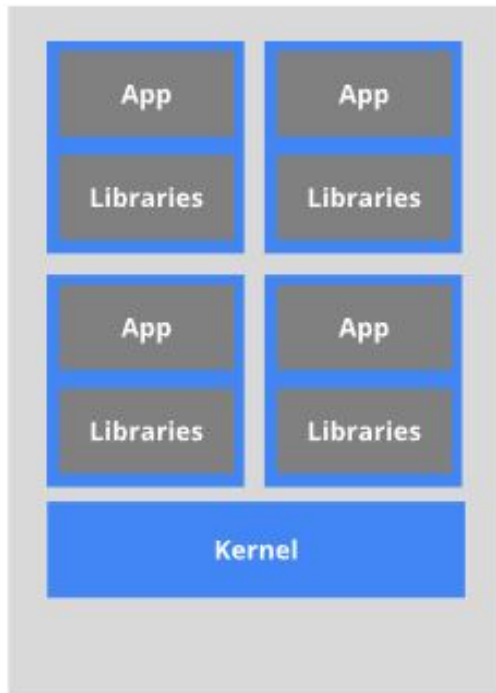
Contenedores

The old way: Applications on host



*Heavyweight, non-portable
Relies on OS package manager*

The new way: Deploy containers



*Small and fast, portable
Uses OS-level virtualization*

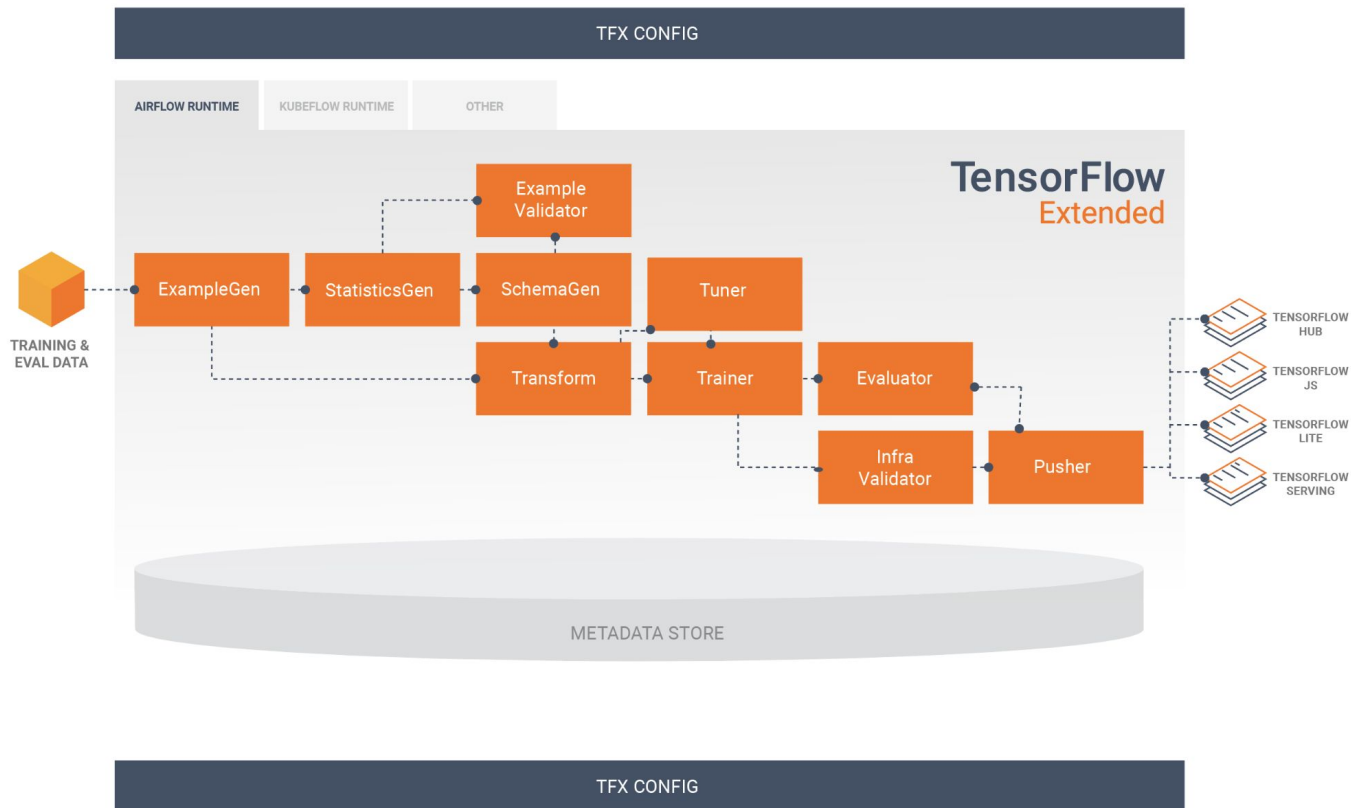
Virtualización a nivel del sistema operativo, en vez del hardware.

Estos contenedores están aislados entre ellos y con el servidor anfitrión: tienen sus propios sistemas de archivos, no ven los procesos de los demás y el uso de recursos puede ser limitado. Son más fáciles de construir que una máquina virtual, y porque no están acoplados a la infraestructura y sistema de archivos del anfitrión, pueden llevarse entre nubes y distribuciones de sistema operativo.

Generar una imagen de contenedor al momento de la compilación permite tener un entorno consistente que va desde desarrollo hasta producción.

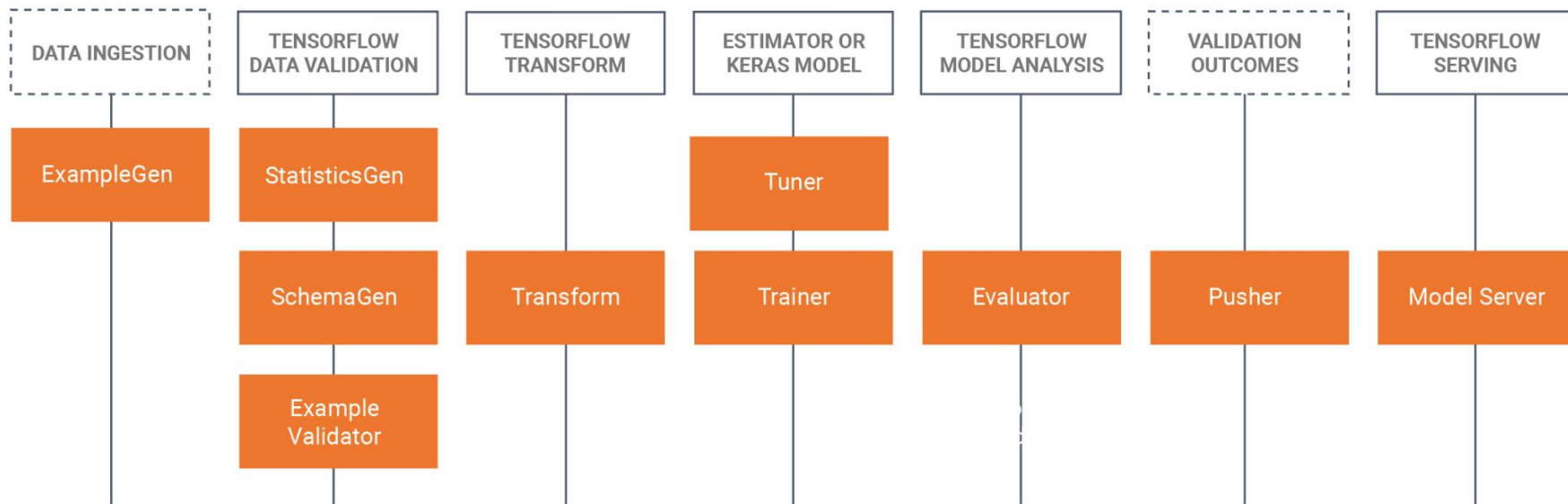
TFX - Tensorflow Extended

Herramienta para crear y gestionar flujos de MLOps.- Basado en Tensorflow.



TFX - Tensorflow Extended

Posee distintos componentes para el armado de flujos de Machine Learning



TFX utiliza [Apache Beam](#) para implementar procesamiento de datos en paralelo.

Opcional: Orquestadores como Apache Airflow y Kubeflow facilitan la configuración, operación, monitoreo y mantenimiento .

Portatil y diseñado para multiples entornos

Material Recomendado de esta semana

[Most Popular Feature Stores In 2023](#)

Chip Huyen, Designing Machine Learning Systems, 2022 - Chapter 4: Training Data - Class Imbalance - Pág. 102-112

Chip Huyen, Designing Machine Learning Systems, 2022 - Chapter 4: Training Data - Data Augmentation - Pág. 113-117

[Kubernetes Tutorial](#)

Otros Recursos

[FEAST - Feature store opensource](#)

[SHAP Interpretable ML Book](#)

[Scott M. Lundberg et. al., A Unified Approach to Interpreting Model](#)

[Predictions, 2017](#)

[SHAP Project API](#)

[Kubeflow Arquitectura](#)

[Apache Airflow Arquitectura](#)

[Docker que es un container](#)

[¿Qué es Kubernetes K8s?](#)

[Borg: The Predecessor to Kubernetes](#)

[TensorFlow Extended TFX](#)