

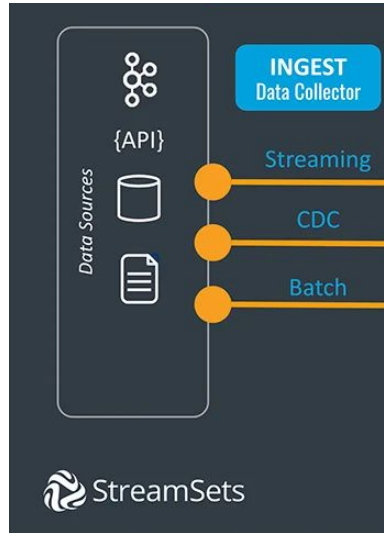
Curso Data Engineer: Creando un pipeline de datos

MÓDULO A - Clase 3

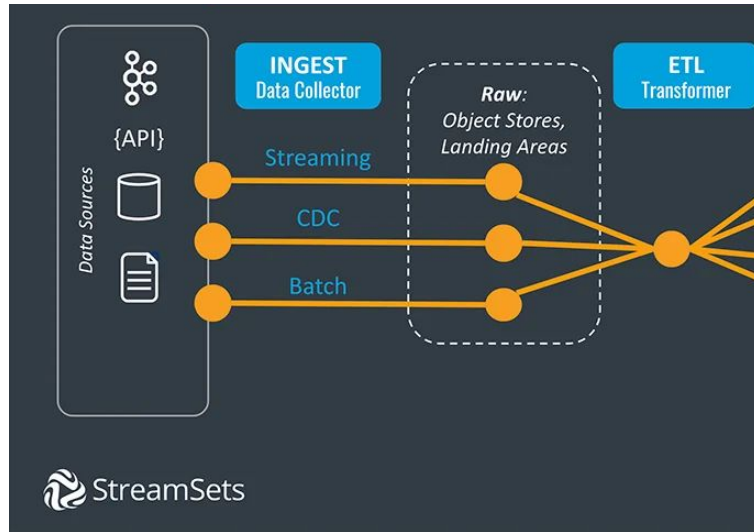


Ecosistema Hadoop

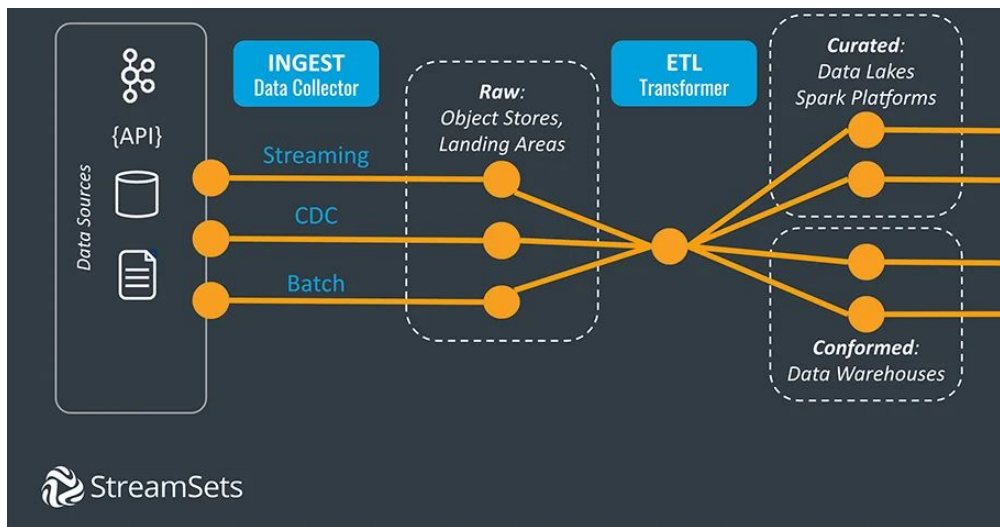
Arquitectura Big Data



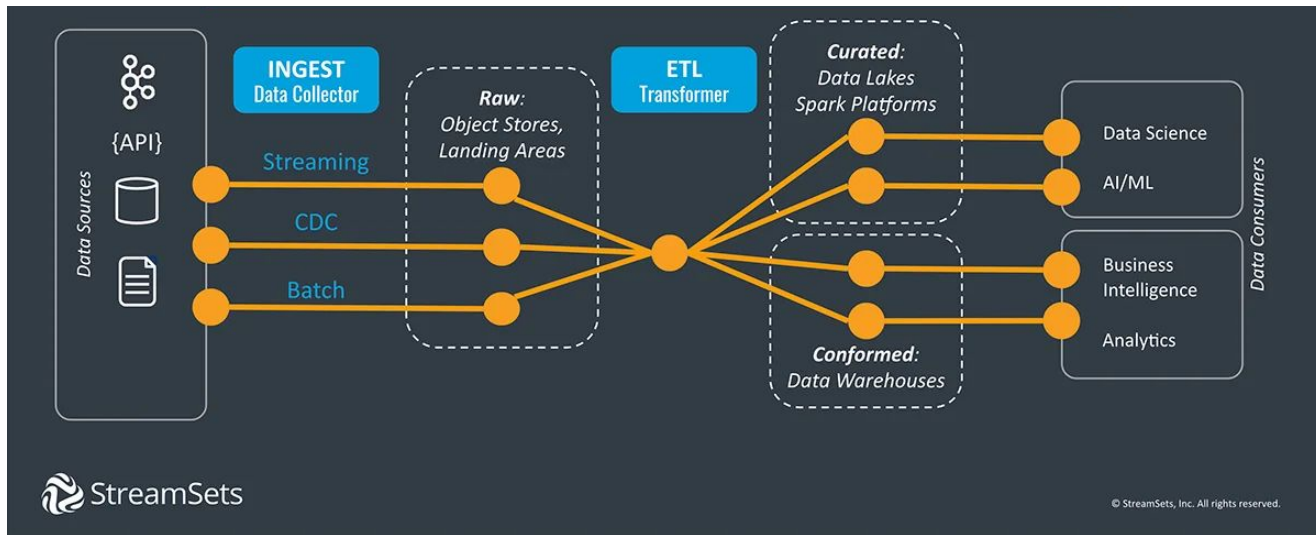
Arquitectura Big Data



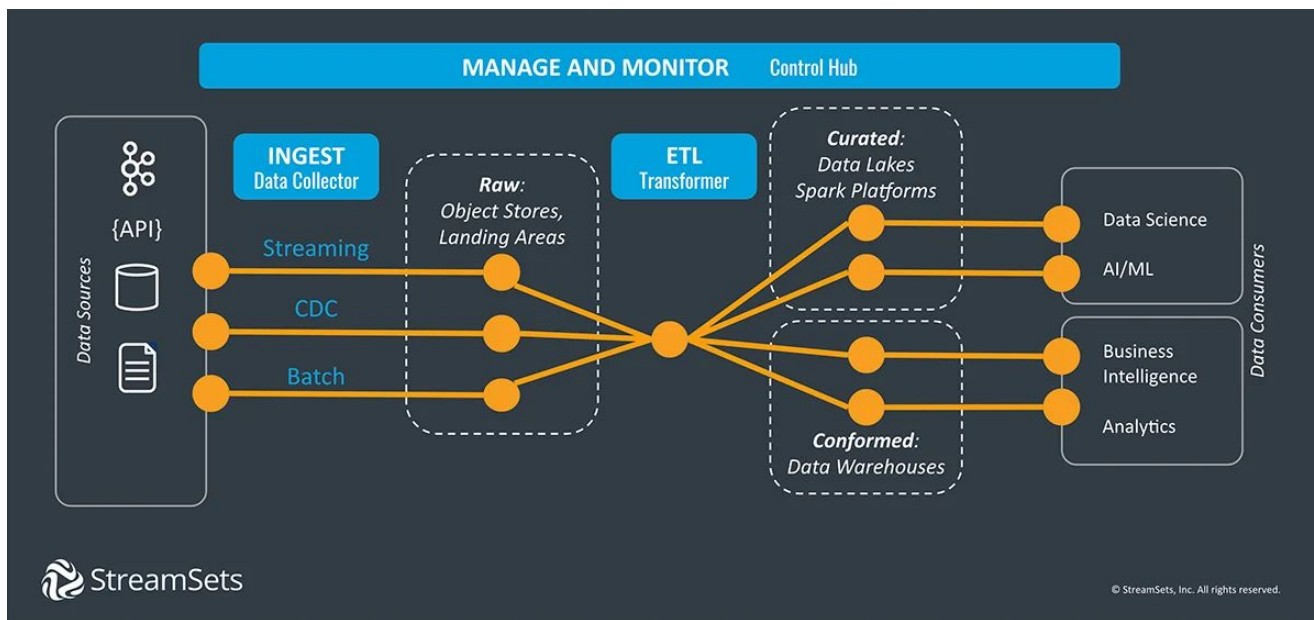
Arquitectura Big Data



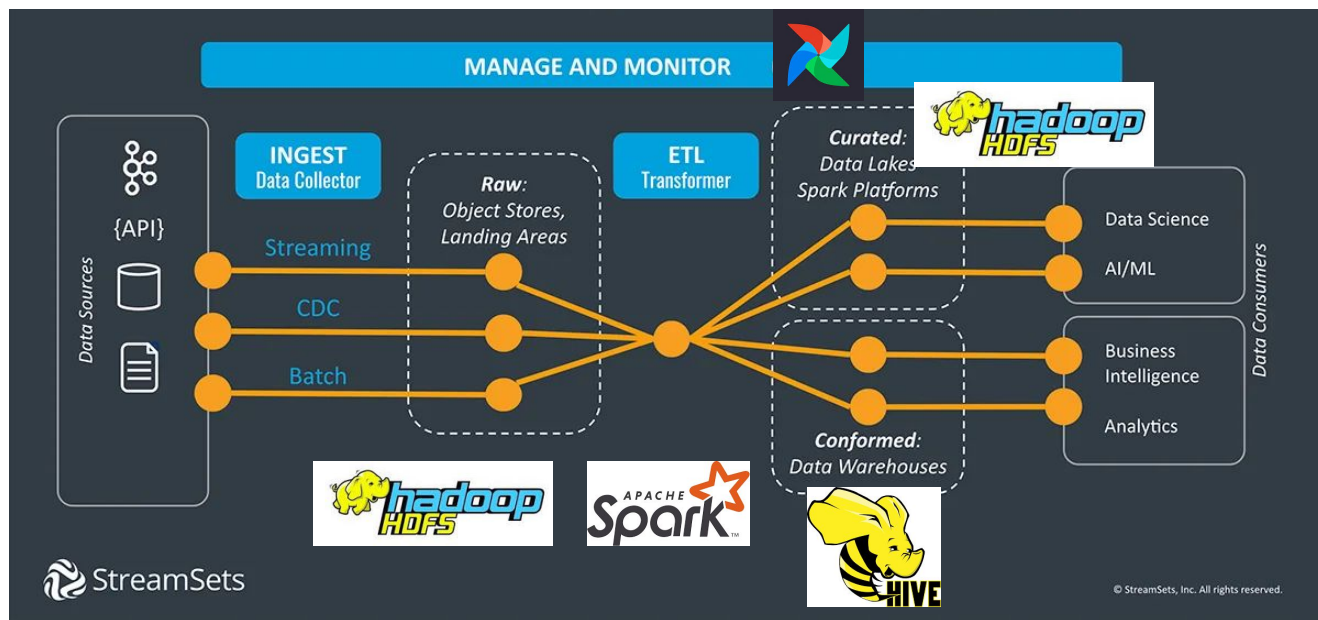
Arquitectura Big Data



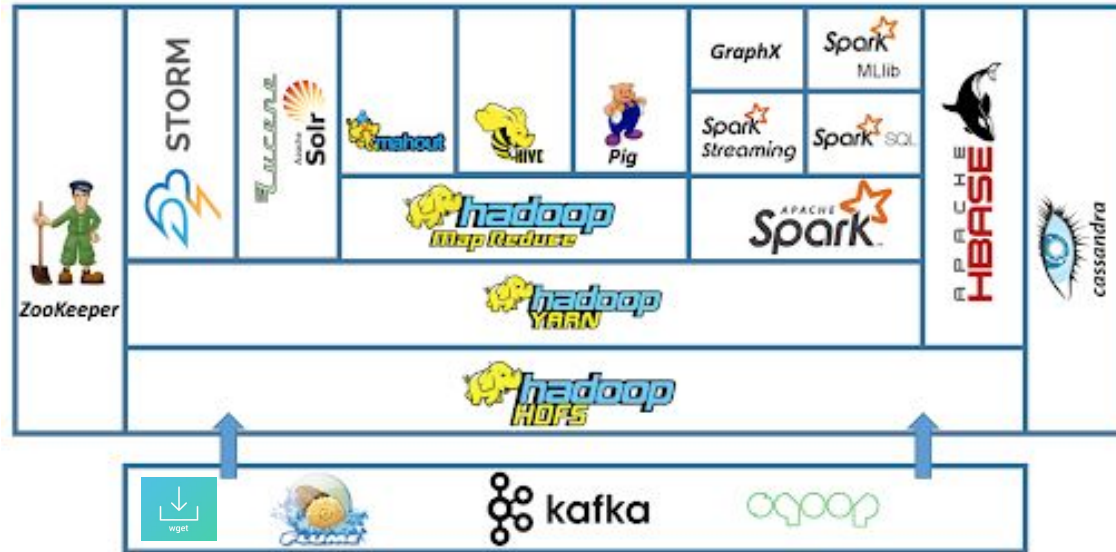
Arquitectura Big Data



Arquitectura Big Data



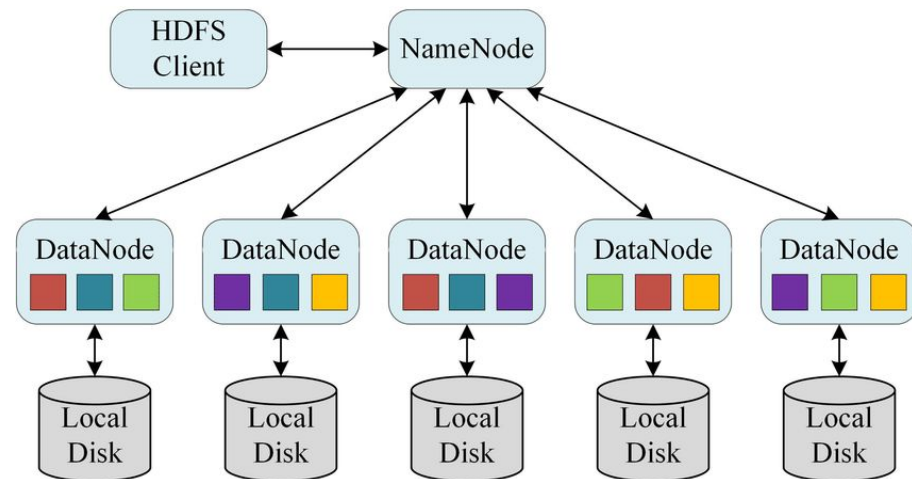
Ecosistema Hadoop



HDFS (Hadoop file system)



- Almacenamiento con tolerancia a fallos
- Almacena en bloques de 128 MB (configurable) en los nodos del cluster
- Escalamiento horizontal (agregar más HDDs o nodos)
- Integridad: almacena 3 copias de cada bloque de datos
- Name Node: gestiona el acceso a los datos y los metadatos, no almacena datos en sí.
- Data Node: nodos del cluster que almacenan información en sus HDDs
- Write once read many: no se pueden editar ficheros almacenados HDFS, pero sí se pueden añadir datos.

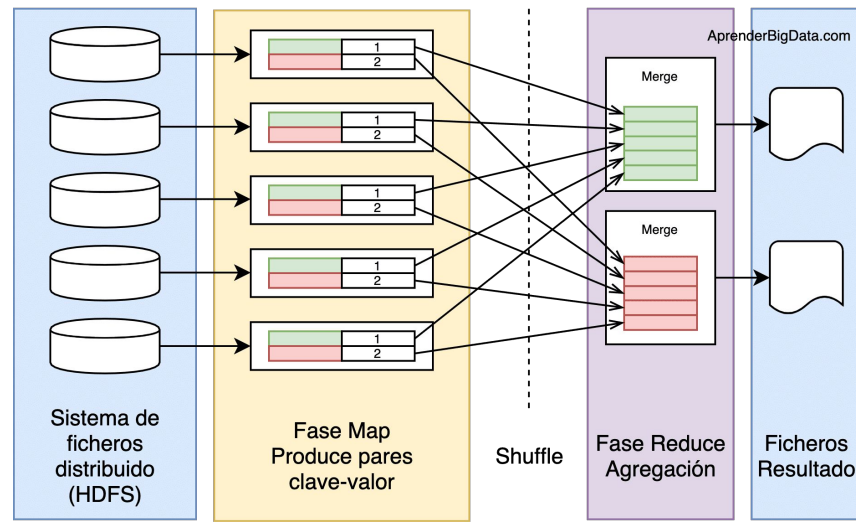


MapReduce

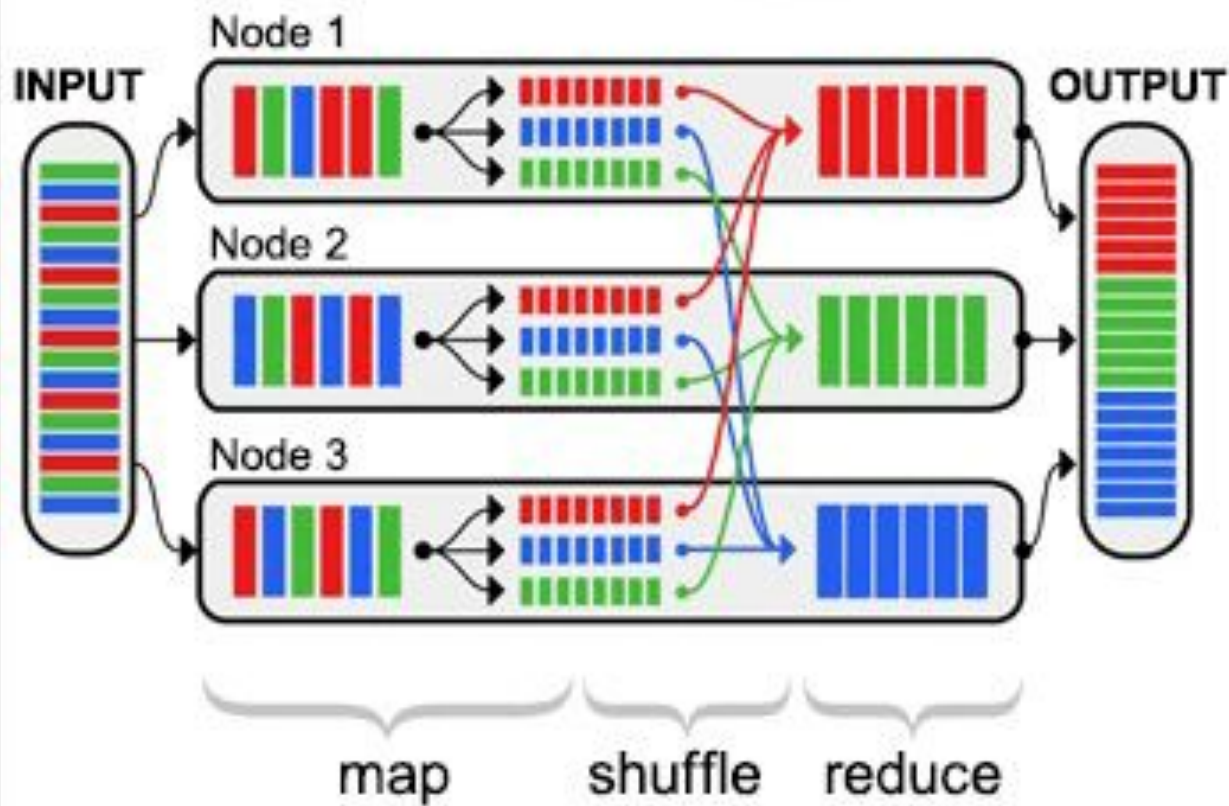


- **Map:** se ejecuta en subtarefas llamadas mappers. Estos componentes son los responsables de **generar pares clave-valor** filtrando, agrupando, ordenando o transformando los datos originales. Los pares de datos intermedios, no se almacenan en HDFS.
- **Shuffle:** (sort) puede no ser necesaria. Es el paso intermedio entre Map y reduce que ayuda a recoger los datos y **ordenarlos** de manera conveniente para el procesamiento. Con esta fase, se pretende agregar las ocurrencias repetidas en cada uno de los mappers.
- **Reduce:** gestiona la **agregación** de los valores producidos por todos los mappers del sistema (o por shuffle) de tipo clave-valor en función de su clave. Por último, cada reducer **genera su archivo** de salida de forma independiente, generalmente **escrito en HDFS**.

Es un paradigma de procesamiento distribuido de datos caracterizado por dividirse en dos fases: Map y Reduce



MapReduce



Map



custId	month	amt	ptype
123098	1	23010.70	Cred
123987	1	1320.50	Cash
123098	2	1500.00	Cash
123098	3	2450.99	Cred
123987	3	1500.00	Cred

Map



```
123098: [23010.70 1500.00 2450.99]  
123987: [1320.50 1500.00]
```

Reduce



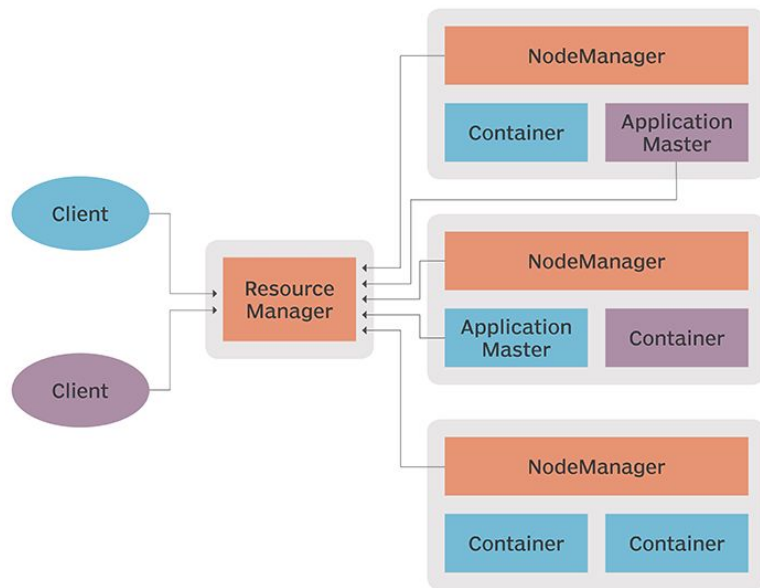
```
123098:26961.69  
123987:2820.50
```

Yarn (Yet Another Resource Negotiator)

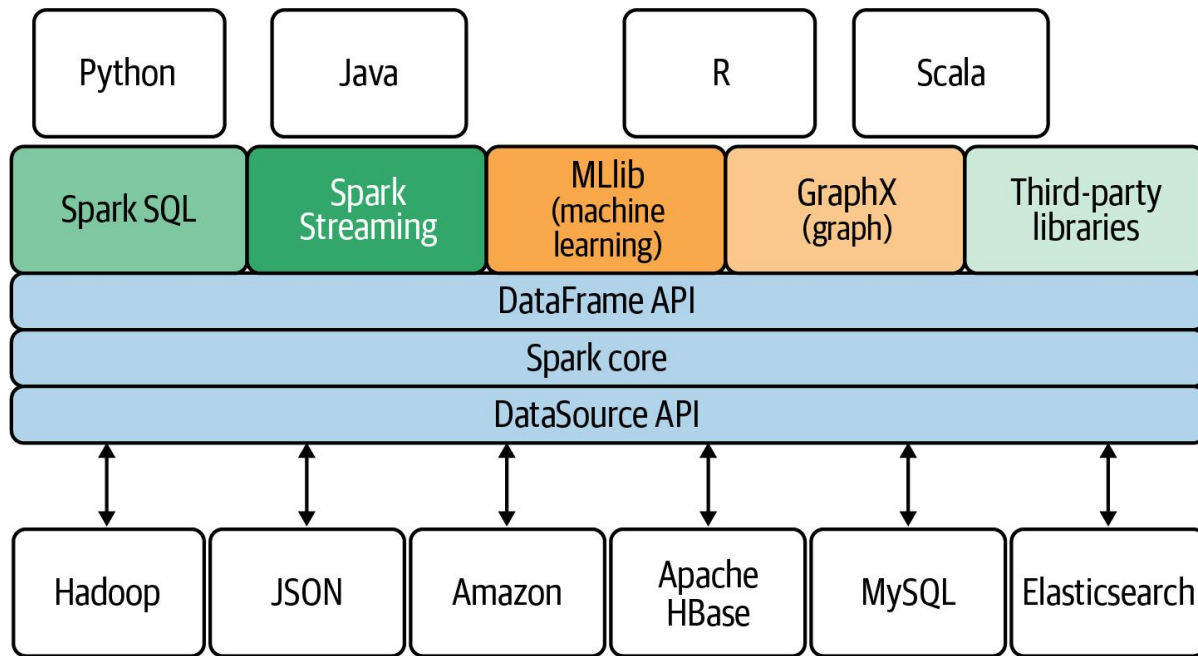


Apache Hadoop YARN **descentraliza la ejecución y el monitoreo de los trabajos** de procesamiento al separar las diversas responsabilidades en estos componentes:

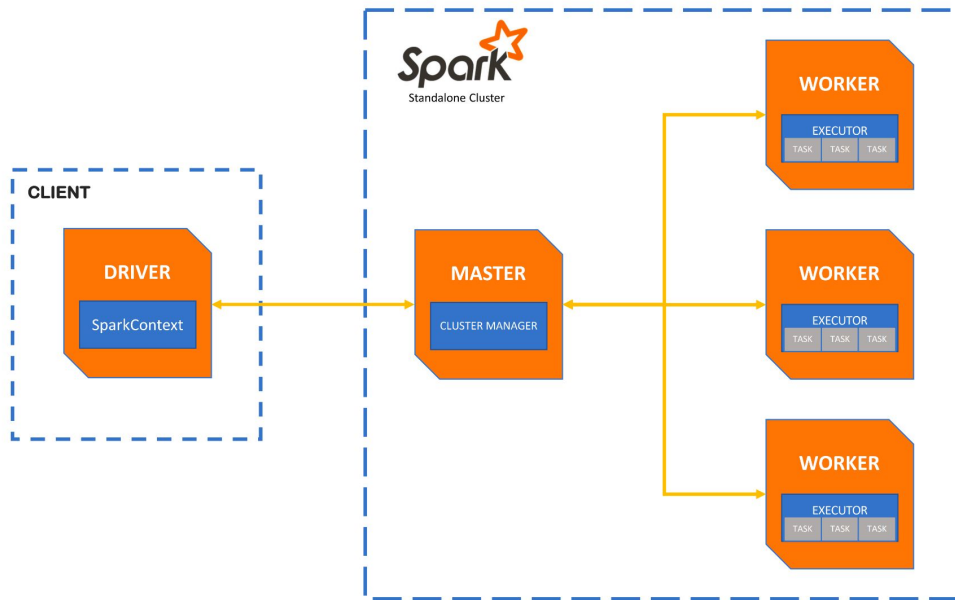
- **ResourceManager:** acepta envíos de trabajos de los usuarios, programa los trabajos y les asigna recursos.
- **NodeManager:** funciona como un agente de supervisión y presentación de informes del ResourceManager
- **ApplicationMaster:** negocia recursos y trabaja con NodeManager para ejecutar y monitorear tareas.
- **Contenedores:** controlados por NodeManagers y asigna los recursos del sistema (CPU cores, RAM, disks) a aplicaciones individuales.



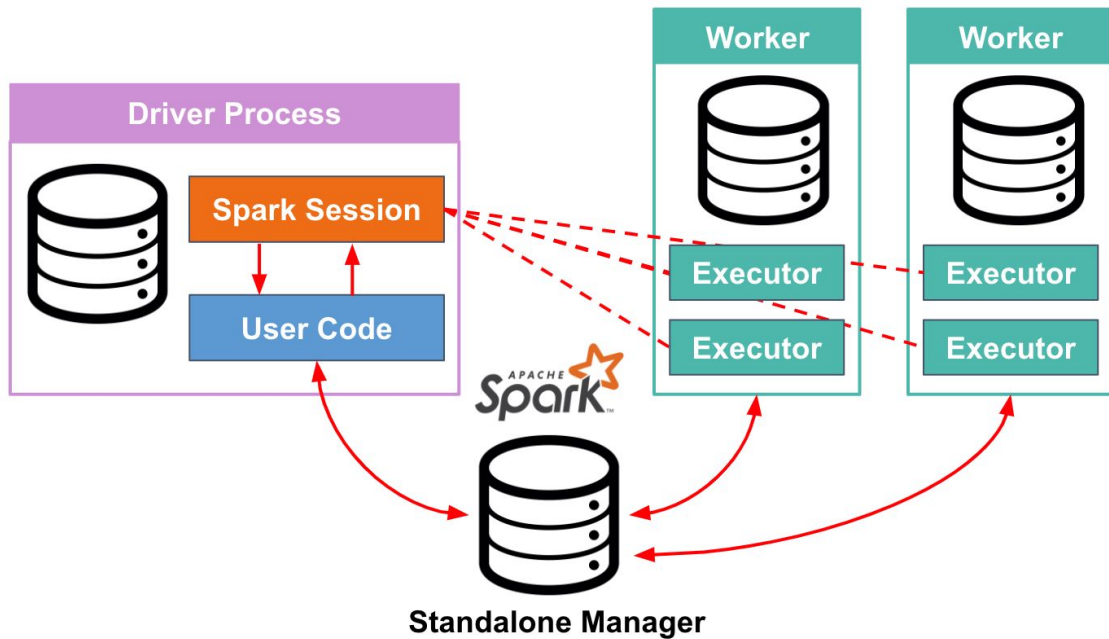
Arquitectura Spark



Spark Master & Workers



Spark Session





Ambiente Hadoop

Docker Hadoop



Bajar la imagen

```
docker pull fedepineyro/edvai_ubuntu:v6
```

Docker Hadoop



Correr la imagen

```
docker run --name edvai_hadoop -p 8081:8081 -p 8080:8080 -p 8088:8088 -p 9870:9870 -p  
9868:9868 -p 9864:9864 -p 1527:1527 -p 10000:10000 -p 10002:10002 -p 50111:50111 -p  
8010:8010 -p 9093:9093 -p 2181:2182 -it --restart unless-stopped  
fedepineyro/edvai_ubuntu:v6 /bin/bash -c "/home/hadoop/scripts/start-services.sh"
```

Docker Hadoop



Ingresar al bash del contenedor

```
docker exec -it edvai_hadoop bash
```

Docker Hadoop



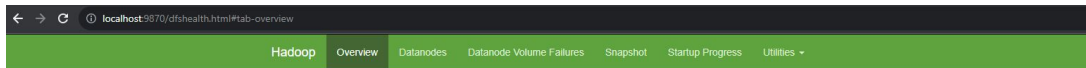
cambiar de usuario (siempre trabajar con el usr hadoop)

su hadoop

Docker Hadoop



Interfaces web: Hadoop HDFS



Overview 'da3d41eed80c:9000' (✓active)

Started:	Tue Oct 24 08:01:35 -0300 2023
Version:	3.3.0, raa9611871bf9858f9bac59c72a81ec470da649af
Compiled:	Mon Jul 06 15:44:00 -0300 2020 by brahma from branch-3.3.0
Cluster ID:	ClD-de8951b9-ed82-4db7-a8d3-6d4d12028e0f
Block Pool ID:	BP-236346611-172.17.0.2-1642895236726

Summary

Security is off.

Safemode is off.

180 files and directories, 68 blocks (68 replicated blocks, 0 erasure coded block groups) = 248 total filesystem object(s).

Heap Memory used 63.62 MB of 170 MB Heap Memory. Max Heap Memory is 982 MB.

Non Heap Memory used 50.7 MB of 54.31 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	1006.85 GB
Configured Remote Capacity:	0 B
DFS Used:	1014.88 MB (0.1%)
Non DFS Used:	9.69 GB
DFS Remaining:	944.95 GB (93.85%)
Block Pool Used:	1014.88 MB (0.1%)
DataNodes usages% (Min/Median/Max/stdDev):	0.10% / 0.10% / 0.10% / 0.00%
Live Nodes	1 (Decommissioned: 0, In Maintenance: 0)

<http://localhost:9870/>


Docker Hadoop



Interfaces web: SPARK

<http://localhost:8080/>

← → ↻ ⓘ localhost:8080

 **Spark Master at spark://da3d41eed80c:7077**

URL: spark://da3d41eed80c:7077
Alive Workers: 1
Cores in use: 2 Total, 0 Used
Memory in use: 2.8 GiB Total, 0.0 B Used
Resources in use:
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

▼ Workers (1)

Worker Id	Address	State	Cores	Memory	Resources
worker-20231024080200-172.17.0.2-43513	172.17.0.2:43513	ALIVE	2 (0 Used)	2.8 GiB (0.0 B Used)	

▼ Running Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------

▼ Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Resources Per Executor	Submitted Time	User	State	Duration
----------------	------	-------	---------------------	------------------------	----------------	------	-------	----------


Docker Hadoop



Interfaces web: HIVE

<http://localhost:10002/>

← → ↻ localhost:10002

 [Home](#) [Local logs](#) [Metrics Dump](#) [Hive Configuration](#) [Stack Trace](#) [Lap Daemons](#)

HiveServer2

Active Sessions

User Name	IP Address	Operation Count	Active Time (s)	Idle Time (s)
Total number of sessions: 0				

Open Queries

User Name	Query	Execution Engine	State	Opened Timestamp	Opened (s)	Latency (s)	Drilldown Link
Total number of queries: 0							

Last Max 25 Closed Queries

User Name	Query	Execution Engine	State	Opened (s)	Closed Timestamp	Latency (s)	Drilldown Link
Total number of queries: 0							

Software Attributes


Attribute Name	Value	Description
Hive Version	2.3.9, r92dd0159f440ca7863be3232f3a683a510a62b9d	Hive version and revision
Hive Compiled	Tue Jun 1 14:02:14 PDT 2021, chao	When Hive was compiled and by whom
HiveServer2 Start Time	Tue Oct 24 08:04:18 ART 2023	Date stamp of when this HiveServer2 was started

Docker Hadoop



Interfaces web: YARN

http://localhost:8088



All Applications

Cluster

About

Nodes

Node Labels

Applications

NEW

NEW SAVING

SUBMITTED

ACCEPTED

RUNNING

FINISHED

FAILED

KILLED

Scheduler

Tools

Cluster Metrics

Apps Submitted	0	Apps Pending	0	Apps Running	0	Apps Completed	0	Containers Running	0 B	Memory Used	4.50 GB	Memory Total	0 B	Memory Reserved	0	VCores Us
----------------	---	--------------	---	--------------	---	----------------	---	--------------------	-----	-------------	---------	--------------	-----	-----------------	---	-----------

Cluster Nodes Metrics

Active Nodes	1	Decommissioning Nodes	0	Decommissioned Nodes	0	Lost Nodes	0	Unhealthy Nodes	0	Reboot
--------------	---	-----------------------	---	----------------------	---	------------	---	-----------------	---	--------

Scheduler Metrics

Scheduler Type	Capacity Scheduler	Scheduling Resource Type	[memory-mb (unit=Mi), vcores]	Minimum Allocation	<memory:1536, vCores:1>	Maximum Allocation	<memory:4608, vCores:4>	0
----------------	--------------------	--------------------------	-------------------------------	--------------------	-------------------------	--------------------	-------------------------	---

Show 20 entries

ID	User	Name	Application Type	Application Tags	Queue	Application Priority	StartTime	LaunchTime	FinishTime	State	FinalStatus	Running Containers	Allocated CPU VCores	Allocated Memory MB	Reserved CPU VCores	Reserved Memory MB	% Q
No data available in table																	

Showing 0 to 0 of 0 entries

Interfaces web: AIRFLOW



← → ↻ localhost:8010/home

Airflow DAGs Security Browse Admin Docs 11:11 UTC

All 33 Active 0 Paused 33

Filter DAGs by tag Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
example-DAG ingest transform	airflow	○○○○	00***		2023-10-22, 00:00:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_branch_operator example example2	airflow	○○○○	00***		2023-10-23, 00:00:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_branch_datetime_operator_2 example	airflow	○○○○	@daily		2023-10-23, 00:00:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_branch_dop_operator_v3 example	airflow	○○○○	*1***		2023-10-24, 11:09:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_branch_labels	airflow	○○○○	@daily		2023-10-23, 00:00:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_branch_operator example example2	airflow	○○○○	@daily		2023-10-23, 00:00:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_branch_python_operator_decorator example example2	airflow	○○○○	@daily		2023-10-23, 00:00:00	○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_complex example example2 example3	airflow	○○○○	None			○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_dag_decorator example	airflow	○○○○	None			○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...
example_external_task_marker_child example2	airflow	○○○○	None			○○○○○○○○○○○○○○○○○○○○	▶ 🗑	...



Ingest

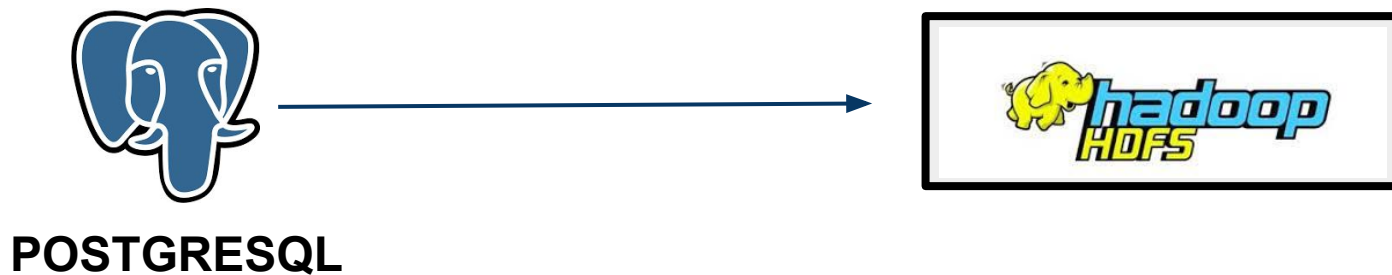
Ingest con scripts



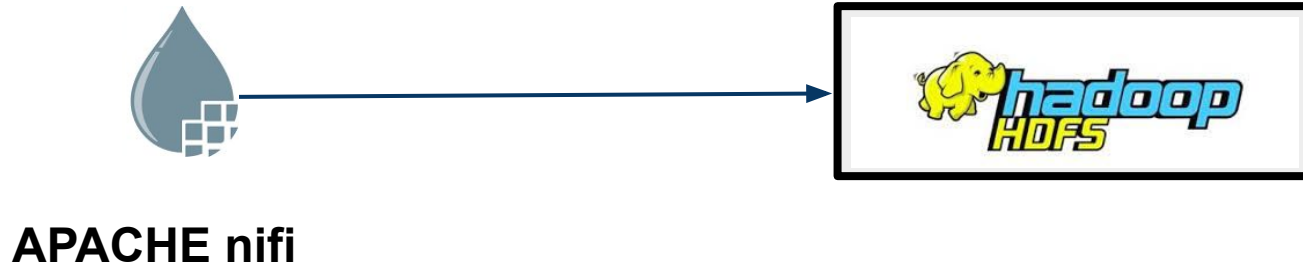
**Blob
Storage**



Ingest con SQOOP



Ingest con APACHE nifi



Ingest mediante scripts



Podemos utilizar algunos comandos de linux para hacer ingest de archivos.

Obtenemos los archivos con WGET:

- **wget -P /home/hadoop/landing**

`https://dataengineerpublic.blob.core.windows.net/data-engineer/yellow_tripdata_2021-01.csv`

Movemos los archivos a HDFS:

- **hdfs dfs -put /home/hadoop/landing/yellow_tripdata_2021-01.csv /ingest**



Ejercicio

Ejercicios



- Habilitar ambiente Hadoop
- Ingest
 - Scripts (WGET y HDFS DFS -PUT)
 - SQOOP (próxima clase)
 - NIFI (próxima clase)