Big Data Project

Synopsis

The goal of the ‘Big Data project’ is to create a simple end-to-end data architecture, including data ingestion, data transformation and data exposition.

In this project, you are free to decide which data you would like to fetch and determine the final output that will create value from your data inputs.
The only necessary thing to respect is a Datalake architecture, which will permit us to structure the data correctly, to have a clean data pipeline and shareable data.

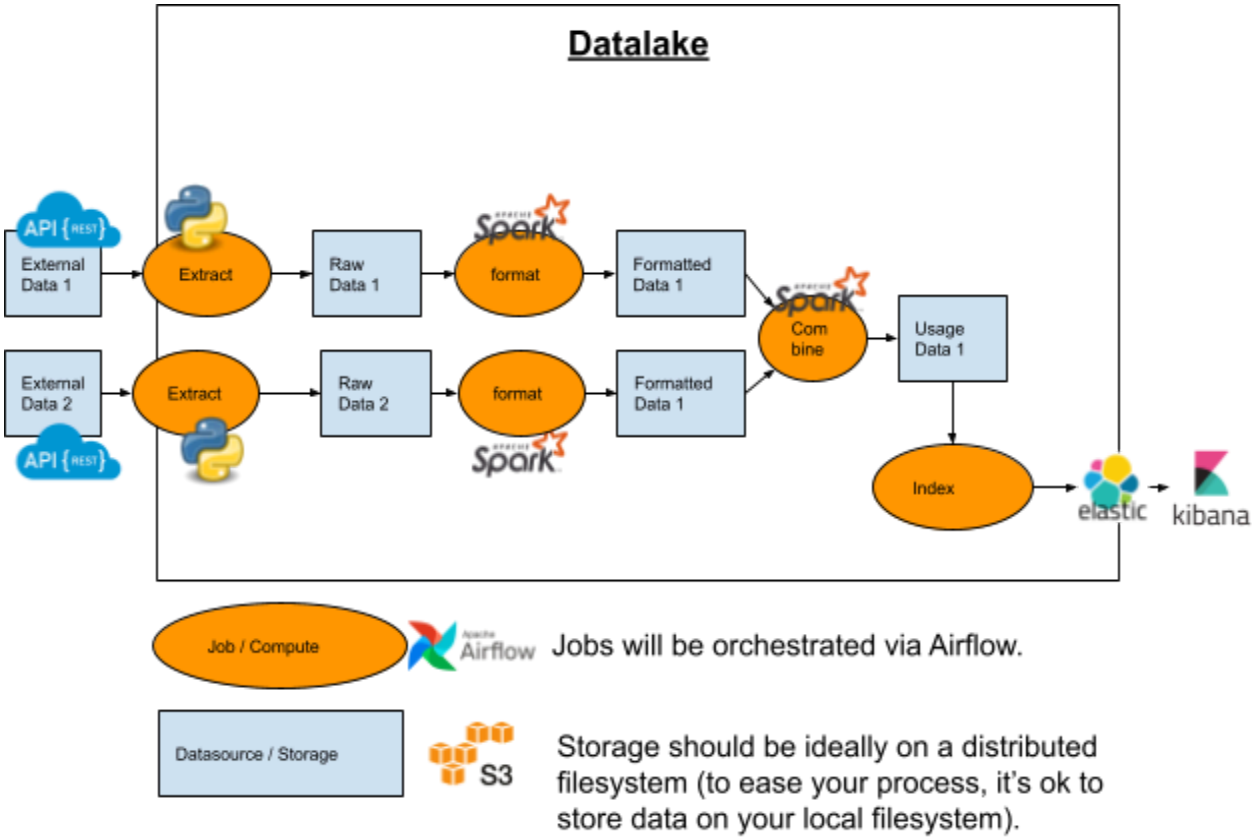
Architecture

Here are the jobs you need to do:

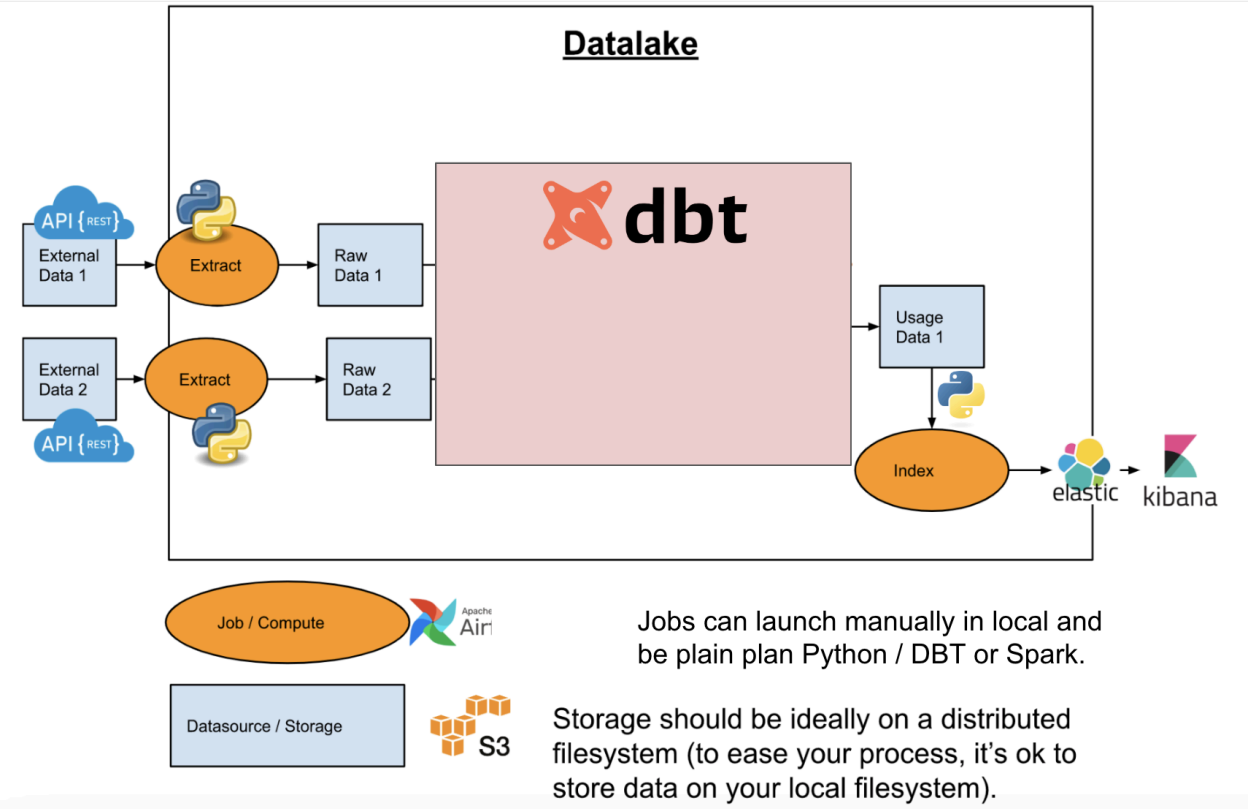
1. Create jobs that extract data from the Internet via REST API.
2. Create jobs that format / normalize these data to transform them into a format suitable to a Datalake / data sharing / data analysis.
3. Create a job that joins / combines your different data sources into a final interesting usage.
4. Index your data in a dashboard to expose your final output to end users.

You have 2 choices of architecture depending on the transformation tool for formatted/combination layers:

1. Using Spark:


Datalake
External Data 1
Extract
Raw Data 1
format
Formatted Data 1
External Data 2
Extract
Raw Data 2
format
Formatted Data 1
Com bine
Usage Data 1
Index
elastic
kibana
Job / Compute
Airflow
Jobs will be orchestrated via Airflow.
Datasource / Storage
S3
Storage should be ideally on a distributed filesystem (to ease your process, it's ok to store data on your local filesystem).

2. Using DBT:



Steps

Here we detailed the necessary steps to realize your project:

Step 1.1: Decide your thema for data

Choose a theme that you like. One of your passions, or something that interests you.

Example: Health, Sport, Cinema, Security (Crimes), Environment, real estate …

Default: If you don't have any idea, we suggest doing it with Cinema.

Step 1.2: Find Datasource to fetch on Internet

Find on Internet data sources you would like to fetch. You should have at least 2 different data sources because the goal of the Datalake is to cross data.
One of the 2 data sources should be refreshed frequently (at least daily refresh), so the dashboard can be interesting to follow every day.

Example:

- open data (https://www.data.gouv.fr/fr/ as example)
- data scraping
- free data apis,
- https://rapidapi.com/
- A list of public apis: https://github.com/toddmotto/public-apis#news
- Social Network data (Twitter tweets, Google trends …)
- …

Step 2: Data Pipeline

We advise you to create your whole data pipeline as 1 Airflow DAG.

Step 2.0: Where to store the data ?

In a real Data Lake, you should store data into a distributed file-system (S3 compatible like OVH storage, GCS …). But for your project, we allow you to use your local file-system (therefore not distributed) in order to ease your life. Using a distributed FS is considered as a bonus.

However in both case, it's very important to respect a clean folder hierarchy as seen in the lecture here Data Lake V2.2, and practical lesson here:
Apache Airflow & Big Data project

Step 2.1: Ingestion

Here you'll create scripts to fetch data on the dedicated APIs and store them into your datalake.

Tips:

- With Python, you can find an SDK to help you fetch data from different platforms. Example with Twitter: https://developer.twitter.com/en/docs/twitter-api/tools-and-libraries/v2 this will be easier than doing manual HTTP requests (not possible since 2023 because of Elon Musk).
- There is many tools to help scrap the web, beautifulSoup is one them: https://python.doctor/page-beautifulsoup-html-parser-python-library-xml

Step 2.2: Formatting

When your data arrives in the Data Lake, it's in CSV, JSON… It's not a good format to do data analysis. You should therefore prepare your data in the layer "formatted/normalize" to convert in the necessary format (parquet ?), apply transformation (date in UTC for example), and also normalize the columns (each column should be clean and have clean values, to be easily re-usable).

You can do it in pure python with the pandas library but the best would be to use parallel scalable processing (Spark or DBT).

Step 2.3: Combination

This is all the goal of the Datalake, being able to cross multiple different data sources to create value.

Your goal is to create a job that combines / joins your different data sources in one final output that creates value.

The job can be a simple join and then do some aggregation to generate KPI (sum, count, average…).

Example:

- Do recommandation
- Anomaly detection
- Fraud analysis
- Resources optimization

Links to example:

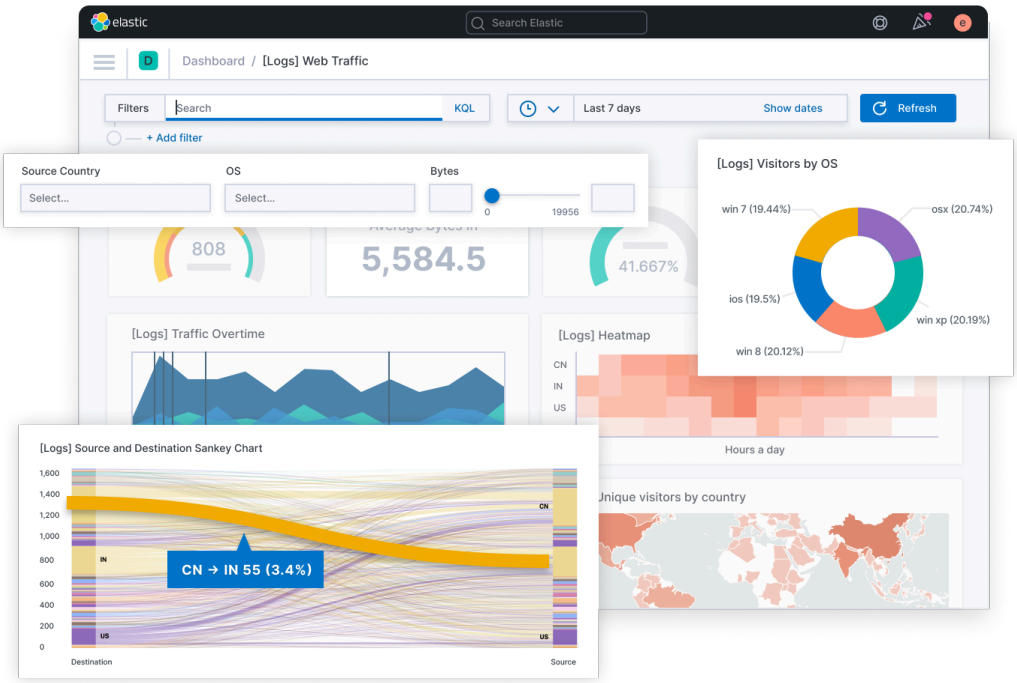
- https://www.upgrad.com/blog/big-data-project-ideas-beginners/
- https://www.projectpro.io/article/top-20-big-data-project-ideas-for-beginners-in-2021/426
- https://medium.com/@LearnBay.co/big-data-is-an-exciting-subject-of-information-technology-cae55cb583ab

Default:

- Recommendation of movies

Step 2.4: Indexing

Index your result in Elasticsearch to expose the results in a Kibana dashboard.



Your deliverable

You should deliver your work in 3 parts:

- One PDF where you explain the different parts of your work (10 pages maximum)
- One video where you show your work (10 min maximum)
- One zip containing all your code

Score / Marks

Note: The sum of the points is more than 20, you are free to pick whatever bonus you'll like. The green cells are the mandatory part to do for the project.

	Beginner	**Advanced**	**Expert**

Data pipeline			
Ingestion	- Ingestion your N data sources in your Data Lake as files (2 points)	- Ingest in a distributed file system (1 point)*	- Realtime** via Kafka (1 point) - Ingestion via Airbyte (1 point)
Formatting	- If using Spark: Transform data sources in parquet format (2 points) - If using DBT: Ingest files to the SQL db	- Add fields normalization like date in UTC, in raw => formatted job (1 point)	- Use Spark (1 point)
Combination	- Join your data source to produce an output creating value (2 points)	- Use Spark (0.5 point)	- Use machine learning to produce the result (1 point)
Indexing	- Index data in Elastic (2 points)		
Data Viz / Dashboarding	- Create a dashboard on top of your result (2 points)		- Do a realtime** dashboard (1 point)
Bonus			
Clean Naming / Convention	- Organize your data lake in a very clean manner, the naming conventions should be perfectly respected (1 point) data/<layer>/<group>/<dataEntity>/<?dateVersion>/<?Partition>*		
Video presentation	0.5 point	1 point	1.5 points
Run all in once	You should be able to launch the DAG and all data is ingested/transform/combined/indexed In one time (1 point)		
Output result Value produced	- Default (0 points)	- Interesting result invented by yourself (1.5 point)	- Very Innovative result (3 points)

Use DBT instead of Spark		- From Formatting to combination (1,5 points)	
Write a blog post about your project and send the link.		(1 point)	
Deploy the project on the cloud and give the link of the running project			1 point

* Ingest in a distributed file system: It could be S3 / OVH storage / GCS. In local you emulate a local S3 with https://github.com/localstack/localstack

** Realtime: Equal or Less than 1 min to refresh