**Covid-19: Cases Data Project(Batch processing)**

What is Batch Processing?

Data which is collected over time and then fed into a system where “meaning” is extracted and further it is stored(warehouses) or it is used to process.

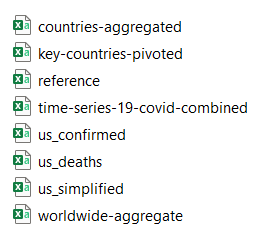
For e.g., *Daily processing of bills in a retail shop*.

**Context:**

* A new coronavirus designated 2019-nCoV was first identified in Wuhan, the capital of China's Hubei province.
* People developed pneumonia without a clear cause and for which existing vaccines or treatments were not effective.
* The virus has shown evidence of human-to-human transmission.
* Transmission rate (rate of infection) appeared to escalate in mid-January 2020
* As of 15th May 2023, approximately 69,27,378 deaths have been confirmed.

**Dataset:**

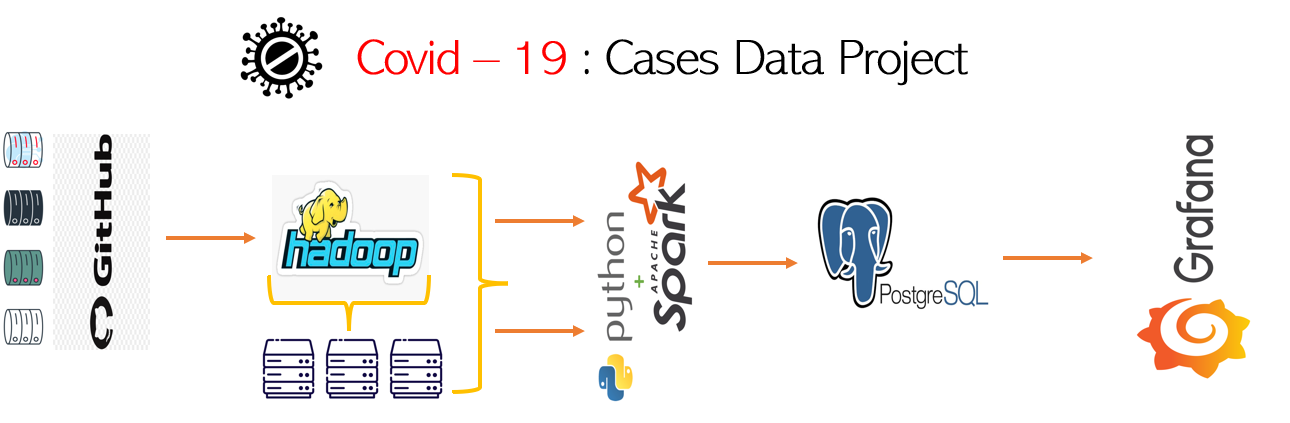
* There are 2,31,745 rows present in the dataset.
* There are 6 columns(Date, Country, State, Confirmed, Recovered, Deaths) in the dataset.
* Data present in the dataset are from **2020 To 2022**.
* We will be considering “time-series-19-covid-combined” table.



**Tech Stack:**

1. Hadoop with two main components(HDFS & YARN) \*
2. Python\*
3. Anaconda(IDE)\*
4. Spark/PySpark\*
5. SQL(PostgreSQL/MySQL)\*
6. NoSQL(Cassandra/HBase)
7. Dashboarding(Tableau, Power-BI, Grafana\*, Kibana)

**High-Level Diagram:**



**Milestones:**

* **Milestone – 1**

- Clone/Download file from Git repo & save it in your local.

- [GitHub - datasets/covid-19: Novel Coronavirus 2019 time series data on cases](https://github.com/datasets/covid-19)

* **Milestone – 2**

- Setup Hadoop in local.

- Here we need to focus on only two main components of Hadoop(HDFS &

YARN)

- HDFS is distributed file system so now copy that file from local to HDFS.

*- hdfs dfs mkdir /dir\_name*

*- hdfs dfs -ls /*

*- hdfs dfs -copyFromLocal “path” /dir\_name\_in\_Hadoop*

*- hdfs dfs -ls /dir\_name*

* **Milestone – 3**

- Setup spark in your local.

* We need to install Spark in our local and further we need to instantiate.
* To initiate we need we can use “spark” in the CMD, or we can use it in Jupyter Notebook.

- Write a spark application which reads file from HDFS and creates a DataFrame.

a. With the help of Spark we could pull the data from HDFS(Hadoop Distributed File

System) and create a DataFrame.

* Just looking a quick view, we can understand how many columns are there and we can plan what type of further transformation we could do.
* **Milestone – 4**
* Following basic transformation on the data:

1. Ingesting only the required columns.
2. Dropping duplicate values
3. Null handling transformation for all the columns.

String: NA

INT: 0

Float: 0.0

Date/Timestamp: 1800-01-01

1. Column renaming
2. Type casting according to the required data type.

* **Milestone – 5**
* Aggregate data in spark application.

1. *How many total cases so far?*
2. *How many recovered cases so far?*
3. *How many deaths so far?*

* **Milestone – 5**
* Write the aggregated data in output table using transactional database like MySQL/Postgres or in NoSQL database.
* **Milestone – 6**
* Use dashboard tool: Grafana
* Plugins: World-map/ Geo-Map
* Dashboard Link: <http://localhost:3000/goto/xd4_gDQ4R?orgId=1>





**Ref Link:**

1. [2 Beginner Level Projects For Aspiring Data Engineers ⌛️Covid-19 Data | Real Time Data Processing 👈 - YouTube](https://www.youtube.com/watch?v=GmR2migiu6A)
2. [PySpark GroupBy Agg | Working of Aggregate with GroupBy in PySpark (educba.com)](https://www.educba.com/pyspark-groupby-agg/)
3. [PySpark Groupby Agg (aggregate) - Explained - Spark By {Examples} (sparkbyexamples.com)](https://sparkbyexamples.com/pyspark/pyspark-groupby-agg-aggregate-explained/)
4. [PySpark Aggregate Functions with Examples - Spark By {Examples} (sparkbyexamples.com)](https://sparkbyexamples.com/pyspark/pyspark-aggregate-functions/#grouping)
5. [PySpark Aggregate Functions: A Comprehensive Guide | by Ahmed Uz Zaman | Medium](https://medium.com/@uzzaman.ahmed/pyspark-aggregate-functions-a-comprehensive-guide-8ed821dfe0cc)
6. [PySpark alias() Column & DataFrame Examples - Spark By {Examples} (sparkbyexamples.com)](https://sparkbyexamples.com/pyspark/pyspark-alias-column-examples/)
7. [Pyspark: GroupBy and Aggregate Functions | M Hendra Herviawan (hendra-herviawan.github.io)](https://hendra-herviawan.github.io/pyspark-groupby-and-aggregate-functions.html)
8. [Load DataFrames To PostgreSQL 10x Faster | Towards Data Science](https://towardsdatascience.com/upload-your-pandas-dataframe-to-your-database-10x-faster-eb6dc6609ddf)
9. [python - How to write DataFrame to postgres table - Stack Overflow](https://stackoverflow.com/questions/23103962/how-to-write-dataframe-to-postgres-table)
10. [Pyspark write to postgres - spark write to postgres - Projectpro](https://www.projectpro.io/recipes/save-dataframe-postgresql-pyspark)
11. [postgresql - unable to connect to server for Postgres - Stack Overflow](https://stackoverflow.com/questions/40532399/unable-to-connect-to-server-for-postgres)
12. [python - DataFrame constructor not properly called - Stack Overflow](https://stackoverflow.com/questions/52946094/dataframe-constructor-not-properly-called)
13. [postgresql - OperationalError: (psycopg2.OperationalError) could not translate host name "143@postgres" to address: Unknown server error - Stack Overflow](https://stackoverflow.com/questions/72049617/operationalerror-psycopg2-operationalerror-could-not-translate-host-name-143)