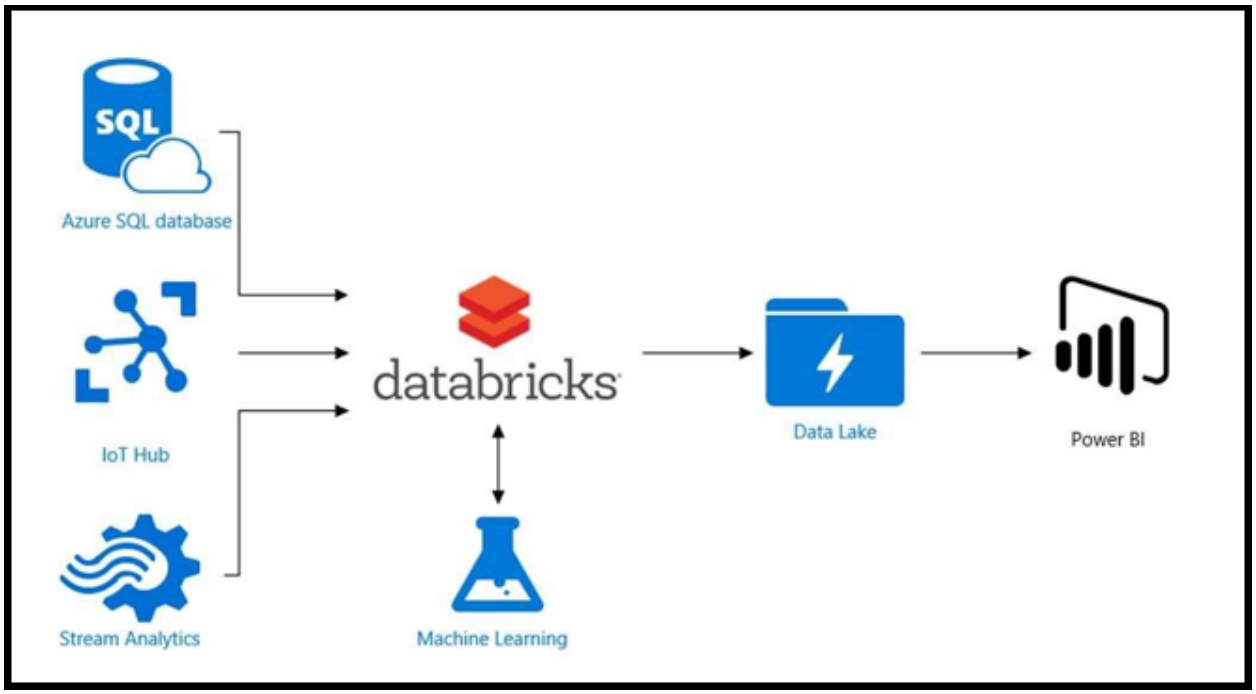
COVID-19 Data Analytics and Reporting with Azure Databricks and Power BI using Python

Problem Statement

In the midst of all of the bad news in COVID era, there are good news reports of the important work done by data engineers and data scientists to get the data of all COVID cases around the world together and provide useful insights which will be useful for pan professionals.

The objective of this article is to focus on a use case that demonstrates the integration between daily changing Source, Azure Databricks and Power BI to deliver insights and data visualizations using a publicly available COVID-19 dataset. While Azure Databricks provides the distributed computing power to process and transform complex datasets, Power BI is a fitting recipient of the transformed dataset that surfaces these insights to business users.

Solution Arhcitecture:



Source

The latest available public data on the geographic distribution of COVID-19 cases worldwide from the European Center for Disease Prevention and Control (ECDC). Each row/entry contains the number of new cases reported per day and per country or region.

Before we start with our exercise, we will need to have the following prerequisites:

1. You need to have an active **Azure Subscription.**

2. **Azure Databricks** – You need to set up both Databricks service and cluster in Azure, you can go over the steps in this article. As shown in this article, we have created a Databricks service.

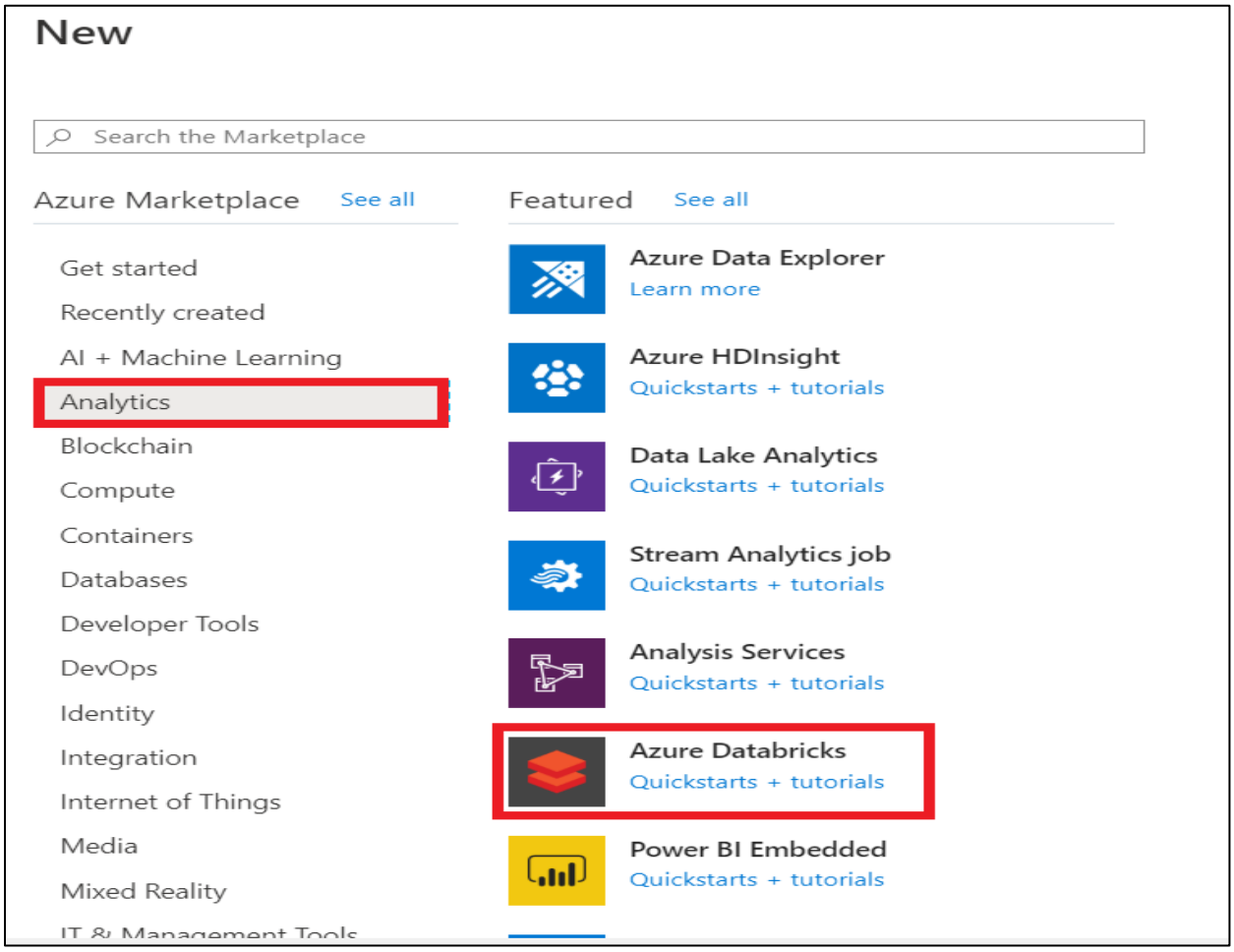
3. **Power BI Subscription**

Let's go ahead and demonstrate the data load into Databricks using Python notebooks from the source mentioned above.

STEP 1: Create Azure Databricks

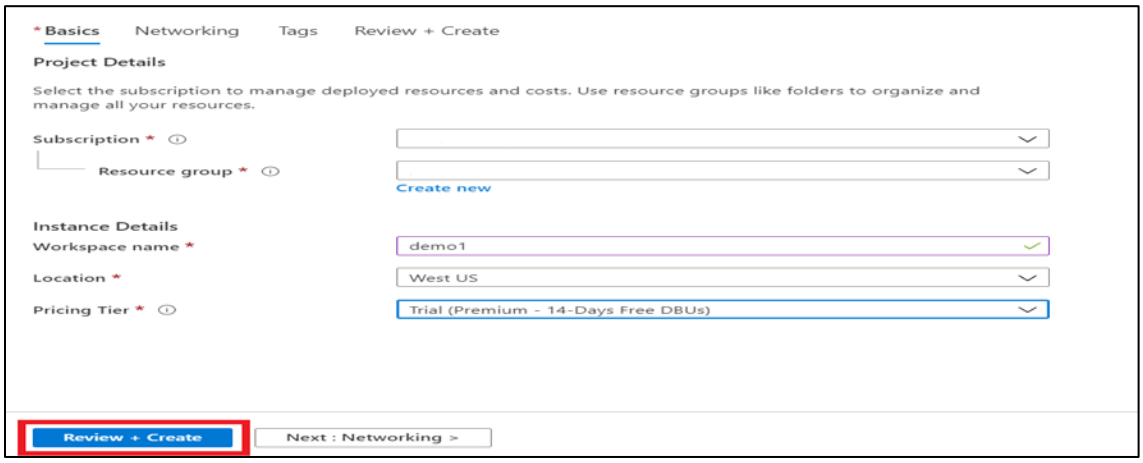
The next step is to create a Databricks Workspace. You can think of the workspace like an application that you are installing within Azure, where you will access all of your Databricks assets. Follow these steps to create a workspace:

- On the Azure home screen, click **'Create a Resource'.**
- In the '**Search the Marketplace'** search bar, type **'Databricks'** and you should see **'Azure Databricks'** pop up as an option. Click that option.
- Click **'Create'** to begin creating your workspace.



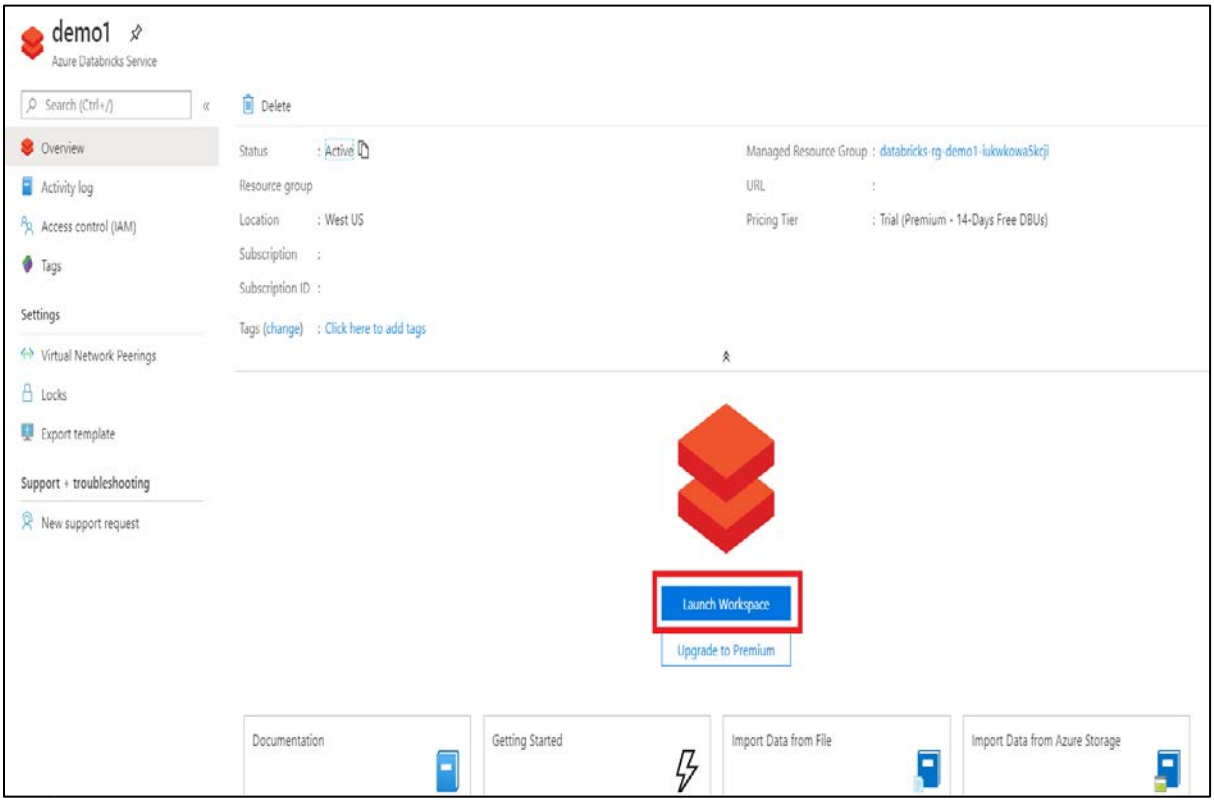
Use the same resource group you created or selected earlier. Then, enter a workspace name. Remember to always stick to naming standards when creating Azure resources, but for now, enter whatever you would like.

You can keep the location as whatever comes default or switch it to a region closer to you. For the **pricing tier**, select **'Trial'**. Finally, select **'Review and Create'**. We can skip-networking and tags for now which are for more advanced set-ups.

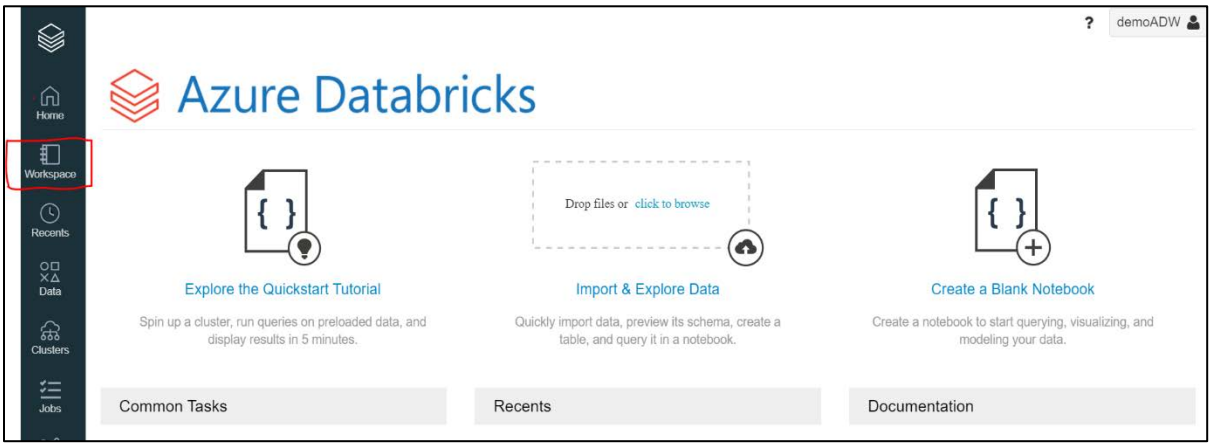


- This should bring you to a validation page where you can click **'create'** to deploy your workspace. This will bring you to a deployment page and the creation of the workspace should only take a couple of minutes.

- Once the deployment is complete, click **'Go to resource'** and then click **'Launch Workspace'** to get into the Databricks workspace.



- Create a Workspace (Notebook) with Python or your choice of language for performing analysis.



STEP 2: Start Scripting in Notebook

- Import relevant libraries and fetch data from the source according to its data type.

```
1  import seaborn as sns
2  import plotly.express as px
3  %matplotlib inline
4  import pandas as pd
5  import numpy as np
6  %matplotlib inline
7  import matplotlib.pyplot as plt
8
9  df = pd.read_csv("https://pandemicdatalake.blob.core.windows.net/public/raw/covid-19/ecdc_cases/latest/ECDCCases.csv")
10 df.head(1000)
```

Out[1]:

	dateRep	day	month	year	cases	deaths	countriesAndTerritories	geoId	countryterritoryCode	popData2019	continentExp	Cumulative_number_for_14_days_of_COVID-19_cases_per_100000
0	08/11/2020	8	11	2020	126	6	Afghanistan	AF	AFG	38041757.0	Asia	3.656508
1	07/11/2020	7	11	2020	58	2	Afghanistan	AF	AFG	38041757.0	Asia	3.538217
2	06/11/2020	6	11	2020	40	0	Afghanistan	AF	AFG	38041757.0	Asia	3.546103
3	05/11/2020	5	11	2020	121	6	Afghanistan	AF	AFG	38041757.0	Asia	3.745884
4	04/11/2020	4	11	2020	86	4	Afghanistan	AF	AFG	38041757.0	Asia	3.782685

- Digging the data to get useful and relevant analytical data
    - Top Countries with cases and deaths.

Top Countries

```
import plotly.graph_objects as go
import plotly.express as px
import matplotlib.pyplot as plt

df.loc[: , ['countriesAndTerritories', 'cases', 'deaths']].groupby(['countriesAndTerritories']).max().sort_values(by='cases',ascending=False).reset_index()[:15].style.background_gradient(cmap='rainbow')
```

Out[42]:

	countriesAndTerritories	cases	deaths
0	United_States_of_America	184813	4928
1	India	97894	2003
2	France	86852	2004
3	Brazil	69074	1595
4	Spain	55019	1623
5	Russia	45188	745
6	Italy	40902	971
7	Chile	36179	1057
8	United_Kingdom	33470	1224
9	Poland	27875	603
10	Germany	23542	315
11	Belgium	22189	322
12	Switzerland	21642	197
13	Kazakhstan	19246	324
14	Argentina	18326	3351

Command took 0.05 seconds -- by ayush.chauhan@iiti.ac.in at 11/20/2020, 3:08:21 PM on democluster

- Calculating total cases, deaths, and death rate around the globe.

Cmd 8

```
cv19_countries_day = df.groupby(by=['dateRep','countriesAndTerritories']).sum()[['cases','deaths']]

Total_confirmed = cv19_countries_day.groupby('dateRep').sum()[['cases','deaths']].sum()['cases']
Total_deaths = cv19_countries_day.groupby('dateRep').sum()[['cases','deaths']].sum()['deaths']

dicc = {'TotalConfirmed' : Total_confirmed, 'TotalDeaths' : Total_deaths, 'DeathRate' : round((Total_deaths/Total_confirmed)*100,2)}
total = pd.DataFrame(dicc,index=['Counter'])[['TotalConfirmed','TotalDeaths','DeathRate']]


total.style.set_properties(**{
    'background-color': 'white',
    'font-size': '20pt',
    'color' : 'red'
})
```

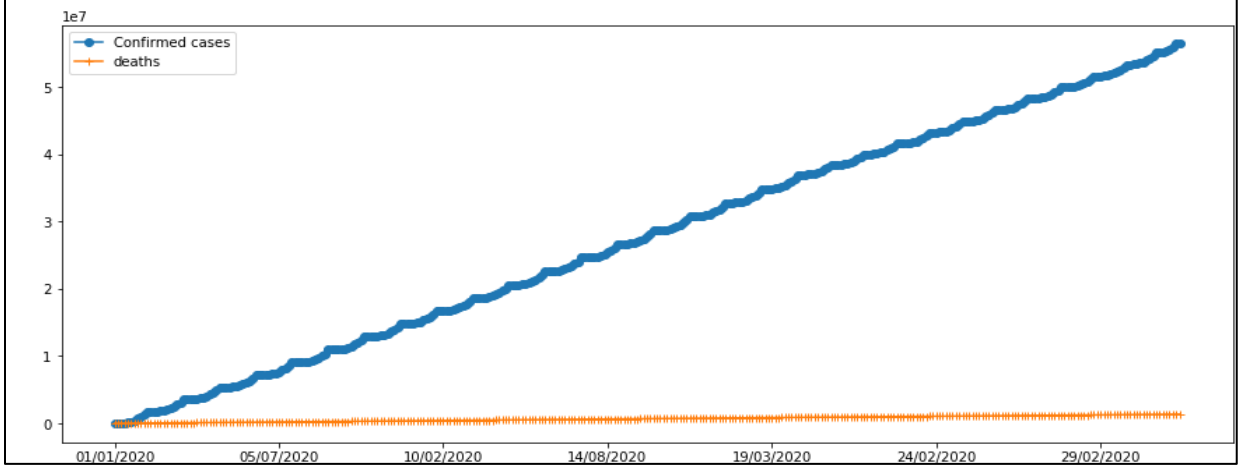
Out[6]:

	TotalConfirmed	TotalDeaths	DeathRate
Counter	56368586	1350713	2.400000

- Visualizing total cases and deaths with a timeline.

Cmd 13

```
cv19_countries_day = df.groupby(by=['dateRep']).sum()[['cases','deaths']]
cv19_countries_day['cases'].cumsum().plot(figsize=(15,6),label='Confirmed cases',marker='o')
cv19_countries_day['deaths'].cumsum().plot(label="deaths",marker='+')
plt.legend()
plt.xlabel('Date')
plt.show()

#cv19_countries_day.sort_values(by='DateRep',ascending = False)
```



- Calculating Death Ratio for top countries.

```
t10['DeathRatio'] = round((t10['deaths'] / t10['cases']) *100,2)
t10.sort_values(by='DeathRatio',ascending = False)
t10f = t10[['DeathRatio']].sort_values(by='DeathRatio',ascending=False).dropna()
t10f.style.apply(highlight_max, color='red', axis=None)
```

Out[17]:

	DeathRatio
Mexico	9.860000
United_Kingdom	4.170000
Spain	2.920000
Brazil	2.880000
United_States_of_America	2.400000
France	2.300000
India	1.480000

- Calculating global change rate per day.

1. Global Rate Change for cases per day.

2. Global Rate Change for deaths per day.

- Calculating Change rate for top countries only.

```
Cmd 29

#Calculate change by country for the 15 first

Impacted_countries = df[['countriesAndTerritories','cases']].sort_values(by=['dateRep','cases'],ascending=False).head(15)['countriesAndTerritories']
Impacted_countries

top_impact = pd.DataFrame()

for country in Impacted_countries:
    top_impact[country] = df[df['countriesAndTerritories']==country]['cases']


top_impact = top_impact.reset_index().sort_values(by='dateRep',ascending=True) #true


#Normalize

#top_impact_norm = top_impact/top_impact.iloc[0] * 100

Command took 0.10 seconds -- by ayush.chauhan@ifi.tech at 11/20/2020, 3:08:20 PM on democluster
```

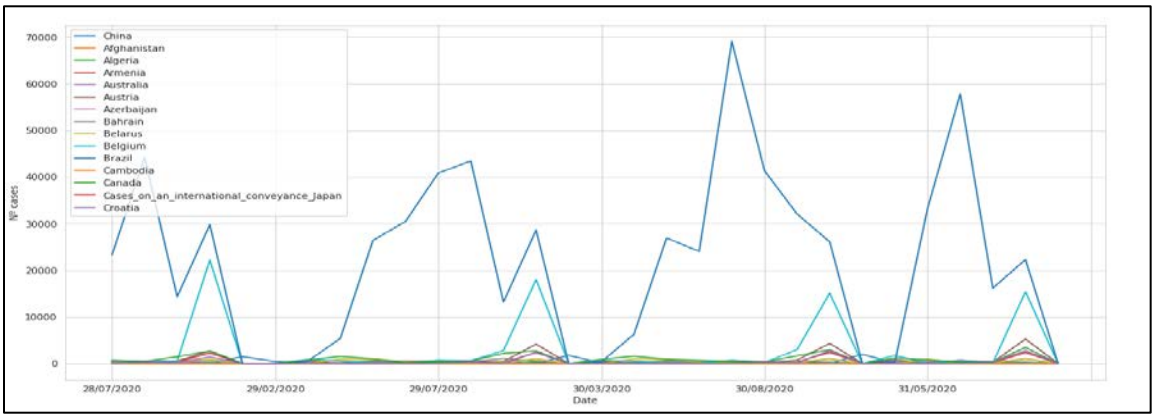
- Top countries Rate Change for cases.

CHANGE COVID RATE

Cmd 26

```
covid19_change_global = cv19_countries_day.cumsum()
covid19_change_global[['Cases Day','Deaths Day']] = cv19_countries_day[['cases','deaths']]
covid19_change_global = covid19_change_global.pct_change(1)
covid19_change_global = covid19_change_global.sort_values(by='dateRep',ascending=False)
covid19_change_global = covid19_change_global.replace([np.inf, -np.inf], np.nan)
covid19_change_global = covid19_change_global.fillna(0)
covid19_change_global = round(covid19_change_global*100,2)
covid19_change_global = covid19_change_global.reset_index()

covid19_change_global_d = cv19_countries_day.cumsum()
covid19_change_global_d[['Cases Day','Deaths Day']] = cv19_countries_day[['cases','deaths']]
covid19_change_global_d = covid19_change_global_d.pct_change(1)
covid19_change_global_d = covid19_change_global_d.sort_values(by='dateRep',ascending=False)
covid19_change_global_d = covid19_change_global_d.replace([np.inf, -np.inf], np.nan)
covid19_change_global_d = covid19_change_global_d.fillna(0)
covid19_change_global_d = round(covid19_change_global_d*100,2)
covid19_change_global_d = covid19_change_global_d.reset_index()

```

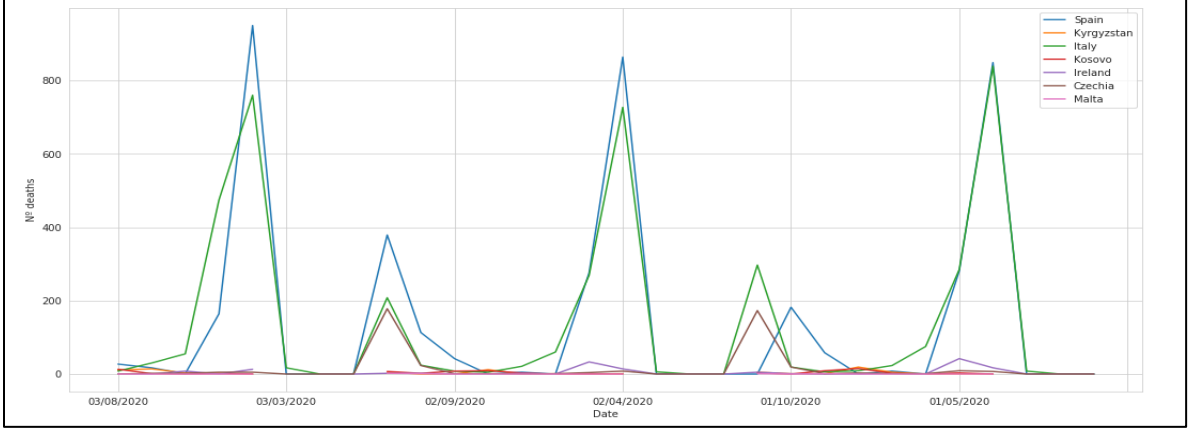
Command took 0.04 seconds -- by ayush.chauhan@ifi.tech at 11/20/2020, 3:08:20 PM on democluster



- Top countries Rate Change for deaths.

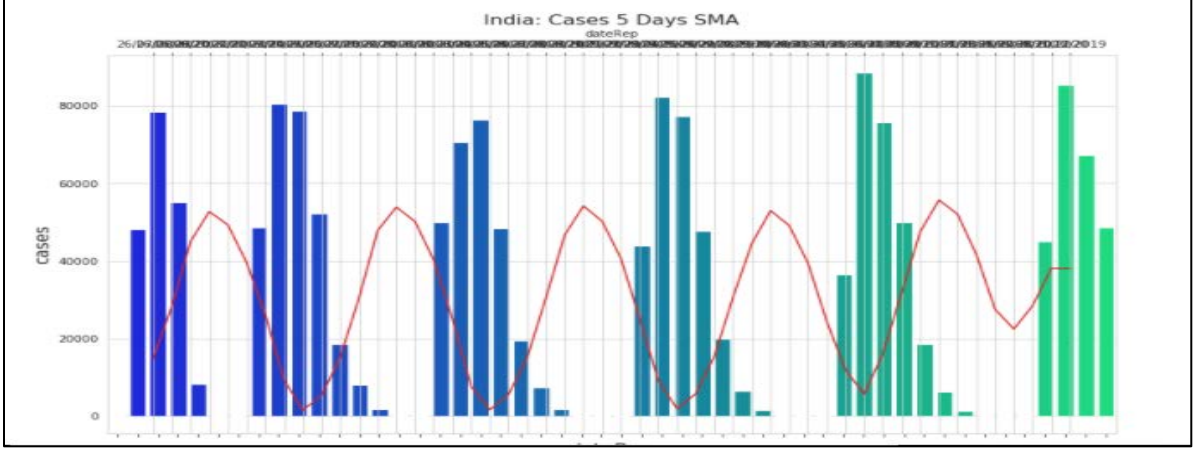
Cmd 39

```
sns.set_style('whitegrid')
top_impact_d.set_index('dateRep').tail(30).plot(figsize=(18,8))
plt.ylabel('Nº deaths')
plt.xlabel('Date')
plt.show()
```

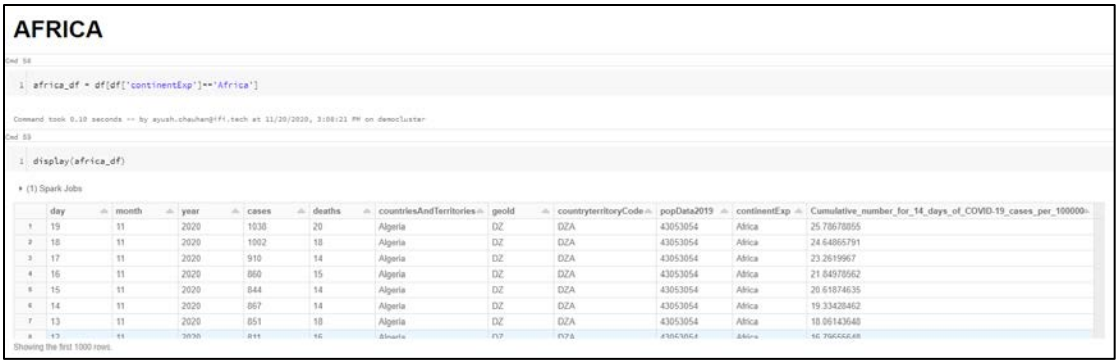


- Fetching Country particular data (e.g. INDIA).

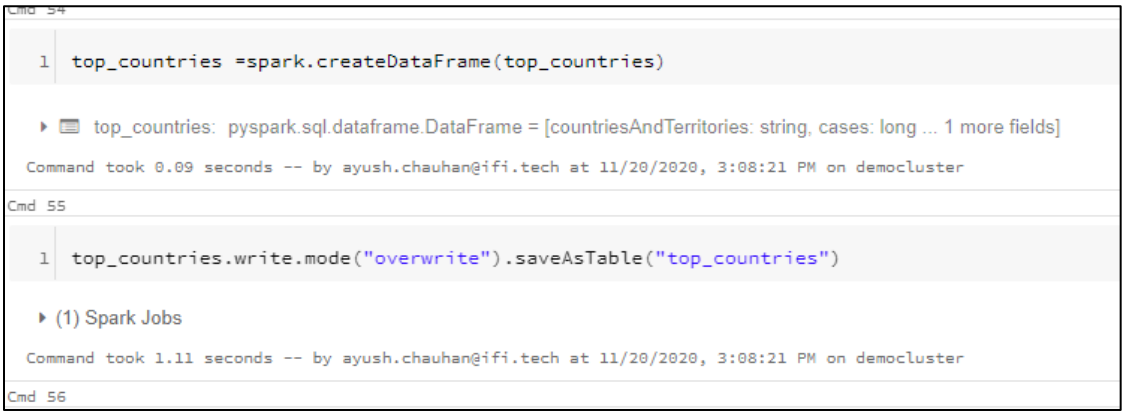
```
#Create combo chart
fig, ax1 = plt.subplots(figsize=(14,8))
color = 'tab:green'
#bar plot creation
ax1.set_title('India: Cases 5 Days SMA', fontsize=16)
ax1.set_xlabel('Date', fontsize=16)
ax1.set_ylabel('Cases', fontsize=16)
ax1 = sns.barplot(x='dateRep', y='cases', data = India_df.reset_index()[:50], palette='winter')

ax1.set_xticklabels(
    ax1.get_xticklabels(minor=True),
    rotation=45,
    horizontalalignment='right',
    fontweight='light',
    fontsize='x-large'
)
ax1.tick_params(axis='y')

#specify we want to share the same x-axis
ax2 = ax1.twiny()
color = 'tab:red'
#line plot creation
#ax2.set_ylabel('5 days SMA', fontsize=16)
ax2 = sns.lineplot(x='dateRep', y='Cases-5-days-SMA', data=India_df.reset_index()[:50], color=color)
ax2.tick_params(axis='y', color=color)
plt.show()

```



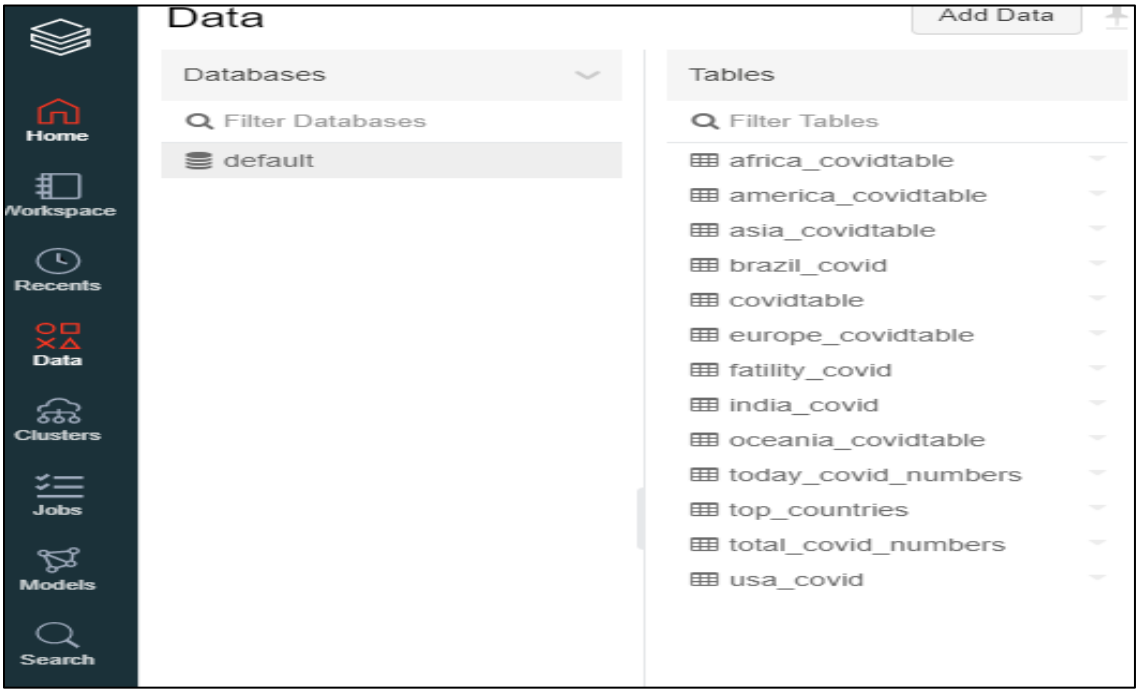
- Fetching Continent particular data.



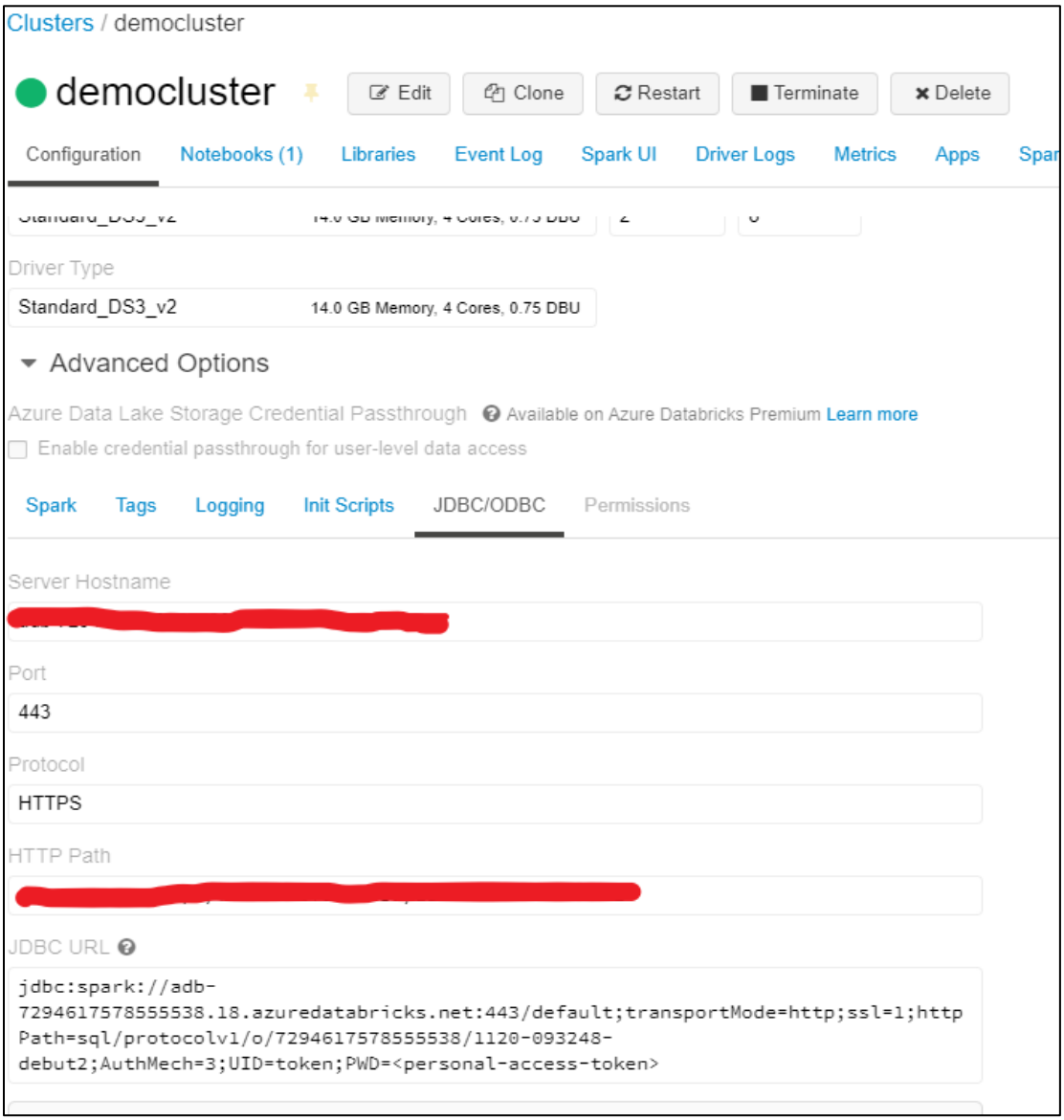
- Save the relevant data frames as tables.



- Check the newly created tables by clicking **'DATA'** in the left panel of data bricks workspace.



- Locate and store the credentials from data bricks clusters (Server Hostname & HTTP Path).

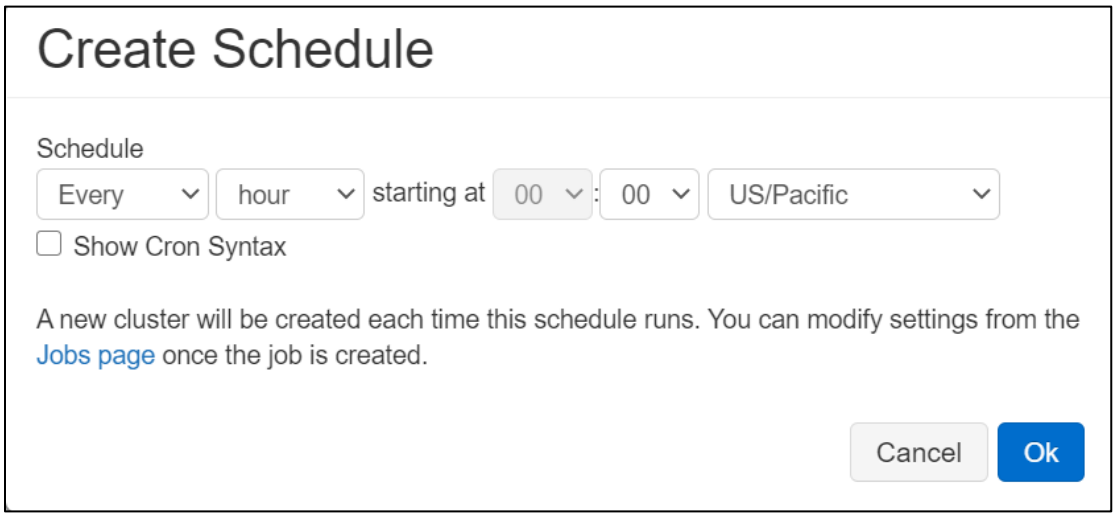


Step 3: Data brick Job Scheduling

The very import scope of this article is to capture the daily changing data, then we must run every other command for every day when a new data will be in our source to avoid this, we will automate the whole process and schedule a job to run our notebook with an interval of a day, so it will check for a new file and apply all the commands accordingly.

- Click on **(calendar like symbol)**, on the top-right corner of your notebook&click on **'+New'** button.

- Schedule a job according to your need and thenclick **'ok'.**



Step 4: Connecting Power Bi to your Databricks.

- Click on' **Get Data' a**nd search for **'Azure data bricks'** to add data bricks credentials (Server Hostname & HTTP Path)

- Click on **'DATA'** to get a view and validation for the tables in Power BI.

- Start making insightful reports by applying different filters and visuals, a sample report to get an overview.

Here you can select Individual countries or group of them to assign them with a particular date with the help of slicers to view analytics accordingly.

This is the end of this article, we successfully automate the process of Loading, Transforming and Writing the data process from a web source and then migrating the processed data to Power BI, it will also read new data and apply all the transformations on it with an interval that was defined by the developer.

Thank you and feel free to try out some different transformations in data bricks and create some awesome visuals with Power BI.

Challenges Faced:

Bringing ML to production is hard.
Most Business Intelligence (BI) workloads are restricted to a fraction of the data that an organization collects. Most of these are done against data warehouses which aggregate and transform data for specific use cases. This can be great for a lot of common reporting and known analysis, but it can also restrict an organization's ability to form new and valuable insights.

Business Benefits :

It can be used to help Medical Hospitals and Doctors to keep a track of patient history
Also with National and International Medical Authorities to analyze the region's covid statistics

Author: Ayush Chauhan, Associate Data Engineer