**Implementation Project**

**on**

**COVID-19**

**BIG-DATA ANALYSIS**

Course: CPSC 531- Advanced Database Management

Section: 3

Team Members:

**Shriya Bannikop 885196238**

**Debdyuti Das 886676550**

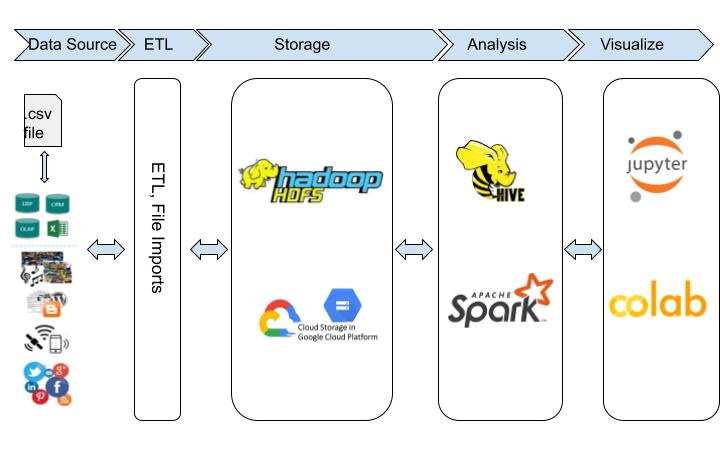
**PROBLEM STATEMENT**

The risk of coronavirus was still increasing even after the government had taken several measures worldwide to minimise the Covid-19 spread. The transmission chain's severity was deemed broken only when no new case was reported in an area. The only way to break the transmission chain is to impose a Lockdown.

The aim of the project is to address, compare and analyse the variation in the number of COVID-19 cases in countries which imposed complete lockdown with restriction rules and observe the following trend:

* To capture the trend in the data based on the increasing number of cases
* Was imposing lockdown a right decision
* Compare countries which imposed lockdown and analyse the variation in the number of covid 19 cases

**ARCHITECTURE**



* Big Data Technologies utilised in Cluster created on Google Cloud Platform i.e. Dataproc
* .csv files are given as input
* Data ingestion is done into Hadoop Distributed File System (HDFS) and stored into Google Cloud Storage Bucket
* Extract Data in Hive,Spark for analysis
* Extracted data using Hive (Hive is used as ETL to connect HDFS and spark) and used Apache Spark to perform the analysis
* The output of the analysed data is visualised using Jupyter Notebook
* The files are stored back into google storage bucket

**TOOLS AND TECHNOLOGIES**

* **Cloud Platform**: Google Cloud Cluster
* **Primary Storage System**: Hadoop Distributed File System
* **Distributed processing System**: Apache spark
* **ETL Tools**: Apache Hive
* **Visualisation**: Jupyter Notebook, Google Colab

**FUNCTIONALITIES**

* To capture the trend in the data collected from multiple datasets based on the increasing number of cases
* To determine if imposing lockdown was a right decision
* To compare countries which imposed lockdown and analyse the variation in the number of Covid-19 cases
* Migration analysis to know the population and cases before and after lockdown

**APPROACH**

**Before implementing with Cluster:**

1. Download files and store in local directory
2. Start all daemons in HDFS using

* **hdfs namenode -format**
* **start-dfs.sh**

1. Verify if all components are running

* **jps**

1. Move .csv files to HDFS

* Make a directory in HDFS:

**hadoop fs -mkdir -p /home/hadoop/directory\_name**

* Copy the .csv file from Local to HDFS:

**hadoop fs -put /home/debdyuti/bigdata/covid\_19\_data.csv /home/hadoop/directory\_name**

* Check if its copied:

**hadoop fs -ls /home/hadoop/directory\_name**

1. Create tables in hive and use MapReduce

* **Cd $HIVE\_HOME/bin**
* Open hive-CLI: **hive**
* Create database:

**CREATE SCHEMA IF NOT EXISTS database\_name;**

**USE database\_name;**

* Create table:

**CREATE TABLE IF NOT EXISTS database\_name.covid\_details(SNo INTEGER, ObservationDate STRING, State STRING, Country STRING, LastUpdate STRING, Confirmed DOUBLE, Deaths DOUBLE, Recovered DOUBLE)**

**ROW FORMAT DELIMITED**

**FIELDS TERMINATED BY ',';**

* Load Dataset from HDFS to HIVE Table:

**LOAD DATA INPATH '/home/hadoop/directory\_name/covid\_19\_data.csv' INTO TABLE database\_name.covid\_details ;**

* To see records in the HIVE table:

**SELECT \* FROM database\_name.covid\_details ;**

1. Extract data into Spark for analysis
2. Read files from Spark and visualise using Google Colab

**After implementing with Cluster:**

1. Create cluster in Google Cloud Platform
2. Open console (SSH) on master node
3. Download data from internet into the Hadoop cluster (HDFS location) using **wget** command
4. Copy files from hadoop (HDFS location) into Google storage bucket using “**gsutil cp migration\_population.csv us-central1 gs://dyutishriya-bucketdbms/Data1;**”
5. Using web interface analyse and data in Jupyter Notebook
6. Read from Google Storage Bucket. After analysis and visualisation store it back into the bucket.

**STEPS TO RUN THE PROJECT**

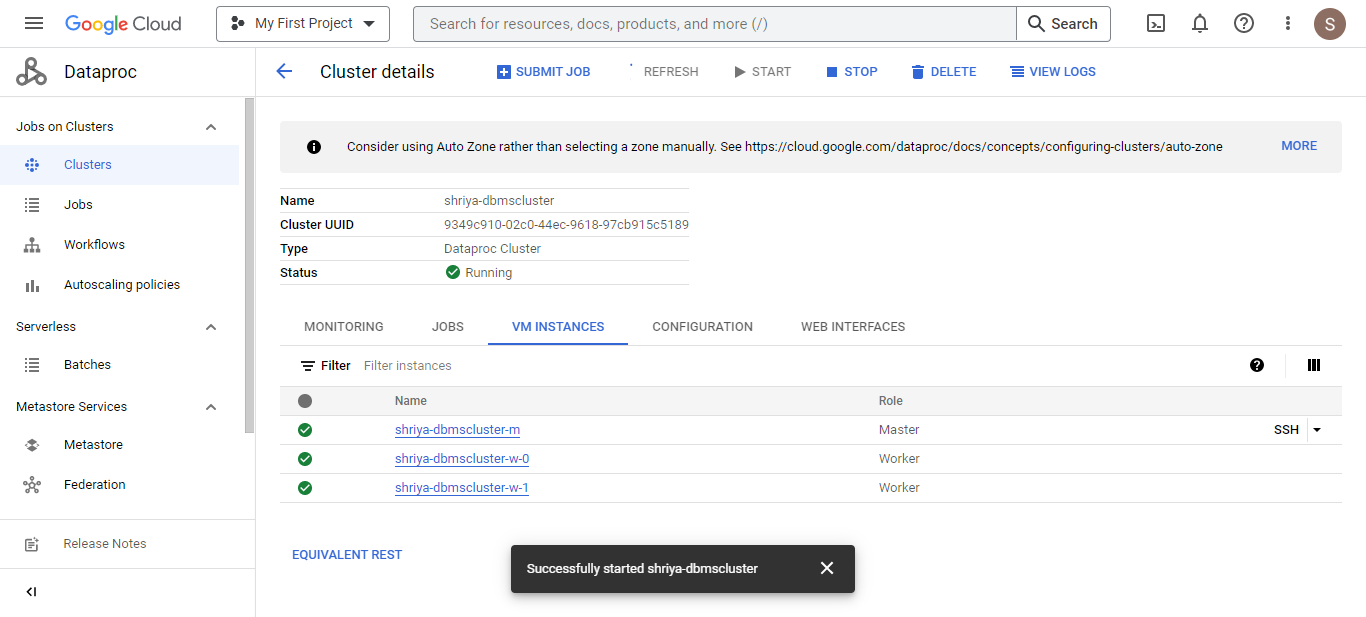
**Github Location of Code :**

https://github.com/Debdyuti-01/Covid-19-Big-Data-Analysis

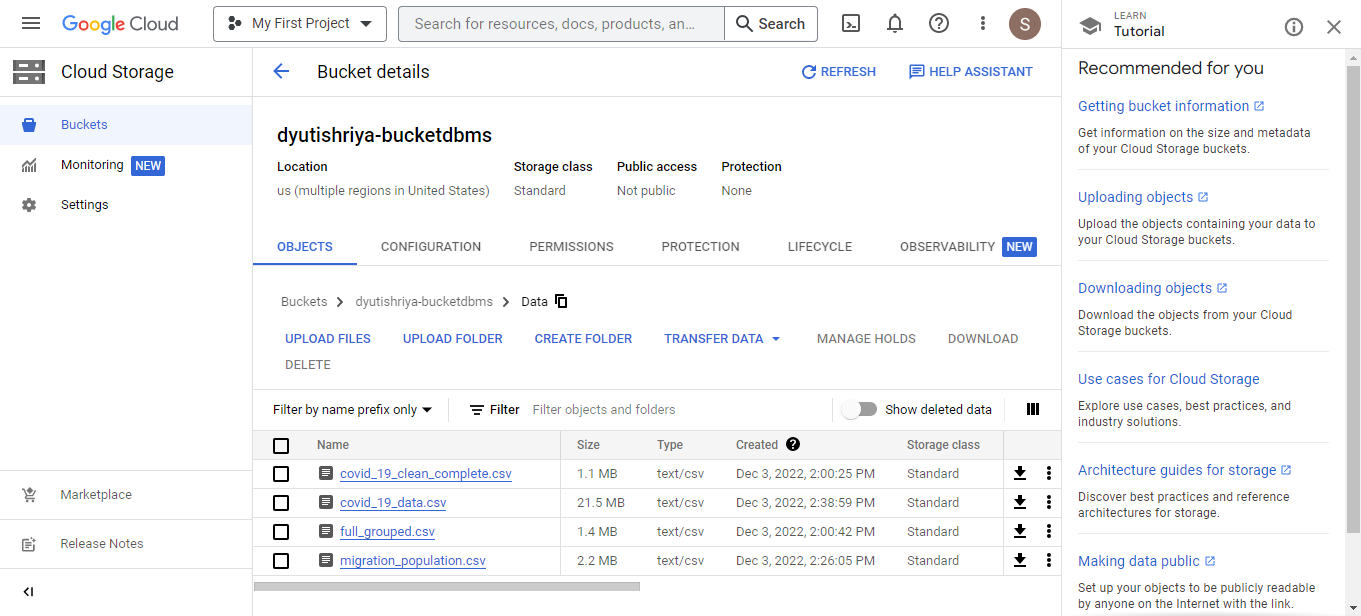
1. Start the Cluster
2. Start demons (start-dfs command) by opening SSH shell in master node
3. Run the .ipynb file on jupyter notebook to check visualisations

**TEST RESULTS OF SPARK ANALYSIS**

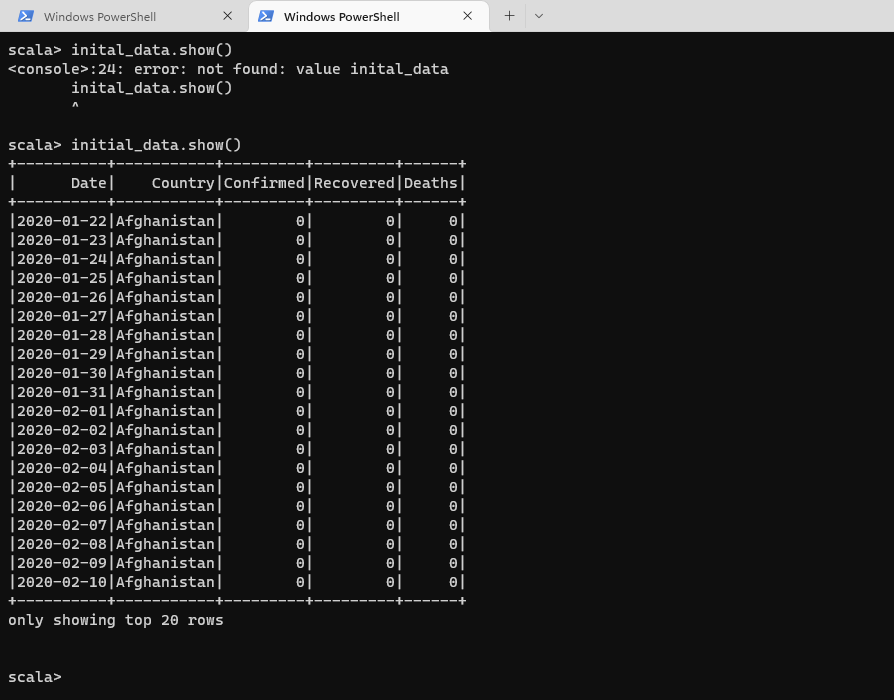
* **Cluster Creation:**

****

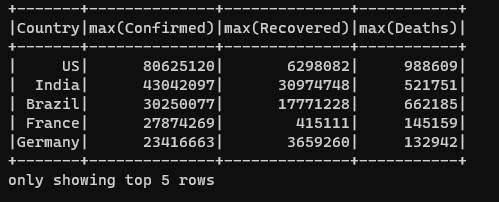
* **Bucket Creation and Loading data:**

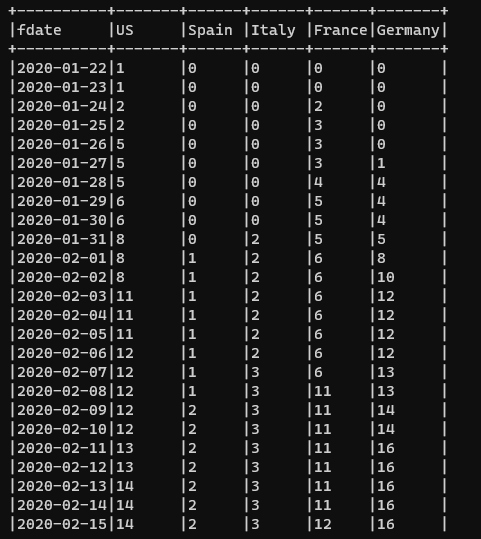
****

* **Data Cleaning:**
* Filled blank fields with ‘unknown’
* Filtered data
* Converted String datatype to Date datatype of Date attribute

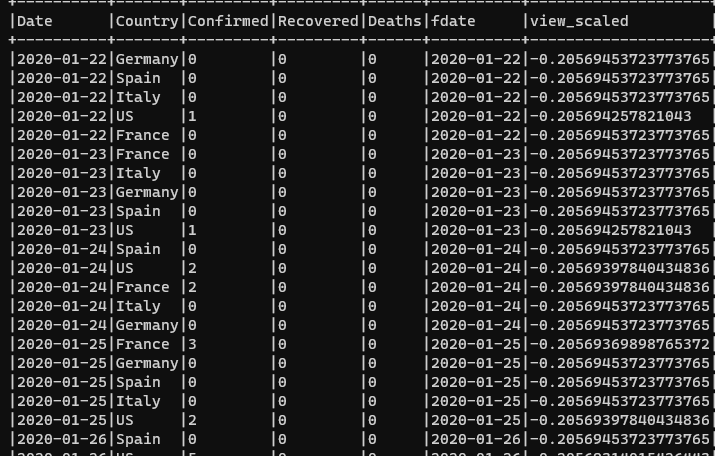
****

* **Data Exploration:**
* To find top 5 countries which were leading with Covid-19 cases
* Pivoted the table by Country attribute
* Total number of recovered, confirmed and death cases of Covid of top 5 leading countries

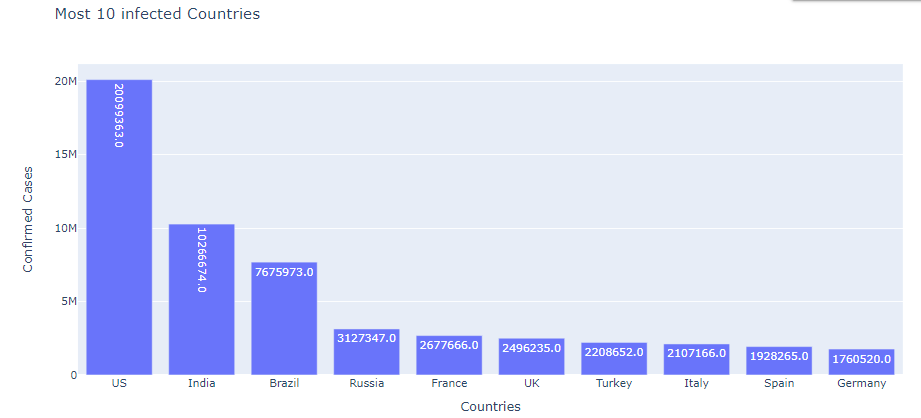




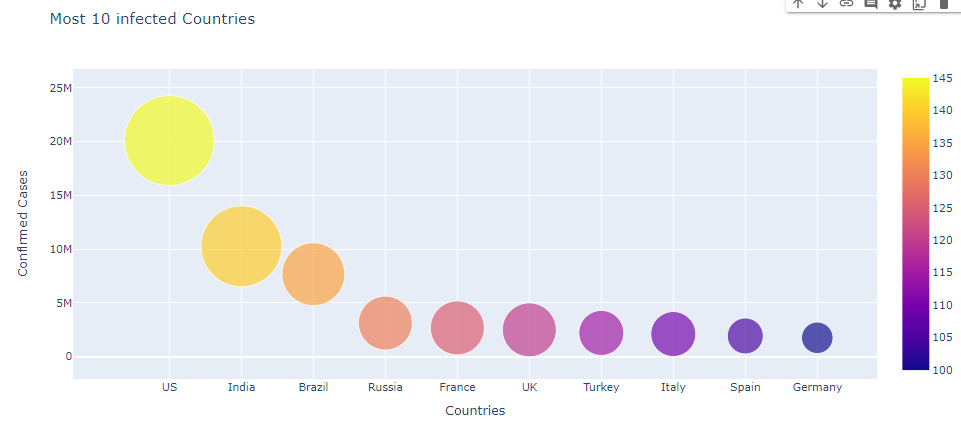
* **Data Preparation**
* Scaled values to remove outliers



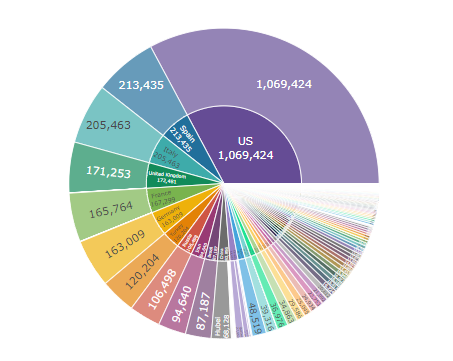
* **Analysis**



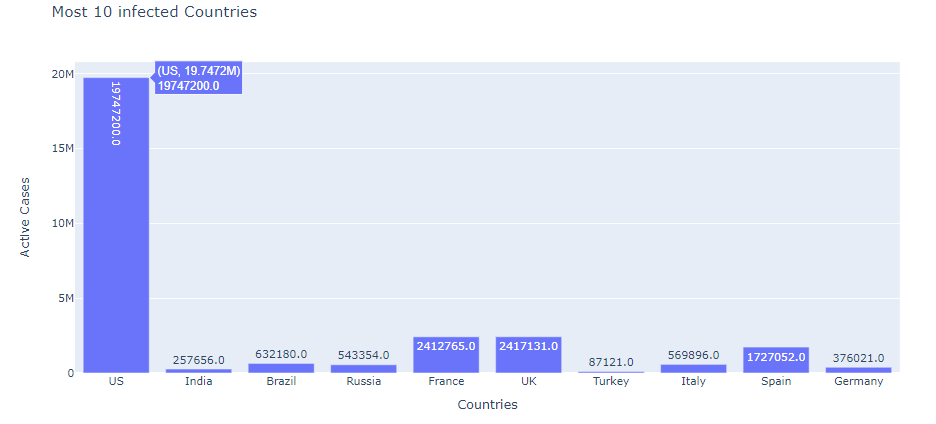
Confirmed covid cases in year 2020 for each country

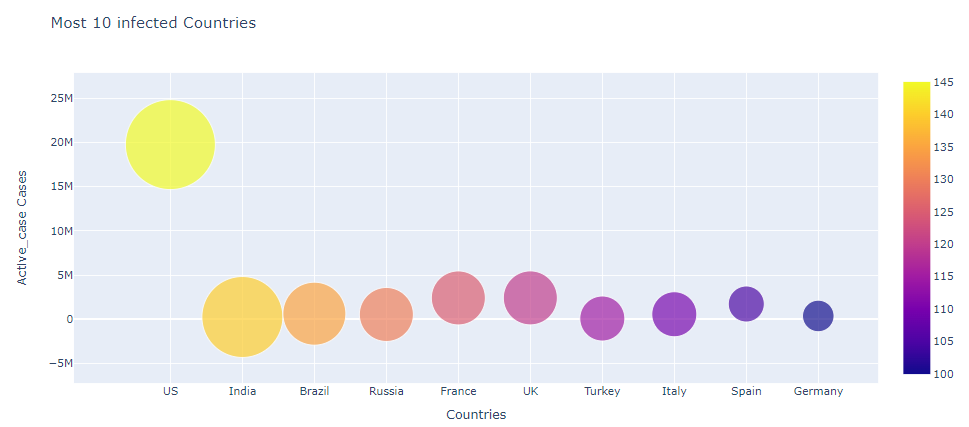


Top 10 countries sorted by maximum number of confirmed covid cases

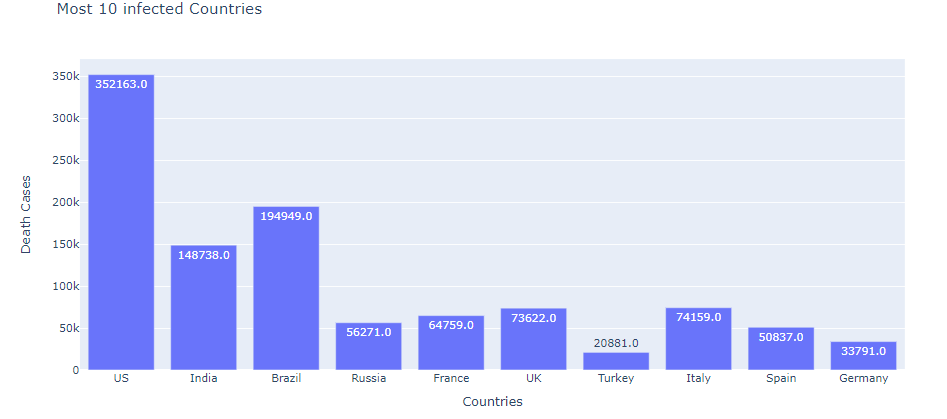


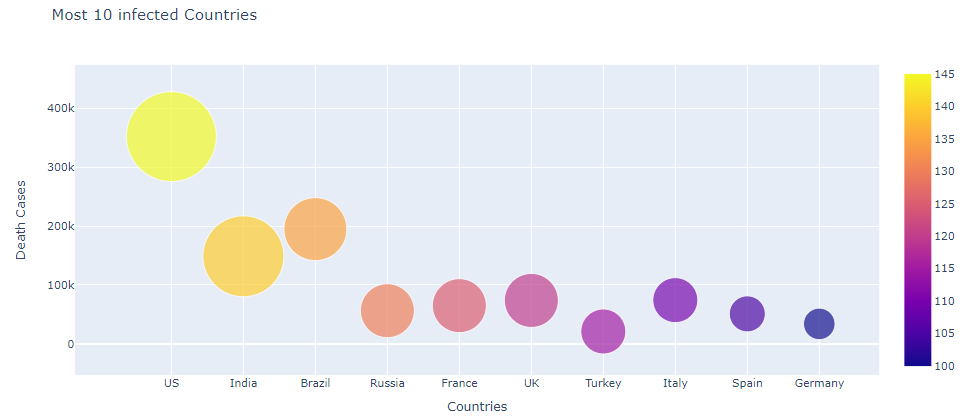
* Total number of active cases in each country
* Top 10 countries sorted by number of active covid cases



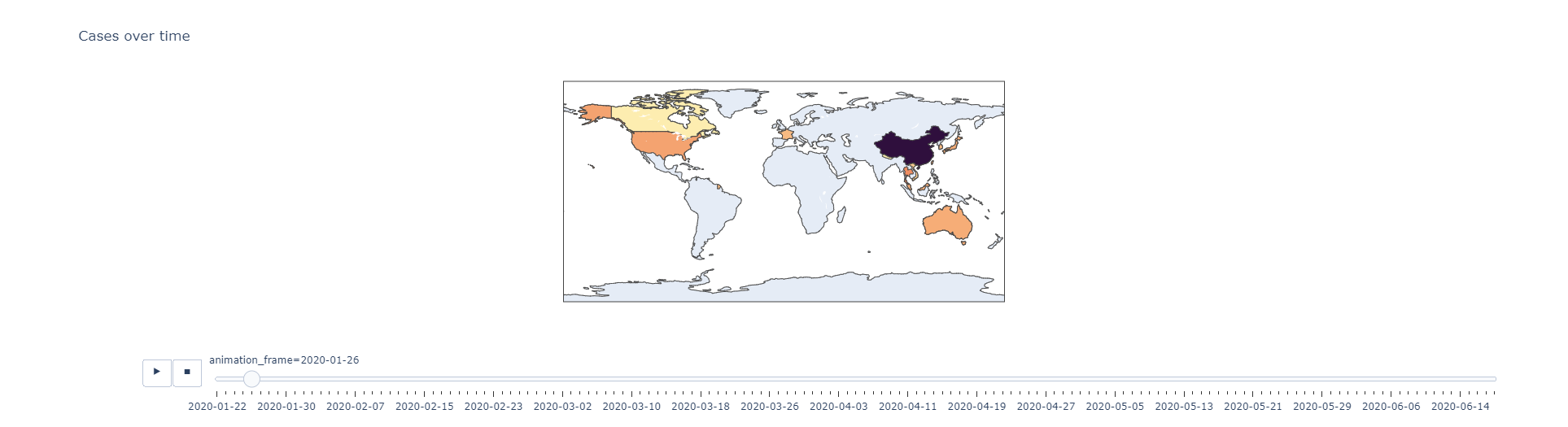


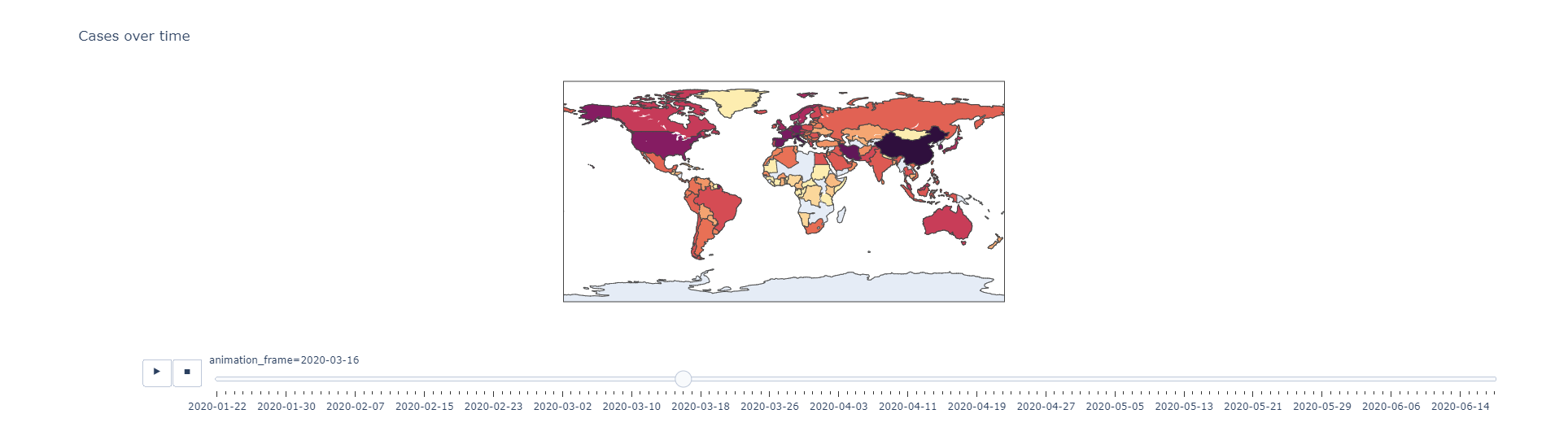
* Total number of death cases in each country
* Top 10 country sorted by covid death rate

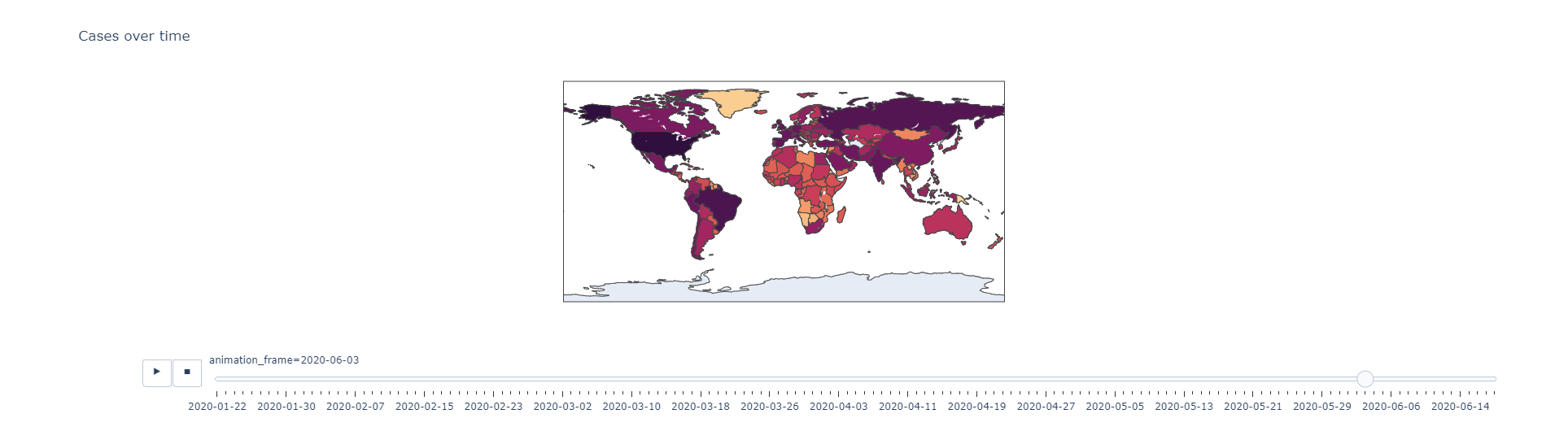




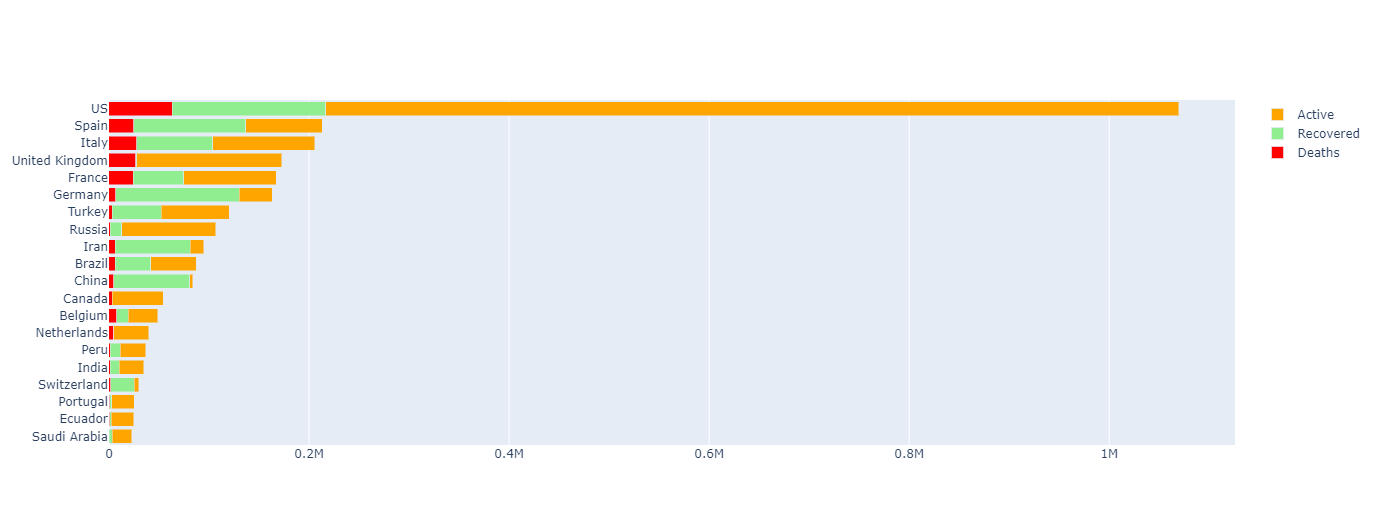
* Plot showing the increases in cases by date(animation plot)



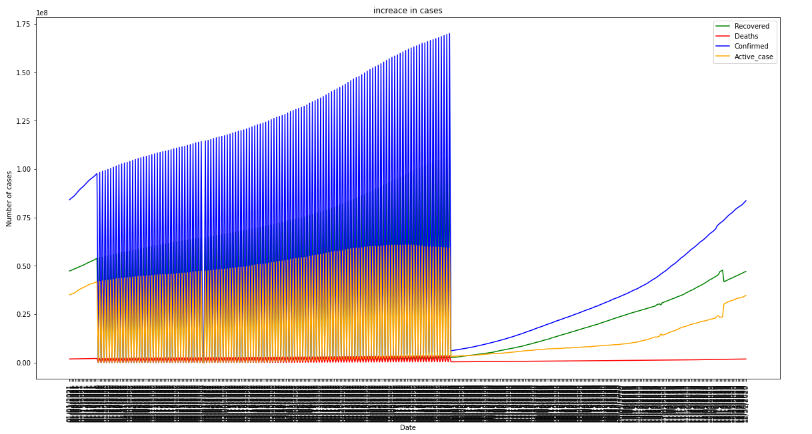




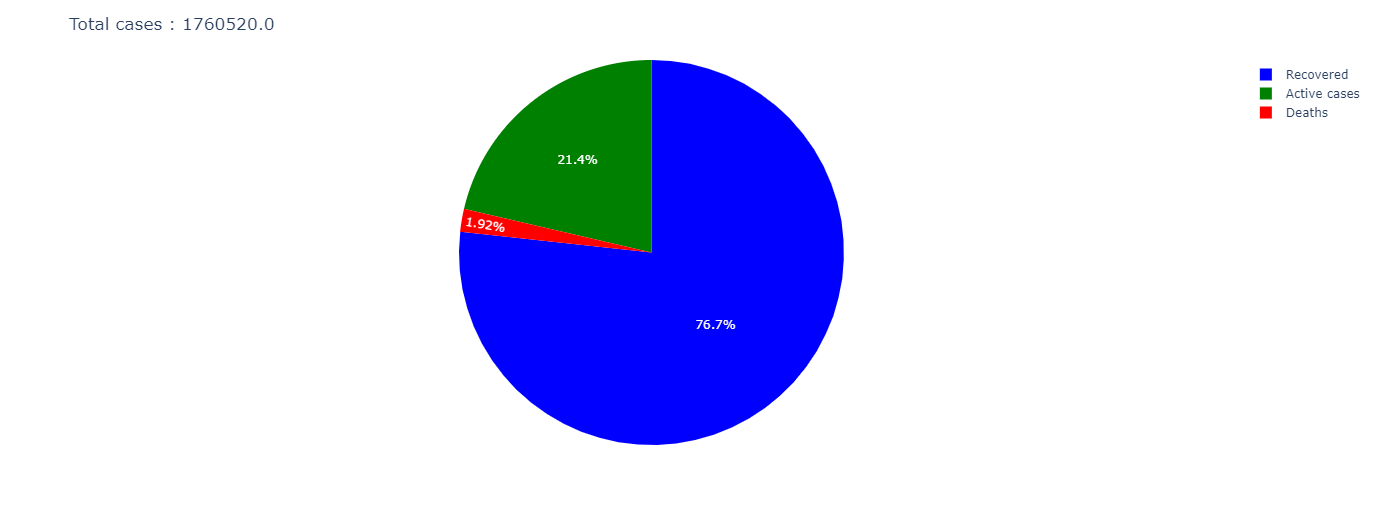
* Top 20 countries sorted based on the total number of cases

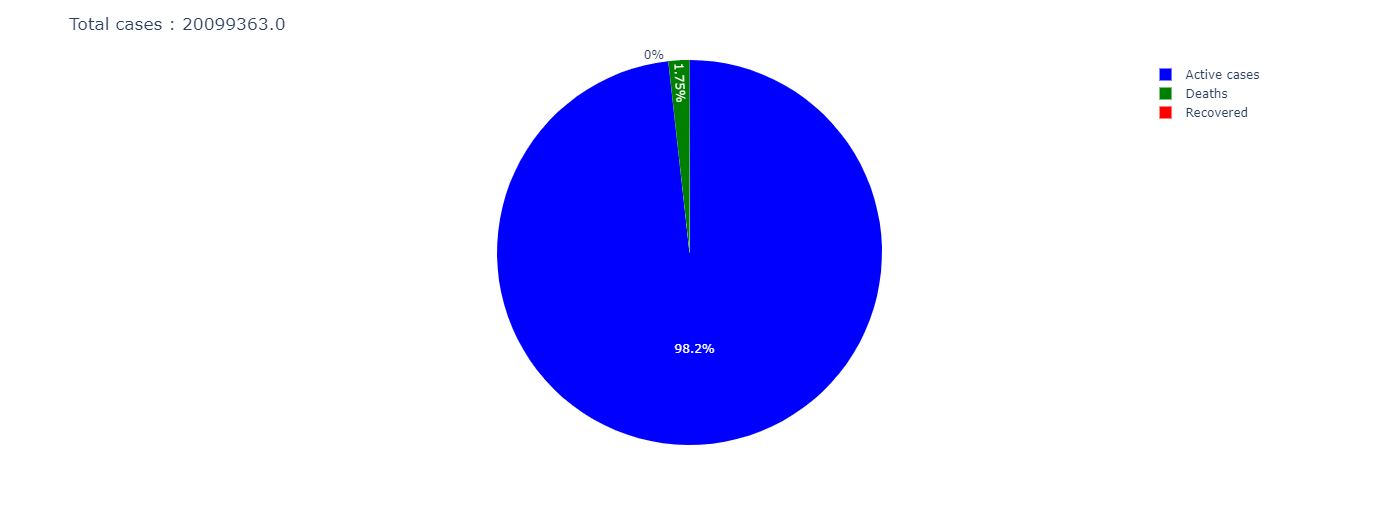


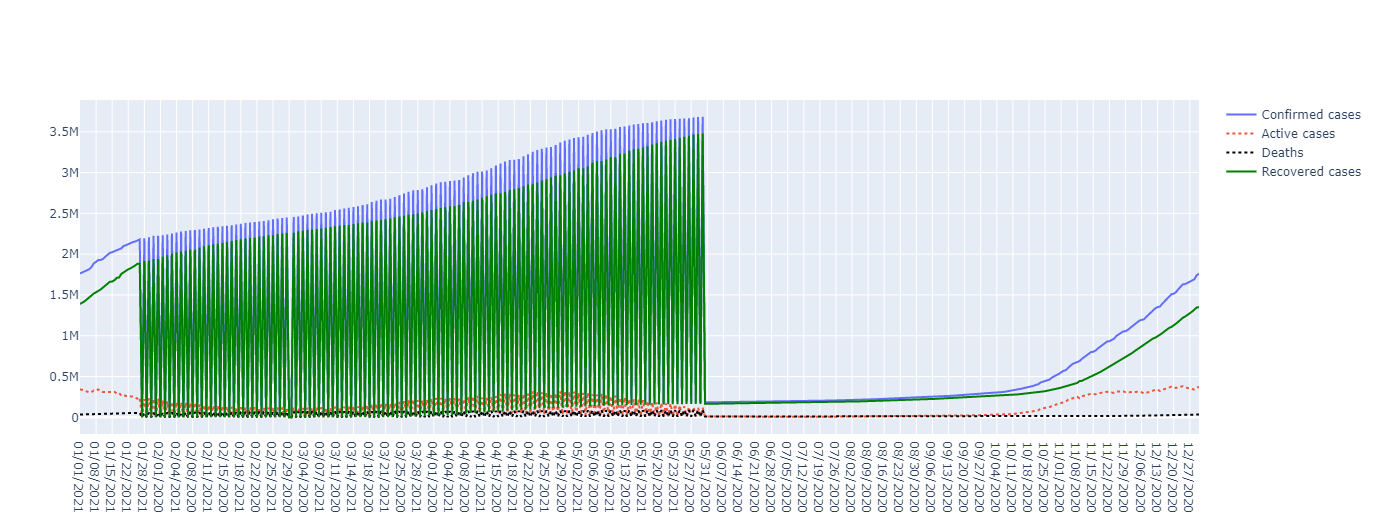
* Line Chart of increase in 'Recovered', 'Deaths', 'Confirmed', 'Active\_case'

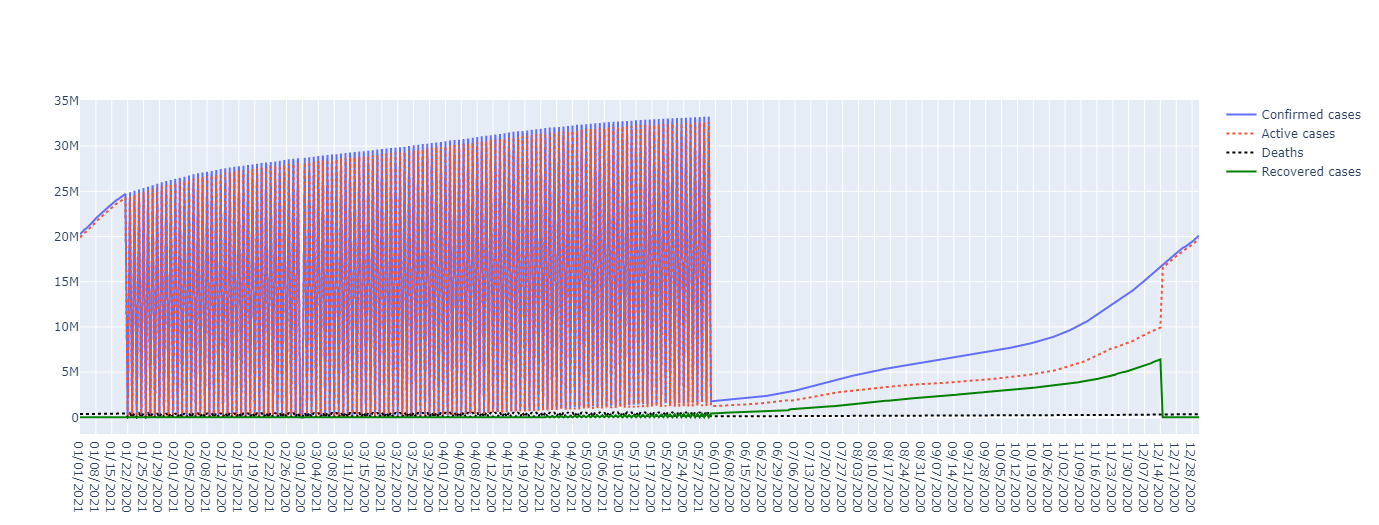
****

* **Comparisions**
* **Germany VS USA**

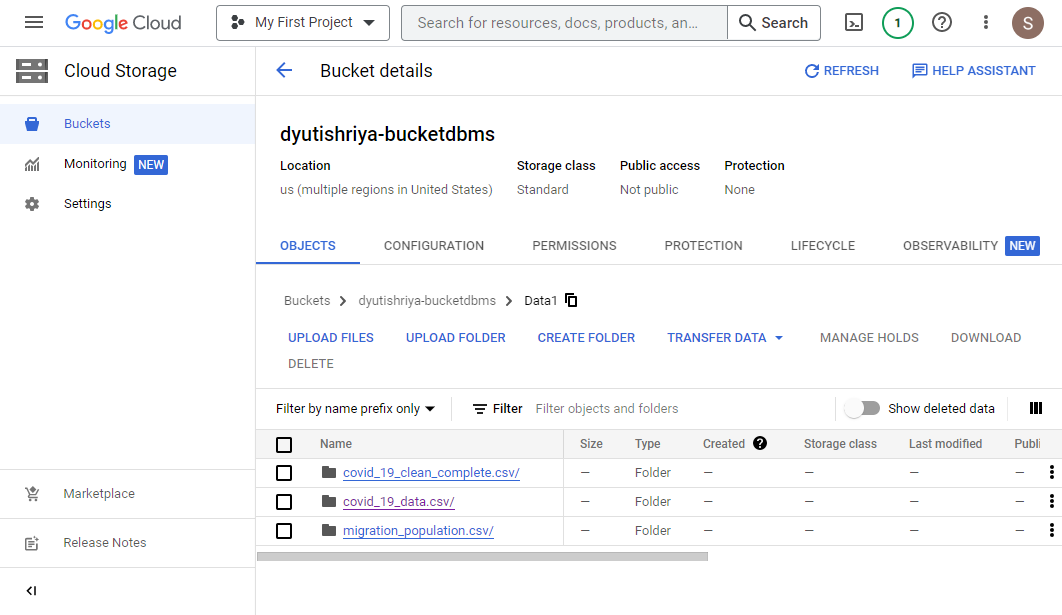
****

****

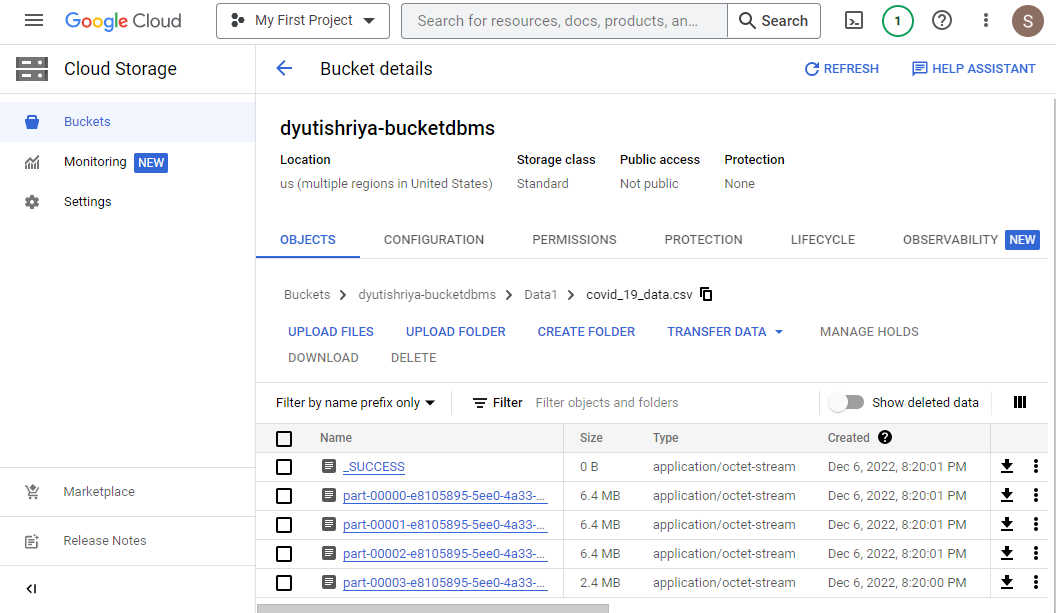
****

****

* Comparing the plots for USA and Germany its is noticeable that USA(figure down) had more active cases(blue) than Germany(figure on top)
* **Storing data back into bucket**
* Data(csv files) stored in google storage bucket where subfolder is created by hadoop to save partitioned data.

****

* **Data is partitioned by hadoop into smaller chunks and saved**

****