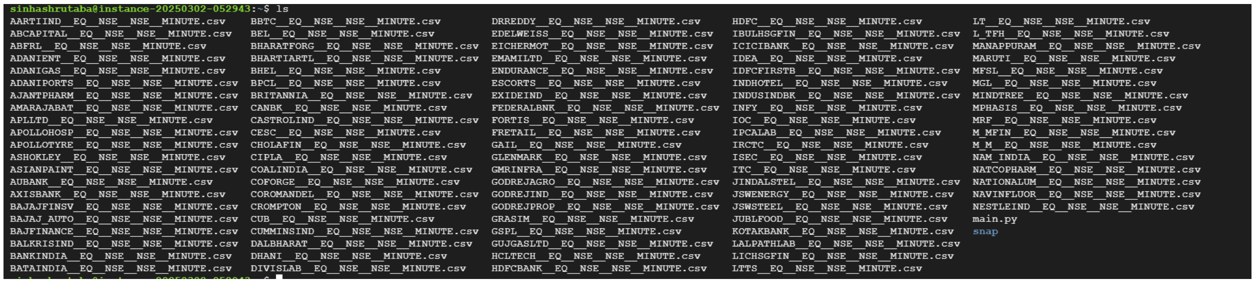
Submitted by Shramana Sinha, 23f1002703

Preparing the input files

Downloading Data from GitHub

This step fetches the stock price data files from the specified GitHub repository into the vm machine.

```Python
repo_url =
"https://github.com/ShabbirHasan1/NSE-Data/tree/main/NSE%20Minute%20Data/NSE_St
ocks_Data"
response = requests.get(repo_url)
soup = BeautifulSoup(response.content, "html.parser")
file_links = []
for link in soup.find_all("a"):
    href = link.get("href")
    if href and href.endswith(".csv"):
        file_links.append(
            f"https://raw.githubusercontent.com{href.replace('/blob', '')}"
        )
for file_link in file_links:
    try:
        filename = os.path.basename(file_link)
        response = requests.get(file_link)
        with open(filename, "wb") as f:
            f.write(response.content)
    except Exception as e:
        print(f"Error processing file {filename}: {e}")
```



```
sinhashrutaba@instance-20250302-052943:~$ ls
AARTIIND__EQ__NSE__NSE__MINUTE.csv    BBTC__EQ__NSE__NSE__MINUTE.csv        DRREDDY__EQ__NSE__NSE__MINUTE.csv     HDFC__EQ__NSE__NSE__MINUTE.csv        LT__EQ__NSE__NSE__MINUTE.csv
ABCAPITAL__EQ__NSE__NSE__MINUTE.csv   BEL__EQ__NSE__NSE__MINUTE.csv         EDELWEISS__EQ__NSE__NSE__MINUTE.csv   IBULHSGFIN__EQ__NSE__NSE__MINUTE.csv  L_TFH__EQ__NSE__NSE__MINUTE.csv
ABFRL__EQ__NSE__NSE__MINUTE.csv       BHARATFORG__EQ__NSE__NSE__MINUTE.csv  EICHERMOT__EQ__NSE__NSE__MINUTE.csv   ICICIBANK__EQ__NSE__NSE__MINUTE.csv   MANAPPURAM__EQ__NSE__NSE__MINUTE.csv
ADANIENT__EQ__NSE__NSE__MINUTE.csv    BHARTIARTL__EQ__NSE__NSE__MINUTE.csv  EMAMILTD__EQ__NSE__NSE__MINUTE.csv    IDEA__EQ__NSE__NSE__MINUTE.csv        MARUTI__EQ__NSE__NSE__MINUTE.csv
ADANIGAS__EQ__NSE__NSE__MINUTE.csv    BHEL__EQ__NSE__NSE__MINUTE.csv        ENDURANCE__EQ__NSE__NSE__MINUTE.csv   IDFCFIRSTB__EQ__NSE__NSE__MINUTE.csv  MFSL__EQ__NSE__NSE__MINUTE.csv
ADANIPORTS__EQ__NSE__NSE__MINUTE.csv  BPCL__EQ__NSE__NSE__MINUTE.csv        ESCORTS__EQ__NSE__NSE__MINUTE.csv     INDHOTEL__EQ__NSE__NSE__MINUTE.csv    MGL__EQ__NSE__NSE__MINUTE.csv
AJANTPHARM__EQ__NSE__NSE__MINUTE.csv  BRITANNIA__EQ__NSE__NSE__MINUTE.csv   EXIDEIND__EQ__NSE__NSE__MINUTE.csv    INDUSINDBK__EQ__NSE__NSE__MINUTE.csv  MINDTREE__EQ__NSE__NSE__MINUTE.csv
AMARAJABAT__EQ__NSE__NSE__MINUTE.csv  CANBK__EQ__NSE__NSE__MINUTE.csv       FEDERALBNK__EQ__NSE__NSE__MINUTE.csv  INFY__EQ__NSE__NSE__MINUTE.csv        MPHASIS__EQ__NSE__NSE__MINUTE.csv
APLLTD__EQ__NSE__NSE__MINUTE.csv      CASTROLIND__EQ__NSE__NSE__MINUTE.csv  FORTIS__EQ__NSE__NSE__MINUTE.csv      IOC__EQ__NSE__NSE__MINUTE.csv         MRF__EQ__NSE__NSE__MINUTE.csv
APOLLOHOSP__EQ__NSE__NSE__MINUTE.csv  CESC__EQ__NSE__NSE__MINUTE.csv        FRETAIL__EQ__NSE__NSE__MINUTE.csv     IPCALAB__EQ__NSE__NSE__MINUTE.csv     M_MFIN__EQ__NSE__NSE__MINUTE.csv
APOLLOTYRE__EQ__NSE__NSE__MINUTE.csv  CHOLAFIN__EQ__NSE__NSE__MINUTE.csv    GAIL__EQ__NSE__NSE__MINUTE.csv        IRCTC__EQ__NSE__NSE__MINUTE.csv       M_M__EQ__NSE__NSE__MINUTE.csv
ASHOKLEY__EQ__NSE__NSE__MINUTE.csv    CIPLA__EQ__NSE__NSE__MINUTE.csv       GLENMARK__EQ__NSE__NSE__MINUTE.csv    ISEC__EQ__NSE__NSE__MINUTE.csv        NAM_INDIA__EQ__NSE__NSE__MINUTE.csv
ASIANPAINT__EQ__NSE__NSE__MINUTE.csv  COALINDIA__EQ__NSE__NSE__MINUTE.csv   GMRINFRA__EQ__NSE__NSE__MINUTE.csv    ITC__EQ__NSE__NSE__MINUTE.csv         NATCOPHARM__EQ__NSE__NSE__MINUTE.csv
AUBANK__EQ__NSE__NSE__MINUTE.csv      COFORGE__EQ__NSE__NSE__MINUTE.csv     GODREJAGRO__EQ__NSE__NSE__MINUTE.csv  JINDALSTEL__EQ__NSE__NSE__MINUTE.csv  NATIONALUM__EQ__NSE__NSE__MINUTE.csv
AXISBANK__EQ__NSE__NSE__MINUTE.csv    COROMANDEL__EQ__NSE__NSE__MINUTE.csv  GODREJIND__EQ__NSE__NSE__MINUTE.csv   JSWENERGY__EQ__NSE__NSE__MINUTE.csv   NAVINFLUOR__EQ__NSE__NSE__MINUTE.csv
BAJAJFINSV__EQ__NSE__NSE__MINUTE.csv  CROMPTON__EQ__NSE__NSE__MINUTE.csv    GODREJPROP__EQ__NSE__NSE__MINUTE.csv  JSWSTEEL__EQ__NSE__NSE__MINUTE.csv    NESTLEIND__EQ__NSE__NSE__MINUTE.csv
BAJAJ_AUTO__EQ__NSE__NSE__MINUTE.csv  CUB__EQ__NSE__NSE__MINUTE.csv         GRASIM__EQ__NSE__NSE__MINUTE.csv      JUBLFOOD__EQ__NSE__NSE__MINUTE.csv    main.py
BAJFINANCE__EQ__NSE__NSE__MINUTE.csv  CUMMINSIND__EQ__NSE__NSE__MINUTE.csv  GSPL__EQ__NSE__NSE__MINUTE.csv        KOTAKBANK__EQ__NSE__NSE__MINUTE.csv   snap
BALKRISIND__EQ__NSE__NSE__MINUTE.csv  DALBHARAT__EQ__NSE__NSE__MINUTE.csv   GUJGASLTD__EQ__NSE__NSE__MINUTE.csv   LALPATHLAB__EQ__NSE__NSE__MINUTE.csv
BANKINDIA__EQ__NSE__NSE__MINUTE.csv   DHANI__EQ__NSE__NSE__MINUTE.csv       HCLTECH__EQ__NSE__NSE__MINUTE.csv     LICHSGFIN__EQ__NSE__NSE__MINUTE.csv
BATAINDIA__EQ__NSE__NSE__MINUTE.csv   DIVISLAB__EQ__NSE__NSE__MINUTE.csv    HDFCBANK__EQ__NSE__NSE__MINUTE.csv    LTTS__EQ__NSE__NSE__MINUTE.csv
```

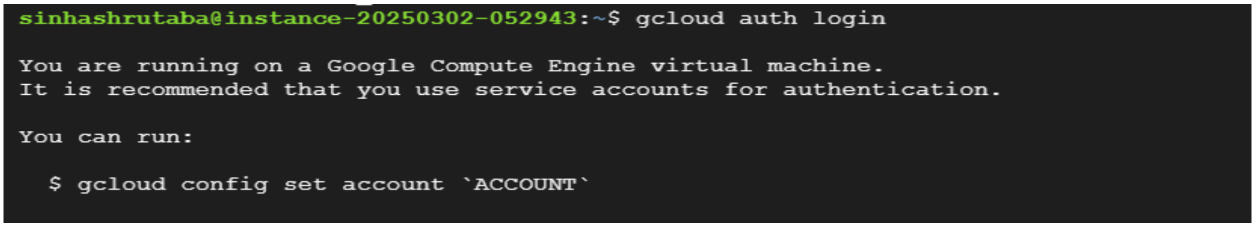
Screenshot: The files that were downloaded from github to the vm’s local machine

Authenticating with Google Cloud

This step authenticates the vm machine to upload files in the gcs buckets.

Unset

```
gcloud auth login
```



```
sinhashrutaba@instance-20250302-052943:~$ gcloud auth login

You are running on a Google Compute Engine virtual machine.
It is recommended that you use service accounts for authentication.

You can run:

  $ gcloud config set account `ACCOUNT`
```

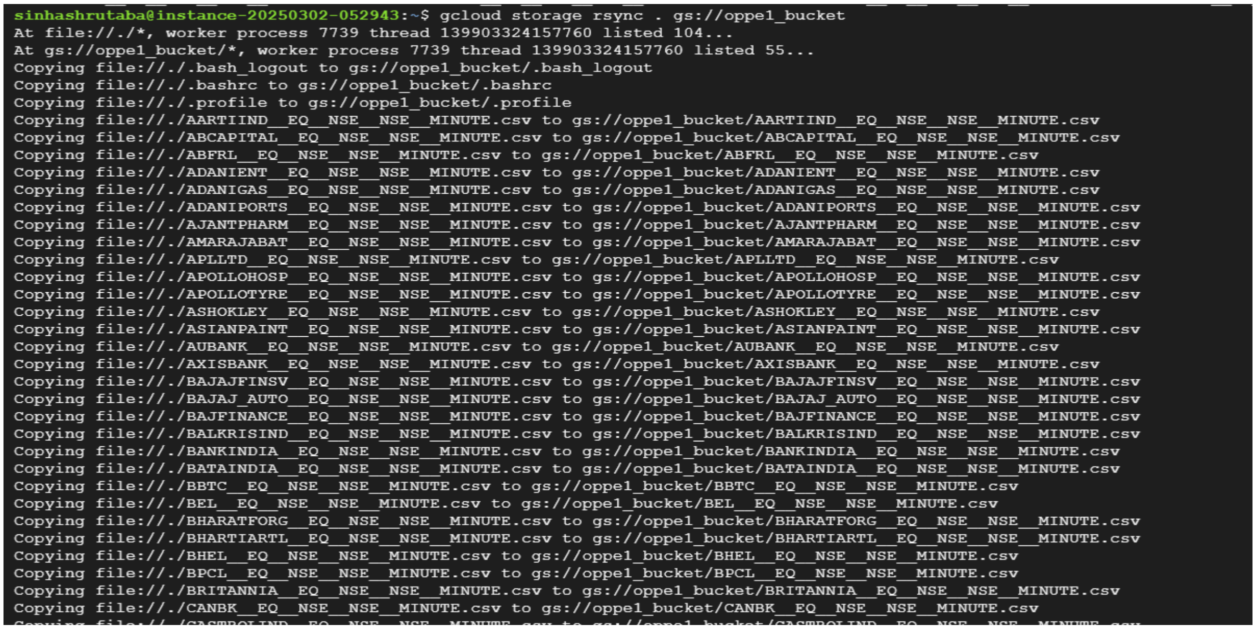
Screenshot: The google cloud storage authentication

Transferring Files to GCS Bucket

In this step, the files were transferred from the vm machine's local storage to the gcs bucket.

Unset

```
gcloud storage rsync . gs://oppe1_bucket
```



```
sinhashrutaba@instance-20250302-052943:~$ gcloud storage rsync . gs://oppe1_bucket
At file://./*, worker process 7739 thread 139903324157760 listed 104...
At gs://oppe1_bucket/*, worker process 7739 thread 139903324157760 listed 55...
Copying file://./.bash_logout to gs://oppe1_bucket/.bash_logout
Copying file://./.bashrc to gs://oppe1_bucket/.bashrc
Copying file://./.profile to gs://oppe1_bucket/.profile
Copying file://./AARTIIND__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/AARTIIND__EQ__NSE__NSE__MINUTE.csv
Copying file://./ABCAPITAL__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ABCAPITAL__EQ__NSE__NSE__MINUTE.csv
Copying file://./ABFRL__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ABFRL__EQ__NSE__NSE__MINUTE.csv
Copying file://./ADANIENT__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ADANIENT__EQ__NSE__NSE__MINUTE.csv
Copying file://./ADANIGAS__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ADANIGAS__EQ__NSE__NSE__MINUTE.csv
Copying file://./ADANIPORTS__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ADANIPORTS__EQ__NSE__NSE__MINUTE.csv
Copying file://./AJANTPHARM__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/AJANTPHARM__EQ__NSE__NSE__MINUTE.csv
Copying file://./AMARAJABAT__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/AMARAJABAT__EQ__NSE__NSE__MINUTE.csv
Copying file://./APLLTD__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/APLLTD__EQ__NSE__NSE__MINUTE.csv
Copying file://./APOLLOHOSP__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/APOLLOHOSP__EQ__NSE__NSE__MINUTE.csv
Copying file://./APOLLOTYRE__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/APOLLOTYRE__EQ__NSE__NSE__MINUTE.csv
Copying file://./ASHOKLEY__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ASHOKLEY__EQ__NSE__NSE__MINUTE.csv
Copying file://./ASIANPAINT__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/ASIANPAINT__EQ__NSE__NSE__MINUTE.csv
Copying file://./AUBANK__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/AUBANK__EQ__NSE__NSE__MINUTE.csv
Copying file://./AXISBANK__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/AXISBANK__EQ__NSE__NSE__MINUTE.csv
Copying file://./BAJAJFINSV__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BAJAJFINSV__EQ__NSE__NSE__MINUTE.csv
Copying file://./BAJAJ_AUTO__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BAJAJ_AUTO__EQ__NSE__NSE__MINUTE.csv
Copying file://./BAJFINANCE__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BAJFINANCE__EQ__NSE__NSE__MINUTE.csv
Copying file://./BALKRISIND__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BALKRISIND__EQ__NSE__NSE__MINUTE.csv
Copying file://./BANKINDIA__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BANKINDIA__EQ__NSE__NSE__MINUTE.csv
Copying file://./BATAINDIA__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BATAINDIA__EQ__NSE__NSE__MINUTE.csv
Copying file://./BBTC__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BBTC__EQ__NSE__NSE__MINUTE.csv
Copying file://./BEL__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BEL__EQ__NSE__NSE__MINUTE.csv
Copying file://./BHARATFORG__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BHARATFORG__EQ__NSE__NSE__MINUTE.csv
Copying file://./BHARTIARTL__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BHARTIARTL__EQ__NSE__NSE__MINUTE.csv
Copying file://./BHEL__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BHEL__EQ__NSE__NSE__MINUTE.csv
Copying file://./BPCL__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BPCL__EQ__NSE__NSE__MINUTE.csv
Copying file://./BRITANNIA__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/BRITANNIA__EQ__NSE__NSE__MINUTE.csv
Copying file://./CANBK__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/CANBK__EQ__NSE__NSE__MINUTE.csv
Copying file://./CASTROLIND__EQ__NSE__NSE__MINUTE.csv to gs://oppe1_bucket/CASTROLIND__EQ__NSE__NSE__MINUTE.csv
```

Screenshot: The files are uploaded from vm to gcs bucket

Dataproc Cluster Setup

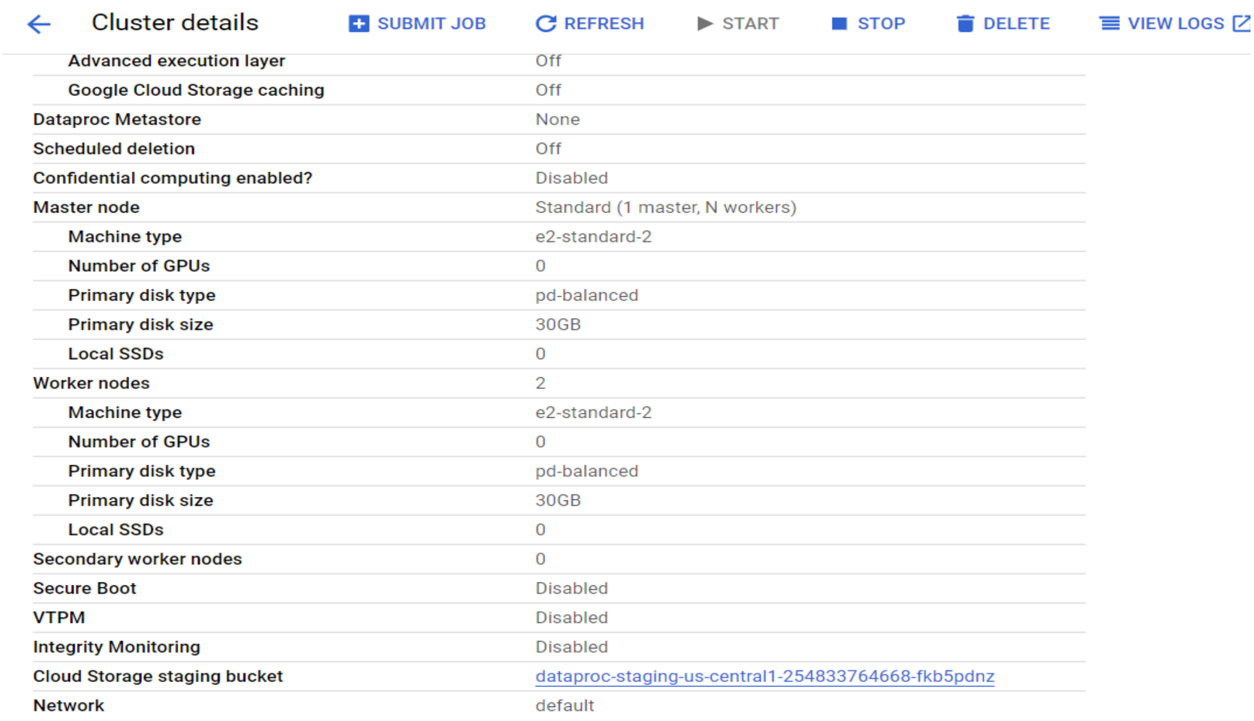
A Dataproc cluster on Compute Engine was created to run the PySpark job, with the following configuration:

Manager Node:

- Machine Series: E2
- Machine Type: e2-standard-2
- Primary Disk Size: 30 GB

Worker Nodes:

- Number of Nodes: 2
- Machine Series: E2
- Machine Type: e2-standard-2
- Primary Disk Size: 30 GB



Advanced execution layer	Off
Google Cloud Storage caching	Off
Dataproc Metastore	None
Scheduled deletion	Off
Confidential computing enabled?	Disabled
Master node	Standard (1 master, N workers)
Machine type	e2-standard-2
Number of GPUs	0
Primary disk type	pd-balanced
Primary disk size	30GB
Local SSDs	0
Worker nodes	2
Machine type	e2-standard-2
Number of GPUs	0
Primary disk type	pd-balanced
Primary disk size	30GB
Local SSDs	0
Secondary worker nodes	0
Secure Boot	Disabled
VTPM	Disabled
Integrity Monitoring	Disabled
Cloud Storage staging bucket	dataproc-staging-us-central1-254833764668-fkb5pdnz
Network	default

Screenshot: Dataproc Cluster Configuration

File Upload

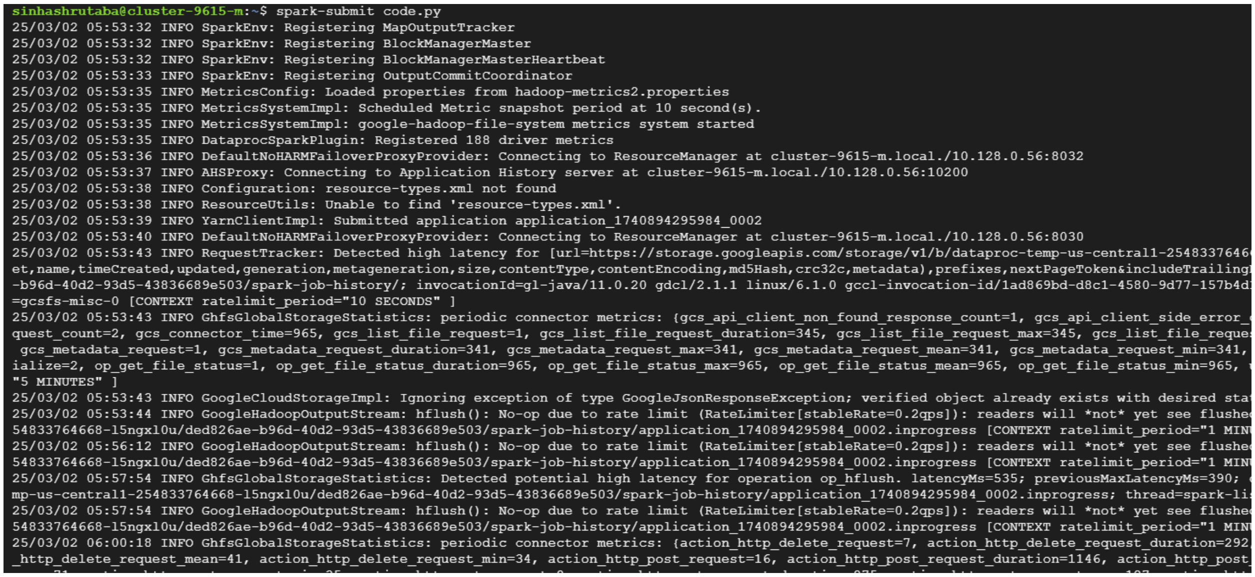
The required files were uploaded to the cluster's Master Node, using SSH on the cloud console.

- **Files:** `main.py` (PySpark script)

PySpark Script Execution

The PySpark job was executed on the cluster's Master Node, using the following command:

Unset
```
spark-submit main.py
```



```
sinhashrutaba@cluster-9615-m:~$ spark-submit code.py
25/03/02 05:53:32 INFO SparkEnv: Registering MapOutputTracker
25/03/02 05:53:32 INFO SparkEnv: Registering BlockManagerMaster
25/03/02 05:53:32 INFO SparkEnv: Registering BlockManagerMasterHeartbeat
25/03/02 05:53:33 INFO SparkEnv: Registering OutputCommitCoordinator
25/03/02 05:53:35 INFO MetricsConfig: Loaded properties from hadoop-metrics2.properties
25/03/02 05:53:35 INFO MetricsSystemImpl: Scheduled Metric snapshot period at 10 second(s).
25/03/02 05:53:35 INFO MetricsSystemImpl: google-hadoop-file-system metrics system started
25/03/02 05:53:35 INFO DataprocSparkPlugin: Registered 188 driver metrics
25/03/02 05:53:36 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-9615-m.local./10.128.0.56:8032
25/03/02 05:53:37 INFO AHSProxy: Connecting to Application History server at cluster-9615-m.local./10.128.0.56:10200
25/03/02 05:53:38 INFO Configuration: resource-types.xml not found
25/03/02 05:53:38 INFO ResourceUtils: Unable to find 'resource-types.xml'.
25/03/02 05:53:39 INFO YarnClientImpl: Submitted application application_1740894295984_0002
25/03/02 05:53:40 INFO DefaultNoHARMFailoverProxyProvider: Connecting to ResourceManager at cluster-9615-m.local./10.128.0.56:8030
25/03/02 05:53:43 INFO RequestTracker: Detected high latency for [url=https://storage.googleapis.com/storage/v1/b/dataproc-temp-us-central1-2548337646
et,name,timeCreated,updated,generation,metageneration,size,contentType,contentEncoding,md5Hash,crc32c,metadata),prefixes,nextPageToken&includeTrailing
-b96d-40d2-93d5-43836689e503/spark-job-history/; invocationId=gl-java/11.0.20 gdcl/2.1.1 linux/6.1.0 gccl-invocation-id/1ad869bd-d8c1-4580-9d77-157b4d
=gcsfs-misc-0 [CONTEXT ratelimit_period="10 SECONDS" ]
25/03/02 05:53:43 INFO GhfsGlobalStorageStatistics: periodic connector metrics: {gcs_api_client_non_found_response_count=1, gcs_api_client_side_error_
quest_count=2, gcs_connector_time=965, gcs_list_file_request=1, gcs_list_file_request_duration=345, gcs_list_file_request_max=345, gcs_list_file_reque
 gcs_metadata_request=1, gcs_metadata_request_duration=341, gcs_metadata_request_max=341, gcs_metadata_request_mean=341, gcs_metadata_request_min=341,
ialize=2, op_get_file_status=1, op_get_file_status_duration=965, op_get_file_status_max=965, op_get_file_status_mean=965, op_get_file_status_min=965,
"5 MINUTES" ]
25/03/02 05:53:43 INFO GoogleCloudStorageImpl: Ignoring exception of type GoogleJsonResponseException; verified object already exists with desired sta
25/03/02 05:53:44 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will *not* yet see flushe
54833764668-15ngxl0u/ded826ae-b96d-40d2-93d5-43836689e503/spark-job-history/application_1740894295984_0002.inprogress [CONTEXT ratelimit_period="1 MIN
25/03/02 05:56:12 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will *not* yet see flushe
54833764668-15ngxl0u/ded826ae-b96d-40d2-93d5-43836689e503/spark-job-history/application_1740894295984_0002.inprogress [CONTEXT ratelimit_period="1 MIN
25/03/02 05:57:54 INFO GhfsGlobalStorageStatistics: Detected potential high latency for operation op_hflush. latencyMs=535; previousMaxLatencyMs=390;
mp-us-central1-254833764668-15ngxl0u/ded826ae-b96d-40d2-93d5-43836689e503/spark-job-history/application_1740894295984_0002.inprogress; thread=spark-li
25/03/02 05:57:54 INFO GoogleHadoopOutputStream: hflush(): No-op due to rate limit (RateLimiter[stableRate=0.2qps]): readers will *not* yet see flushe
54833764668-15ngxl0u/ded826ae-b96d-40d2-93d5-43836689e503/spark-job-history/application_1740894295984_0002.inprogress [CONTEXT ratelimit_period="1 MIN
25/03/02 06:00:18 INFO GhfsGlobalStorageStatistics: periodic connector metrics: {action_http_delete_request=7, action_http_delete_request_duration=292
_http_delete_request_mean=41, action_http_delete_request_min=34, action_http_post_request=16, action_http_post_request_duration=1146, action_http_post_
```

Screenshot: The execution of the code

Code Explanation

Import Required Libraries

```
Python
from pyspark.sql import SparkSession
from pyspark.sql import Window
from pyspark.sql.functions import col, lag, abs, expr
import pyspark.sql.functions as F
```

Create Spark Session

This creates a new Spark session or gets the existing one with the application name "Stock Price Analysis".

```
Python
spark = SparkSession.builder.appName("Stock Price Analysis").getOrCreate()
```

Load the data from GCS

The code reads all CSV files from the specified GCS bucket, automatically detecting headers and inferring the schema of the data.

Python

```
gcs_path = "gs://oppe1_bucket/*.csv"
df = spark.read.option("header", "true").option("inferSchema",
"true").csv(gcs_path)
```

Clean and Preprocess Data

This filters out rows with null values in any of the required columns to handle bad data. Data types are explicitly cast to ensure proper handling of calculations.

Python

```
df_clean = df.filter(
    col("timestamp").isNotNull()
    & col("close").isNotNull()
    & col("open").isNotNull()
    & col("high").isNotNull()
    & col("low").isNotNull()
    & col("volume").isNotNull()
)
df_clean = (
    df_clean.withColumn("timestamp", col("timestamp").cast("timestamp"))
    .withColumn("close", col("close").cast("double"))
    .withColumn("open", col("open").cast("double"))
    .withColumn("high", col("high").cast("double"))
    .withColumn("low", col("low").cast("double"))
    .withColumn("volume", col("volume").cast("long"))
)
```

Extract Stock Identifier

This extracts information about which file each row came from, serving as a stock identifier.

Python

```
df_clean = df_clean.withColumn("stock_ticker", F.input_file_name())
```

Sort Data for Time-Series Analysis

The data is sorted by stock ticker and timestamp to ensure correct calculation of price changes over time.

Python

```
df_sorted = df_clean.orderBy("stock_ticker", "timestamp")
```

Calculate Price Change Percentage

A window function creates partitions by stock ticker, then calculates the previous close price within each partition. The percentage change is calculated using the absolute value of the difference between current and previous close prices.

Python

```
windowSpec = Window.partitionBy("stock_ticker").orderBy("timestamp")

df_with_change = (
    df_sorted.withColumn("prev_close", lag("close", 1).over(windowSpec))
    .filter(col("prev_close").isNotNull())
    .withColumn(
        "pct_change", abs((col("close") - col("prev_close")) /
col("prev_close") * 100)
    )
)
```

Cache Data for Performance

The DataFrame is cached in memory to improve performance for subsequent operations.

Python

```
df_with_change.cache()
```

Calculate Percentile Values

This calculates multiple percentiles (95th, 99th, 99.5th, 99.95th, and 99.995th) in a single operation.

Python

```
percentiles_expr = expr(
    "percentile(pct_change, array(0.95, 0.99, 0.995, 0.9995, 0.99995))"
)
percentile_values = df_with_change.select(
    percentiles_expr.alias("percentiles")
).collect()[0][0]
```

Extract Individual Percentile Values

The individual percentile values are extracted from the result array.

Python

```
p95_value = percentile_values[0]
p99_value = percentile_values[1]
p995_value = percentile_values[2]
p9995_value = percentile_values[3]
p99995_value = percentile_values[4]
```

Count Trades Exceeding Thresholds

For each percentile threshold, the code counts how many trades exceed that threshold.

Python

```
count_exceeding_p95 = df_with_change.filter(col("pct_change") >
p95_value).count()
count_exceeding_p99 = df_with_change.filter(col("pct_change") >
p99_value).count()
count_exceeding_p995 = df_with_change.filter(col("pct_change") >
p995_value).count()
count_exceeding_p9995 = df_with_change.filter(col("pct_change") >
p9995_value).count()
count_exceeding_p99995 = df_with_change.filter(col("pct_change") >
p99995_value).count()
```

Create Results DataFrame

The results are organized into a structured DataFrame with appropriate column names.

Python

```
result_data = [
    ("95th", p95_value, count_exceeding_p95),
    ("99th", p99_value, count_exceeding_p99),
    ("99.5th", p995_value, count_exceeding_p995),
    ("99.95th", p9995_value, count_exceeding_p9995),
    ("99.995th", p99995_value, count_exceeding_p99995),
]

result_df = spark.createDataFrame(
    result_data,
    [
        "Percentile of % change in stock price",
        "Value of % change in stock price",
        "Number of trades exceeding this value",
    ],)
```

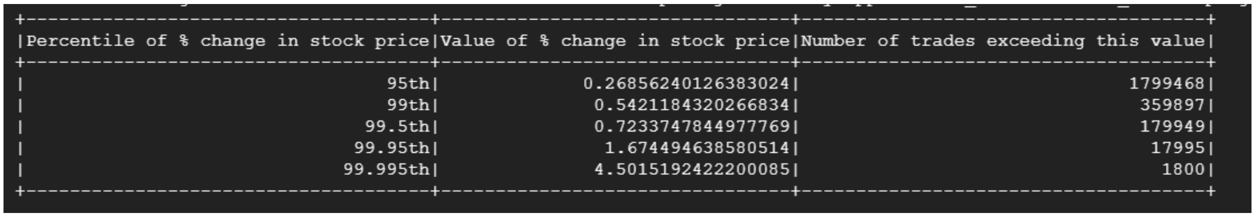
Display and Save Results

The final results are displayed in the console and saved as a CSV file with headers to an "output" location.

```Python
result_df.show()
result_df.write.csv("output", header=True)
```

Output

```
+-------------------------------------+--------------------------------+-------------------------------------+
|Percentile of % change in stock price|Value of % change in stock price|Number of trades exceeding this value|
+-------------------------------------+--------------------------------+-------------------------------------+
|                                 95th|             0.26856240126383024|                              1799468|
|                                 99th|              0.5421184320266834|                               359897|
|                               99.5th|              0.7233747844977769|                               179949|
|                              99.95th|               1.674494638580514|                                17995|
|                             99.995th|              4.5015192422200085|                                 1800|
+-------------------------------------+--------------------------------+-------------------------------------+
```



```
+-------------------------------------+--------------------------------+-------------------------------------+
|Percentile of % change in stock price|Value of % change in stock price|Number of trades exceeding this value|
+-------------------------------------+--------------------------------+-------------------------------------+
|                                 95th|             0.26856240126383024|                              1799468|
|                                 99th|              0.5421184320266834|                               359897|
|                               99.5th|              0.7233747844977769|                               179949|
|                              99.95th|               1.674494638580514|                                17995|
|                             99.995th|              4.5015192422200085|                                 1800|
+-------------------------------------+--------------------------------+-------------------------------------+
```

Screenshot: The output of the execution