Partner Summit 2022 - Workshop Student Guide

Version : 1.0.0 + date : 2022/09/05

This document guides students through the Hands on lab for Partner Summit 2022. It will take you step by step to completing the Prerequisites and deliver this demo.

Introduction

The purpose of this repository is to enable the easy and quick setup of the Partner Summit workshop. Cloudera Data Platform (CDP) has been built from the ground up to support hybrid, multi-cloud data management in support of a Data Fabric architecture. This worshop provide an introduction to CDP, with a focus on the data management capabilities that enable the Data Fabric and Data Lakehouse.

Overview

In this exercise, we will work get stock data from Alpha Vantage, offers free stock APIs in JSON and CSV formats for realtime and historical stock market data,

- Data ingestion and streaming—provided by ***Cloudera Data Flow (CDF)*** and ***Cloudera Data Engineering (CDE)**.
- Global data access, data processing and persistence—provided by ***Cloudera Data Hub (CDH)***.
- Data visualization with ***CDP Data Visualization***.

Cloudera DataFlow (CDF) is a scalable, real-time streaming analytics platform that ingests, curates, and analyzes data for key insights and immediate actionable intelligence. CDF's Flow Management is powered by Apache NiFi, a no-code data ingestion and management solution. Apache NiFi is a very mature open source solution meant for large scale, high velocity enterprise data ingestion use cases.

Cloudera Data Engineering (CDE) is a serverless service for Cloudera Data Platform that allows you to submit batch jobs to auto-scaling virtual clusters. CDE enables you to spend more time on your applications, and less time on infrastructure. CDE allows you to create, manage, and schedule Apache Spark jobs without the overhead of creating and maintaining Spark clusters. With Cloudera Data Engineering, you define virtual clusters with a range of CPU and memory resources, and the cluster scales up and down as needed to run your Spark workloads, helping to control your cloud costs.

CDP Data Visualization enables data engineers, business analysts, and data scientists to quickly and easily explore data, collaborate, and share insights across the data lifecycle—from data ingest to data insights and beyond. Delivered natively as part of Cloudera Data Platform (CDP), Data Visualization delivers a consistent and easy to use data visualization experience with intuitive and accessible drag-and-drop dashboards and custom application creation.

Pre-requisites

1. Laptop with a supported OS (Windows 7 not supported) or Macbook.
2. A modern browser - Google Chrome (IE, Firefox, Safari not supported).
3. Wifi Internet connection.
4. Git installed.

Step 1: Get github project

You can use the workshop project cloning this github repository : Workshop github repo

```
git clone https://github.com/bguedes/partner-summit-2022
```

Step 2: Get Alpha Vantage Key

Go to website Alpha Vantage
Choose link -> 'Get Your Free Api Key Today'

Choose 'Student' for description
Choose your own organisation
Fill up your professional email address

Get the license and keep it on a notepad or whatever, you will need it later on.

Store your given key, you will need it later.

Step 3: Access CDP Public Cloud Portal

Please use the login url Workshop login

Enter the username and password shared by your instructor.

You should be able to get the following home page of CDP Public Cloud.

Step 4: Define Workload Password

You will need to define your workload password that will be used to parameters the Data Services. + Please keep it with you, if you have forget it, don't panic, you will be able to repeat this process and define another one. + + Click on your profile.

Define your password. + Click button -> "Set Workload Password".

+

Check that you have this mention -> "Workload password is currently set".

+

Step 5: Create the flow to ingest stock data via API to Object Storage

CDP Portal

Choose CDF icon.

Create a new CDF Catalog

On the left menu choose -> "Catalog".
Then select the button -> "Import Flow Definition".

Fill up those parameters :

Flow Name

> (yourUserName)_stock_data

Nifi Flow Description

> Upload the file "**Stocks_Intraday_Alpha_Template.json**"

Click button "Import"

The new catalog has been added

Now let's deploy it.

Deploy DataFlow

Click on the catalog you just finished to create. + Click on "Deploy" button.

Click on "Deploy" button.

You will need to select the wokshop environment "se-workshop-1-env".

Give a name to this dataflow
Flow Name

> (user)_stock_data

Let parameters by default. Click "Next"

CDP_Password

> Fill up your CDP worload password here

CDP_User

> your user

S3_Path

> stocks

api_alpha_key

> your Alpha Vantage key

stock_list

> IBM
> GOOGL
> AMZN
> MSFT

Nifi Node Sizing

> Extra Small

Enable "Auto scaling"

> Let parameters by default

Click "Next"

You can defined KPI's in regards what has been specified in your dataflow, but we will skip this for simplication. + Click "Next"

Click "Deploy" to launch the deployment

Deployment on the run.

+

Dataflow is up and running. + In minutes we will start receiving stock information into our bucket! If you want you can check in your bucket under the path s3a://se-workshop-1-aws/user/(yourusername)/stocks/new

View Nifi DataFlow

Click on blue arrow on the right of your deployed dataflow.

+

Select the blue arrow on the right side of the deployed dataflow.

+

Select "Manage Deployment" on top right corner.

+

On this windows, choose "Action" -> "View Nifi".

You can see the Nifi data flow that has been deployed from the json file. + Let's take a quick look together.

At this stage you can suspend this dataflow, go back to "Deployment Manager" -> "Action" -> "Suspend flow". We will add a new stock later on and restart it.

Create Iceberg Table

Now we are going to create the Iceberg table. + From the CDP Portal or CDP Menu choose "Data Warehouse".

From the CDW Overview window, click the "HUE" button on the corner left.

Now you're accessing to the sql editor called "HUE".

Let's ***select the Impala engine*** that you will be using for interacting with database.
Create database using your login user050, for example replace (user) by user050 for database creation :

```
CREATE DATABASE <user>_stocks;
```

See the result

After create a Iceberg table, change (user) with your login :

```
CREATE TABLE IF NOT EXISTS <user>_stocks.stock_intraday_1min (
  interv STRING,
  output_size STRING,
  time_zone STRING,
  open DECIMAL(8,4),
  high DECIMAL(8,4),
  low DECIMAL(8,4),
  close DECIMAL(8,4),
  volume BIGINT)
PARTITIONED BY (
  ticker STRING,
  last_refreshed string,
  refreshed_at string)
STORED AS iceberg;
```

See the result

Let’s now create our engeneering process.

+

Step 6: Process and Ingest Iceberg using CDE

Now we will use Cloudera Data Engineering to check the files in the object storage, compare if it’s new data, and insert them into the Iceberg table.

From the CDP Portal or CDP Menu choose "Data Engineering".

Let’s create a job -> click Create Job".

Job Type

> Choose Spark 3.2.0

Name

> (user)-StockIceberg

Application File

> Select StockIcebergResource -> stockdatabase_2.12-1.0.jar

Main Class

> com.cloudera.cde.stocks.StockProcessIceberg

Arguments

> (user)_stocks
> s3a://se-workshop-1-aws/
> stocks
> (user)

Create it, not run it yet

This application will:

- Check new files in the new directory;
- Create a temp table in Spark/cache this table and identify duplicated rows (in case that NiFi loaded the same data again)

- MERGE INTO the final table, INSERT new data or UPDATE if exists
- Archive files in the bucket

After execution, the processed files will be in your bucket but under the "processed"+date directory

On step7, we will query data.

But right now, let show you how to create a simple dashboard, using CDP DataViz.

Step 7: Create Dashboard using CDP DataViz

Go back to CDW window.

On the menu on the left choose Data Vizualisation.

Then click the "Data Viz" button on the right.
You will access to the following window :

Choose "Data" on the upper menu.

Click "New Connection" button on the left upper corner.

Name

> (user)_dataset

Dataset Source

> From Table

Select Database

> (user)_stocks

Select Table

> stock_intraday_1min

Select "Create".

Select "New Dashboard" ->

Let’s drag from Data on the "Dashboard Designer" to Visuals.

Dimansions -> ticker

> Move it to Visuals -> Dimensions

Measures -> #volume

> Move it to Visuals -> Measures

Then on Visuals choose "Packed Bubbles"

Make it public + You have succed in a simple way your dashboard, well done + Now let's query our data and see the time travel and snapshoting capabilties of Iceberg

Step 8: Query Iceberg Tables in Hue and Cloudera Data Visualization

Iceberg Architecture

Apache Icebeg is an open table format, originally designed at Netflix in order to overcome the challenges faced when using already existing data lake formats like Apache Hive.

The design structure of Apache Iceberg is different from Apache Hive, where the metadata layer and data layer are managed and maintained on object storage like Hadoop, s3, etc.

It uses a file structure (metadata and manifest files) that is managed in the metadata layer. Each commit at any timeline is stored as an event on the data layer when data is added. The metadata layer manages the snapshot list. Additionally, it supports integration with multiple query engines,

Any update or delete to the data layer, creates a new snapshot in the metadata layer from the previous latest snapshot and parallelly chains up the snapshot, enabling faster query processing as the query provided by users pulls data at the file level rather than at the partition level.

Our example will load the intraday stock daily since the free API does not give real-time data, but we can change the Cloudera Dataflow Parameter to add one more ticker and we've scheduled to run hourly the CDE process. After this we will be able to see the new ticker information in the dashboard and also **perform time travel using Iceberg!**

Iceberg snapshots

Let's see the Iceberg table history

```
DESCRIBE HISTORY <user>_stocks.stock_intraday_1min;
```

Copy and paste the snapshot_id and use it on the following impala querie :

```
SELECT count(*), ticker
FROM <user>_stocks.stock_intraday_1min
FOR SYSTEM_VERSION AS OF <snapshot_id>
GROUP BY ticker;
```

Add new stock

Go to CDF, choose Actions and Suspend the flow. Add in parameters called (stock_list) the stock NVDA (Nvidia)

Let’s add on the parameter "stock_list" the stock NVDA (NVIDIA) + Apply changes

Start again the flow.

Check new snapshot history

Now let check again the snapshot history :

As CDF has ingested a new stock value and then cde has merge those value it has created new Iceberg snapshots Copy and paste the new snapshot_id and use it on the following impala query :

```
SELECT count(*), ticker
FROM <user>_stocks.stock_intraday_1min
FOR SYSTEM_VERSION AS OF <new_snapshot_id>
GROUP BY ticker;
```

Now, we can see that this snapshot retreive the count value for stock NVDA that has been added in the cdf dataflow stock_list parameter.

If we run this query without snapshot, we get all values, because all parents and child snapshots :

```
SELECT count(*), ticker
FROM <user>_stocks.stock_intraday_1min
GROUP BY ticker;
```

Show Data Files

```
show files in <user50>_stocks.stock_intraday_1min
```