ClueAnalytics

Datasets

Walid Darwish



# What is a Dataset

A dataset is a collection of formatted data files. They can be related or not related. The data inside these files can be columns of different data types in tabular format. Null values can exist and no guarantee of the consistency of data types in each column.

# Anatomy of a Dataset

## File Formats

A Dataset can contain any of the following file types:

|  |  |
| --- | --- |
| File Type | Comment |
| CSV Files | A file of data values separated by commas, tabs, semicolons, colons, spaces a predefined separator. Each value will be double quoted. Each line will represent a row and the first row is usually describes the names of the columns but not necessarily present. In the cases of the absence of the columns’ names row, the columns will be referred to as col1, col2, …etc. |
| XML Files | An xml formatted file where the data points are represented by either xml tags or xml attributes. In case of tags, the column names are the tag names, in case of data represented in xml attributes, the columns’ names are the attribute names |
| JSON Files | A JSON data file will contain a JSON array of objects where each object represents a row. Each row will contain a |
| ZIP Files | A zipped collection of files of any other type. A zip file can contain other zipped files. |
| Parquet Files | A data file in the apache parquet format. These files can be compressed and encrypted. In case of the encrypted files, a pair of private and public keys should be associated with the file |
| XLS/XLSX File | A Microsoft Excel portable file format. These files can have multiple sheets each represent tabular data. The file format contains metadata information and should be respected when handling these types of files. |
| Images/Infographics | Data extracted from images using image recognition machine learning algorithm. The image can be a chart or a table or an infographics format |

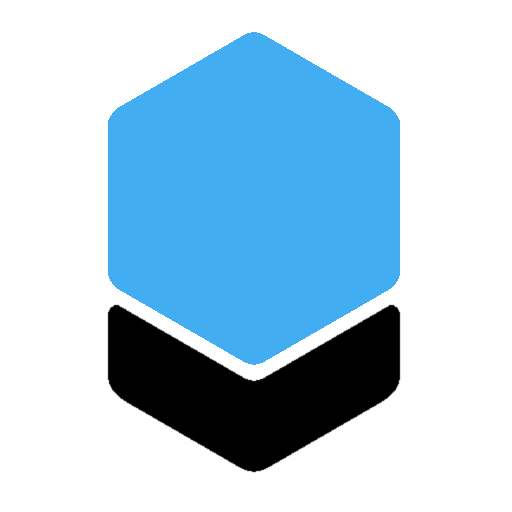
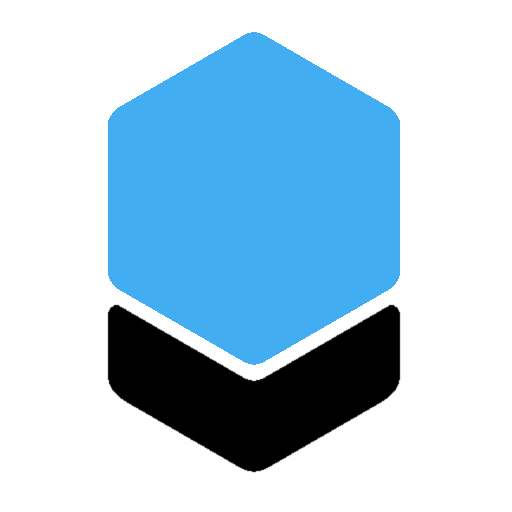
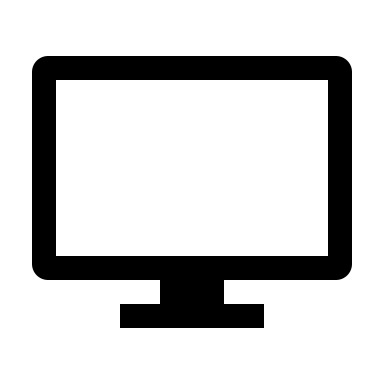
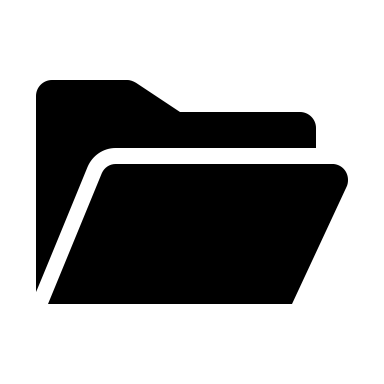
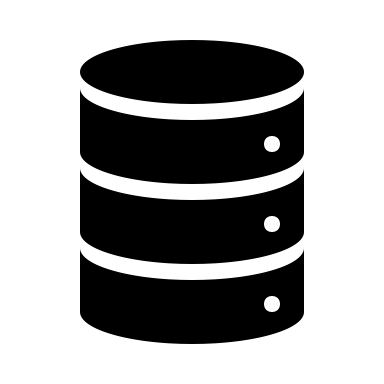
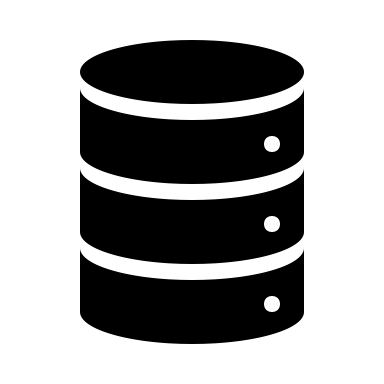
## Dataset Processing in ClueAnalytics

ClueAnalytics processing of dataset will be handled by the following components:

* ClueUI
* Orchestrator service
* Dataset Controller Service
* Datasets store (cockroachdb)
* Clue Catalog (MongoDB)
* Clue-DFS (seaweed distributed file system)

## Architecture

The following diagram describe the different involved components and the interactions between them



ClueUI

Orchestrator

Dataset Controller

Datasets Store

(cockroachdb)

ClueDFS

(Seaweed)

Clue Catalog

(MongoDB)

To guarantee data isolation, a different database for each tenant will be automatically created in Datasets Store once the tenant creates his first dataset. A separate database user will be created for each tenant along with the required certificate files. The certificate files will be saved in ClueDFS. Each time the Dataset Controller will access the datasets store, it will use the certificate files related to the tenant.

When creating databases, dataset controller will use generated readable unique names to related to the tenant. This generated name will live inside the tenant record in ClueCatalog.

When a dataset created, an implicit model will also be created with it. This model will have no model objects, but it will get them once files are uploaded.

Datafiles’ names belonging to each dataset will be saved in a list under dataset in Clue Catalog. The corresponding table name inside Dataset Store will also be saved in Clue Catalog.

When a file is uploaded, it will be saved in ClueFDS by the orchestrator. The Orchestrator will then move the file to ClueDFS (it will be better to directly upload to ClueDFS if possible, otherwise a temp space will be required). The Orchestrator will add the file name and generate a table name for the file and store them under the dataset in Clue Catalog. The Orchestrator will trigger the Dataset controller to import the file using cockroachdb IMPORT command. The file can then be deleted from Clue DFS and the orchestrator will update the dataset corresponding model to reflect the new file.

When a file is deleted, the orchestrator will delete the file and the table name from the dataset in Clue Catalog and will trigger the Dataset Controller to drop the corresponding table. The model will be updated to reflect the change.

## Use Cases

All use cases can be visualized/modified at <https://www.websequencediagrams.com/>. Use the text provided in the text box for each case.

* + 1. **Initialization – build time**

A screenshot of a cell phone

Description automatically generated

title Initialization

Build->DatasetStore: Create root user and certs

Build->DatasetStore: Create super tenant database

Build->DatasetStore: Create super tenant user (dsuser) and certs

Build->ClueDFS: Store super tenant dataset database information and certs

Build->ClueCatalog: Update super tenant record

* + 1. A screenshot of a cell phone

       Description automatically generatedC**reate a Dataset**

title Dataset Creation

CLueUI->Orch: Create a dataset

Orch->ClueCatalog: Check if tenant dataset database record exists

Orch->DatasetController: If no, create database, pass certs to dataset controller

DatasetController->DatasetStore: Create a database for tenant

DatasetController->DatasetStore: Create a user for tenant

DatasetController->DatasetStore: Generate certs for tenant

DatasetController->Orch: Done

Orch->ClueCatalog:Create tenant dataset database record

Orch->ClueCatalog:Create dataset record

Orch->ClueCatalog:Create dataset model

Orch->ClueCatalog:Attach model to dataset

Orch->ClueUI: Success

* + 1. **Upload Data File**

A screenshot of a cell phone

Description automatically generated

title Datafile upload

ClueUI->Orch: Upload a new file to dataset

Orch->ClueDFS: Load file directly to DFS - Avoid double copying

Orch->ClueCatalog: Add datafile to Dataset document - Status pending

Orch->Dataset Controller: Import file into dataset directly from DFS (Import command)

Dataset Controller->Orch: Done

Orch->ClueCatalog: Update file status - uploaded

Orch->ClueCatalog: Update dataset model

Orch->ClueUI: Done

* + 1. **Delete Data File**

A screenshot of a cell phone

Description automatically generated

title Delete Data file

ClueUI->Orch: Delete a data file

Orch->ClueCatalog: Remove data file from dataset

Orch->ClueCatalog: Remove from the dataset model

Orch->Dataset Controller: Drop the file coresspondeing table

Dataset Controller->Orch: Done

Orch->ClueUI: Done

* + 1. **Refresh data file**

TBD (Demo?)

* + 1. **Delete dataset**

A screenshot of a cell phone

Description automatically generated

title Delete Dataset

ClueUI->Orch: Delete a dataset

Orch->ClueCatalog: Remove dataset record

Orch->ClueCatalog: Remove dataset model

Orch->Dataset Controller: Drop all tables coressponding to dataset

Dataset Controller->Orch: Done

Orch->ClueUI: Done

* + 1. **Delete tenant**

A screenshot of a cell phone

Description automatically generated

title Delete Tenant

ClueUI->Orch: Delete tenant

Orch->ClueCatalog: Remove all datasets

Orch->ClueCatalog: Remove all datasets models

Orch->Dataset Controller: Drop tenant database

Dataset Controller->Orch: Done

Orch->ClueUI: Done