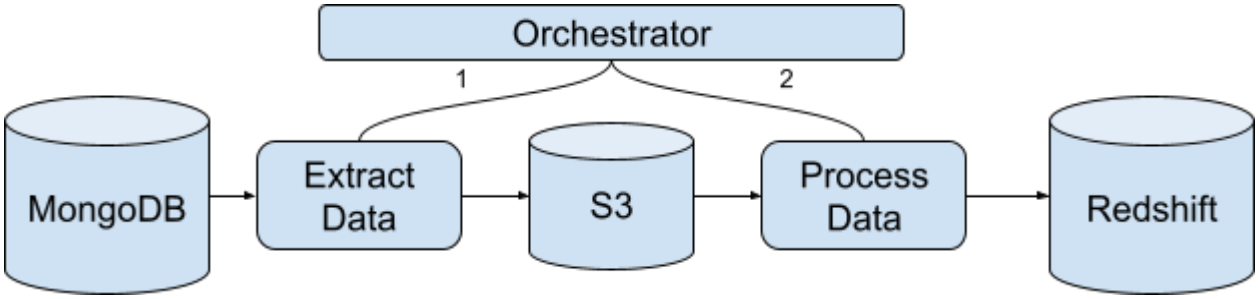
Transfer data from a MongoDB DB into Redshift

Overview of the solution / High level specification

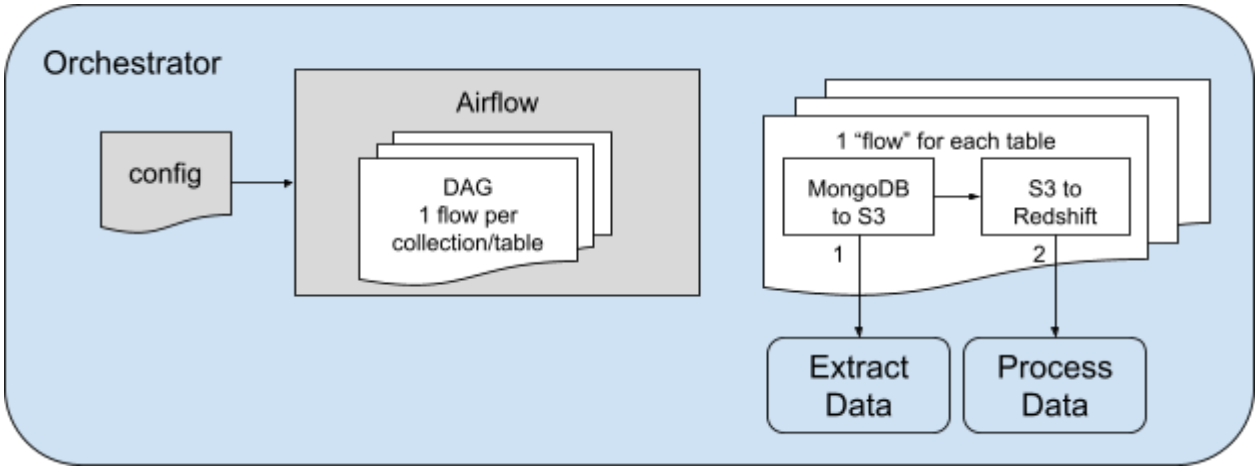


This is a very simplified overview that shows the main pieces of work that need to be done for this project. Some more detail will be discussed in this document, for each element of the data flow.

Orchestrator

This is the service that will “orchestrate” the data flow process. When it runs and what steps run in which order

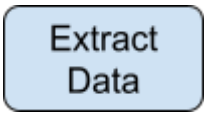
An open suggestion of what this step could look like:



MongoDB

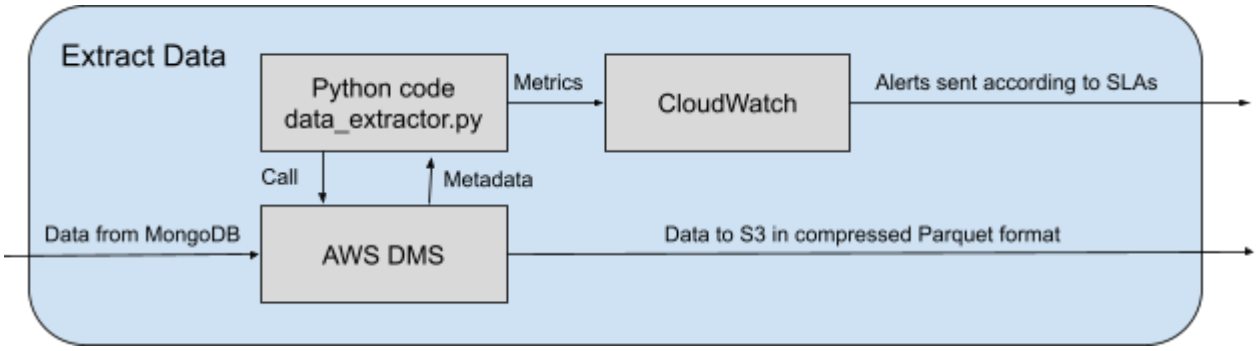
This step might look like “just MongoDB”, but ideally it should be a **read replica** and it should have **versioning enabled**, so that any change to documents is not lost. I am also making the assumption that **all documents in a given collection have the same schema**.

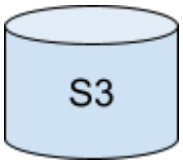
Extract Data



This step should extract CDC data from The MongoDB database on a scheduled way.
It should also log status (success/failures) and other metrics (number of documents extracted, volume, …) into a logging server from which NRT alerts can be triggered. It is very important that this step is able to extract data from a MongoDB collection and is able to export that data into a tabular format (e.g. parquet).

An open suggestion of what this step could look like:



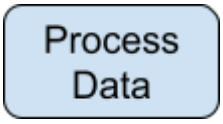


S3

Although this is actually just an S3 bucket, some precautions should be taken into consideration:

- Versioning should be enabled
- Data at rest should be encrypted
- Logging should be enabled for all operations

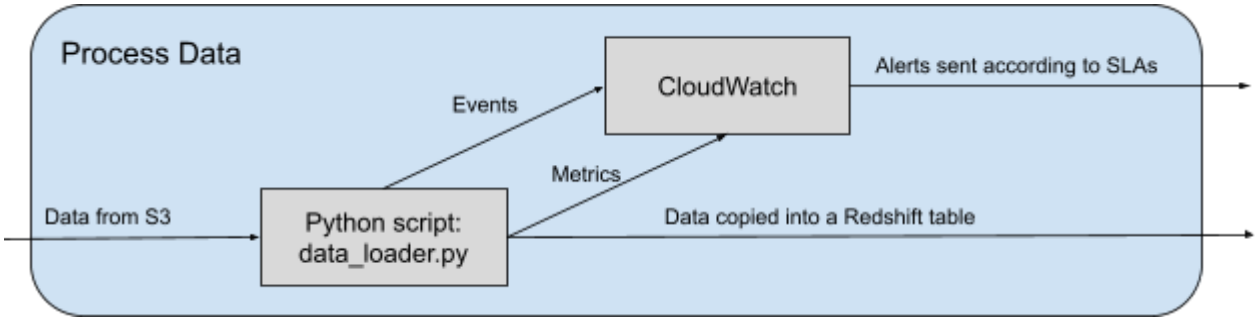
Process Data



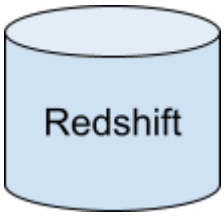
This step should be Python code that is triggered from the same orchestrator that also calls “Extract data” and should start as soon as the previous finishes (it’s also possible to run it as a trigger from an S3 event). The code should run in a Docker container (e.g. using Batch and spot instances).
The Python code needed for this step is addressed in more detail on the code repository where this document is also located.
A few key aspects of the deliverable:

- Any schema changes should be detected at this point and a decision should be made. For now and thinking in doing an initial MVP, any schema changes should stop data ingestion from that collection.
- New schemas should be auto-detected and tables created on Redshift that are able to ingest that data.

An open suggestion of what this step could look like:



Redshift



Again, this is simply a Redshift cluster and “normal” actions should be taken, regarding security, logging and monitoring.