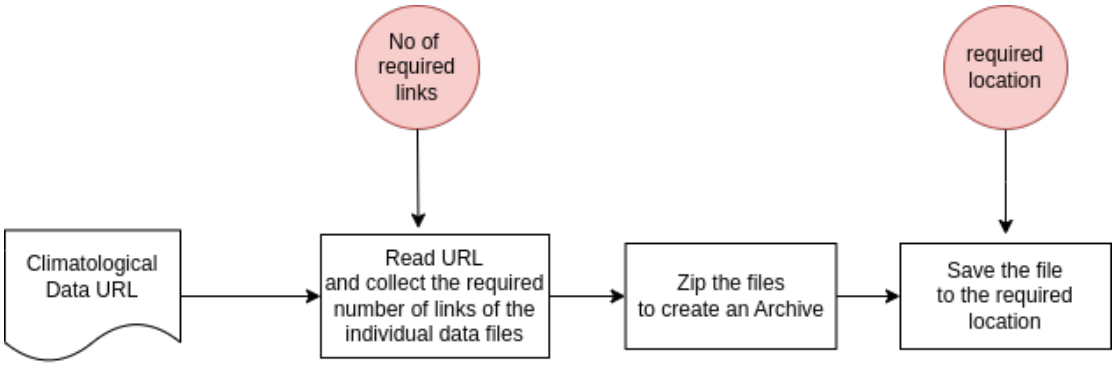
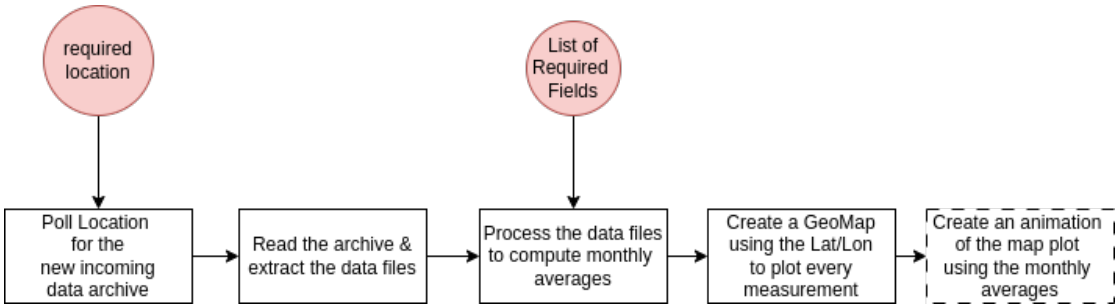
Setup a Data Engineering Pipeline

We are going to setup a pipeline for acquiring public domain climatological data from **National Centers for Environmental Information** (https://www.ncei.noaa.gov/). The archive https://www.ncei.noaa.gov/data/local-climatological-data/access/YYYY/) contains data collected from over 13400 stations from **YYYY=1901** to **YYYY=2024**. Each year in every station the data is collected every hour on Altimeter, DewPointTemperature, DryBulbTemperature, Precipitation, PresentWeatherType, PressureChange, PressureTendency, RelativeHumidity, SkyConditions, SeaLevelPressure, StationPressure, Visibility, WetBulbTemperature, WindDirection, WindGustSpeed, WindSpeed,Sunrise, Sunset readings. We want to setup another pipeline that will process the data to generate geomaps for different parameters.

Pipeline 1



Pipeline 2



Task 1 (DataFetch Pipeline) **(4x5=20 points)**

Given the base URL (https://www.ncei.noaa.gov/data/local-climatological-data/access/) and the year YYYY, create a pipeline in Apache Airflow for

1. Fetch the page containing the location wise datasets for that year. (Bash Operator with *wget* or *curl* command)
2. Based on the required number of data files, select the data files randomly from the available list of files. (Python Operator)
3. Fetch the individual data files (Bash or Python Operator)
4. Zip them into an archive. (Python Operator)
5. Place the archive at a required location. (Bash/Python Operator)

Task 2 (Analytics Pipeline) **(6x5 = 30 points)**

Knowing the place to fetch the data file, let's unzip the archive into individual data files in CSV format. Followed by processing each of the data files in parallel using the Apache Beam framework (with DirectRunner) to generate data visualization using geomaps. The Airflow pipeline should have the following operations. The Airflow pipeline should be set to an autotrigger that get activated every 1 min.

1. Wait for the archive to be available (with a timeout of 5 secs) at the destined location. If the wait has timed out, stop the pipeline. (FileSensor)
2. Upon the availability (status=success), check if the file is a valid archive followed by unzip the contents into individual CSV files. (BashOperator)
3. Extract the contents of the CSV into a data frame and filter the dataframe based on the required fields such Windspeed or BulbTemperature, etc. Extract also the Lat/Long values from the CSV to create a tuple of the form <Lat, Lon, [[ArrayOfHourlyDataOfTheReqFields]]>. (PythonOperator using Apache Beam)
4. Setup another PythonOperator over ApacheBeam to compute the monthly averages of the required fields. The output will be of the form <Lat, Long, [[Avg_11, ..., Avg_1N] .. [Avg_M1, ..., Avg_MN]]> for N fields and M months.
5. Using 'geopandas' and 'geodatasets' packages, create a visualization where you plot the required fields (one per field) using heatmaps at different lat/lon positions. Export the plots to PNG. (PythonOperator using ApacheBeam)
6. Using a suitable tool like 'ffmpeg' or equivalent create a GIF animation using the 12 months data in PNG image format (BashOperator) [Optional]
7. Upon successful completion, delete the CSV file from the destined location.

Important Pointers

1. You are expected to submit *.py files (DAGs and Beam pipelines) instead of Jupyter notebooks.
2. Here is the allocation of points per task
   20% for a clean coding style
   50% for the correctness of the implementation
   20% for readable comments
   10% for input arguments' validation/boundary check

Best Wishes.