Real-Time Analytics on unstructured data sets

Background

At Big Wind LLC, we provide timely analytics related to weather data to our customers.

One of our leading solutions allows our customers to query high-resolution weather forecast data that is provided by the UK Met Office. This solution provides these core capabilities to our customers:

- High availability
- Quick response
- Timely availability of new data

Because of the significant financial losses they would incur, our customers cannot accept any downtime. Our solution provides them with the data they need to plan and the flexibility to make sure downtime does not occur. For example, airport and airline operators use our solution to plan their flights or predict delays caused by weather.

Goal

You are tasked to provide reporting and dashboard access to the forecast data with Amazon QuickSight so that Data Analysts can query the data and show the data insights in tabular format.

In this exercise, you will design, implement, and showcase the secured and scalable data pipelines that ingest, transform, and support visualization of the data in Amazon QuickSight. All new data should be available to the Data Analysts as fast as possible.

You can choose any variables in the dataset to support one or more business use cases of your choice. For example, your dashboard could focus on airport operators and report the forecast on specific variables used by airline operators (dew point, air temperature, wind speed) during takeoff.

Data

Your solution should use the MOGREPS-UK dataset and assume that the files are available in netCDF format every few minutes with 20–30 MBs of data.

A sample dataset is available at the S3 location referenced in link [1] and more details on the dataset can be found at the location referenced in link [2]. These links are available in the Links section.

Each file contains weather forecasts that predict variables such as air temperature, humidity, and rainfall. Each variable is predicted for these dimensions: latitude, longitude, altitude (when applicable), pressure (when applicable), and time.

Latitude and longitude values use a RotatedGeogCS coordinates system. Time is stored as number of time periods (for example, hours) since epoch.

You may use libraries such as Iris to explore the file contents, but, make sure to note the library and dependencies size when you deploy it to production. The Sample Code section includes a code sample to help you explore the file contents. Distributed frameworks also have libraries that support netCDF manipulation.

Compute

Build the solution using AWS services. You can use any services or frameworks that show our core capabilities, security, and cost effectiveness. The $10 AWS credit code provided in the email you received should cover the cost of this exercise.

Note

- **Stop all EC2 instances, terminate all clusters and disable any computation triggers after you complete the exercise.**
- **Delete all data and resources after your recruiting process concludes.**
- We recommend that you tag every resource used in the exercise with this key/value: *environment/awstakehome*.
- If there are AWS services you want to use that are not yet available in the region(s) you chose, make sure to indicate that in the architecture document and use a region where the service is available.

Output

Send us a single PDF document that includes this information:

- IAM User credentials (AccessKey/SecretKey; username/password) with read access to the console, services, and resources you used to create the solution. We will use the credentials to review the implementation.
- A document that details your architecture, thought processes, and any code samples and Amazon QuickSight dashboard(s). In your architecture, make sure to highlight how you met our core capabilities, security, and cost effectiveness.

Sample Code

```
### To install iris, install conda and run "conda install -y -c conda-forge iris"

import iris
filename='prods_op_mogreps-uk_20140717_03_11_015.nc'
#Load the list of cubes. See [3] in the links section for more details.
listofcubes = iris.load(filename)
```

```
print(listofcubes)
#Extract one cube, air_temperature measure in Kelvin. See [3] in the links section for more details.
cube=listofcubes[8]
print(cube)
#Number of dimensions for the time coordinate
cube.coord('time').points
#Number of dimensions for the pressure coordinate
cube.coord('pressure').points
#Number of dimensions for the longitude coordinate
cube.coord('grid_longitude').points.size
#Convert from kelvin to Celsius
cube.convert_units('celsius')
#Print the cube contents
print(cube.data)
#Get a subset of the data
print(cube.data[0][0][0][0:10])
```

Links

[1] s3://awsbigdatatakehome/ (region: eu-west-2)

[2] https://aws.amazon.com/public-datasets/mogreps/

[3] https://scitools.org.uk/iris/docs/latest/userguide/iris_cubes.html and https://scitools.org.uk/iris/docs/latest/userguide/loading_iris_cubes.html

Measurement Criteria

1. *Work Sample (70)*
    a. *Modular design*
    b. *Scalable Architecture*
    c. *High Availability*
    d. *Timely availability of new data (ingest, transform)*
    e. *Responsive (data model, index)*
    f. *Security*
    g. *Cost Effective Solution (storage, compute)*
2. *Presentation (30)*
    a. *Clarity of explanation (15)*
    b. *Focus on business outcome (15)*
3. *Extra Points*
    a. *Well documented solution with code samples*
    b. *Spark / Flink implementation*

c. *Think Big - architecting solution with future work in mind (visualization, data lakes, AI/ML)*