Designing an Analytics Pipeline using Airflow

Learning Journey

Solve a Pandemic Management Case Study

Need a DB on the cloud to store data	Learn how to setup & use MySQL DB on the cloud
Need a way to send message notifications	Learn how to send automated messages on Slack
Need a way to keep my code modular	Learn how to create plugins to keep your code clean
Need a way to send data across tasks	Learn how to use Xcoms in Airflow

Agenda

- SQLDB on the Cloud
  - Setting up a SQL database on the Cloud
  - Connecting to the Cloud DB with MySQL Workbench
  - Creating tables and importing data into MySQL
  - Connecting the Cloud DB with Python
- Creating a Slack App and sending messages using a webhook
- Using plugins to make your code modular
- Sharing metadata across tasks in a DAG
- Pandemic Management Case Study
  - Context, Objective and Data Understanding
  - Solution Approach
  - Implementation
  - DAG and Plugin codes
  - Hands-on demo & Simulation
- Summary

Pandemic Management Case Study

Context: To tackle public health emergencies, clear and reliable communication is critical. Having communication systems in place at this time can help a country prepare for a pandemic. We'll take historical data from a zone to construct an automated pipeline that analyses daily case statistics and sends a summary message to an open channel to keep officials informed so they can make timely decisions.

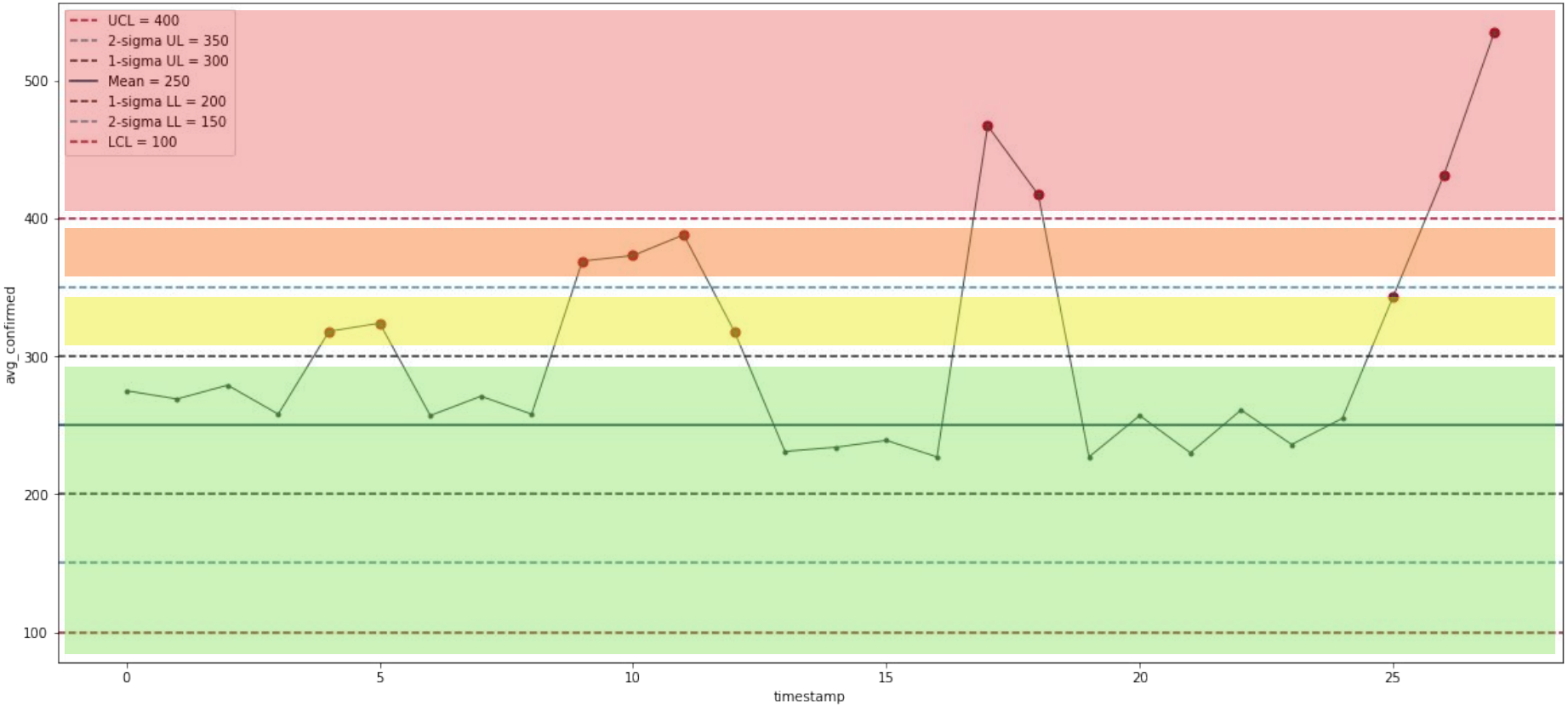
Objective: As an analytics engineer, our goal is to create an airflow DAG that can read hourly statistics in a specific area, such as new confirmed COVID cases, hospitalizations, and so on, and calculate metrics like the number of beds available and the amount of O2 (litres) remaining in the zone's care centre. We also need to send the zone's end-of-day status (**GREEN**, **YELLOW**, **ORANGE**, **RED**)

Solution Approach

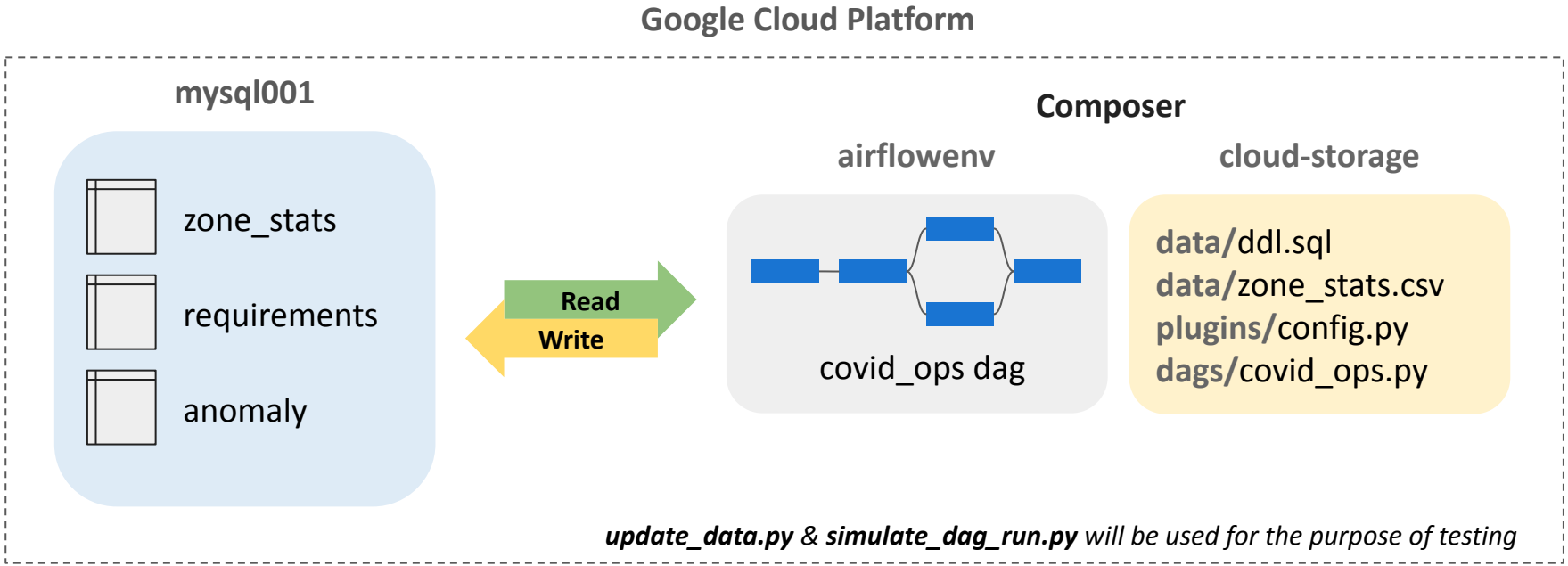
The Airflow DAG must be designed to do the following:

- Read the zone statistics that arrive every hour from a SQL database, compute the % beds available & the % O2 in liters remaining in the care center and send this as a notification on a Slack channel
- At the end of every 24 hours, aggregate the day's data, check the status of the zone depending on the average confirmed cases for the day and send this as an update on the Slack channel. The color of the zone will depend on the hypothesised mean value of the cases (250) and the standard deviation (50)

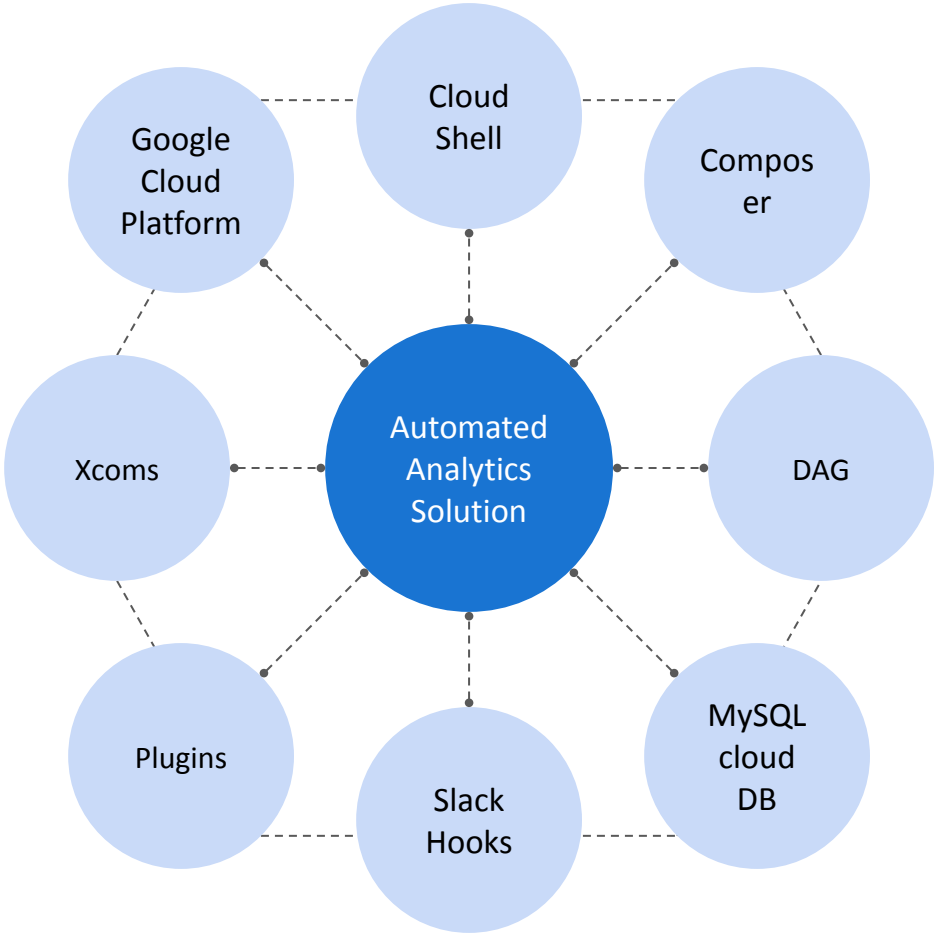
Solution Approach



Implementation



Summary


Cloud Shell
Google Cloud Platform
Compos er
Xcoms
Automated Analytics Solution
DAG
Plugins
Slack Hooks
MySQL cloud DB

Great Learning

Happy Learning !