


Google Planning

12 May 2023

 Quantiphi availability: 6AM - 2PM EST

Partnership goals.

 We are open to guidance and partnership on this journey; please show us what good MLOps is and we'll learn and grow from that. Ideally we'll be learn from these running examples and be able to reuse the MLOps patterns in our solutions.

And...

All proposed designs should mitigate risk by utilizing the good work already in place that terraform offers for cloud infrastructure. We're aiming to refine and revise the DS user experience clearly within the bounds of broader ITG, not step outside that; we're not aiming to replace ITG in any way.

80% (or more) - Forecasting Factory & DS User Experience improvements

The primary focus to help the Data Science team is to design and build a running MVP of a Forecasting Factory in an HCA GCP project, using HCA user roles and permissions, with non-PHI data. The Forecasting Factory generally consists of two parts: 1) well architected data engineering patterns utilizing GCP Dataflow and Vertex AI Pipelines for ETL/ELT, and 2) running examples of reusable vertex AI pipelines demonstrating MLOps patterns. The data engineering would ideally support quickly and easily adding new features to a feature store. The first HCA example use case for this should be staffing forecasts; this example should modularize location specific mappings so that a change to one facilities ETL/ELT or modeling technique does not impact others.

A growth proposal for the Forecasting Factory could walk a path similar to the following.

- Crawl: Example Forecasting MLOps with public datasets
- Walk: Extend patterns from Crawl stage to incorporate additional features (Weather, search trends, etc.) to showcase advanced Forecasting MLOps solutions

Primary deliverables:

- A Running non-production Forecasting Factory in an HCA GCP Project.
- Goal: self-service Backstage (HCA internal name is the Evolve Internal Developer Platform) template that:
 - Creates a github PR into an existing github repo (that contains terraform). The contents of that PR includes: scripts or terraform to be run by GH actions, one time. That one-time running GH action will setup an example of a Level 1 MLOps vertex ai pipeline in an existing GCP project. That pipeline does multivariate time-series forecasting on a readily available public data set. In addition to the vertex AI pipeline, we need to readily be able to assess if a model is ready for deployment—that its predictive performance is better than a certain baseline - in a graphical interface (not in a notebook, perhaps with GCP metrics graphs?). This architecture will incorporate python classes to logically group reusable functions, and make extensive use of comments throughout the example pipeline to point a mlops engineer where changes are expected to be made to adapt the reference architecture to a real world forecasting problem.
- Code for the above:
 - python, scripts, Notebooks, cloud asset definitions (.yaml's), GitHub Actions, Readme's, configs
 - Terraform Modules w/ examples
 - These could be potentially be generated via AutoMLOps - <https://github.com/GoogleCloudPlatform/automlops>

Secondary asks - 20% to the Forecasting Factory would be to improve the overall DS user experience.

Overall the DS user experience could be improved at the end of this project, and the time it takes a DS to take a new problem from idea to production should be reduced to 1/2 day (4 hours) or less. Specifically, we'd like the following DS user pain points to be improved as part of this work as well 1) easier solution CI/CD, by using Cloud Build instead of GH Actions (what triggers those builds, ???, open to solutions) 2) utilization of appropriate ETL/ELT and deployed DS solution arch by primarily using vertex ai pipelines or GCP Dataflow patterns instead of cloud run. Hopefully the majority of DS user tasks to go from "new idea to deployed solution" should be made available to data scientists via GCP console (mostly Vertex AI), or through self-service automation; these tasks *ideally don't require* data scientists to learn and use terraform.

With that said, see note above about utilizing terraform to mitigate risk.

If needed to improve the DS User Experience, revisions and updates to the existing TDD's (Data Science and Analytics Infrastructure & MLOps) would be helpful. They are key documents that align all parties, and are a way to capture feedback and incorporate agreed upon designs from interested groups.

Hoped for secondary deliverables:

- Self-service automation for the following:

- Adding a notebook to an existing GCP project
 - Adding a starter MLOps pipeline, complete with build and solution deployment GCP components - Use this example from AutoMLOps of a deployed inference solution as a base repo structure - <https://github.com/GoogleCloudPlatform/automlops/tree/main/examples/inferencing/AutoMLOps>
 - Stretch goal - Generating a whole new GCP project, and provisioning a starter MLOps solution within it.
-

Interactions and Deliverables

Author: Nick Selleh Date: 5/19/2023

