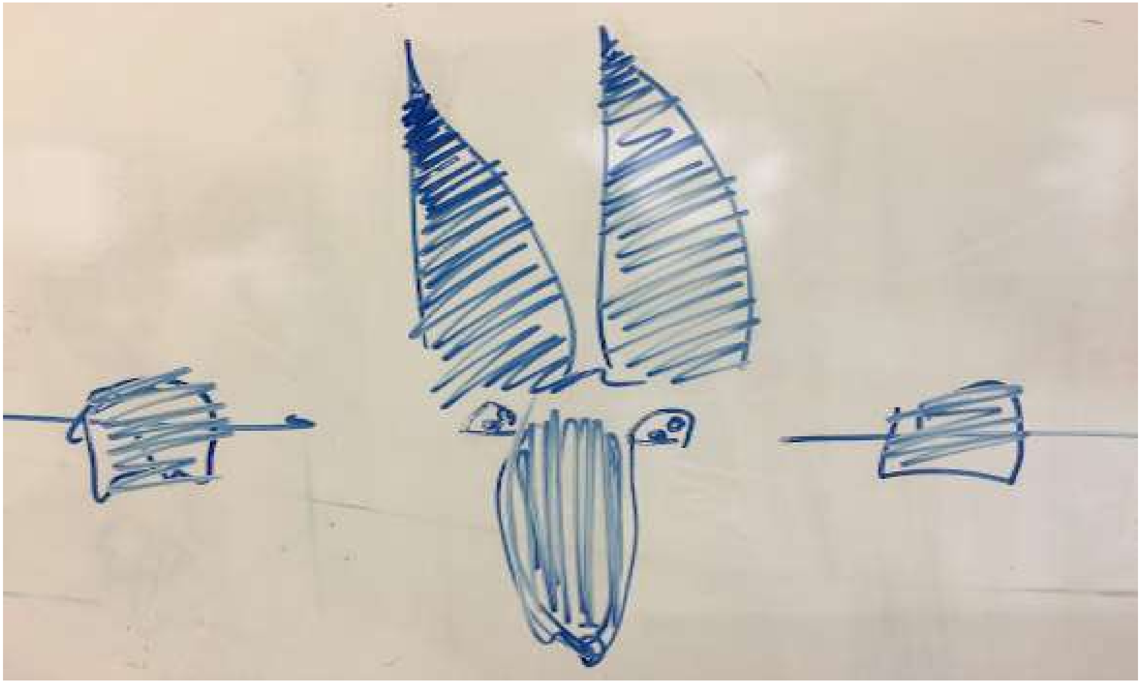
Anubis

A "framework/tool/service" that enables you to perform benchmarks of your machine learning models/frameworks/hardware in order to evaluate functionality, performance and detect regressions.



Anubis provides a simple, self service solution for teams to schedule and run benchmarks for machine learning workloads. Abubis lets teams to easily spin-up the required infrastructure and automation to perform benchmarks. Consider Anubis a measuring tool, a "ruler" if you willl, that lets you measure the efficacy of your machine learning model, framework and/or computing infrastructure. Anubis provides a simple, declartive, input descriptor, and produces operational and user metrics, and sophisticated alerts for detecting regressions. Logged output from the model are stored and indexed for deeper inspection, allowing for more effective root cause investigation.

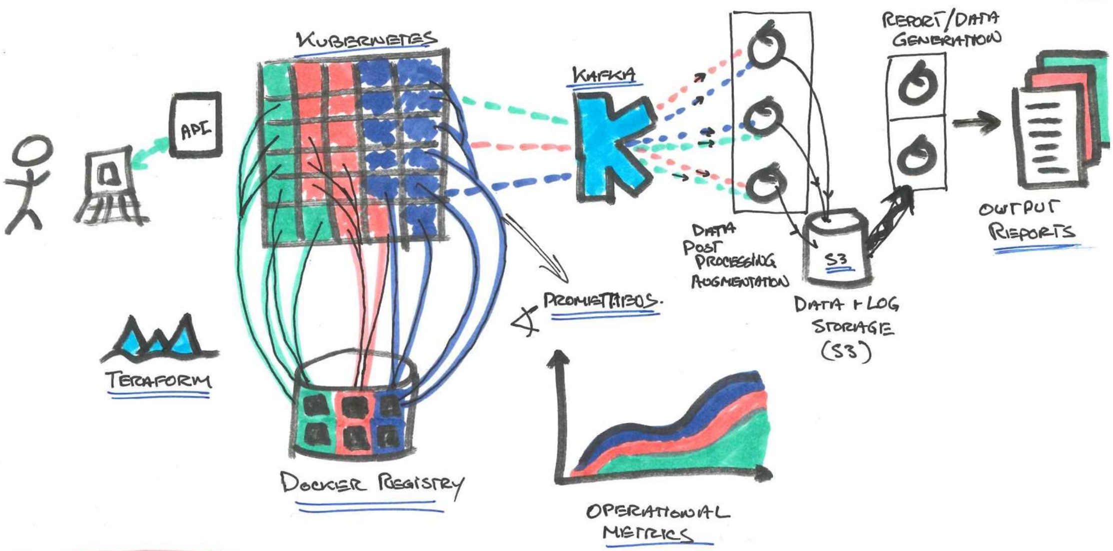
Design and architecture

The system is built to embody a few guiding tenets (see details):

- Self Service
- "Push Button" Installation
- Complete Delivery
- Clear Mental Model (Transparency)
- Use best of breed technology that has significant mind share

Many design and implementation decisions were made, all of them well informed. Please read our Anubis Design FAQ if you are curious.

The picture below shows the "infrastructure" layer of Anubis.



ENGINEERING BIAS

- FULLY FLESHED OUT DOCKER CONTAINERS./THIN MACHINE IMAGES
- OPEN TECHNOLOGIES WITH LARGE MINDSHARE
- MONITOR ALL OF THE THINGS!
- SEPARATION OF CONCERNS VIA LOOSE COUPLING VIA EVENTS

The orchestration layer consists of services running on K8s, sending events to each other to coordinate the activities necessary to shepherd model runs. More specifically events are sent to kafka on specified topics. Kafka facilitates the event sending throughout Anubis.

Technologies

This project is an exercise in the amalgamation and orchestration of several technologies to create a tool that adds value to our users.

- Python: (https://www.python.org/)
- Clojure: (https://clojure.org/), (https://www.braveclojure.com/do-things/)
- Bash: (https://www.tldp.org/LDP/abs/html/)
- Jq: (https://stedolan.github.io/jq/)
- Conda: (https://docs.conda.io/projects/conda/en/latest/index.html)
- Docker: (https://www.docker.com/)
- Kafka: (http://kafka.apache.org/intro) (MSK)
- Kubernetes: (https://kubernetes.io/) (EKS)
- Prometheus: (https://prometheus.io/)
- Terraform: (https://www.terraform.io/)

- Zookeeper: (https://zookeeper.apache.org/)
- ElasticSearch: (https://www.elastic.co/products/elasticsearch) (Managed ElasticSearch)

Supported cloud providers

(currently)

- AWS