Going Cloud Native In Baseball



Matt Dering & Long Nguyen

KubeCon | CloudNativeCon
North America 2026

Matt Dering

- Principal Software Engineer - ML Engineering
- Started December 2023



Long Nguyen

- ML Engineer
- Former Quantitative Analyst
- Started May 2021



Lay of the Land

- Approximately 25 data scientists
  - 13 SWEs, 2 in our ML Engineering group
- Variable technical abilities
- Over 100 different statistical products
- Our job: put the power of the cloud in the palms of their hands
  - Without spending all day on it

Tech Stack

- 95% Python shop
- Data ingested to mysql and copied to Bigquery, or BQ directly
- Models pull data from Bigquery and push inferences to bigquery
- Frontend pulls inference data from RDB, copied from BQ
- Deployed work via Airflow, runs in K8s

Containers

Challenges:

- “Variable Technical Ability”
- Most code in a single repo
- Most code in a single **image**
    - Meaning projections use the same set of dependencies as our web services
- Completely locked up. Unable to upgrade past Python 3.9 without breaking production environments

Solutions:

- Basic siloing system within our monorepo
- Each project gets access to shared code, plus its own dockerfile, dependencies, etc.
- Templated structure: cookiecutter
- Map cleanly from Docker Compose to Kubernetes pod definitions.
- CLI for creation and launching
- Simple bespoke build system

Jobs

Challenges:

- Workloads run in preallocated VMs.
- Prototyping in individual VMs.
- Clunky resource management.
- Manual GPU management.

Solutions:

- Migrate to Kubernetes jobs.
- Autoscale clusters by resource needs.
- Provision GPUs for accelerated workloads.
- Support distributed workloads via KubeRay.

Models & Inference Services

Challenges:

- Model runs are bespoke scripts by analysts with no standardized entrypoint.
- Not composable.
- Support only batch processing.

Solutions:

- Develop a standardized model interface.
- Enable meta-training, e.g. hyperparameter tuning and surrogate modelling.
- Training and inference model runs via Kubernetes jobs and KubeRay.
- Deploy as inference services via KServe with autoscaling and canary rollout.
- Composable into inference pipelines.

Microservices

Challenges:

- Not enough backend developers to support growing product demands.
- Outdated patterns with limited concurrency.
- Manual deployment in dedicated VMs.

Solutions:

- Develop framework to build APIs with familiar Python classes, functions, and type hints.
- Deploy as FastAPI servers via Kubernetes deployments and services.
- Rollout via CI/CD.
- Concurrency at app and deployment level.
- Automated schema compatibility checks.
- Auto-generated clients and CLIs.

Events & Functions

Challenges:

- ETL from multiple data sources
- Loosely coupled python scripts writing to mysql DB
- backfills/reruns would run on a single node for ~2 weeks
- Entirely batch
- Data updated asynchronously

Solutions:

- Knative stack
- Lambdas do extraction, validation, transformation, etc
  - Simplification of code
- Eventing does loading, coordinates dependencies
  - No more loosely coupled scripts
- Serving brings us scalability
  - No more single node 2 week jobs
- Exactly the same live as in batches
- Allows us to adopt industry standard patterns