Paperspace

Productionizing ML & DL Models At Scale

STAC Summit June 6, 2019

Rajesh Vadlamudi
Solutions Manager

About Paperspace

Our Team

- Founded by Dillon Thompson Erb and Daniel Kobran in 2015
- Based in Brooklyn, NY, with 30+ team members worldwide

Our Goal

- Provide a high-performance cloud infrastructure and integrated development/deployment environment for deep learning

Paperspace

Agenda

1. The State of ML/DL Development
2. Barriers to Entry
3. CI/CD for Traditional Apps vs. ML/DL
4. Emerging Best Practices

“

By 2023, AI and deep-learning techniques will be the most common approaches for new applications of data science”

Alexander Linden

VP, Artificial Intelligence & Machine Learning at Gartner

The State of ML/DL Today

Raw components are here.

- Algorithms
- Massive data streams
- Parallel compute

Currently limited to select few.

Only accessible to Uber/Facebook that have built large pipelines internally.

Primed to enter mainstream.

Every company stands to gain. Most are struggling with putting all the pieces together to start.

Challenges

No tooling, ML-Ops is nascent.

Nothing for developers to plug into. Loss of productivity.

AI hardware is rapidly evolving.

Infrastructure is quickly outdated and difficult to maintain.

Hurdles to productionize.

Significant upfront investment, time to market is 6 to 18 months!

The biggest barrier to AI adoption is an infrastructure and tooling problem, not an algorithm problem

The field of ML / DL / AI has quickly outpaced the tools available to it's practitioners

Business Objective

Infrastructure

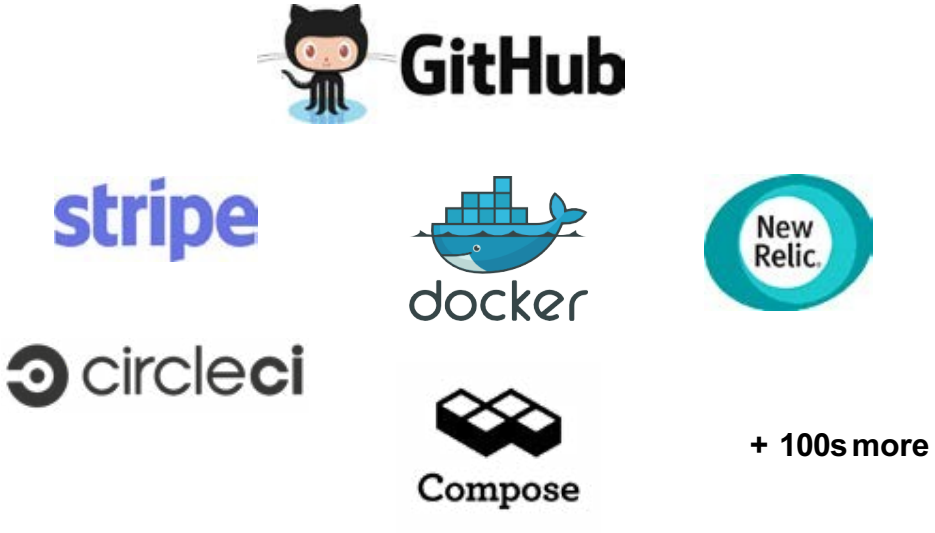
BI
Prediction
Optimization
Recommender systems
Any heuristic

GPUs
Datastore
Algorithms (CNNs, RNNs, ...)
Toolsets (Pytorch, TensorFlow, etc)

Paperspace

The Software Development ecosystem is rich

Traditional Software



Storage, CDN, deploy, monitor, VPC, Load balance, IPsec, CI/CD, DNS . . .

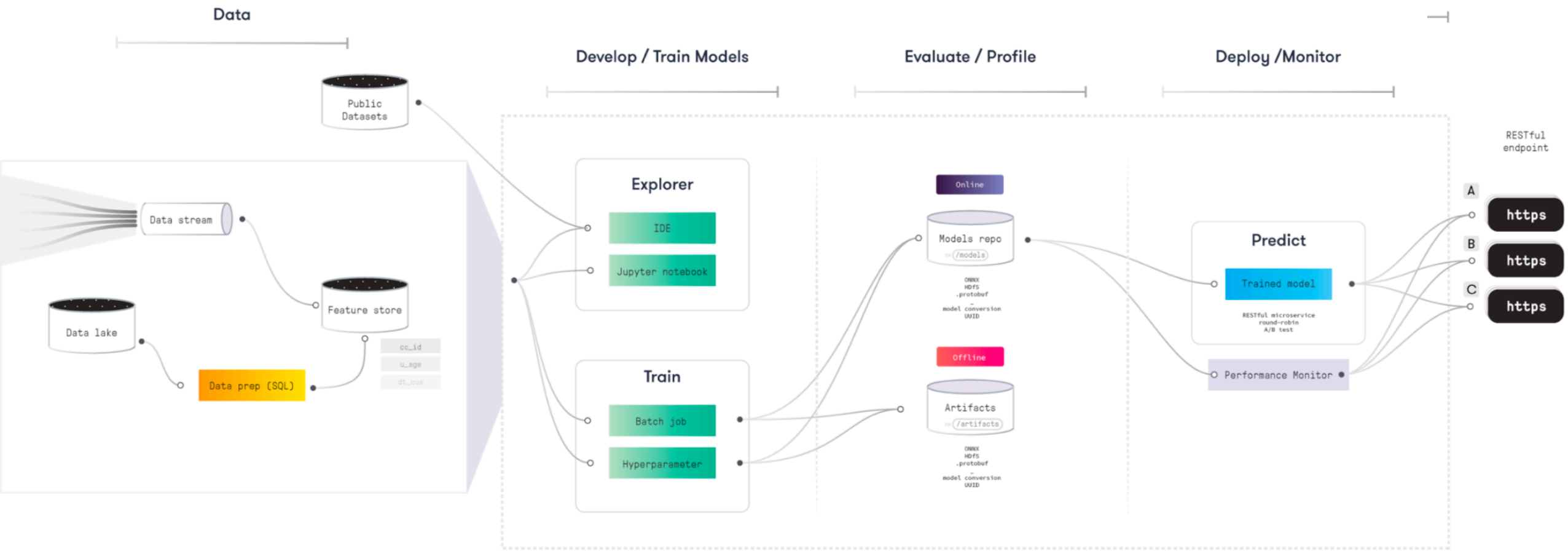
Deep Learning

data, notebooks, train, visualize, collaborate, version, hyperparameters . . .

Paperspace

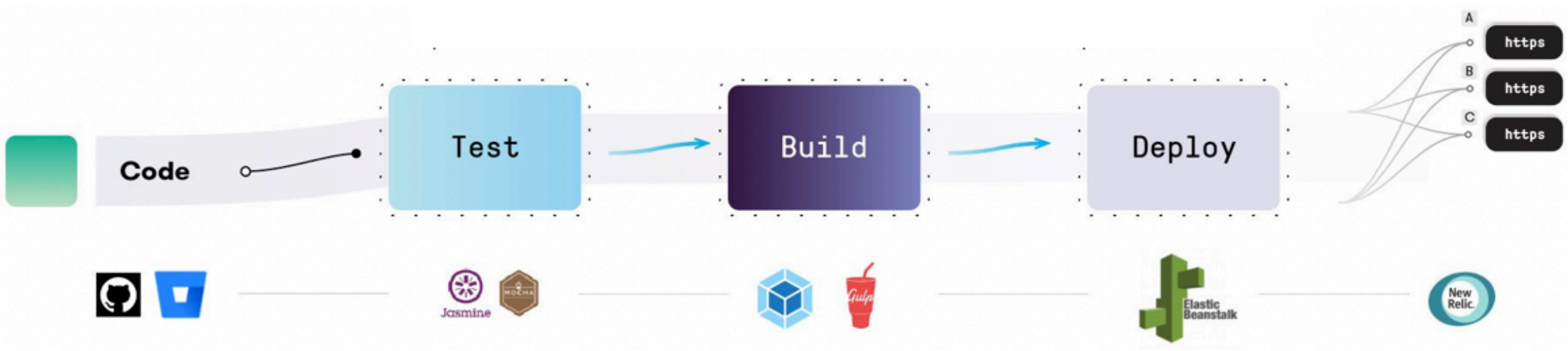
ML engineers spend 80% of their time managing infrastructure and tooling.

The ML/DL Development Lifecycle

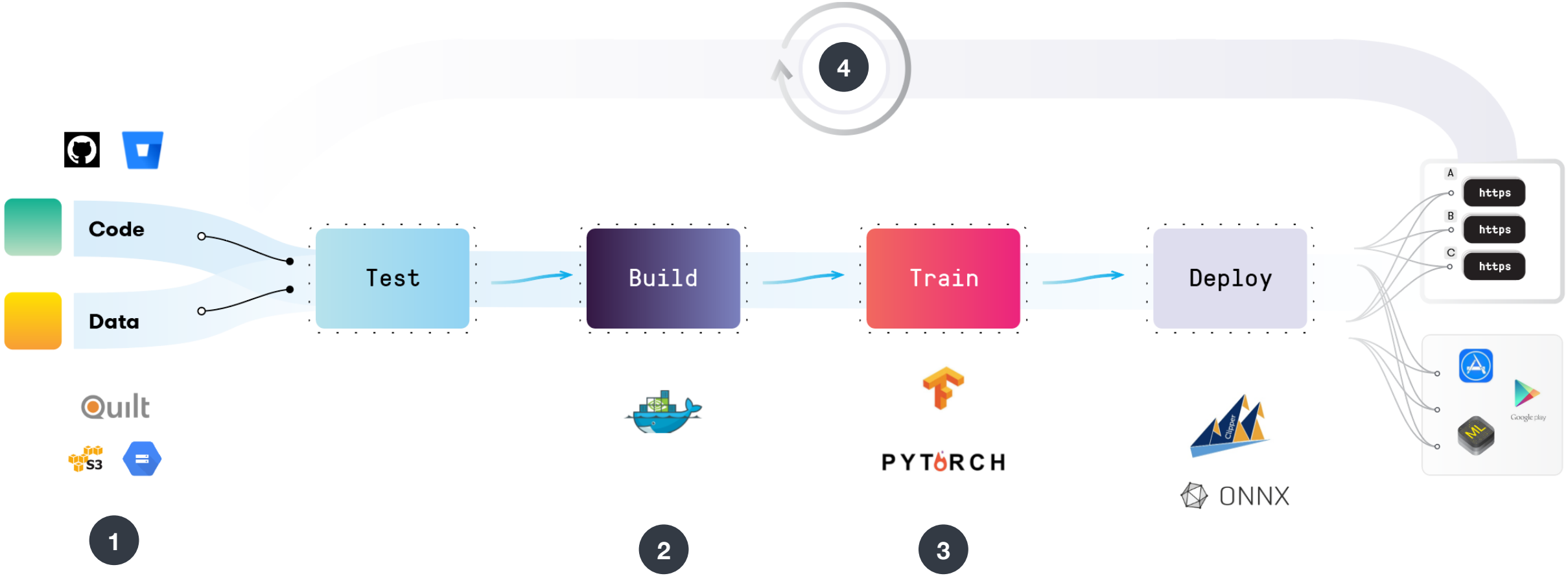


Paperspace

CI/CD Workflow for Traditional Apps



CI/CD Workflow for ML/DL



Paperspace

Semantics

Input

1. Data governance
2. Data versioning

Triggers

1. Data drift
2. Model drift

Output

1. Artifact management
2. Multi-armed bandit
3. Edge vs datacenter

Concourse CI - Docs

https://concourse-ci.org

Concourse

tutorial reference discuss download

Docs
1 About
2 Setup & Operations
3 Concepts
4 Tutorials
5 Reference
6 Contribute
7 Community
8 Trademarks

Concourse is an open-source continuous thing-doer.

Built on the simple mechanics of resources, tasks, and jobs, Concourse presents a general approach to automation that makes it great for CI/CD.

Quick Start

Configuration As Code

```
resources:
- name: booklit
  type: git
  source: {uri: "https://github.com/vito/booklit"}

jobs:
- name: unit
  plan:
  - get: booklit
    trigger: true
  - task: test
    file: booklit/ci/test.yml
```

You can think of a pipeline as a distributed, higher-level, continuously-running `Makefile`.

Each entry under `resources` is a dependency, and each entry under `jobs` describes a `plan` to run when the job is triggered (either manually or by a `get` step).

Jobs can depend on resources that have `passed` through prior jobs. The resulting sequence of jobs and resources is a dependency graph that continuously pushes your project forward, from source code to production.

help improve this site

powered by Booklit

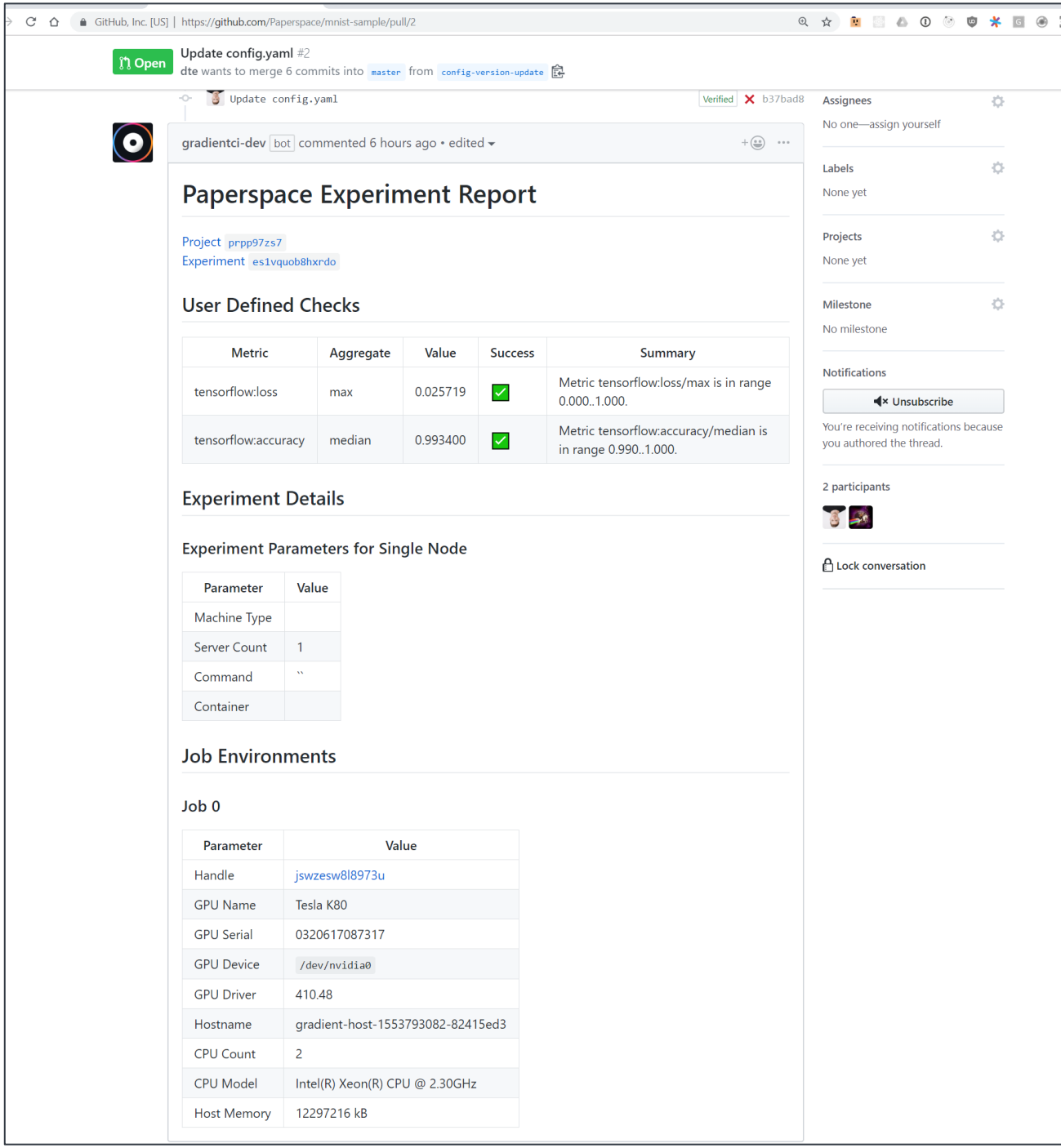
Observations

1. Tooling will change to accommodate more complicated semantics.

2. Binary (pass/fail) will give way to richer abstractions.

3. Tools will go wide and deep to close the ML-Ops feedback loop.

4. From a hardware perspective, the edge and the datacenter will merge.



Paperspace

Emerging Trends

- Jupyter
  - Standard for learning. Seeing traction for training and some deployments.
- Containers and Kubernetes
  - Silo concerns of data scientists and dev ops
  - K8s for container orchestration
- Model drift monitoring
  - Monitoring several models and their performance over time.
- Rollback models and model repos
  - Input new data to existing models
  - Pre-trained models for emerging use cases specific for your business.

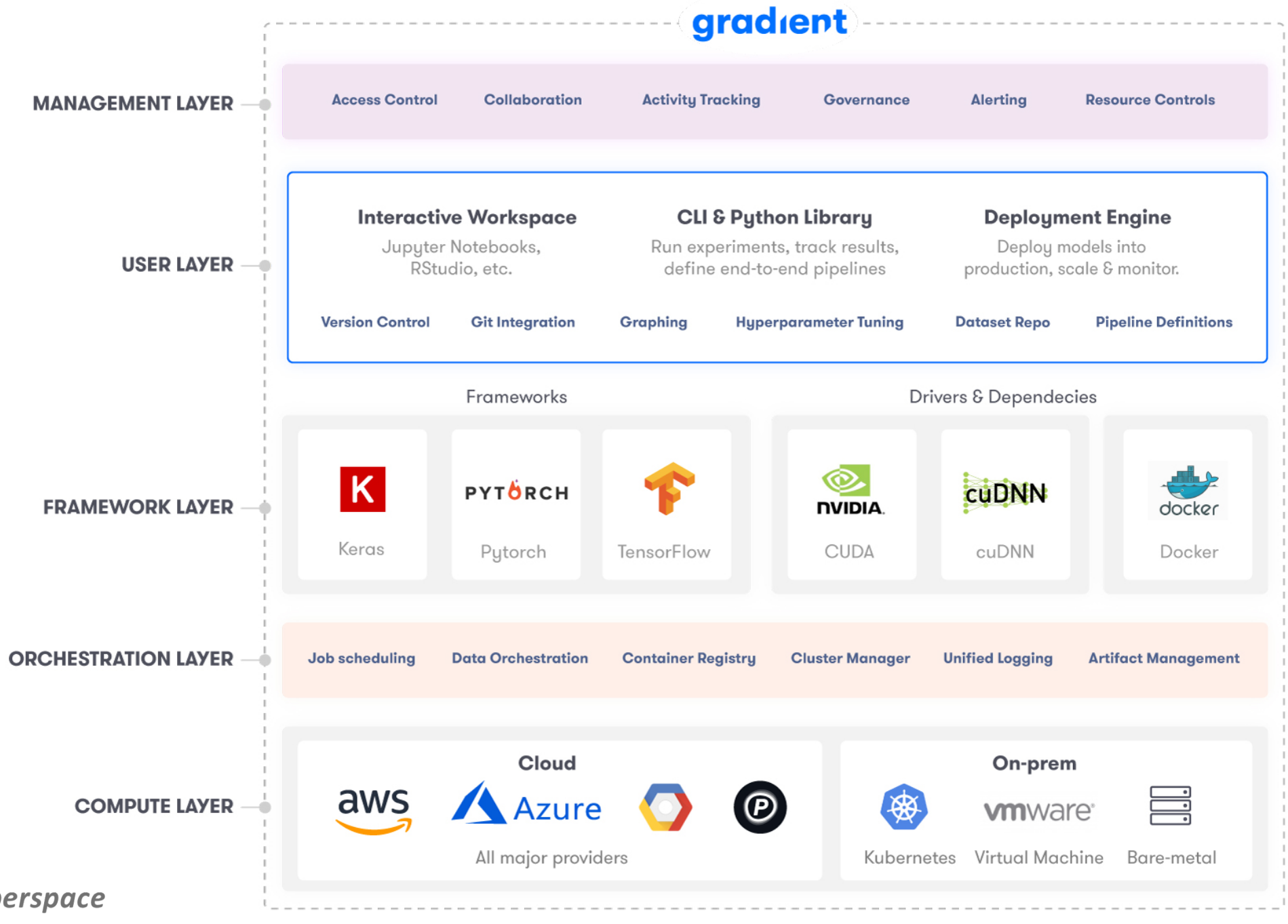
In the Field: Uber's Michelangelo Platform

End-to-End Workflow
Build, train, refine, and deploy models

Model Developer Velocity
Innovation comes from lots of iteration.

ML-as-Engineering
Apply CI/CD methods from software to ML

Modular & Tiered
Assembled for specialized use cases.

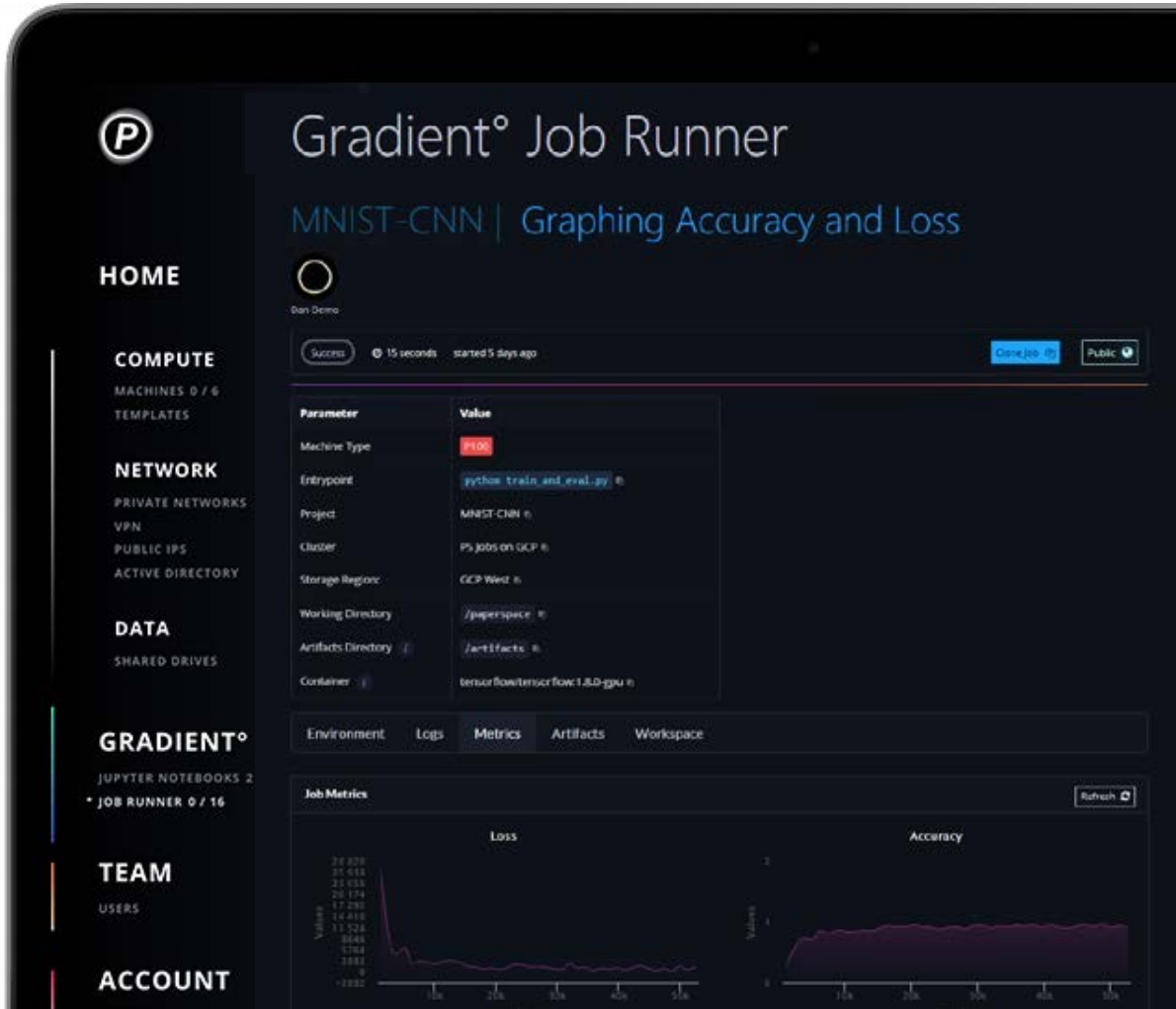

gradient
MANAGEMENT LAYER
Access Control
Collaboration
Activity Tracking
Governance
Alerting
Resource Controls
USER LAYER
Interactive Workspace
Jupyter Notebooks, RStudio, etc.
CLI & Python Library
Run experiments, track results, define end-to-end pipelines
Deployment Engine
Deploy models into production, scale & monitor.
Version Control
Git Integration
Graphing
Hyperparameter Tuning
Dataset Repo
Pipeline Definitions
FRAMEWORK LAYER
Frameworks
Keras
Pytorch
TensorFlow
Drivers & Dependecies
CUDA
cuDNN
Docker
ORCHESTRATION LAYER
Job scheduling
Data Orchestration
Container Registry
Cluster Manager
Unified Logging
Artifact Management
COMPUTE LAYER
Cloud
All major providers
On-prem
Kubernetes
Virtual Machine
Bare-metal
Paperspace

Build, train, and deploy your ML/DL models at scale.

gradient°
by Paperspace

Learn more at www.gradient.cloud

Email me at raj@paperspace.com



Thanks for listening!

Paperspace