

AI on Kubernetes Without the Chaos

Building Reproducible ML Environments with Argo and Kubeflow

Nourhan Mohamed

About Me



Golden KubeStronaut



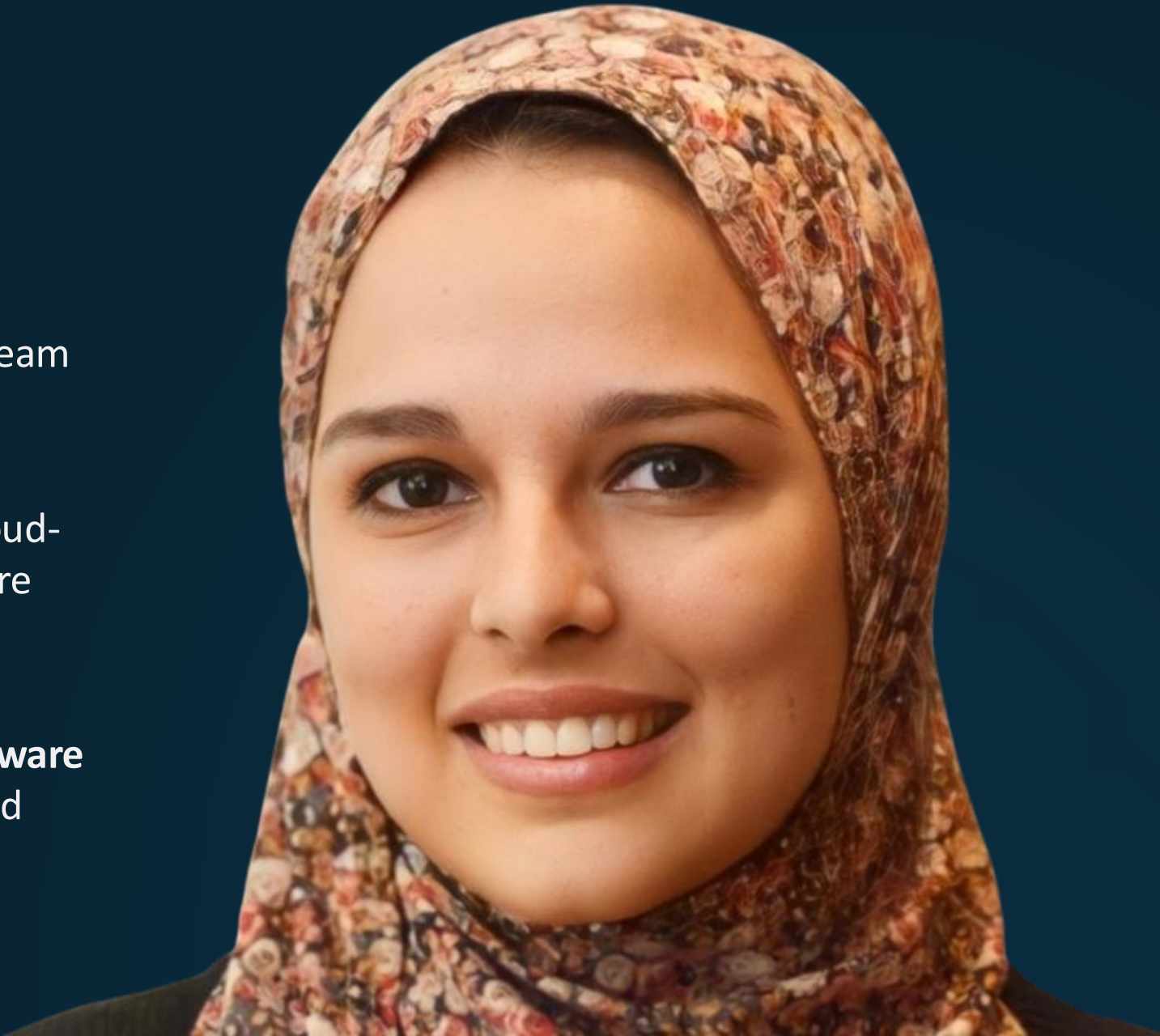
DevOps Lead and leads the Labs team
at **KodeKloud**



Deep expertise in Kubernetes, cloud-native platforms, and infrastructure automation.



With prior leadership roles at **VMware** and **Dell Technologies** in Tanzu and infrastructure support



~300

ML Papers compromised by
Data Leakage affecting
reproducibility

Kapoor & Narayanan, Patterns, 2023

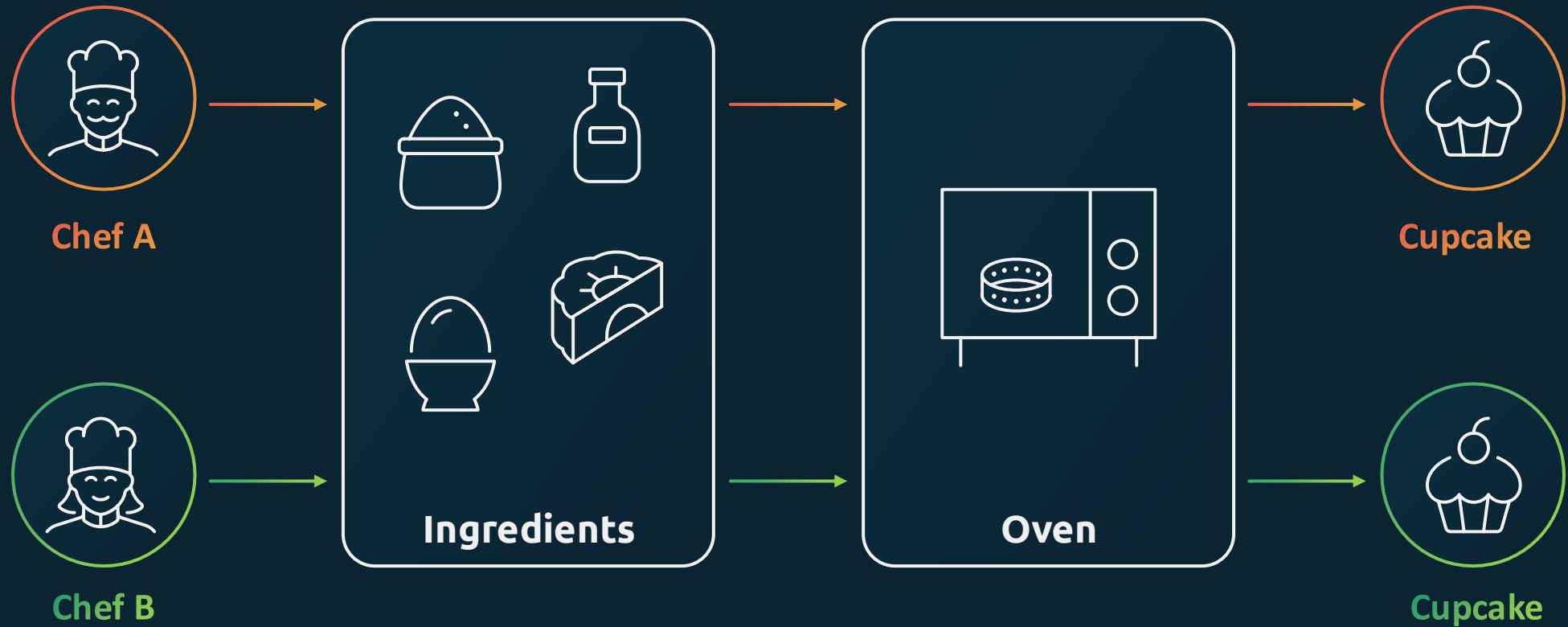
The Reproducibility Problem



The Reproducibility Problem

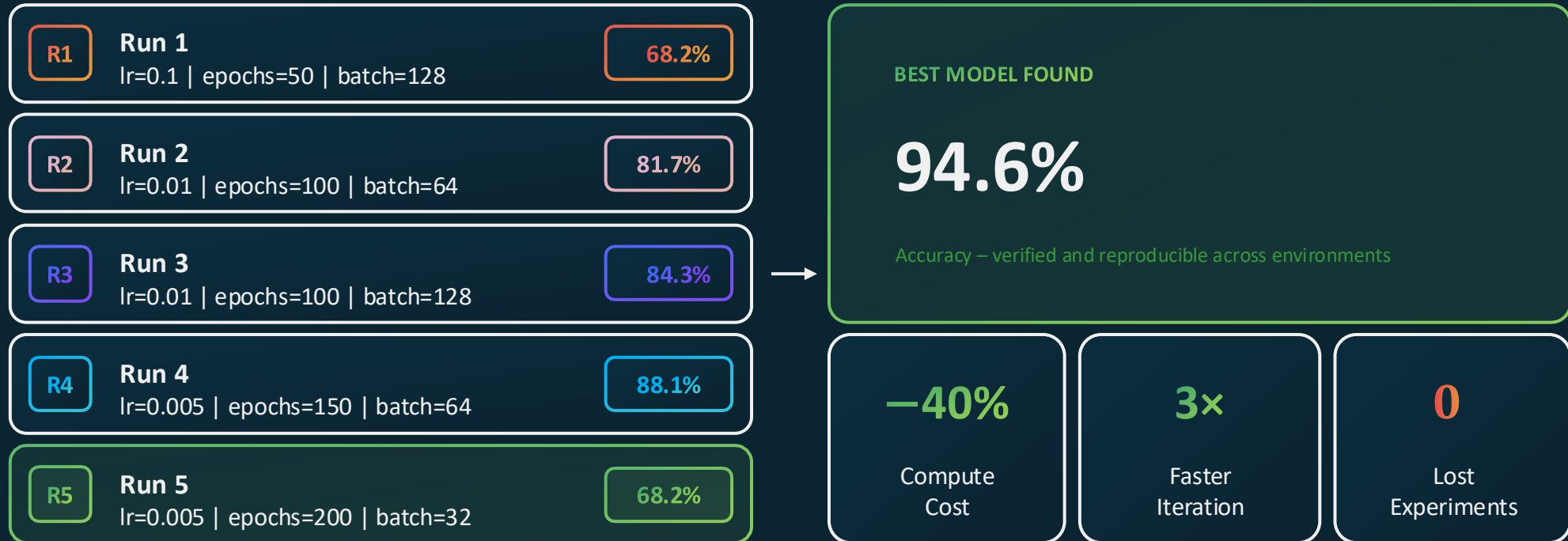


The Reproducibility Problem



Why Chase Reproducibility?

Systematic exploration finds the best model - faster and cheaper

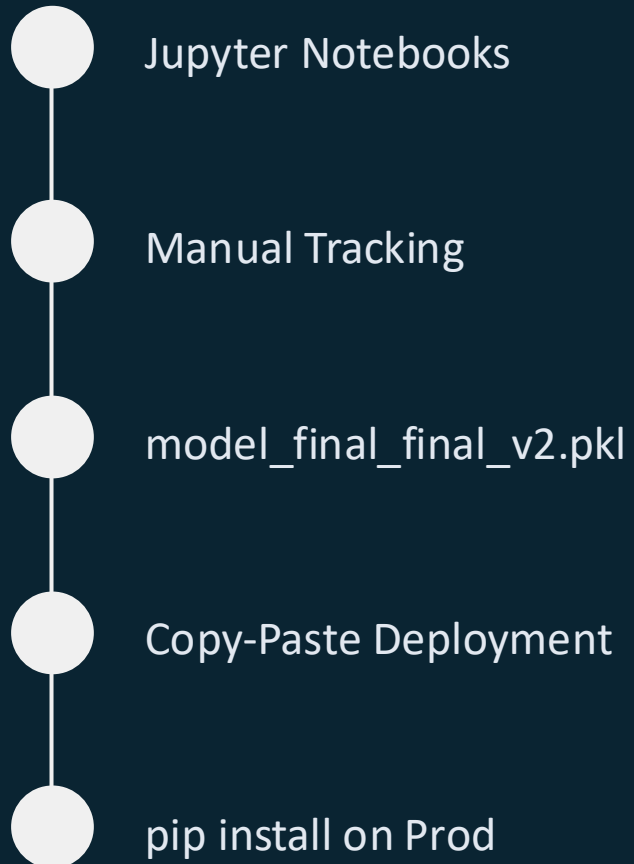


Without reproducibility, you can't compare runs.

Without comparison, you can't converge on the best model.

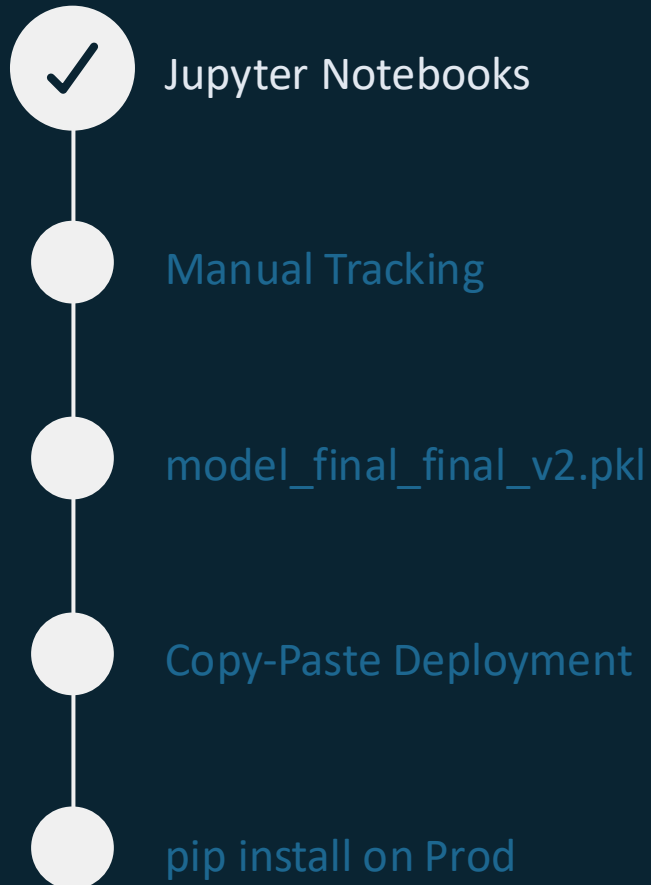
The Traditional ML Workflow

The workflow that leads to chaos



The Traditional ML Workflow

The workflow that leads to chaos



Experiments run interactively.

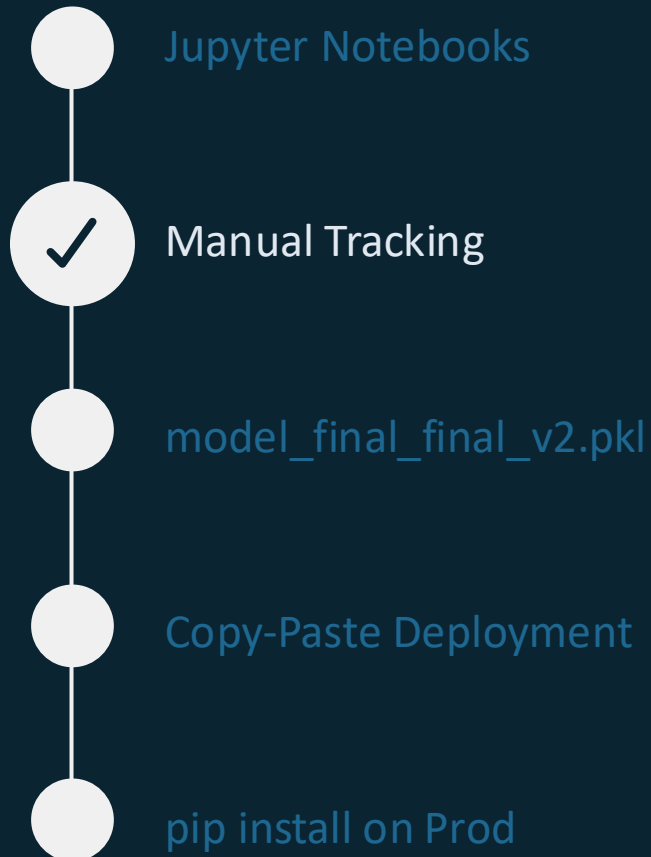
State lives in memory.

Results lost on restart.

NO REPRODUCIBILITY

The Traditional ML Workflow

The workflow that leads to chaos



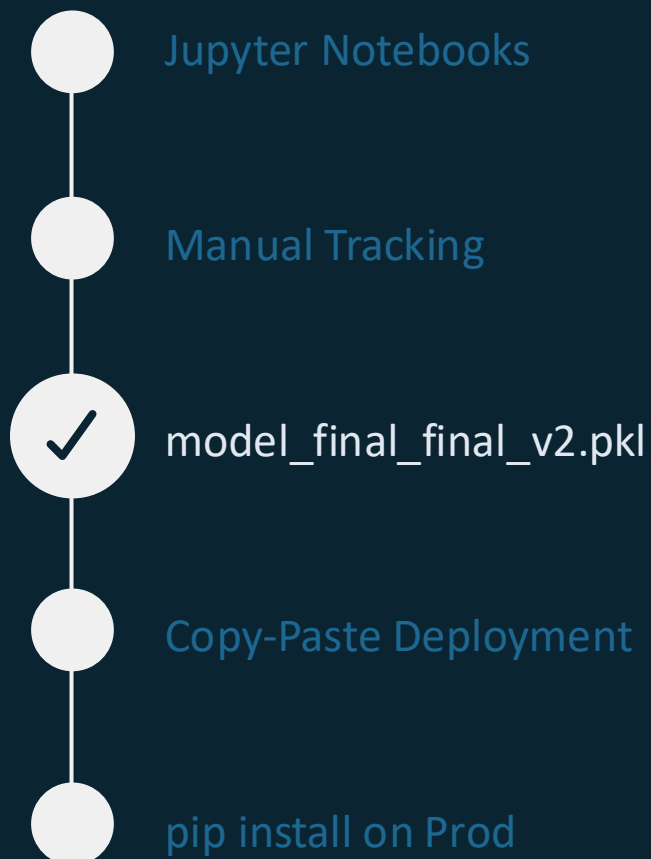
Hyperparameters in spreadsheets.

"Which learning rate was best?"

NO EXPERIMENT HISTORY

The Traditional ML Workflow

The workflow that leads to chaos



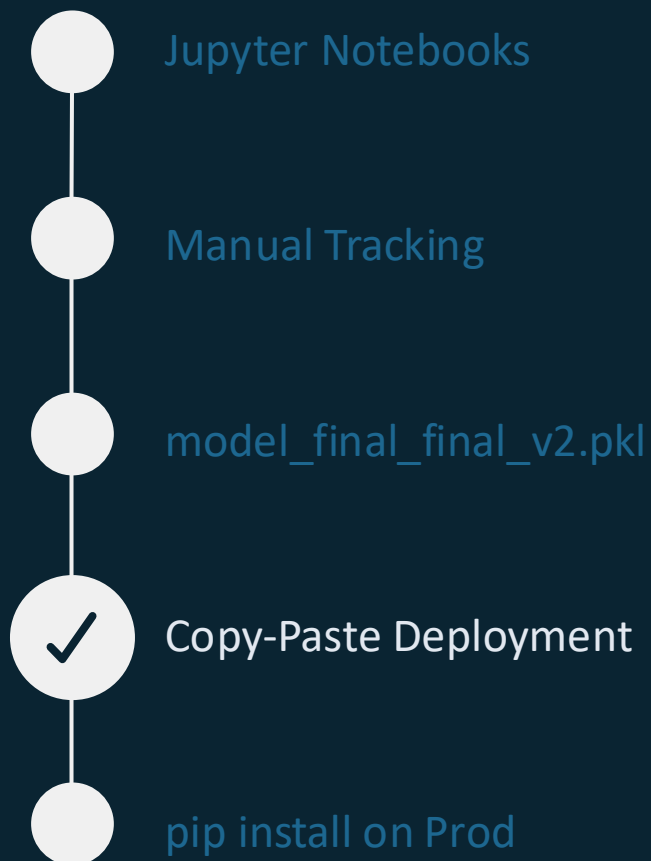
Models saved locally with arbitrary names.

No link to data or code.

NO LINEAGE

The Traditional ML Workflow

The workflow that leads to chaos



SCP model to server

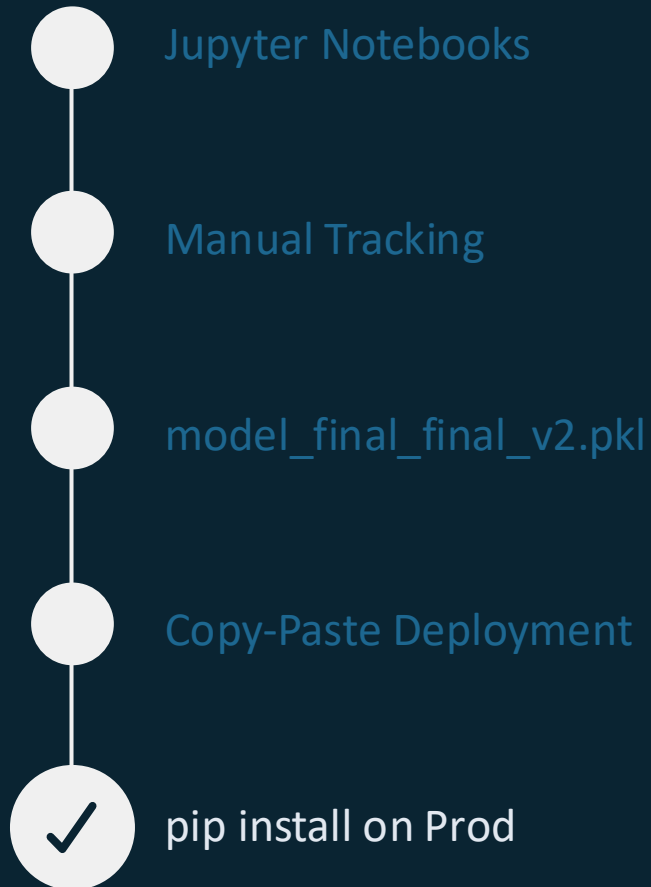
Restart Service

Hope it works

NO ROLLBACK

The Traditional ML Workflow

The workflow that leads to chaos



Dependencies installed ad-hoc.

Works today, breaks tomorrow.

ENVIRONMENT DRIFT

On Kubernetes, this becomes chaotic

01



Version drift across
packages, nodes and
environments

02



Broken pipelines
from undeclared
dependencies

03



GPU scheduling
conflicts causing
silent failures

04



**"It worked on my
laptop"** syndrome
at scale

Two Problems That Break Everything

01

Environment and Version Drift

Unpinned packages change between runs, altering model behavior silently. Different outcomes across different environments (development, staging, production)

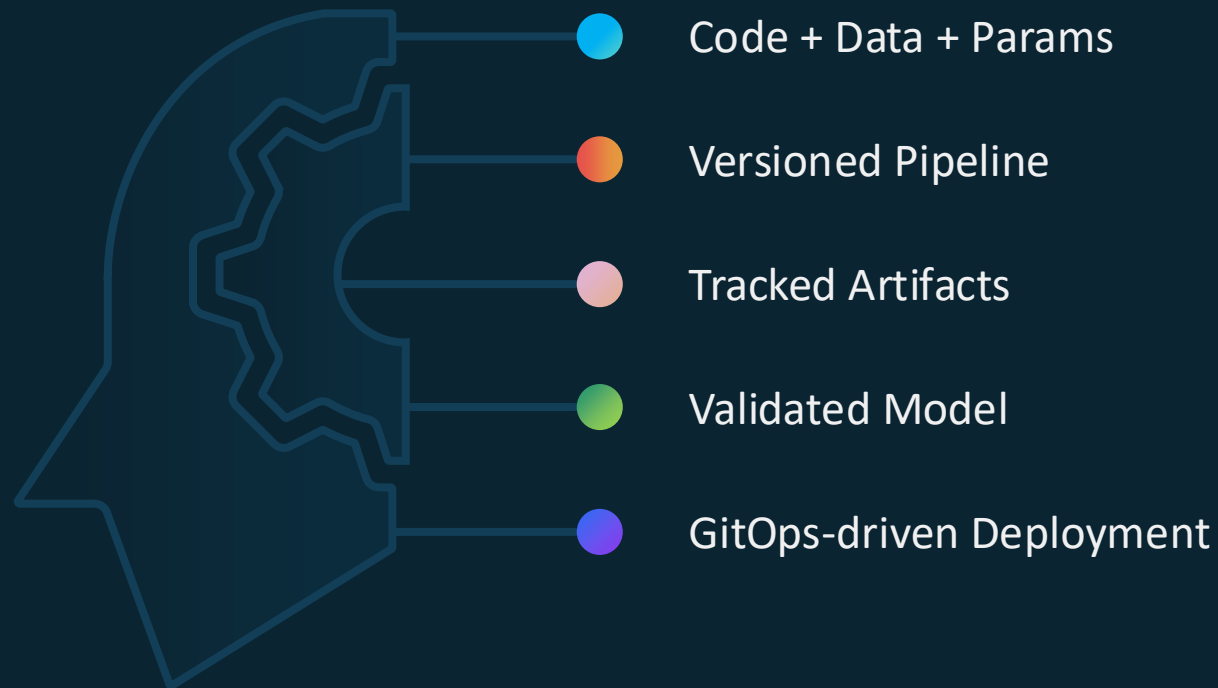
02

No Lineage

Can't trace which data, code, and params produced a certain model

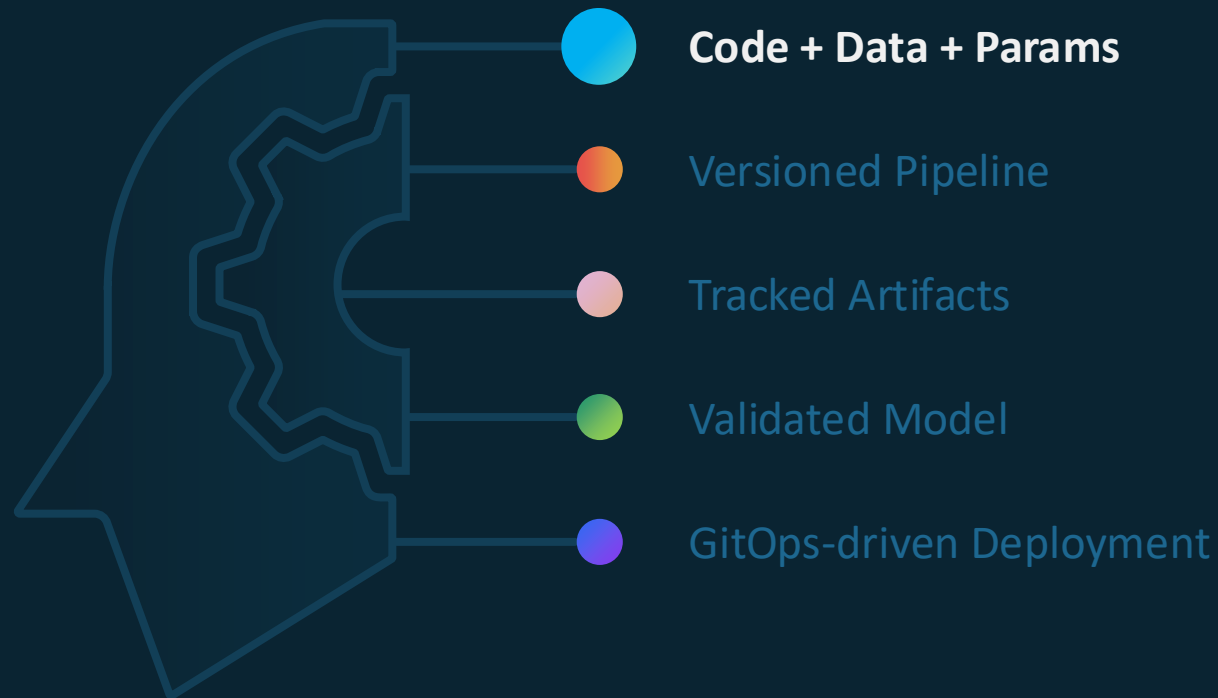
What a Reproducible ML System Actually Looks Like

A reproducible ML environment requires



What a Reproducible ML System Actually Looks Like

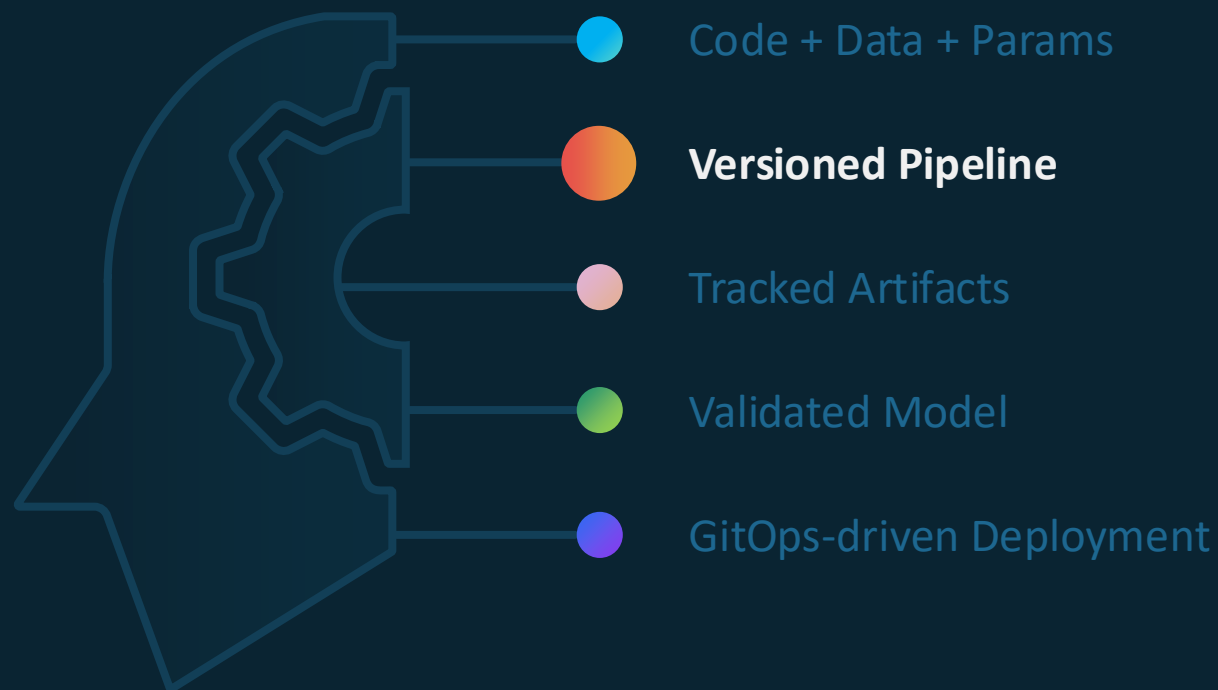
A reproducible ML environment requires



Versioned Inputs
hyperparameters, data, thresholds

What a Reproducible ML System Actually Looks Like

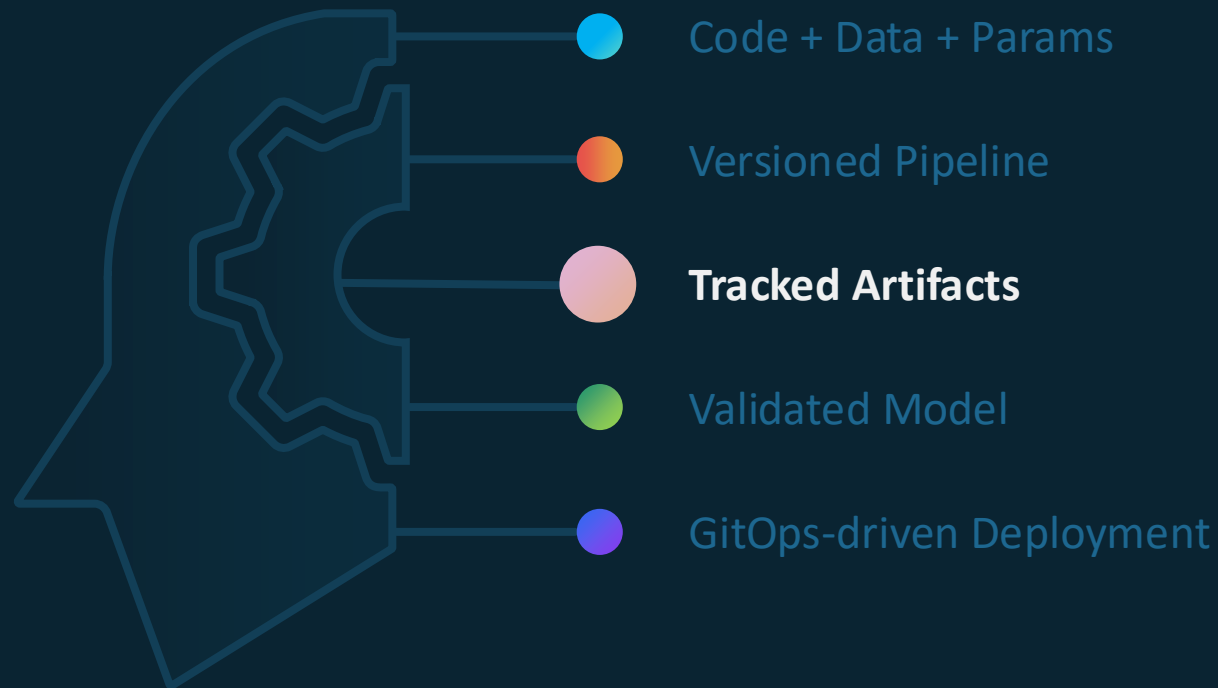
A reproducible ML environment requires



Orchestrated & Immutable
Containerized Workflow Steps

What a Reproducible ML System Actually Looks Like

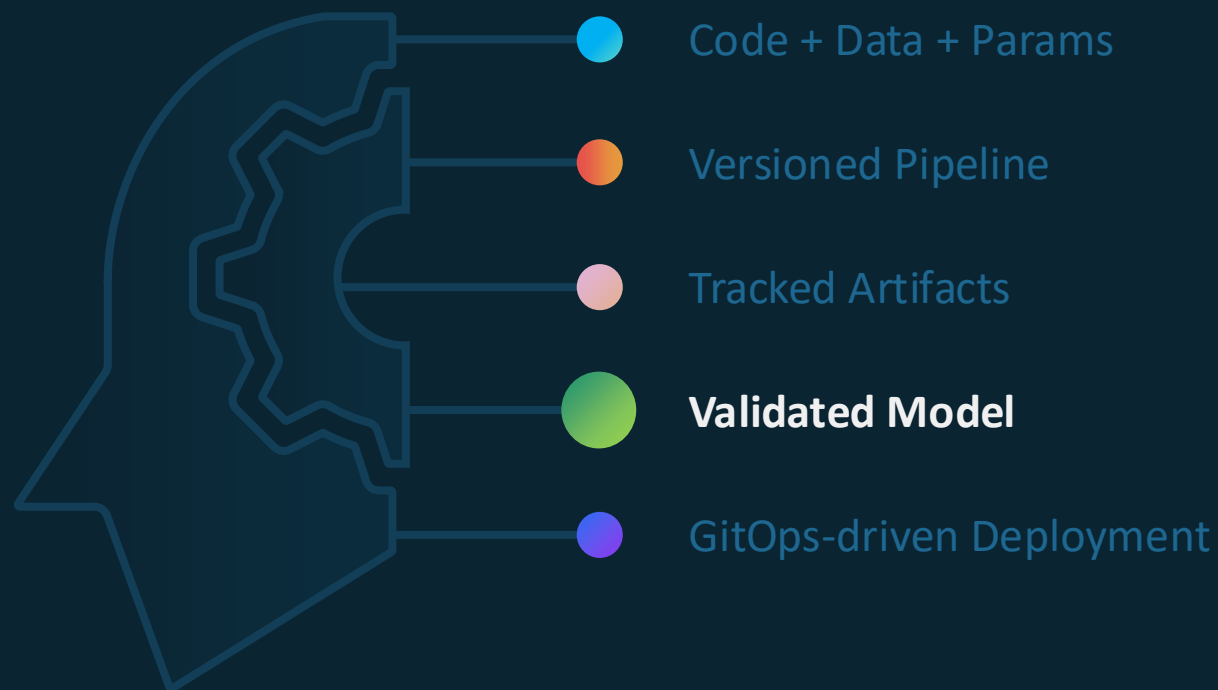
A reproducible ML environment requires



Metrics, Models & Lineage
Logged & Stored

What a Reproducible ML System Actually Looks Like

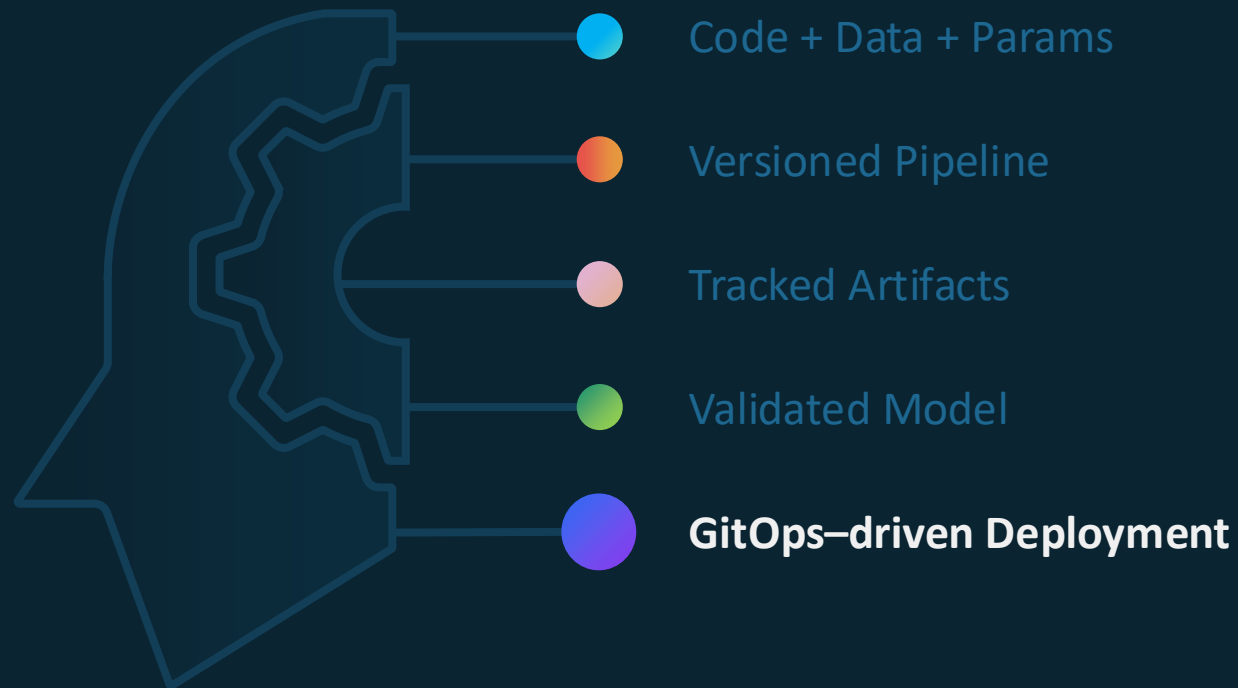
A reproducible ML environment requires



Automated Quality Gates
Tests & Approvals

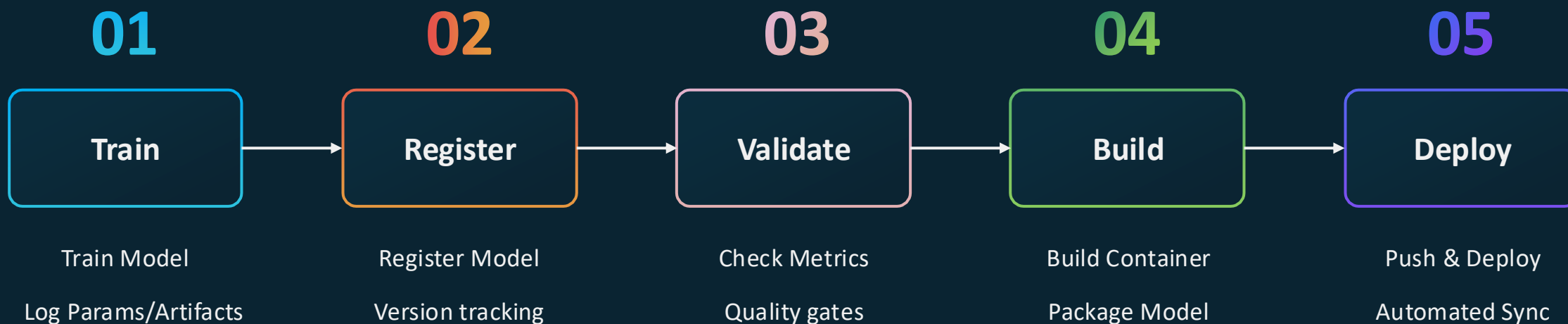
What a Reproducible ML System Actually Looks Like

A reproducible ML environment requires

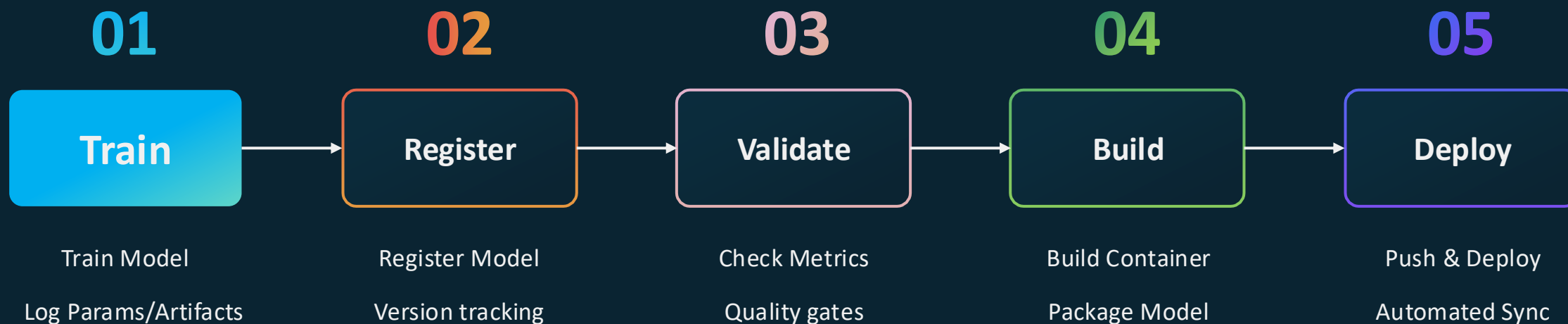


Continuous Reconciliation
Audit & Rollback Safe

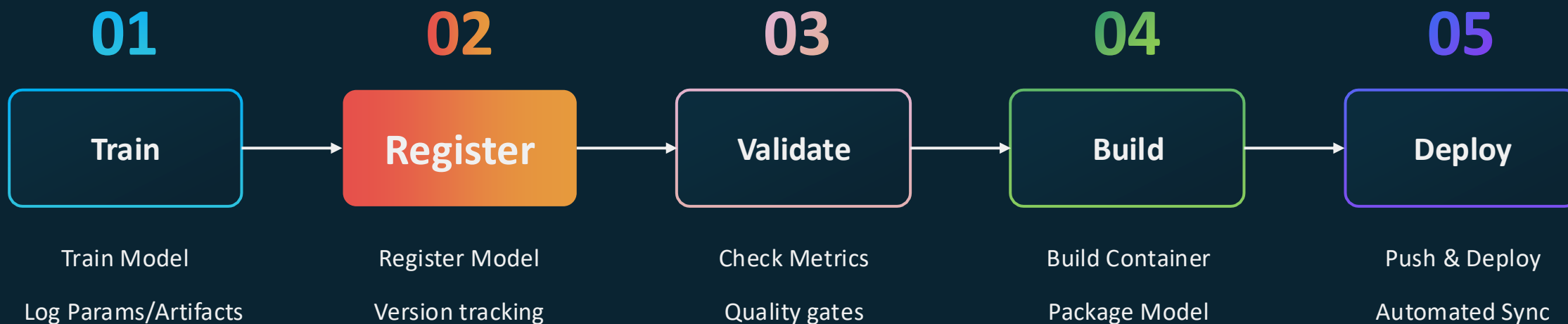
The Pipeline – 5 Steps to Production



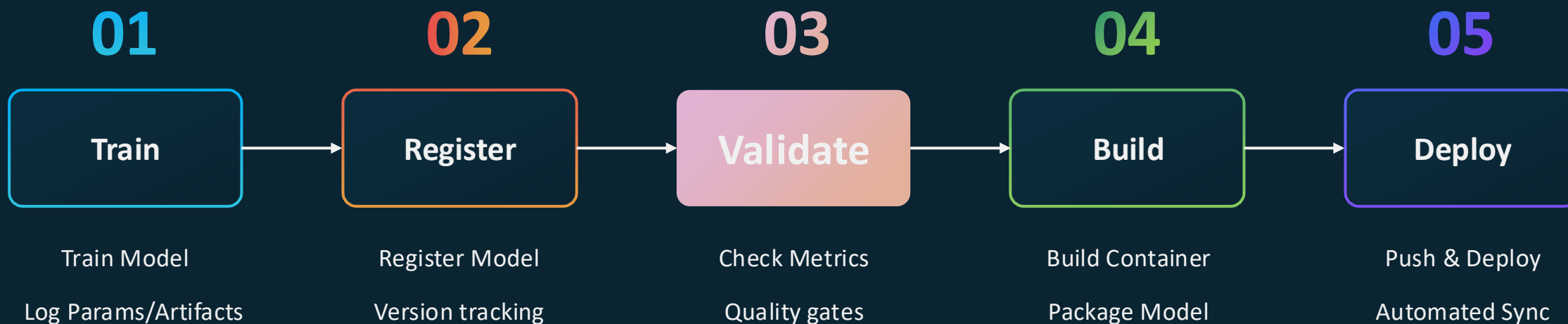
The Pipeline – 5 Steps to Production



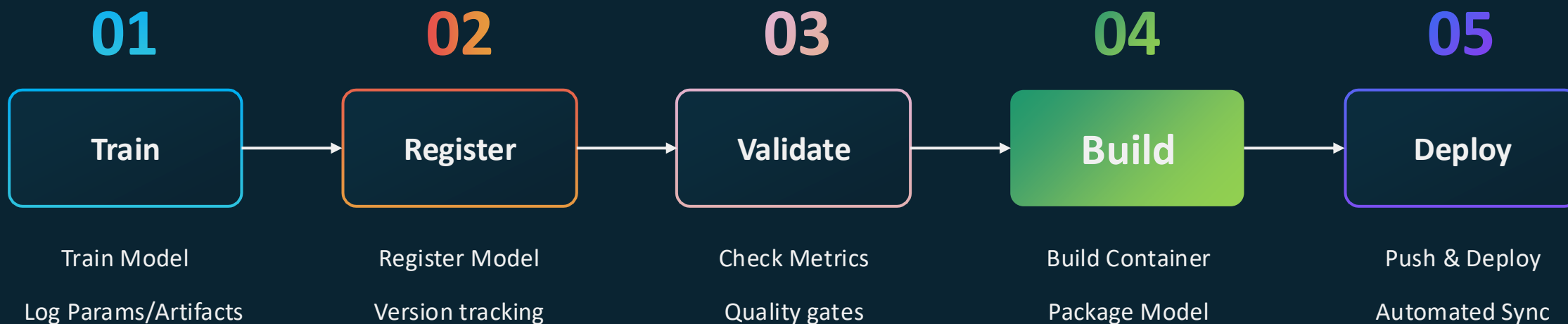
The Pipeline – 5 Steps to Production



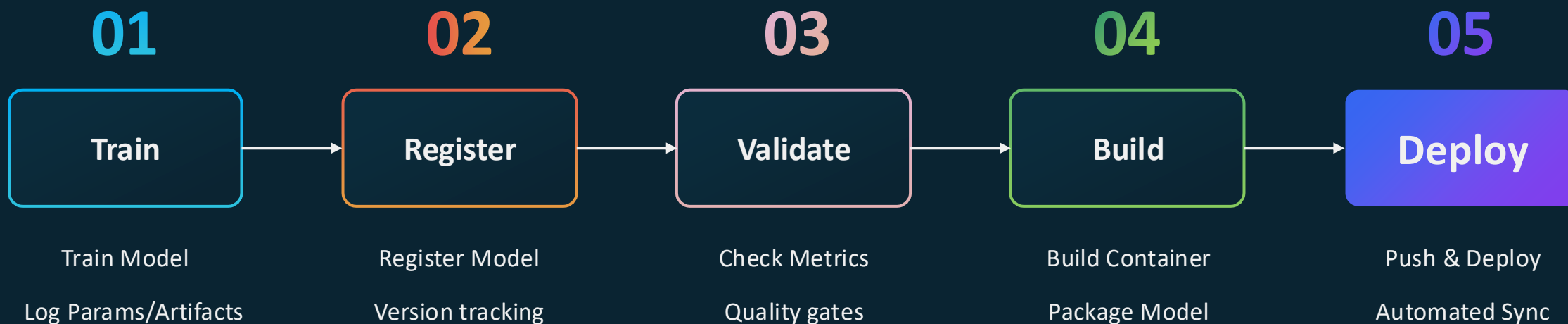
The Pipeline – 5 Steps to Production



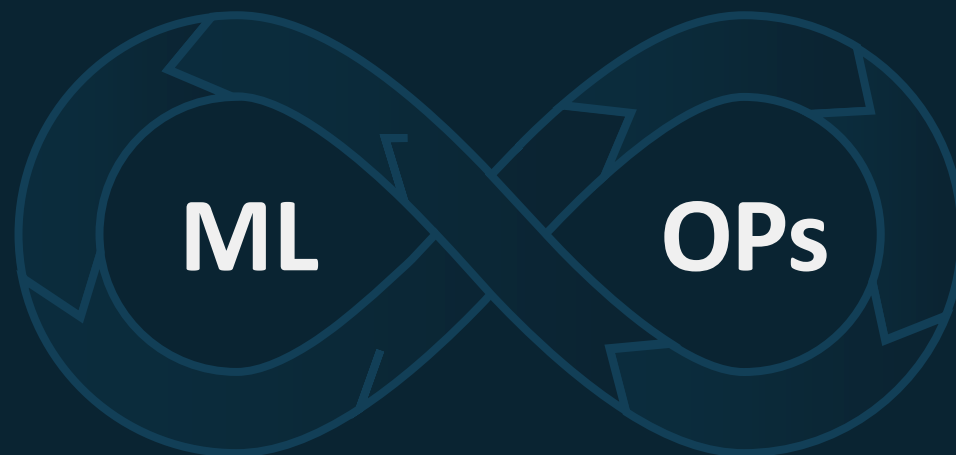
The Pipeline – 5 Steps to Production



The Pipeline – 5 Steps to Production



Our Building Blocks

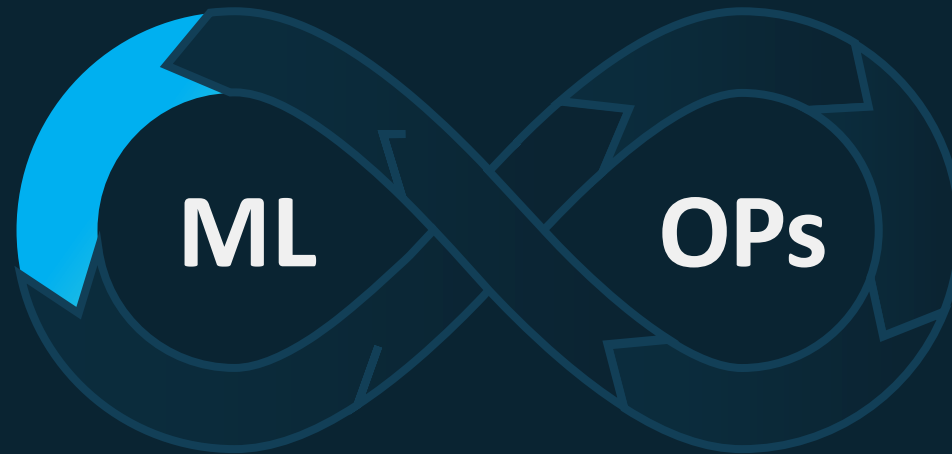


Our Building Blocks

ORCHESTRATION

Kubeflow Pipelines

ML workflow orchestration with Argo
Workflows + ML Metadata



Our Building Blocks

EXPERIMENT TRACKING

MLflow

Metrics, parameters, model registry



Our Building Blocks



ARTIFACT STORAGE

MinIO

S3-compatible object storage for artifacts

Our Building Blocks



IMAGE BUILD

Kaniko

In-cluster container builds

Our Building Blocks

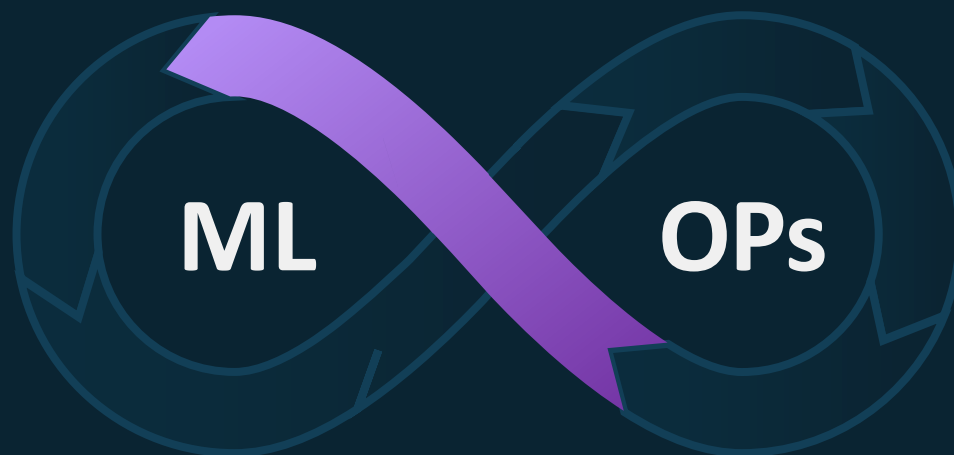


SOURCE CONTROL

[Gitea](#)/[Github](#)/[Gitlab](#)

Git repository

Our Building Blocks



GITOPS

ArgoCD

Continuous delivery with drift
detection

Our Building Blocks

ORCHESTRATION

Kubeflow Pipelines

ML workflow orchestration with Argo
Workflows + ML Metadata

EXPERIMENT TRACKING

MLflow

Metrics, parameters, model registry

ARTIFACT STORAGE

MinIO

S3-compatible object storage for artifacts

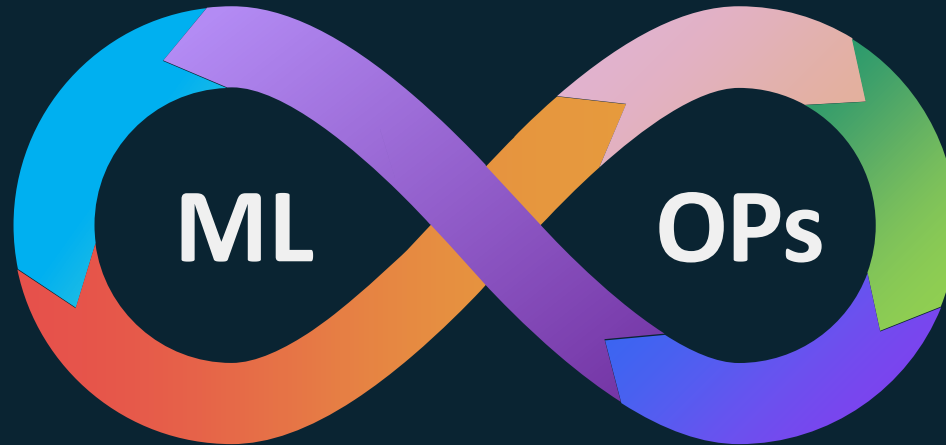


IMAGE BUILD

Kaniko

In-cluster container builds

SOURCE CONTROL

Gitea/Github/Gitlab

Git repository

GITOPS

ArgoCD

Continuous delivery with drift
detection

The Dataset

20,000

Records

39

Numeric Features Used

5%

Expected Anomaly Rate

0

Labels Needed (Unsupervised)

Biometrics

Age · Weight · Height
BMI · Body Fat % · Lean Mass

Heart Rate

Max BPM · Avg BPM
Resting BPM · % HRR · % Max HR

Exercise

Workout Frequency · Duration
Calories Burned · Sets · Reps

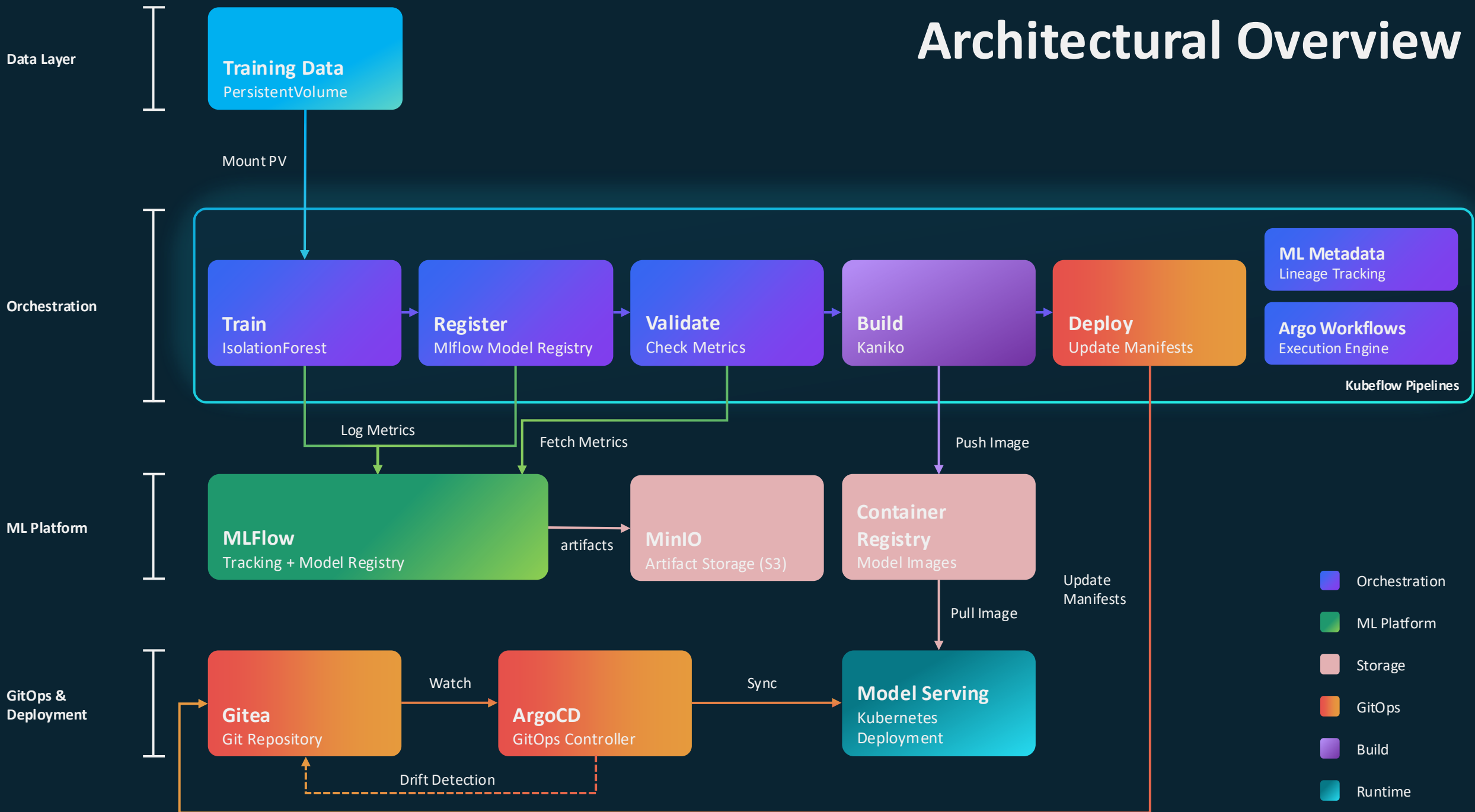
Nutrition

Calories · Carbs · Protein · Fat
Caloric Balance · Sodium · Sugar

We're detecting statistically unusual combinations across all 39 features.

Anomaly rate must be within **3% - 8%**, otherwise the pipeline fails.

Architectural Overview



Live Demo – Hands on Lab

Use the link below to access the workshop environment

<https://kode.wiki/ai-on-k8s>

Key Takeaways

Pin everything – images, packages, and base layers

Track lineage – MLFlow and ML Metadata connects data to model to deployment

GitOps deploys – declarative state, automatic drift correction

Validate before ship – quality gates prevent bad models from being deployed in production

Immutable artifacts – same inputs always produce the same traceable output

Questions?

Thank you for attending!

Scan here if you'd like to rate
the session or provide feedback



Workshop materials and code

<http://github.com/NourhanMohamed/reproducible-ml-workflows-demo>