DATABRICKS ASSET BUNDLES:

A UNIFYING TOOL FOR FOR DEPLOYMENT ON DATABRICKS

Pieter Noordhuis (Databricks) & Connor Brown (84.51)
June 13th, 2024

PRODUCT SAFE HARBOR STATEMENT

ABOUT

Pieter Noordhuis

Sr. Staff Software Engineer, Databricks

Connor Brown

Cloud Engineer, 84.51˚

MOTIVATION

MOTIVATION

Building reliable software or services typically involves:

- Code review
- Testing
- Automated deployment
- Environment separation (e.g. development/staging/production)

MAPPING TO DATABRICKS

MAPPING TO DATABRICKS

Code

"I swear this cell was working yesterday... Did someone change it?"

- Databricks Notebooks
- Libraries (e.g. Python Wheels)
- SQL files
- JARs
- ...

Resource configuration

"The refresh job failed this morning... Was it updated to use the most recent set of expected parameters?"

- Databricks Workflows
- Delta Live Table Pipelines
- Model Serving Endpoints
- Lakehouse Monitoring
- ...

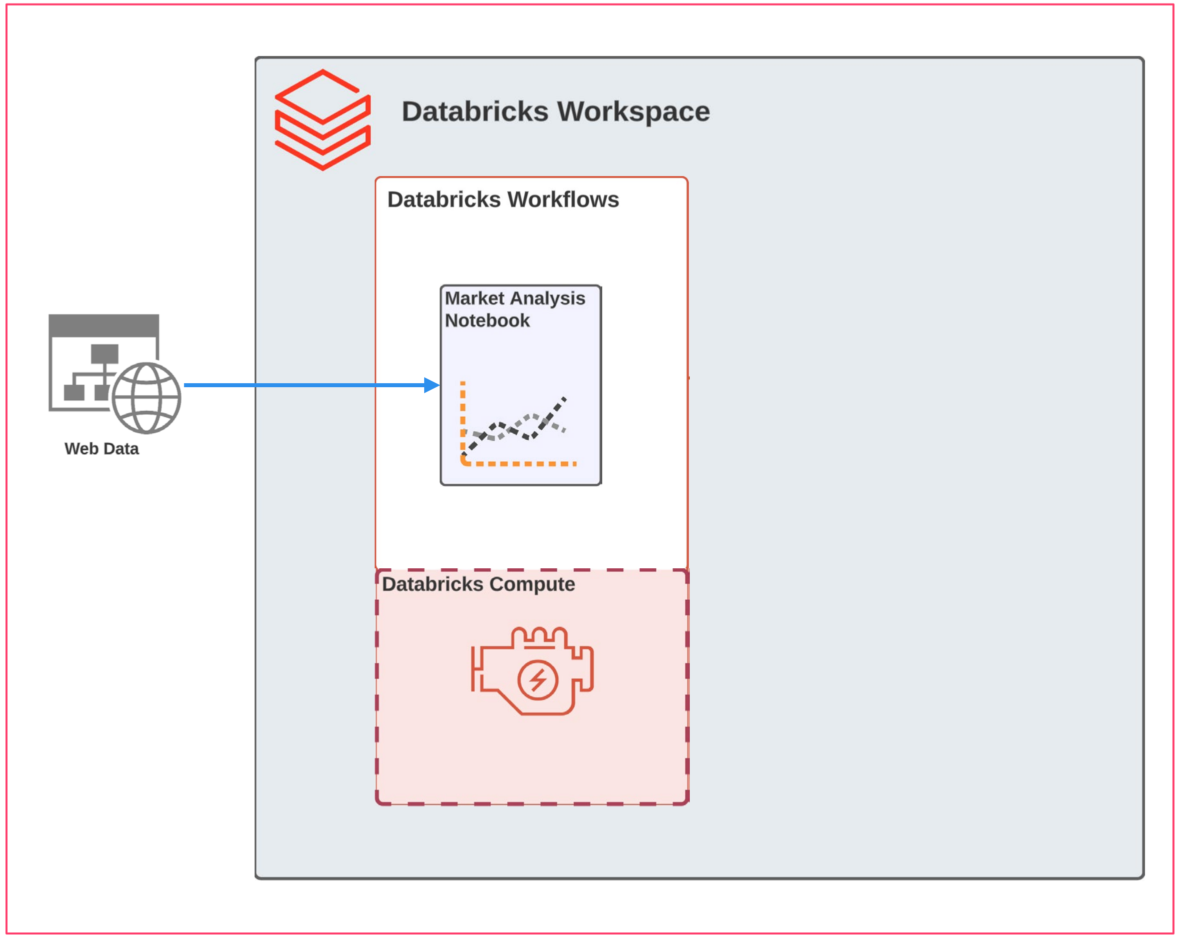
Environment separation

"We need a larger cluster in prod, who has access to update it?"

- Databricks Workspaces
- Unity Catalog (e.g. catalogs)
- Execution identity
    - User or service principal

LET'S BREAK THIS DOWN FURTHER

A simple example



Goal: Market trends report

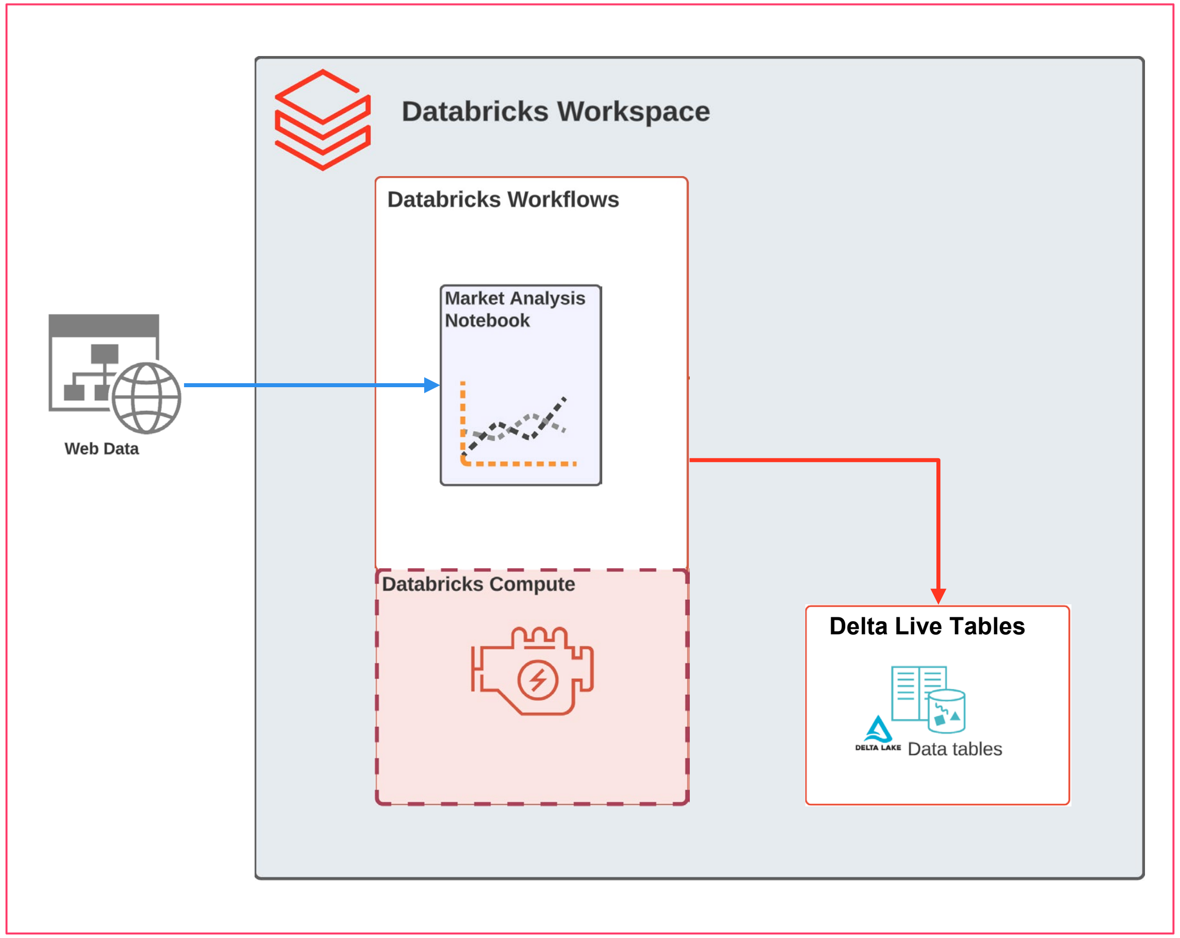
Code: Notebook

Resources: Databricks Workflow

Environment: Databricks Workspace

LET'S BREAK THIS DOWN FURTHER

What if our goal changes?



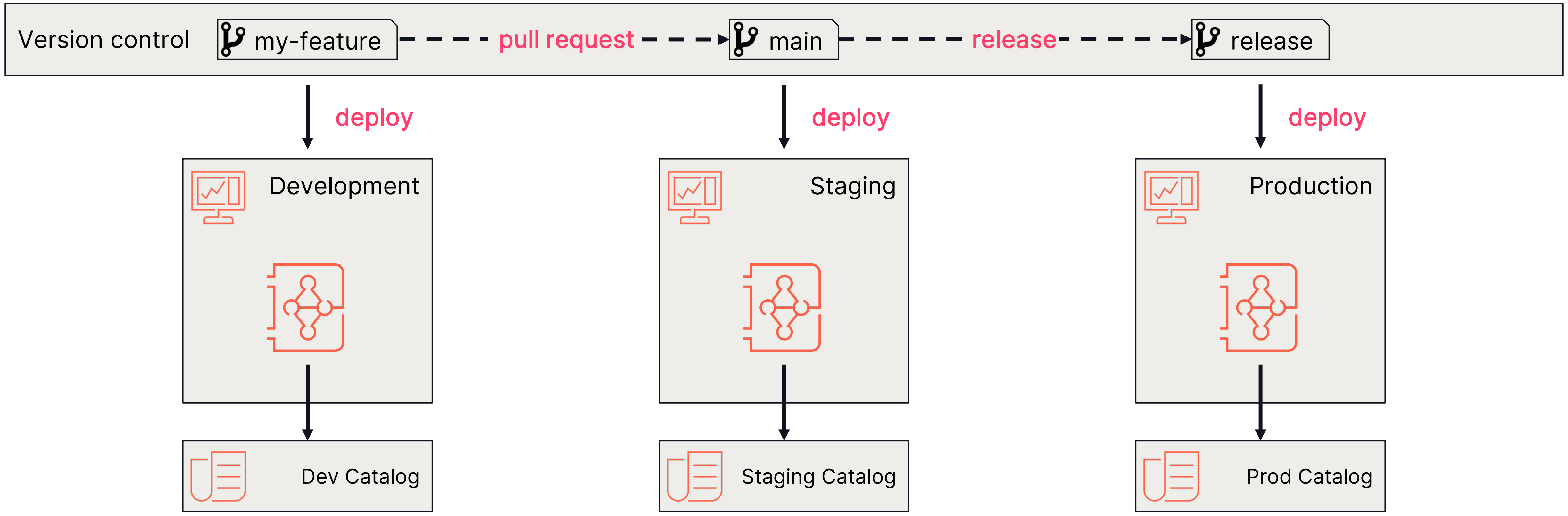
Goal: Market trends report
& persisted table

Code: Notebook
& pipeline source

Resources: Databricks Workflow
& Delta Live Tables Pipeline

Environment: Databricks Workspace
& Unity Catalog Schema

FROM DEVELOPMENT TO PRODUCTION



PROPERTIES WE'RE LOOKING FOR

- Single source of truth for code and resource configuration
- Define resources using an easy to understand format (e.g., YAML)
- Specialize resources based for target environment (dev/staging/prod)
- Ensure deployment isolation
- Easily automate deployment (CI/CD)

Enter Databricks Asset Bundles

DATABRICKS ASSET BUNDLES

What?	How?	Where?
Define Databricks resources, deployment targets, and per-target overrides, in YAML configuration. Co-locate this with your project code for a single unit of collaboration and deployment.	The `bundle` subcommand of the **Databricks CLI** lets you interact with this definition. You can validate the configuration, deploy the configuration, and interact with resources it defines.	Commands are executed client-side, so really anywhere. You can do development on your laptop and/or run automated production deployments from CI/CD systems such as GitHub Actions or Azure DevOps.

TAKING A LOOK AT databricks.yml (1)

- Bundle definition
- Use include to split configuration
- Workspace authentication
- Resource definitions (after API model)
- Local relative paths to code

```yaml
bundle:
 name: market_analysis

include:
 - resources/*.yml

workspace:
 host: https://my-workspace.cloud.databricks.com

resources:
 jobs:
   market_analysis_refresh:
     name: Run market analysis
     tasks:
     - task_key: run_notebook
       job_cluster_key: cluster
       notebook_task:
         notebook_path: ./src/market_analysis.py
```

TAKING A LOOK AT databricks.yml (2)

- Define deployment `targets`
- Optional deployment modes
- Per-target overrides

```yaml
targets:
 dev:
   mode: development
   workspace:
     host: https://dev-ws.cloud.databricks.com

 prod:
   mode: production
   workspace:
     host: https://prod-ws.cloud.databricks.com

   resources:
     jobs:
       market_analysis_refresh:
         job_clusters:
           - job_cluster_key: cluster
             new_cluster:
               num_workers: 32
```

THE bundle CLI COMMANDS

- **Validate** to check for issues with the bundle configuration
- **Deploy** to synchronize code to the workspace and materialize the defined resources
- **Run** to interact with the deployed resources

```
$ databricks bundle validate -t dev
Warning: unknown field: num_workerzzz
  at resources.jobs.my_job.tasks[0].new_cluster
  in resources/job.yml:17:13

Name: market_analysis
Target: dev
Workspace:
  Host: https://dev-ws.cloud.databricks.com
  User: pieter.noordhuis@databricks.com
  Path: ~/.bundle/market_analysis/dev

Found 1 warning
```

BUNDLE STATE

- Bundles use the Databricks workspace for file and state storage
- Defaults to `~/.bundle/${bundle.name}/${bundle.target}`
  - If it **does not yet exist**; full file sync, create resources
  - If it **exists**; incremental file sync, create/update/delete resources
- Default can be modified to incorporate additional dimensions
  - E.g. include branch name for 1 deployment per feature branch

GETTING STARTED WITH BUNDLES

From scratch

- Execute `databricks bundle init`
- Select a built-in template to get started
- Leverage IDE auto-completion on the configuration schema to find what you need

From existing code and resources

- Start with an empty bundle
- Use `databricks bundle generate` to import code and resource configuration from your workspace
- Use `databricks bundle bind` to attach the bundle resource to the pre-existing resource in the workspace

LIVE WALKTHROUGH

WE DIDN'T COVER

- Fine grained permissions and ownership
- How to author DAB templates
- How to use (complex) variables
- Setting up CI/CD
- Authoring DABs with Python
- Managing Lakeview dashboards
- ...

DABS CUSTOMER JOURNEY: 84.51°

About 84.51˚

“We are a **retail data science, insights and media company.** We help The Kroger Co., consumer packaged goods companies, agencies, publishers and affiliates **create more personalized and valuable experiences** for shoppers across the path to purchase.

Powered by cutting-edge science, we utilize first-party retail data from over 62 million U.S. households sourced by the Kroger Plus loyalty program to fuel a more customer-centric journey using 84.51° Insights, 84.51° Loyalty Marketing and our retail media advertising solution, Kroger Precision Marketing.”

DABs at 84.51˚

- 84.51˚ has rolled out DABs to ML, Data Engineering, and Adhoc Data Science use cases
- Before DABs was released, 84.51˚ had an internal tool called “SciCLOps” that was already covering CI/CD of Databricks assets, with over 100 users.
- This established SciCLOps user base became key to success in rolling out DABs at 84.51˚

SciCLOps

SciCLOps Quick Facts

- SciCLOps (Science Creation Lite Operations) is both the name of a Platform Engineering team and the product they maintain.
- SciCLOps is used to provision “landing zones” for Databricks users.
- SciCLOps is used for most data science, data engineering, and ML projects at 84.51˚.
- By using SciCLOps, users can go from “idea to deployed” in the Databricks development environment in under 10 minutes.
- SciCLOps has been used over 200 times since it’s release just over a year ago.

SciCLOps

How does SciCLOps work?

Service Portal

"I want a project"

- Users fill out a form providing details about the kind of data access they need, name their project, etc.
- The service portal uses the automation engine API to create the infrastructure

Automation (Terraform)

"I am getting a project"

- Creates infrastructure for the user
- Applies updates to the infrastructure (day two operations)
- Provides an overview of all environments

GHA* Reusable Workflow

"I want to deploy my project"

- Deploys starter code and future commits to Databricks (via DABs)
- Allows the SciCLOps team to update the workflow with minimal user disruption

** = GitHub Actions*

SciCLOps

What does a SciCLOps user get?

GitHub

- Repo
    - **databricks.yaml – NEW!**
    - Starter notebooks
- Actions Workflow
- Environment Variables
    - DATABRICKS_HOST
    - AZURE_CLIENT_ID
- Secrets
    - DATABRICKS_TOKEN

Databricks

- Service Principal
- Secret Scopes
- Cluster policies
- Directory
- Permissions
    - Instance Pool(s)
    - Directory
    - Unity Catalog

Azure

- Service Principal
- KeyVault
- Entra Groups

SciCLOps

Where does DABs fit in?

Reusable Workflow Before DABs

- Wheel build, job creation, repo deployment were all separate actions to maintain
- Model, DLT, and Experiment resources not yet supported
- Only one job deployed at a time – no support for multiple jobs

Reusable Workflow After DABs

- Wheel build, job creation, repo deployment all one action (bundle deploy)
- Model and DLT resources now supported
- Multiple jobs (or any other resource) can be deployed with one command.

SciCLOps

Bonus Features! “The little things”

databricks.yml

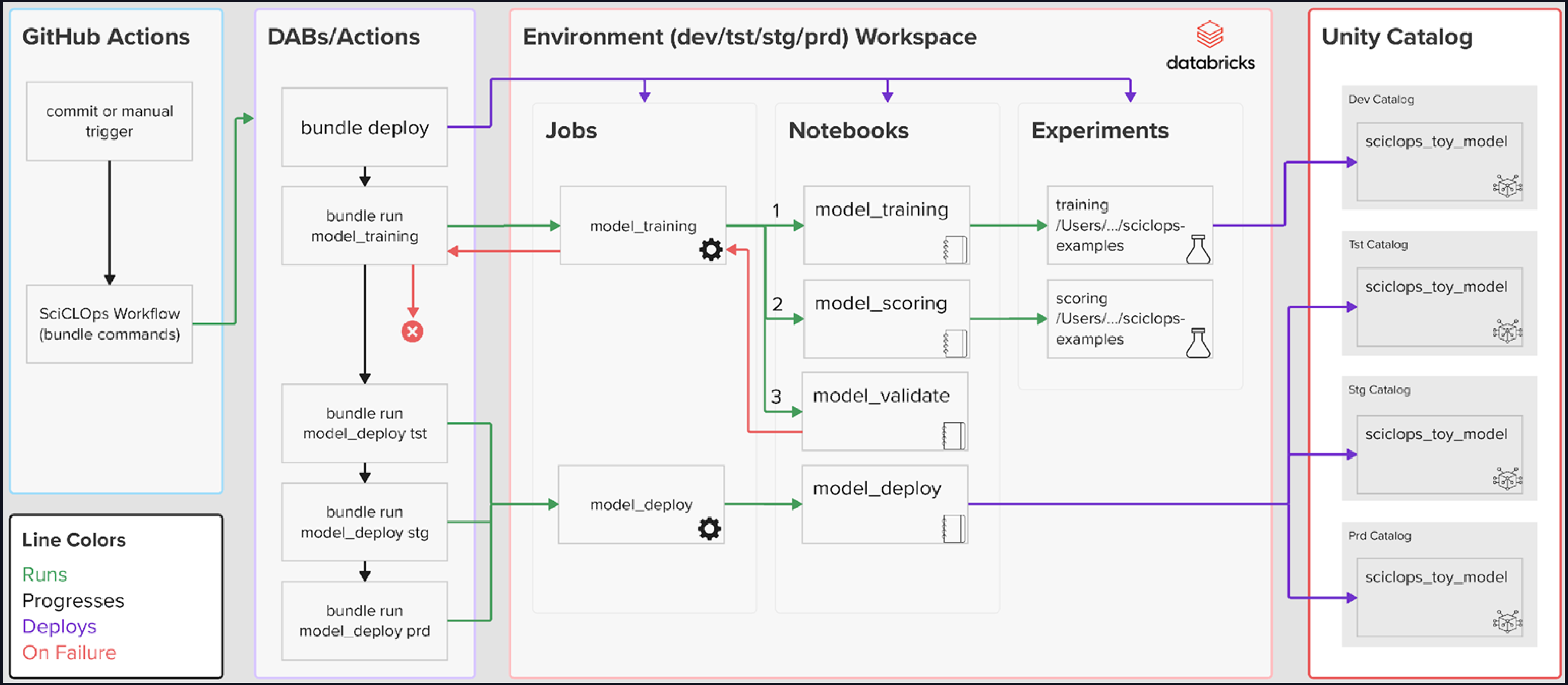
```
# 1. Lookups
bundle:
  name: ”dabs-rocks"

variables:
  INSTANCE_POOL_ID :
    description: ”Instance Pool ID"
    lookup:
      instance_pool: "E8as_v4_SPOT”

  CLUSTER_POLICY_ID:
    description: ”Cluster Policy ID"
    lookup:
      cluster_policy: ”SciCLOps - Single Node Cluster"
```

databricks.yml

```
# 2. Substitutions
bundle:
  name: ”dabs-rocks"
resources:
  experiments:
    training:
      name: ”training"
    permissions:…
  jobs:
    train_and_register_model:
      name: train_and_register_model
      parameters:
        - default: ${resources.experiments.training.name}
          name: training_experiment_name
        tasks: …
```

Example workflow using DABs: 84.51° MLOps

MLOps "Golden Workflow"



Key Takeaways

How was 84.51° able to roll out DABs quickly?

Platform Engineering Team

- The SciCLOps team acts like a Platform Engineering team for Databricks
- Responsible for maintaining a GitHub reusable workflow that makes for fast adoption of workflow changes/features
- Capability to make changes to already-provisioned SciCLOps environments rapidly through Terraform

Community of Practice (CoP)

- Established a biweekly CoP for SciCLOps/Databricks users
- Communicated release notes of SciCLOps during CoP (more effective than email/take questions live)
- Gave space for DABs users to demo their own findings/workflows

Example workflows (dogfood!)

- Though the SciCLOps team does not own any Databricks workflows, we created a “toy model”
- The toy model was maintained as if it was production – making us ”dogfood” SciCLOps/DABs
- The resulting workflow was coined the “golden workflow”, serving as an example for other teams using DABs

WRAPPING UP

- Installing the CLI
  - https://docs.databricks.com/en/dev-tools/cli/install.html
- Getting started with Databricks Asset Bundles
  - https://docs.databricks.com/en/dev-tools/bundles/index.html

WRAPPING UP

Other sessions at DAIS

- **SciCLOps** (Tuesday, Jun 11, 2:50 PM - 3:30 PM)
    - Databricks Quick Start for Machine Learning, Powered by DABs
- **Path to Production** (Thursday, Jun 13, 12:30 PM - 1:10 PM)
    - Databricks Project CICD for Seamless Inner to Outer Dev Loops
- **Code to Production** (Tuesday, Jun 11, 10:10 AM - 10:50 AM)
    - Leveraging Your Favorite IDEs with Databricks

DATA+AI SUMMIT