DATA+AI SUMMIT BY databricks

Efficient MLOps: Developing and Deploying ML Models with Databricks

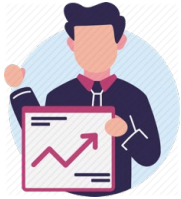
Lavinia Guadagnolo
Alessandro Mazzullo

©2024 Databricks Inc. — All rights reserved

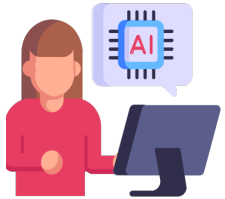
Who is this talk for?

Data scientists

Data engineers



Data analysts



ML Engineers

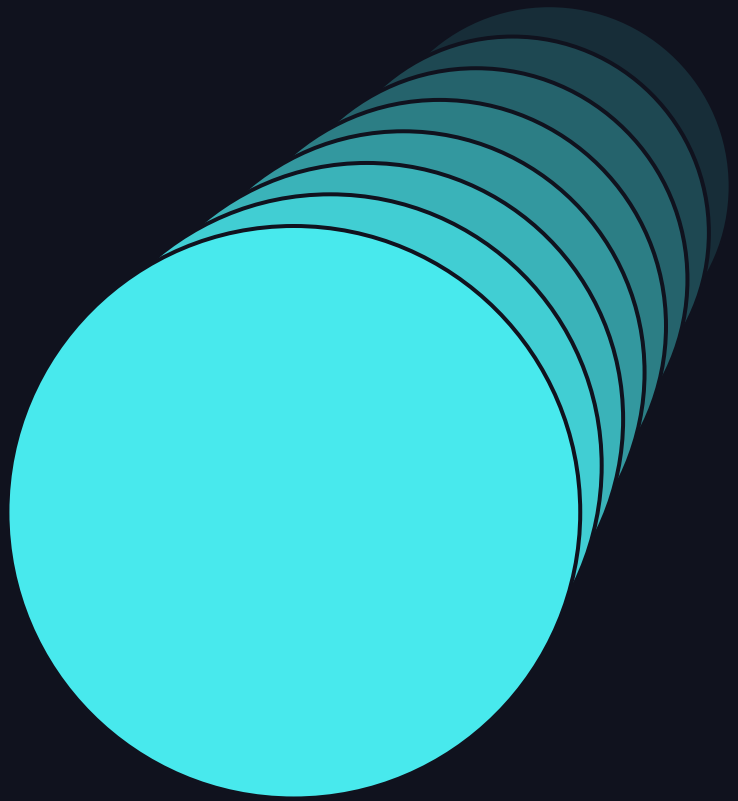
Organizations

©2024 Databricks Inc. — All rights reserved

DATA+AI SUMMIT BY databricks

Agenda:

- About us
- Challenges and motivations
- Main tools
  - Unity Catalog
  - Databricks Asset Bundles
- MLOps for *MLHopes*
  - CI/CD
- Demo



©2024 Databricks Inc. — All rights reserved

About us

©2024 Databricks Inc. — All rights reserved

About us

MLHopes

Alessandro Mazzullo

Product Owner, Data Scientist
Plenitude

Lavinia Guadagnolo

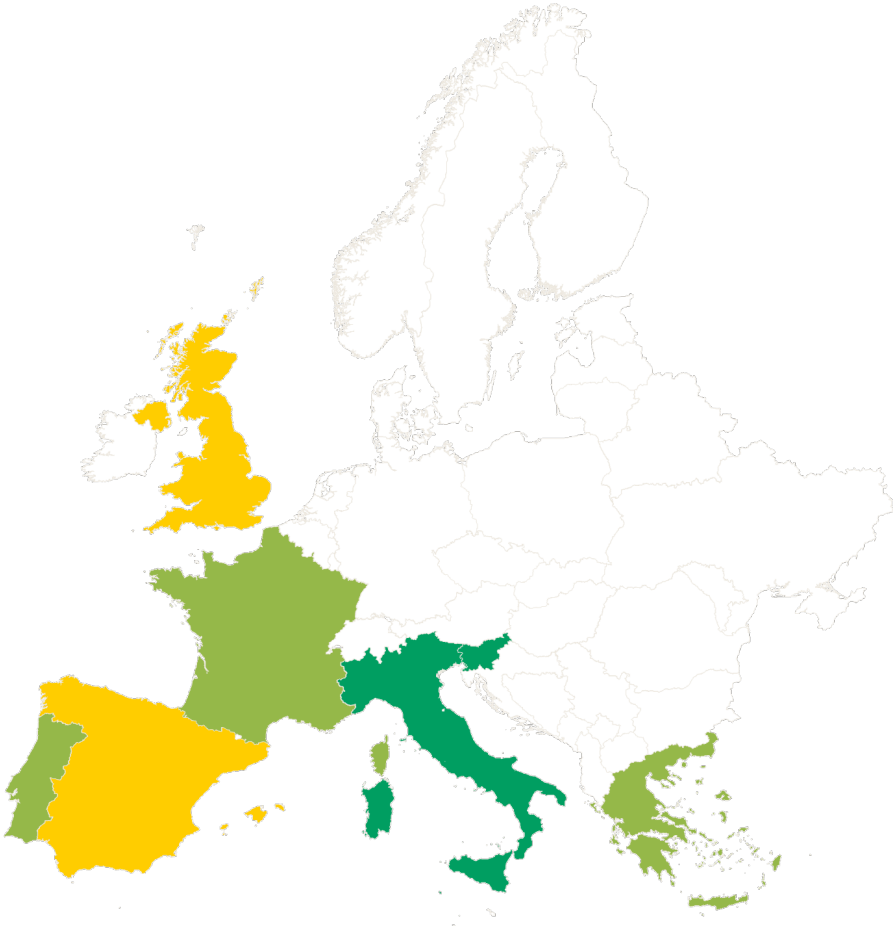
Product Owner, Data Scientist
Plenitude

©2024 Databricks Inc. — All rights reserved

About us

plenitude

- We are an international **Benefit Company** operating in the **energy sector** since 2017
- We are active in the **retail** market offering energy and related services to more than **10 mln customers**
- We operate in energy production from **renewable sources** and in the world of **e-mobility**
- Our approach is focused on **sustainability** and **innovation**

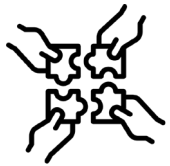


©2024 Databricks Inc. — All rights reserved

Challenges

©2024 Databricks Inc. — All rights reserved

Motivation & Challenges



Collaboration

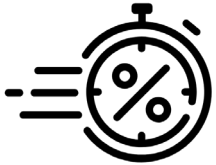
- Team growth
- Different **skills** and background

Agility

- **Manual** steps for industrialisation
- Difficulty in **bug fixing**

Scalability

- Need for easy integration for the many **new models**
- Multiplication of code structures
- Need to **reduce** operating **costs**



Time to Market

- **Long** time-to-market
- Difficult to meet planning and **delivery** expectations

Experiment

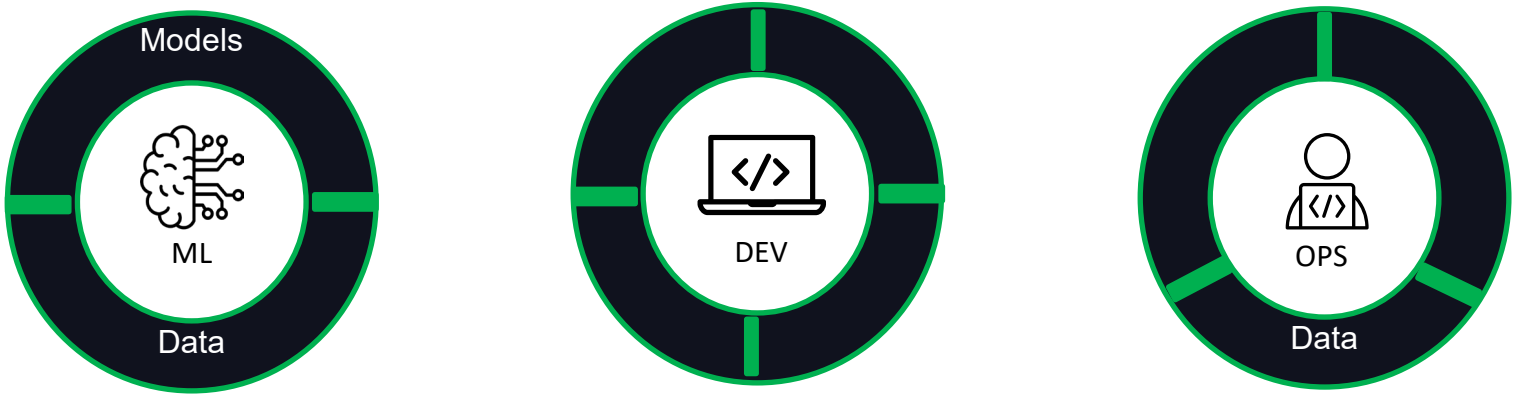
- Difficulty **reproduce** and control experiments
- Complex **Integration** of successful experiments in other environments

©2024 Databricks Inc. — All rights reserved

MLOps

ML + Dev+ Ops = MLOps

Collaborative and experimental in nature |Automate as much as possible |Continuous improvement of ML Models |Standardize and Scale



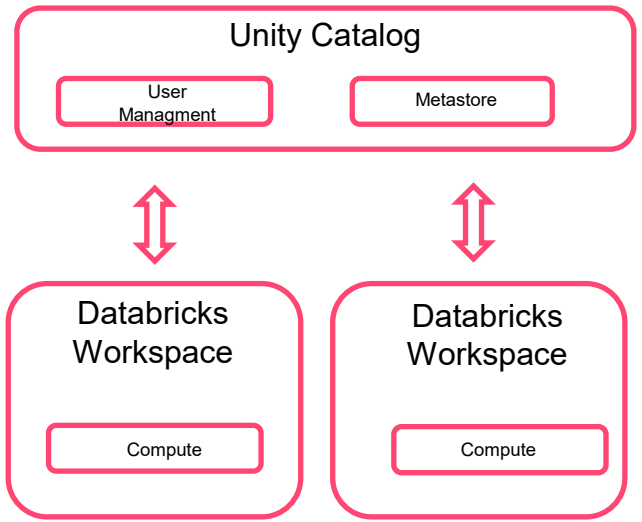
©2024 Databricks Inc. — All rights reserved

Tools

©2024 Databricks Inc. — All rights reserved

Unity Catalog

What enables



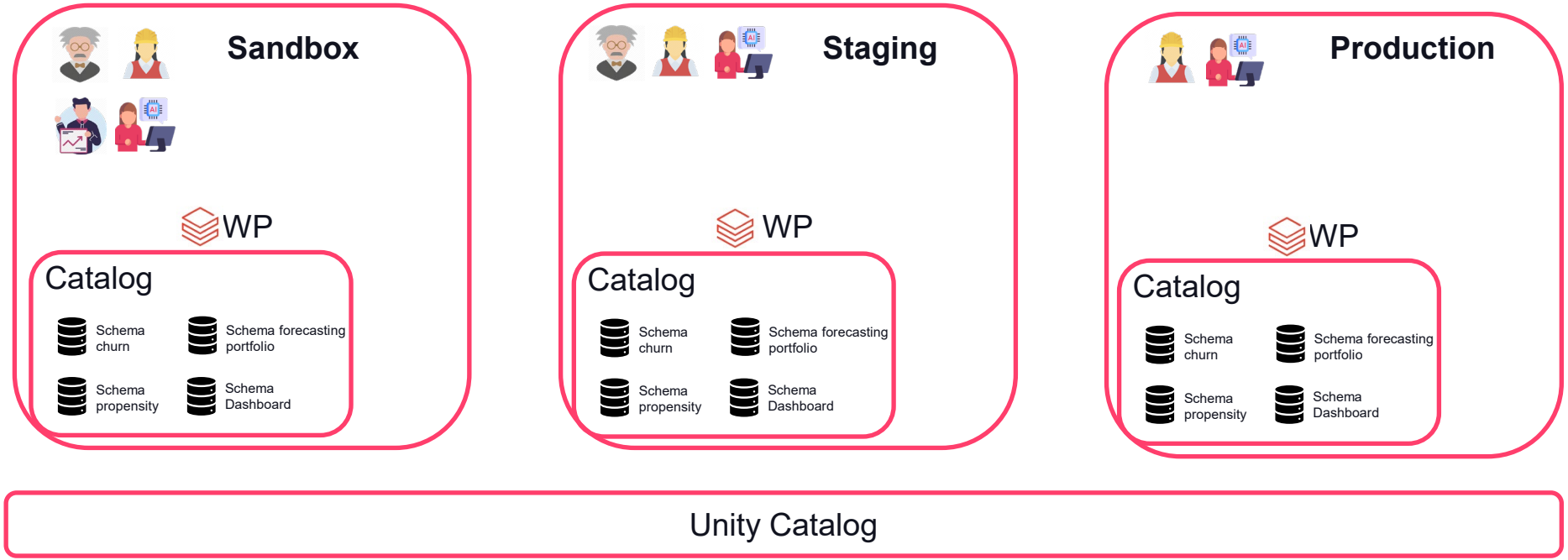
Main Features

- Secure everywhere
- Compliant security mode
- Auditing & Lineage
- Data discovery
- System tables

©2024 Databricks Inc. — All rights reserved

Unity Catalog

How we leverage it



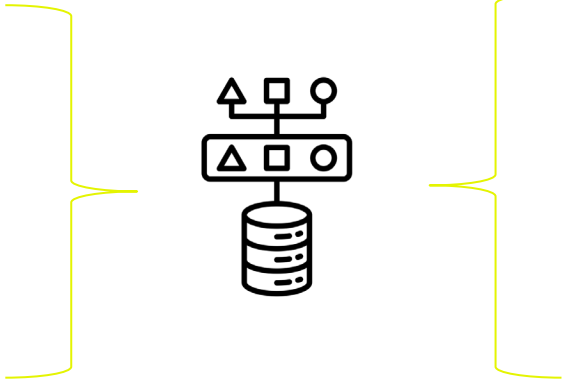
©2024 Databricks Inc. — All rights reserved

Databricks Asset Bundles

Why it is important

COMPONENTS

- Code
- Execution environment
- Mlflow tracking server
- Databricks workflows



DATA PRODUCTS

- Dashboards
- Models
- Tables

©2024 Databricks Inc. — All rights reserved

Databricks Asset Bundles

What it is

YAML files which specify: artifacts, resources, and configurations of a Databricks project.

```yaml
# These is the default bundle configuration if not otherwise overridden in
# the "targets" top-level mapping.
bundle: # Required.
  name: string # Required.
  compute_id: string
  git:
    origin_url: string
    branch: string

# These are for any custom variables for use throughout the bundle.
variables:
  <some-unique-variable-name>:
    description: string
    default: string

# These are the default workspace settings if not otherwise overridden in
# the following "targets" top-level mapping.
workspace:
  artifact_path: string
  auth_type: string
```

It enables the ability to validate, deploy and run Databricks workflows and includes the possibility to manage MLflow assets, through **Databricks CLI**.

```
databricks bundle init <project-template-local-path-or-url>
```

They ease the management of **CI/CD pipelines**.

```yaml
# Deploy bundle to workspace
- task: AzureCLI@2
  displayName: Deploy bundle to ${{ parameters.environment }} deployment target
  inputs:
    scriptType: "ps"
    scriptLocation: "inlineScript"
    workingDirectory: src
    inlineScript: |
      $Env:DATABRICKS_TOKEN = "$(databricks.Principal.AccessToken)"
      $Env:DATABRICKS_HOST = "https://adb-5973735350228664.4.azuredatabricks.net"

      databricks bundle deploy -t ${{ parameters.environment }}
    azureSubscription: ${{ parameters.azureSubscription }}
```

Write code once, deploy to multiple workspace easily

©2024 Databricks Inc. — All rights reserved

Databricks Asset Bundles

Why it is important for us

Collaboration

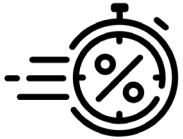
It helps organize and manage various source files efficiently, ensuring smooth collaboration.

Compliance

It helps maintain a versioned history of code and infrastructure work.

Standardization

Set standards for all projects including permission, CI/CD configurations service principals



Time to market

It helps developing ML projects faster, setting best practices standards from the beginning, therefore improves **velocity**

©2024 Databricks Inc. — All rights reserved

Databricks Asset Bundles

How we leverage it

```
repository-name            <- Root directory of the repository
|
├── .gitignore             <- Pre-defined files to ignore in Version Control.
|
├── pre-commit-config.yaml <- Tests which are performed at CI level over the code. Needed for successfully passing the PR step.
|
├── test-requirements.txt  <- Specifies Python dependencies for testing ML code (for example: pytest).
|
├── src             <- Contains python code, notebooks and ML assets related to one ML project.
|   |
|   ├── requirements.txt       <- Specifies Python dependencies for ML code (for example: model training, batch inference).
|   |
|   ├── requirements-custom.txt <- Specifies Plenitude custom Python dependencies for ML code (for example: atum-ndp-dy365, plenitude-metrics).
|   |
|   ├── databricks.yml         <- databricks.yml is the root bundle file for the ML project that can be loaded by databricks CLI bundles. It defines the bundle name, workspace URL and asset config component to be included.
|   |
|   ├── training               <- Training folder contains Notebook that trains and registers the model.
|   |
|   ├── validation             <- Optional model validation step before deploying a model.
|   |
|   ├── monitoring             <- Model monitoring, feature monitoring, etc.
|   |
|   ├── deployment             <- Deployment and Batch inference workflows
|   |   |
|   |   ├── batch_inference    <- Batch inference code that will run as part of scheduled workflow.
|   |   |
|   |   ├── model_deployment   <- As part of CD workflow, deploy the registered model by assigning it the appropriate alias.
|   |
|   |
|   ├── tests                  <- Unit tests for the ML project, including the modules under `features`.
|   |
|   ├── assets              <- ML asset (ML jobs, MLflow models) config definitions expressed as code, across dev/staging/prod/test.
|       |
|       ├── model-workflow-asset.yml               <- ML asset config definition for model training, validation, deployment workflow
|       |
|       ├── batch-inference-workflow-asset.yml     <- ML asset config definition for batch inference workflow
|       |
|       ├── ml-artifacts-asset.yml                 <- ML asset config definition for model and experiment
|
├── .azure/devops-pipelines <- Configuration folder for CI/CD using Azure DevOps Pipelines. The CI/CD workflows deploy ML assets defined in the `./assets/*` folder with databricks CLI bundles.
    |
    ├── ci.yaml              <- Specifies steps for testing code and building the app for each PR.
    |
    ├── cd.yaml              <- Specifies actual steps for deploying and/or running code into the computing environments (uses cd-template.yaml).
    |
    ├── cd-template.yaml     <- Specifies parametrized steps for deploying and/or running code into the computing environments.
```


©2024 Databricks Inc. — All rights reserved

Databricks Asset Bundle

How we leverage it – databricks.yaml

Include Databricks job definitions

Define target workspace definition for each stage

```yaml
# The name of the bundle. run `databricks bundle schema` to see the full bundle settings schema.
bundle:
  name: src

variables:
  experiment_name:
    description: Experiment name for the model training.
    default: /Users/${workspace.current_user.userName}/${bundle.target}-replace-project-name-experiment
  model_name:
    description: Model name for the model training.
    default: replace-project-name-model

include:
  # Assets folder contains ML artifact assets for the ml project that defines model and experiment
  # And workflows assets for the ml project including model training -> validation -> deployment,
  # batch inference, data monitoring, metric refresh, alerts and triggering retraining
  - ./assets/*.yml

# Deployment Target specific values for workspace
targets:
  dev:
    default: true
    workspace:
      # TODO: add dev workspace URL
      host:

  staging:
    workspace:
      host: https://1234567890abcdefghilmnip.azuredatabricks.net

  prod:
    workspace:
      host: https://1234567890abcdefghilmnip.azuredatabricks.net

  test:
    workspace:
      host: https://1234567890abcdefghilmnip.azuredatabricks.net
```


©2024 Databricks Inc. — All rights reserved

Databricks Asset Bundle

How we leverage it – model-workflow-asset.yaml

Define a Databricks job, which can be scheduled and can perform several tasks by orchestrating notebooks execution.

Each task is responsible for executing a different notebook.
In this case:

- first two tasks are related to data preparation and can be executed in parallel
- Last task to be executed is related instead to model training and model validation

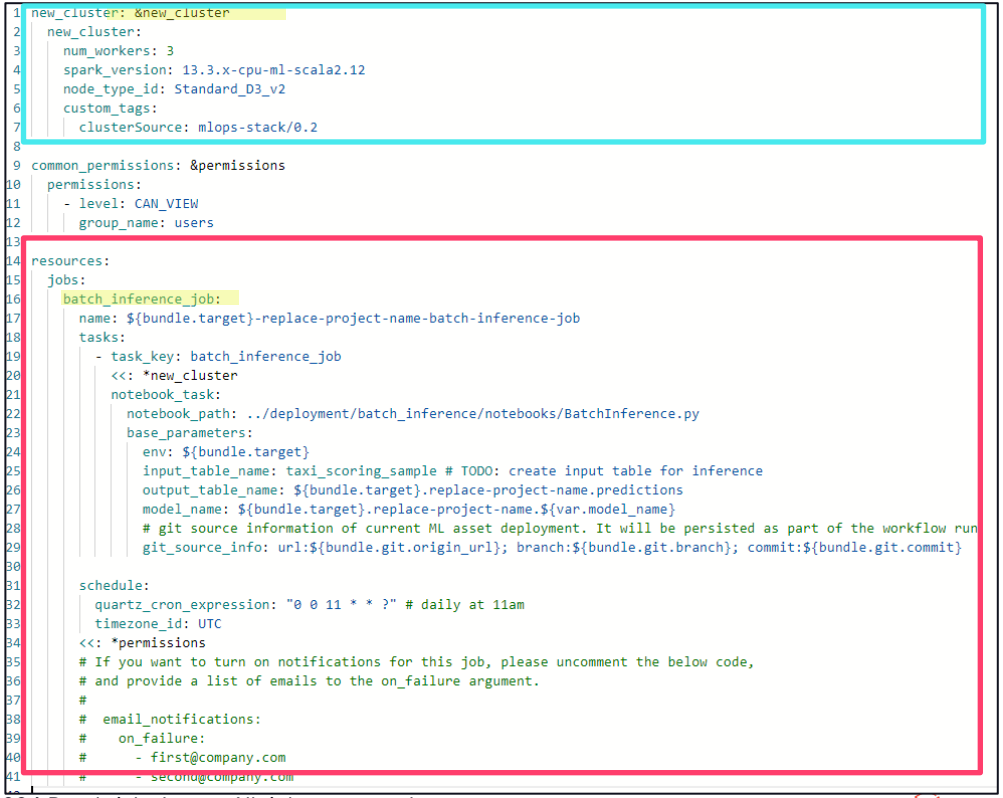
```yaml
resources:
  jobs:
    model_training_job:
      name: ${bundle.target}-replace-project-name-model-training-job
      parameters:
        - name: env
          default: ${bundle.target}
        - name: snapshot_date
          default: "{{job.trigger.time.[iso_date]}}"
      # job_clusters:
      #   - job_cluster_key: model_training_job_cluster
      # ...
      tasks:
        - task_key: MasterCreation
          # job_cluster_key: model_training_job_cluster
          existing_cluster_id: 0129-161127-2vcxccf8 # 0622-083621-e35takbu # TODO: must be removed for production clusters
          notebook_task:
            notebook_path: ../feature_engineering/notebooks/MasterCreation.py
            base_parameters:
              # git source information of current ML asset deployment. It will be persisted as part of the workflow run
              git_source_info: url:${bundle.git.origin_url}; branch:${bundle.git.branch}; commit:${bundle.git.commit}

        - task_key: TargetCreation
          # job_cluster_key: model_training_job_cluster
          existing_cluster_id: 0129-161127-2vcxccf8 # TODO: must be removed for production clusters
          notebook_task:
            notebook_path: ../feature_engineering/notebooks/TargetCreation.py
            base_parameters:
              # git source information of current ML asset deployment. It will be persisted as part of the workflow run
              git_source_info: url:${bundle.git.origin_url}; branch:${bundle.git.branch}; commit:${bundle.git.commit}

        - task_key: ModelTraining
          # job_cluster_key: model_training_job_cluster
          existing_cluster_id: 0129-161127-2vcxccf8 # TODO: must be removed for production clusters
          notebook_task:
            notebook_path: ../training/notebooks/ModelTraining.py
            base_parameters:
              experiment_name: ${var.experiment_name}
              model_name: ${bundle.target}.replace-project-name.${var.model_name}
              # git source information of current ML asset deployment. It will be persisted as part of the workflow run
              git_source_info: url:${bundle.git.origin_url}; branch:${bundle.git.branch}; commit:${bundle.git.commit}
          depends_on:
            - task_key: MasterCreation
            - task_key: TargetCreation
```

Databricks Asset Bundle

How we leverage it - batch-inference-workflow-asset.yml

A job cluster is leveraged for performance reasons for performing model inference

Job related to inference tasks



```yaml
new_cluster: &new_cluster
  new_cluster:
    num_workers: 3
    spark_version: 13.3.x-cpu-ml-scala2.12
    node_type_id: Standard_D3_v2
    custom_tags:
      clusterSource: mlops-stack/0.2

common_permissions: &permissions
  permissions:
    - level: CAN_VIEW
      group_name: users

resources:
  jobs:
    batch_inference_job:
      name: ${bundle.target}-replace-project-name-batch-inference-job
      tasks:
        - task_key: batch_inference_job
          <<: *new_cluster
          notebook_task:
            notebook_path: ../deployment/batch_inference/notebooks/BatchInference.py
            base_parameters:
              env: ${bundle.target}
              input_table_name: taxi_scoring_sample # TODO: create input table for inference
              output_table_name: ${bundle.target}.replace-project-name.predictions
              model_name: ${bundle.target}.replace-project-name.${var.model_name}
              # git source information of current ML asset deployment. It will be persisted as part of the workflow run
              git_source_info: url:${bundle.git.origin_url}; branch:${bundle.git.branch}; commit:${bundle.git.commit}

      schedule:
        quartz_cron_expression: "0 0 11 * * ?" # daily at 11am
        timezone_id: UTC
      <<: *permissions
      # If you want to turn on notifications for this job, please uncomment the below code,
      # and provide a list of emails to the on_failure argument.
      #
      #  email_notifications:
      #    on_failure:
      #      - first@company.com
      #      - second@company.com
```


©2024 Databricks Inc. — All rights reserved

MLOps for *MLHopes*

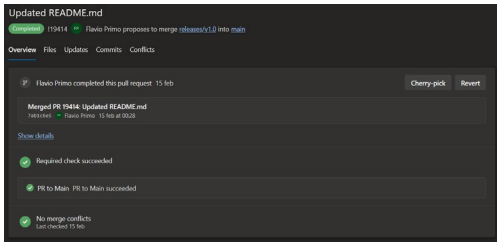
©2024 Databricks Inc. — All rights reserved

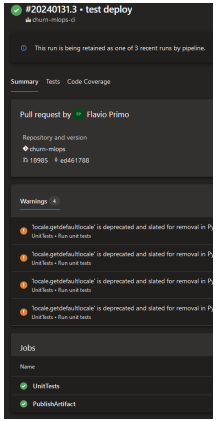
CI / CD high level

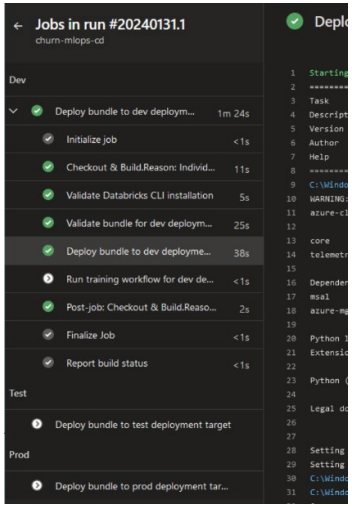
Orchestrate and enforce the testing, development and deployment process

At the core of the solution is the CI/CD pipeline developed on **Azure DevOps**, where the code is versioned on **git** following a **TBD (Trunk-Based Development)** approach for managing branches.

Code development process:







1. **Create a PR** with the proposed changes which performs code checks.

2. **CI** performs tests, build and artifact versioning.

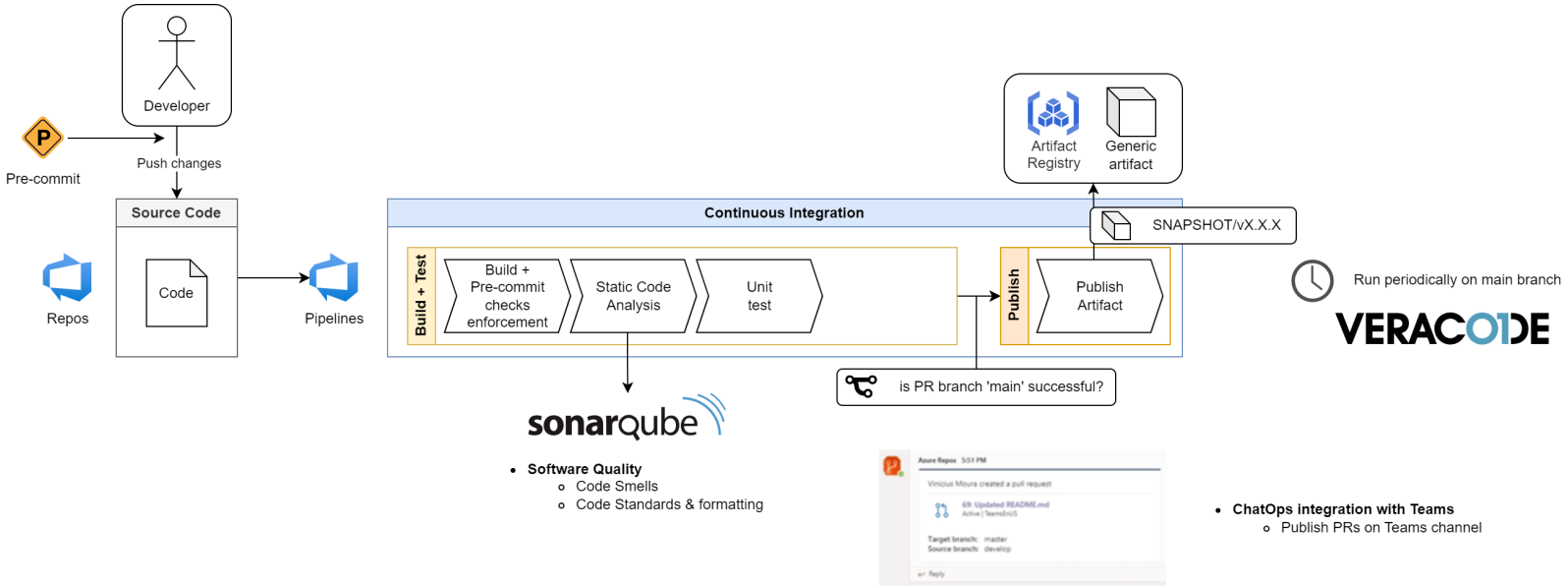
3. **CD** performs bundle validation, deployment and builds on all the environments sequentially.

©2024 Databricks Inc. — All rights reserved

CI

Approve modifications, test, build and version the code artifact

Leverage **Microsoft Teams** to perform **ChatOps** duties, while **SonarQube and Veracode** for static code analysis checks for code **security and quality purposes**.



©2024 Databricks Inc. — All rights reserved

Deep dive CI

Testing

Name	Description	When to perform	Tools / Examples
Static code analysis	Perform static code analysis (without running the code) for potential **defects**, **security vulnerabilities**, and coding standards violations.	Depends on the type of test. Locally, PRs, scheduled, during code pushes, etc.	Ruff , Black , Bandit , SonarQube , VeraCode
Unit testing	Test the **logic** of small, discrete pieces of code (e.g., a function, method, or class).	Locally, at every PR to main and CI.	Pytest
Smoke/integration testing	Ensure that integrated **components work as expected** when combined and that they interact correctly with each other. Allows to test end-to-end the deployed component in the given environment.	After deployment on a given environment.	Pytest , custom scripts
Load testing	Ensure the system's ability to handle the anticipated **workload** without experiencing performance degradation or failures.	Scheduled during non-peak hours.	Gatling , K6

©2024 Databricks Inc. — All rights reserved

Deep dive CI

Focus on static code analysis

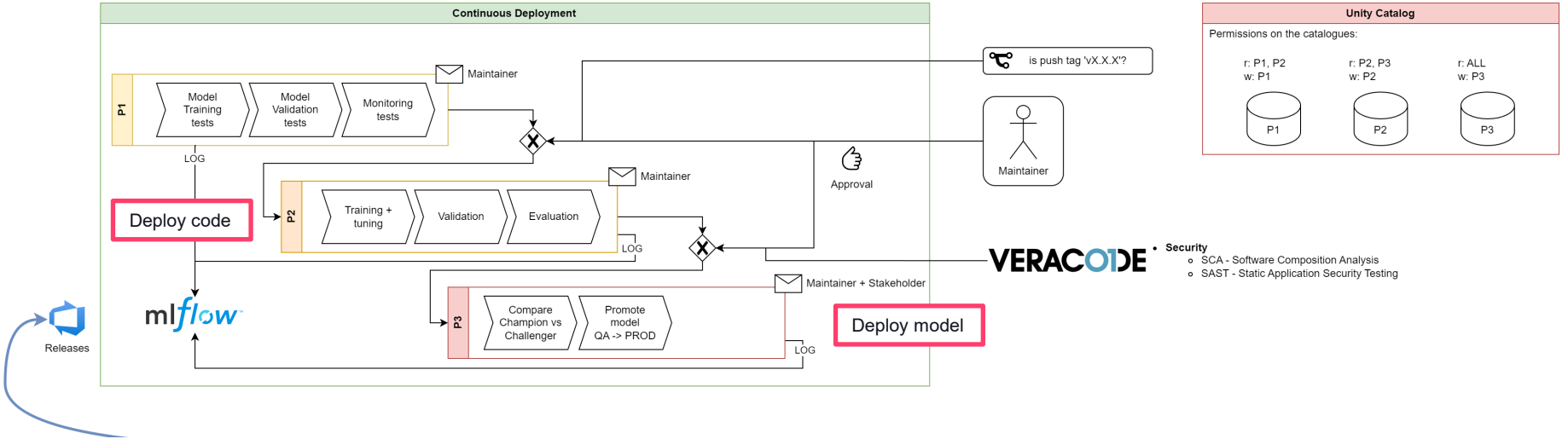
Tool	Description
Ruff	Linting and code formatting tool for **code quality** checking (covers more than 400 rules). Includes auto fix feature.
Black	**Code formatting** tool (compatible with Databricks noebooks with Blackbricks extension).
Bandit	Tool for detecting common **security issues**.
Sonarqube	Provides in-depth **reports** for code **quality** (code duplication, test coverage, code complexity).
Veracode	Cloud-based application to in-depth identify, prioritize and remediate **security vulnerabilities**.

First line of code checks, performed at **commit phase** (via pre-commit tool) and enforced during PRs

In-depth and **comprehensive code** checks with **reports generation**

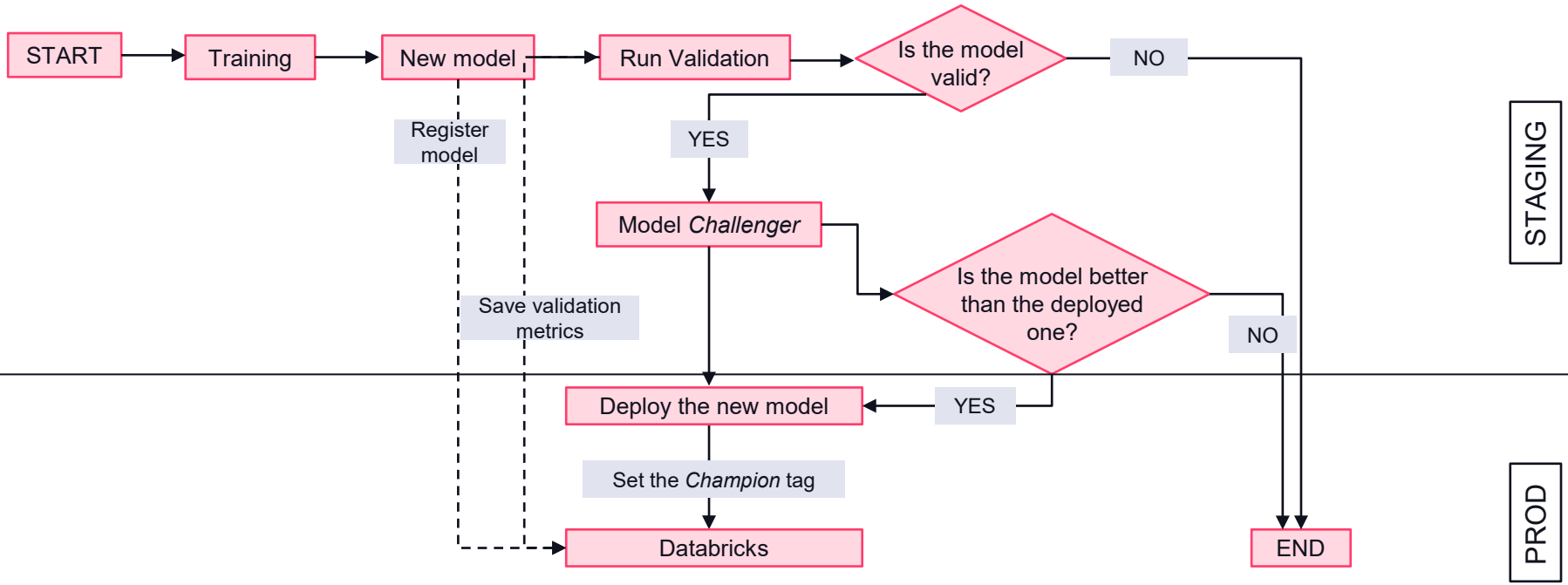
©2024 Databricks Inc. — All rights reserved

CD



©2024 Databricks Inc. — All rights reserved

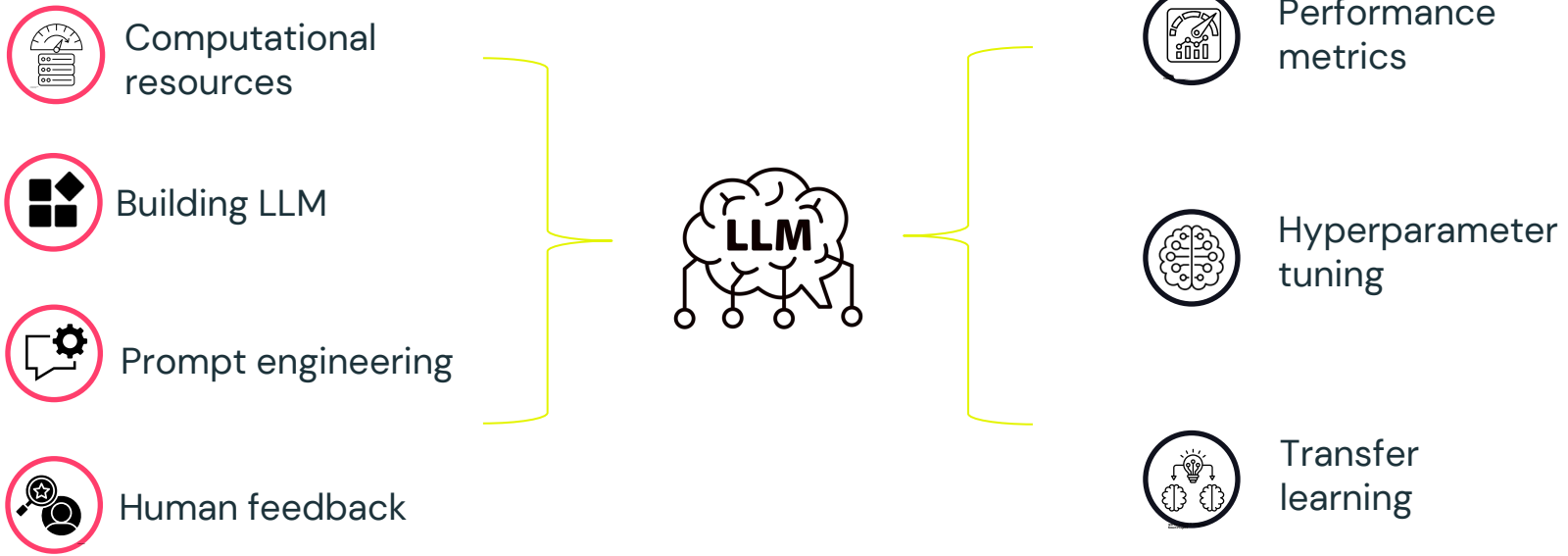
Model evaluation



©2024 Databricks Inc. — All rights reserved

LLMlops

What changes with LLMs



©2024 Databricks Inc. — All rights reserved

Demo

©2024 Databricks Inc. — All rights reserved

Video will be presented

©2024 Databricks Inc. — All rights reserved

Takeways

©2024 Databricks Inc. — All rights reserved

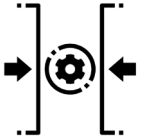
Takeways

Culture

Culture shift into an MLOps approach is equally important as the implementation

Multidisciplinarity

Involvement of several org departments since MLOps is a multidisciplinary feat (Data Science, Operations, Infrastructure, Third-party Vendors)



Constraints

Organizational constraints must not be considered as limitations to shift into an MLOps approach

Investment

Revolutionizing the way projects are carried out is an **investiment**: it requires effort, but in the short-medium term brings benefits, which highly compensate the effort

©2024 Databricks Inc. — All rights reserved

An amazing team and..

Flavio Primo

Cloud & Data Architect, BIP

bip. xTech

©2024 Databricks Inc. — All rights reserved