

mlflow

Platform for Machine Learning Lifecycle

Jules S. Damji
@2twitme



Outline – Introduction to MLflow: How to Use MLflow Projects – Module 2

- MLflow Projects
 - Concepts and Motivations
 - Executing MLflow projects in (DCE)
 - Building MLflow Projects
 - Explore MLflow UI
 - Tutorials & Exercises
- Q & A

<https://github.com/dmatrix/tmhs-workshop>

MLflow Components

mlflow

Tracking

Record and query experiments: code, data, config, and results

mlflow

Projects

Package data science code in a format that enables reproducible runs on many platform

mlflow

Models

Deploy machine learning models in diverse serving environments

mlflow

Model Registry

Store, annotate and manage models in a central repository

databricks.com/mlflow



mlflow.org



github.com/mlflow



twitter.com/MLflow

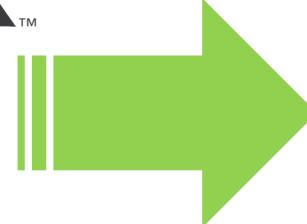


MLflow Projects Motivation

Diverse set of tools



TensorFlow



Diverse set of environments



Challenge: ML results difficult to reproduce

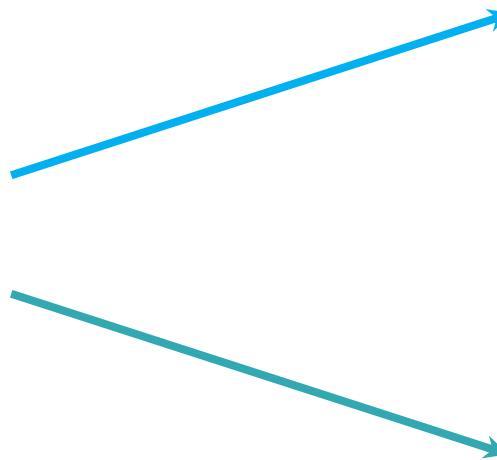


mlflow™
Projects

Package data science
code in a format that
enables reproducible runs
on any platform



MLflow Projects



Local Execution

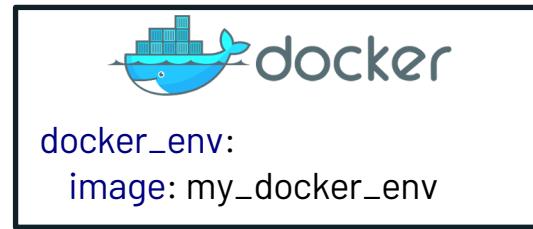


Remote Execution



MLflow Projects Environments

- MLflow Supports three software environments
 - Environments to Execute your Entry point code



- Conda for Python Packages
- Docker for non-Python Packages
- Current System environment

1. Example MLflow Project File

```
my_project/
├── MLproject
|
└── main.py
    ├── conda.yaml
    └── model.py
...
...
```

```
name: tutorial

conda_env: conda.yaml

entry_points:
  main:
    parameters:
      batch_size: {type: int, default: 10}
      epochs: {type: int, default: 100}
    command: "python train_keras.py {batch_size} {epochs}
```

```
$ mlflow run git://<my_project>.git -P epochs=200
mlflow.run("git://<my_project>", parameters={..})

mlflow run . -e main -P epochs=200
```

2. Example Conda.yaml

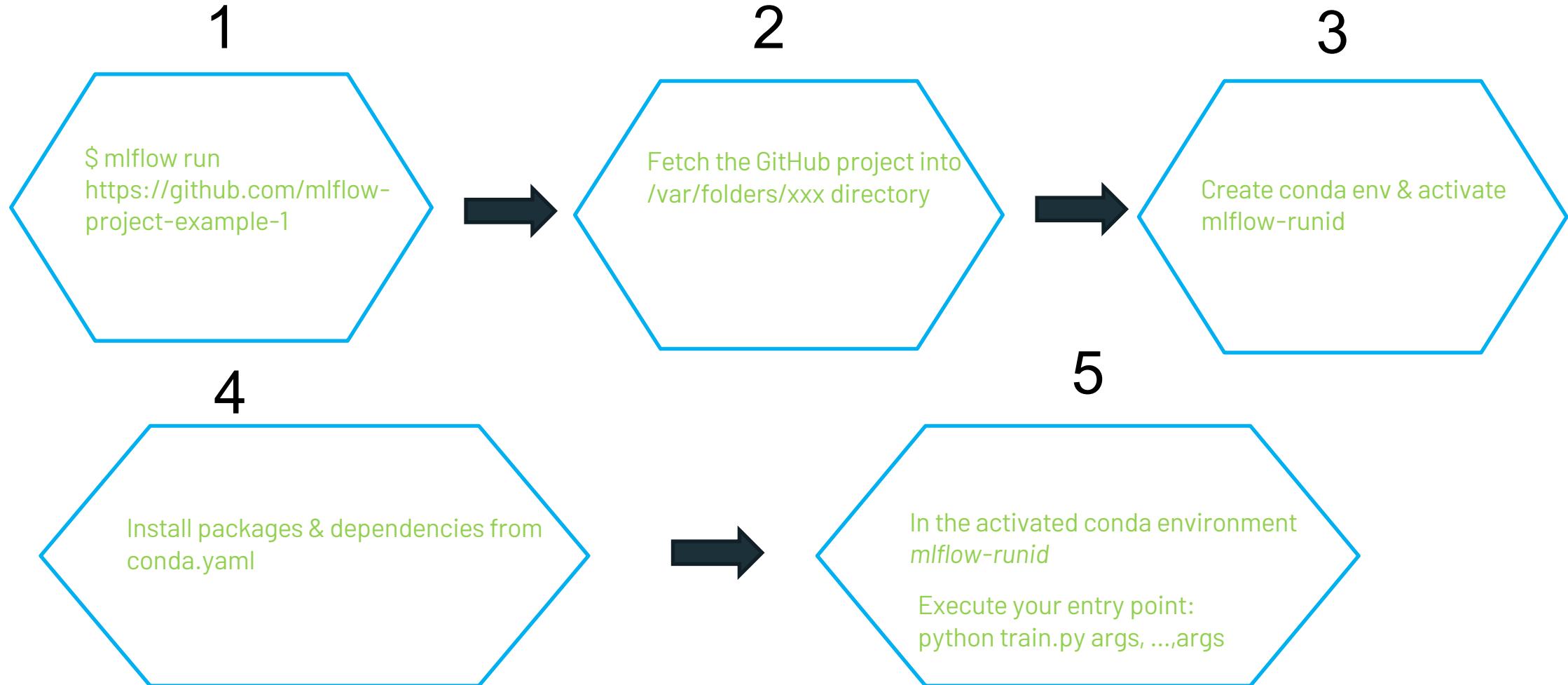
```
my_project/
└── MLproject
    ├── conda.yaml
    ├── main.py
    └── model.py
    ...

```

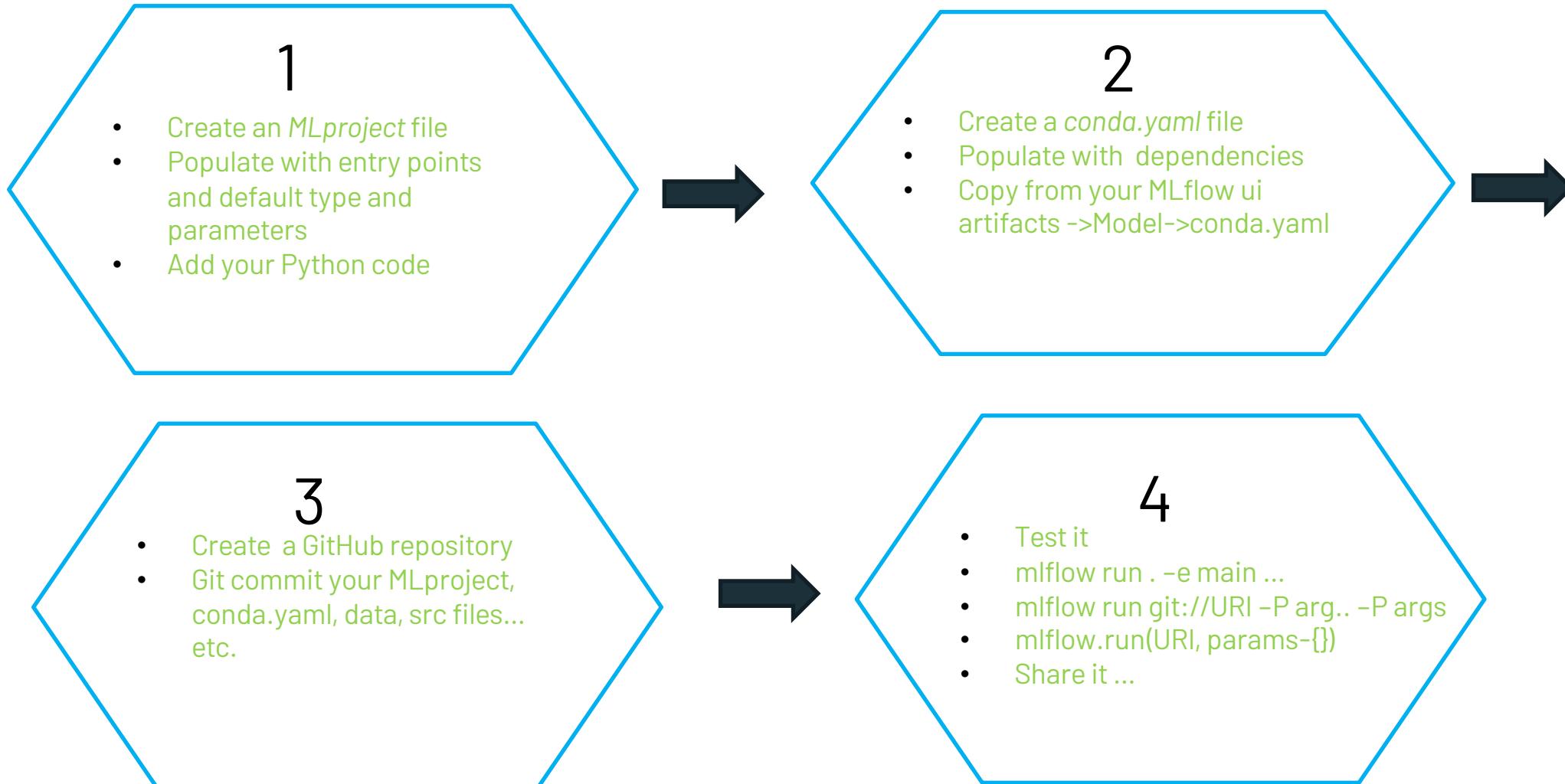
```
channels:
- defaults
- conda-forge
dependencies:
- python=3.7.5
- pip
- pip:
  - mlflow
  - keras==2.3.1
  - tensorflow==2.0.0
name: mlflow-env
```



Anatomy of MLflow Project Execution



How to build an MLflow Project



MLflow Projects

Packaging format for reproducible ML runs

- Any code folder or GitHub repository
- MLproject file with project configuration

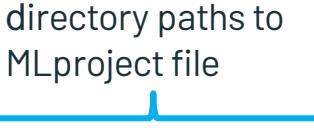
Defines dependencies for reproducibility

- Conda (+ R, Docker, ...) dependencies can be specified in MLproject
- Reproducible in (almost) any environment

Execution API for running projects

- CLI / Python / R / Java
- Supports local and remote execution
 - mlflow run –help (CLI)
 - mlflow run <https://github.com/dmatrix/jsd-mlflow-examples.git#keras/imdbclassifier> (CLI)
 - mlflow.run (<project_uri>, parameters={}) or mlflow.projects.run(<project_uri>, parameters={}) (API)

directory paths to
MLproject file



MLflow Projects Usage

- Reproducibility and Experimentation ✓

- Reproduce experiments in a target environment
- Experiment with different hyperparameters
- Use of CLI and Programmatic interface

- Shareability ✓

- Use other people's models and experiment via a MLFlow Project GitHub

- Flexibility ✓

- Execute on three different software environments
 - Conda
 - Docker
 - System

- Extensibility

- Create complicated Project Workflows

MLflow Project Tutorials

<https://github.com/dmatrix/tmhs-workshop>

Thank you! 😊

Q & A

jules@databricks.com
@2twitme

<https://www.linkedin.com/in/dmatrix/>