

Building an ML Experimentation Platform for Easy Reproducibility



lakeFS



About Me

- I am [Vino](#) Duraisamy.
- SWE -> Data/ML Engineer -> Developer Advocate
- Open-source contributor @lakeFS



@vinodhini_sd

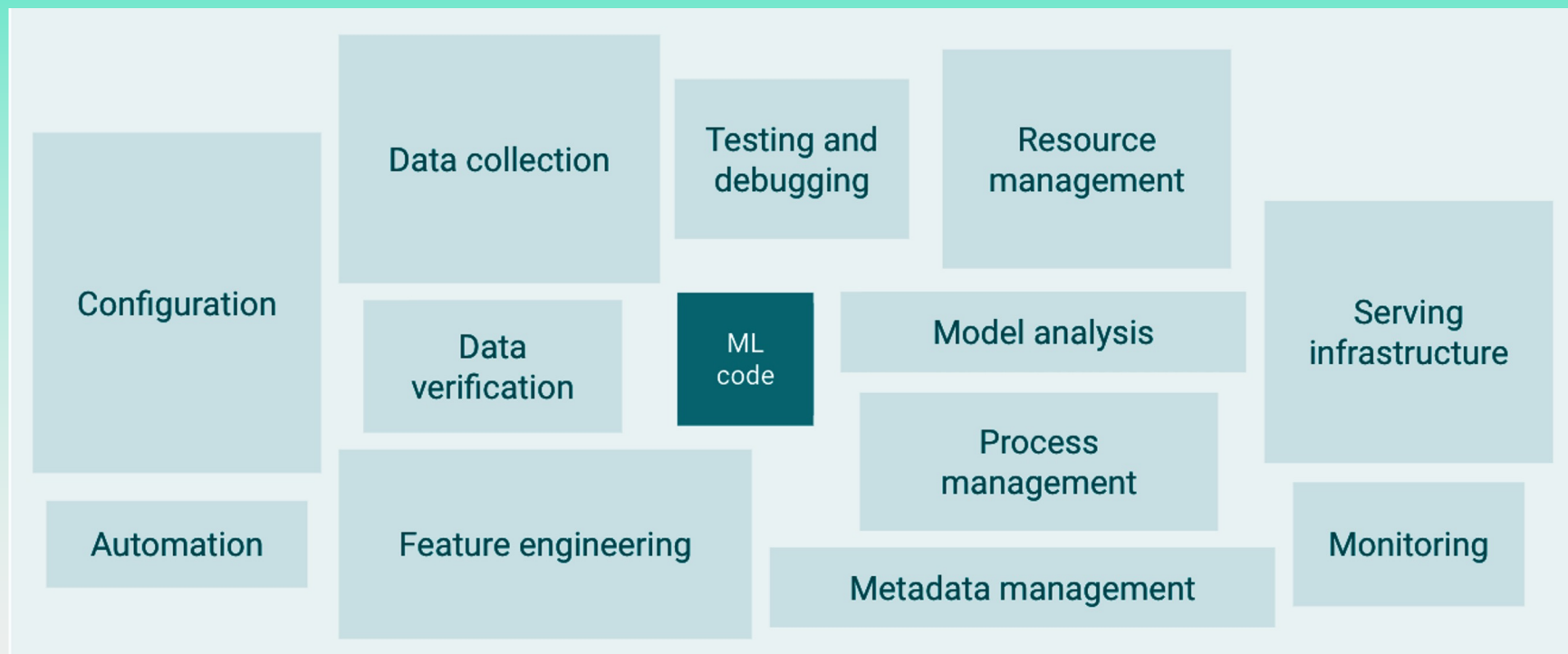


vinodhini-sd.medium.com

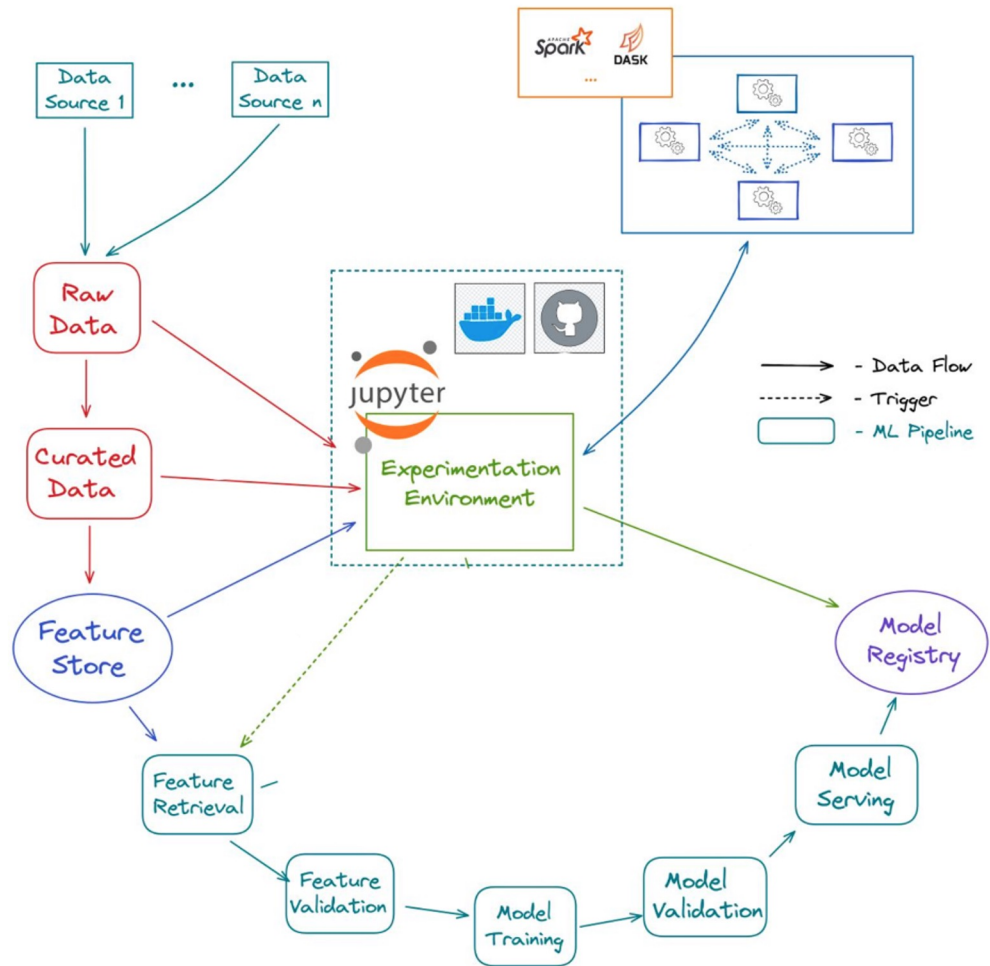


@vinodhini-sd

MLOps in a Complex Data World



MLOps in a Complex Data World



Uncovering the Gaps in MLOps

- ML experimentation infrastructure
 - Automation of experiments
 - Crisis of Reproducibility
 - Explainability of ML models
- Data/Feature ownership and Lineage
- Collaboration
 - MLEs working with same training data/feature set
- Version Control all ML assets atomically
 - Data, Model Artifacts, Code, Config, Metrics

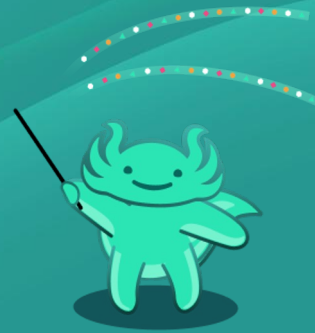
MLOps landscape today



1. ML Experimentation

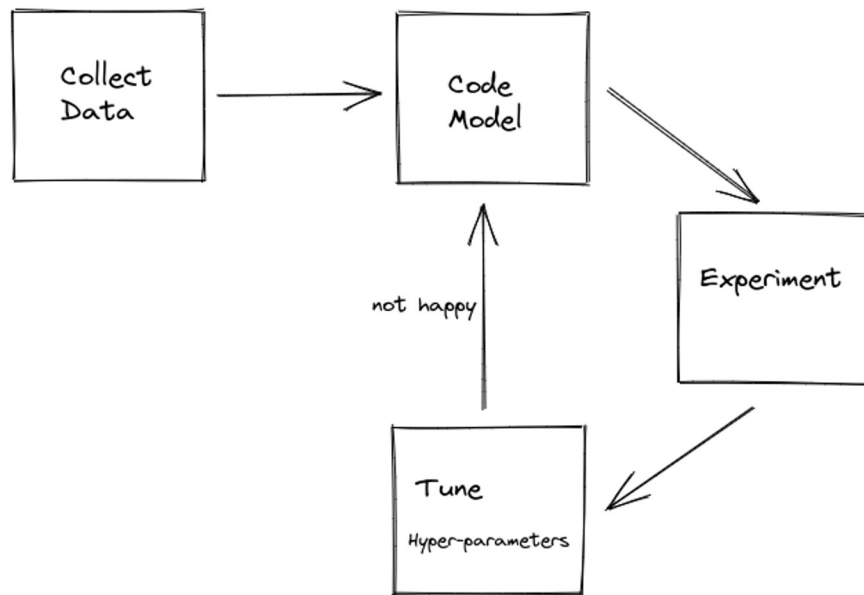


lakeFS

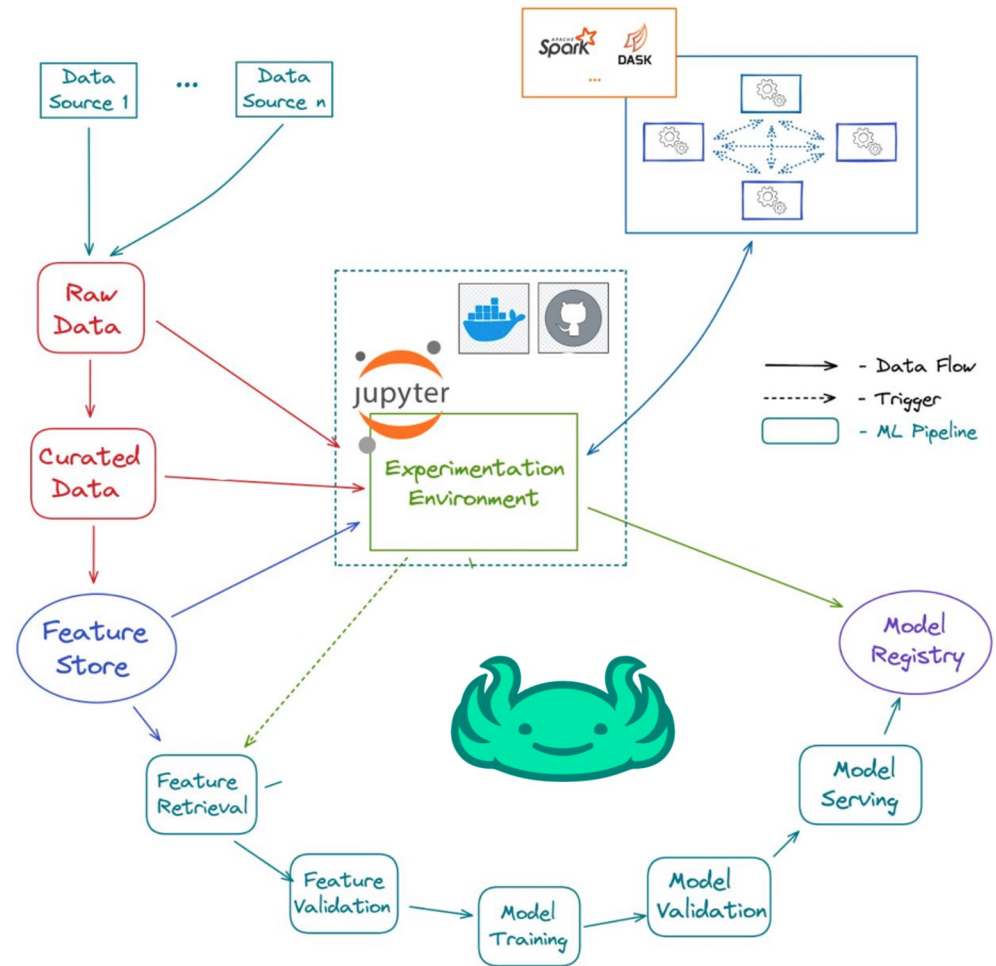


ML Experimentation: Challenges

- Iterative development
- Ensure data consistency
- Trust your ML pipeline
- Compliance to Regulations
- Reproducibility of ML training
- Explainable ML models

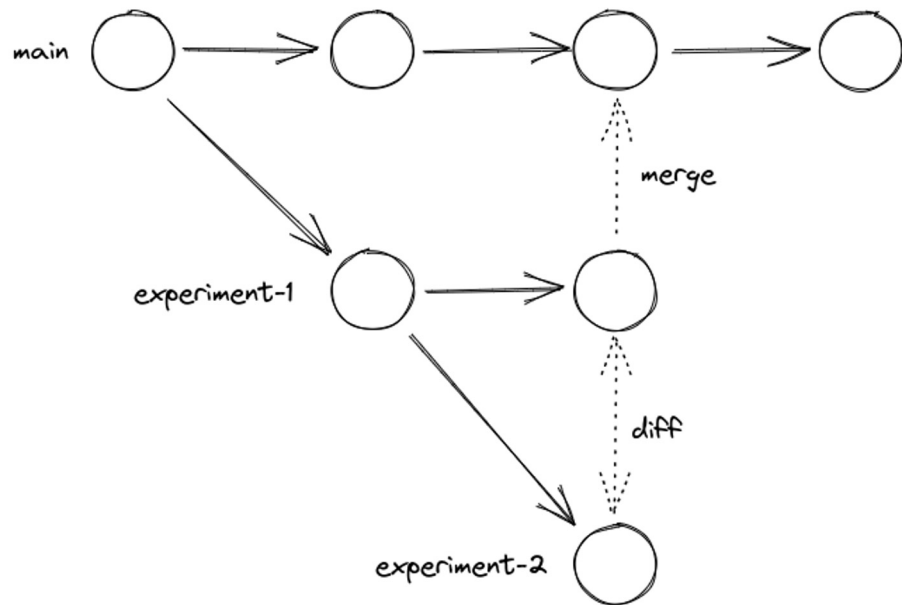


MLOps with lakeFS

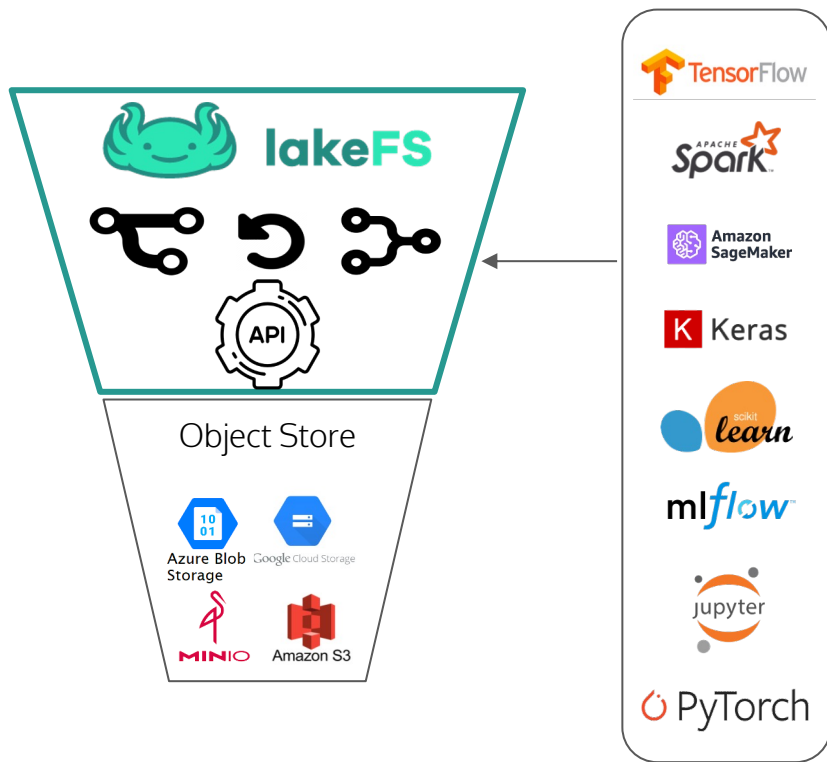


ML Experimentation Infrastructure

Git-like branching and
experimenting **with**
lakeFS



Git for Data – lakeFS



s3://repo/collections/foo



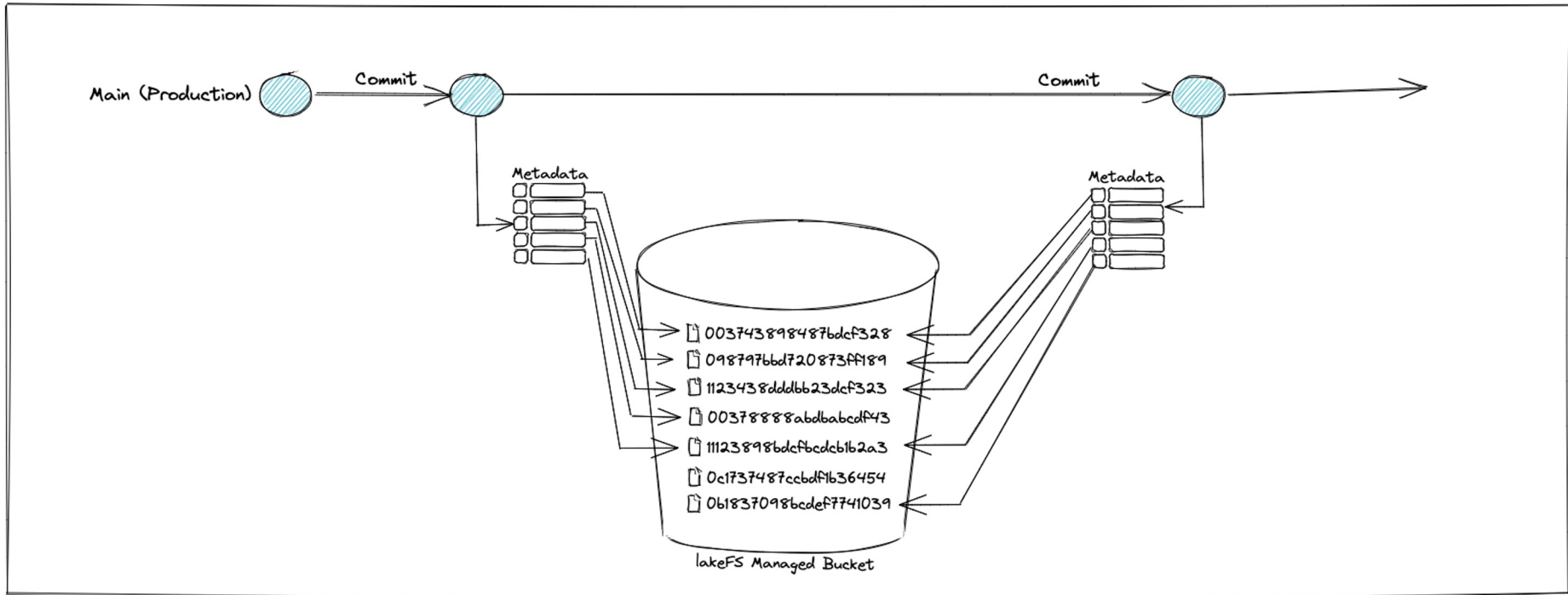
lakefs://repo/main/collections/foo

```
lakectl branch create \  
lakefs://repo/testing-spark-3 \  
--source lakefs://repo/main
```

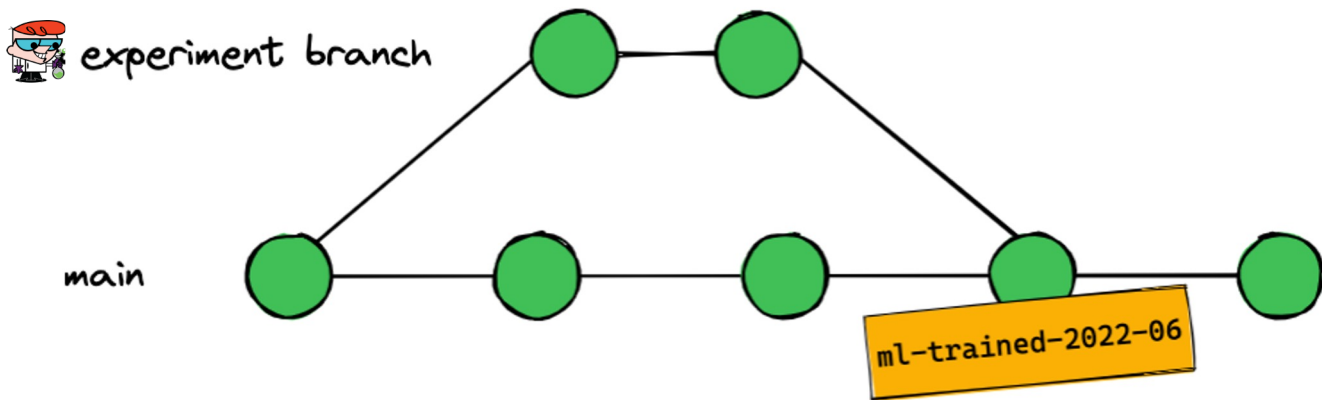
```
# output:  
# created branch 'testing-spark-3.0',  
# pointing to commit ID:  
'd1e9adc71c10a'
```



How does lakeFS work?



lakeFS for ML experimentation



```
df = spark.read.parquet(  
    'lakefs://my-repo/ml-trained-2022-06/inputs/vists/')
```

Merged estimation model using new learning set

hyper-params

{ ... }

training-job-uri

github.com/my-ml/a34902

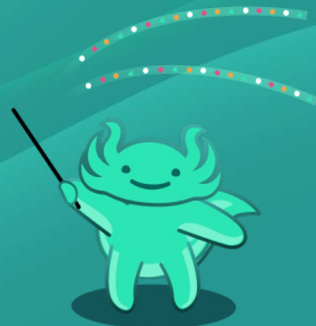
infra-uri

github.com/my-terraform/..

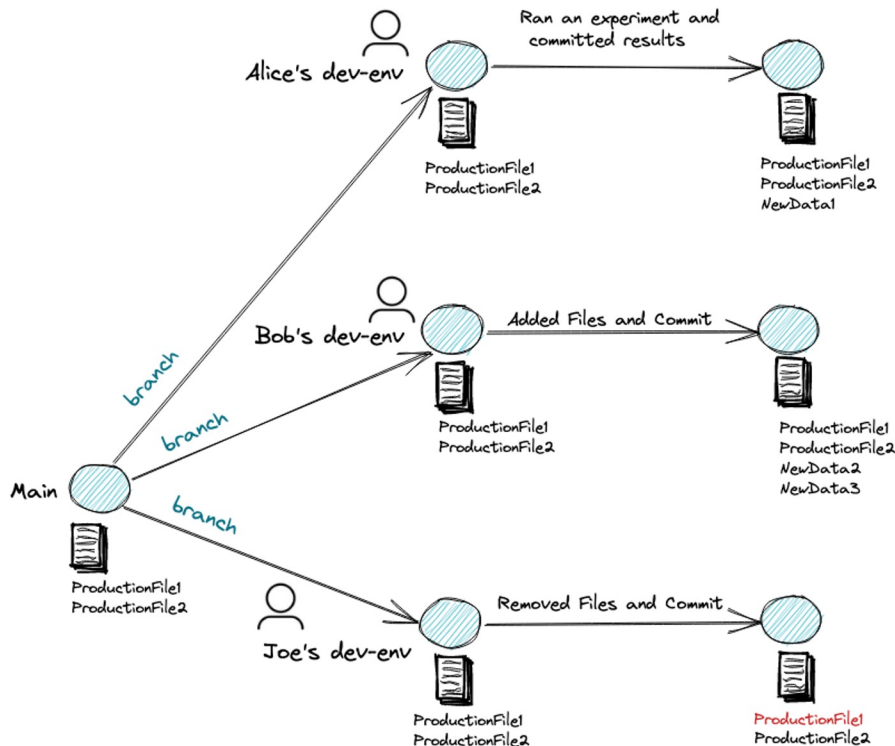
2. Unlock Collaboration



lakeFS



Modularize & Collaborate at each step

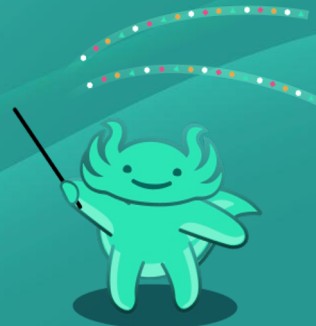


Enable ML engineers to work on same training data and features without duplication **using lakeFS**

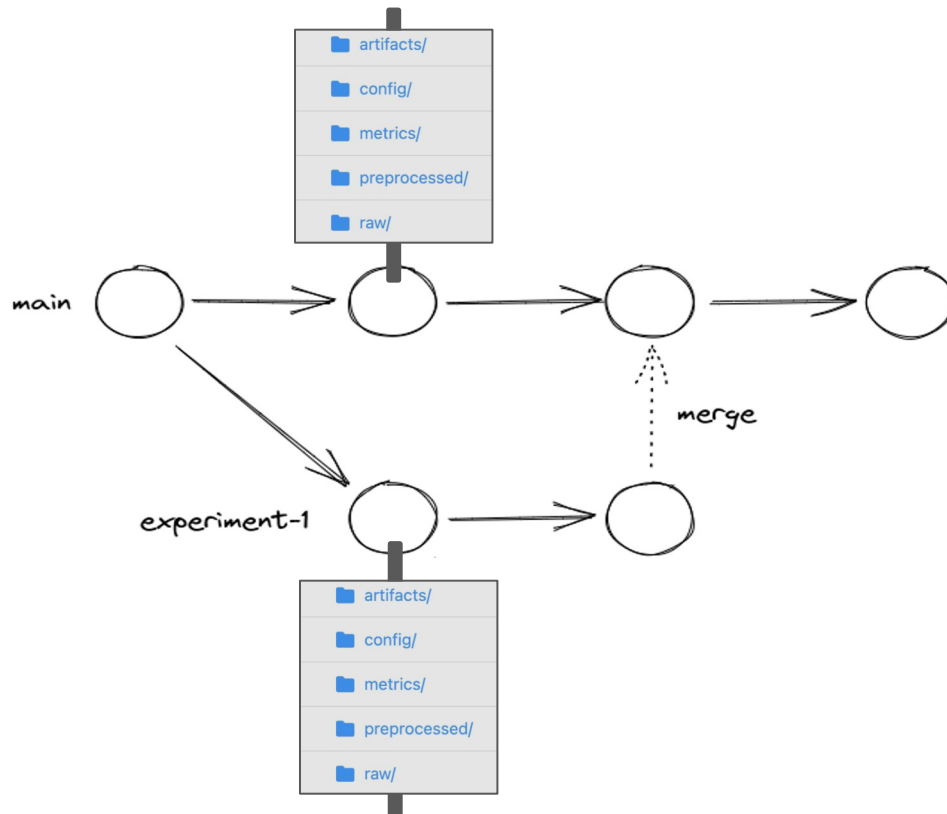
3. Version Control all ML assets atomically



lakeFS



Version control all ML assets atomically



Using lakeFS



Experiment in Isolation



\$ lakectl **branch create** ml-test



Tag



\$ lakectl **tag create** exp-v1 a83b1c



Reproduce



\$ spark.read.parquet
('lakefs://my-repo/**exp-v1**')



Reprocess



\$ lakectl **branch create** new-logic
\$ lakectl **merge** new-logic main



Undo a change

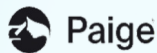


\$ lakectl **revert** main^1

Demo Time!



lakeFS Community



lakefs.io/slack

