Dev/Stage/Prod is an Anti-Pattern for Data Pipelines

Oz Katz, March 2024
T-3h
T+1m

Oh boy
T+12h

THIS IS FINE.
Hi👋

Oz Katz
CTO & Co-Creator of lakeFS

github.com/ozkatz

@ozkatz100
So how did we get here?
So how did we/I get here?

running my untested code directly on prod

Data Lakehouse™
Code: It broken? Roll back the last version

Data: Best effort

Code: CI/CD in place. Promote from UAT to Production

Data: Hope

STAND BACK
WE'RE TRYING THIS IN PRODUCTION
But data isn’t code
Hi👋

Lottie
CCO & spirit animal @ lakeFS

github.com/treeverse/lakeFS

@lakeFS
Object Store

- lakeFS
- lakeFS
- s3://data-repo/collections/foo
- s3://data-repo/main/collections/foo

- Object Store
- Google cloud storage
- Amazon S3
- Azure blob storage

- lakectl branch create
- "lakefs://data-repo@my-experiment"
- --source "lakefs://data-repo/main"

// output:
// created branch 'my-experiment',
// pointing to commit ID: 'd1e9adc71c10a'
lakeFS Capabilities

Development

- **Branches**
  - **Experimentation** - Try tools and code in isolation
  - **Debug** - Create an isolated snapshot of the data at the time of the failure
  - **Collaborate** - Tools, code or different versions of your data

Deployment

- **Merges & Hooks**
  - **Version control** - point data consumers to newly deployed data.
  - **Best Practices & Data Quality** Enforced by pre-merge hooks

Production

- **Commits & Reverts**
  - **Roll Back** - Recover from errors by instantly reverting data to a former, consistent snapshot of the data lake.
  - **Troubleshoot** - Investigate production errors by starting with a snapshot of the inputs to the failed process
How Does lakeFS Work?

Create an isolated environment WITHOUT copying the data
So, Dev/Stage/Prod?
Sandboxed Pipelines

main (Stage)

sandbox (Pipeline Execution)

- ✓ validate_schema
- ✓ foreign_key_check
- ✓ dod_vol_threshold
Sandboxed Pipelines
Sandboxed Pipelines

- Why
- Who
- When
- How
- What
Sandboxed Pipelines

https://github.com/treeverse/lakeFS-samples/

This sample repository captures a collection of notebooks, dockerized applications and code snippets that demonstrate how to use lakeFS.

lakeFS is a popular open-source solution for managing data. It provides a consistent and scalable data management layer on top of cloud storage, such as Amazon S3, Azure Blob Storage, or Google Cloud Storage. It allows users to create and manage data in a version-controlled and immutable manner; and offers features such as data governance, data lineage, and data access controls. lakeFS is compatible with a wide range of data processing frameworks and tools.

Let’s Get Started 🐠

Clone this repository

```
$ git clone https://github.com/treeverse/lakeFS-samples.git
$ cd lakeFS-samples
```

You now have two options:

Run a Notebook server with your existing lakeFS Server
T-3h
Sandboxed Pipelines

main (Stage)

sandbox (Pipeline Execution)

- validate_schema
- foreign_key_check
- !dod_vol_threshold

<
Sandboxed Pipelines

- **main** (Stage)
- **sandbox** (Pipeline Execution)
lakeFS Community

Trusted by more than **1K Companies**

Liked by more than **4K Members**

[lakefs.io/slack]