

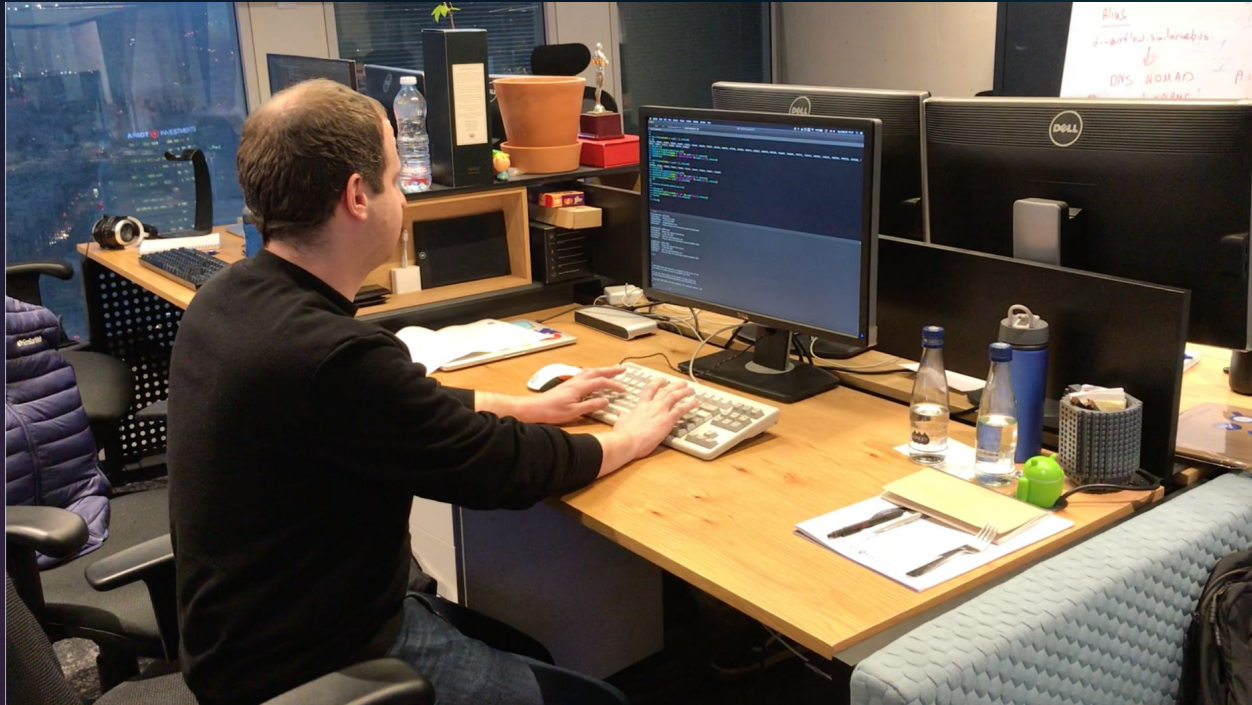
# Dev/Stage/Prod is an Anti-Pattern for Data Pipelines

---

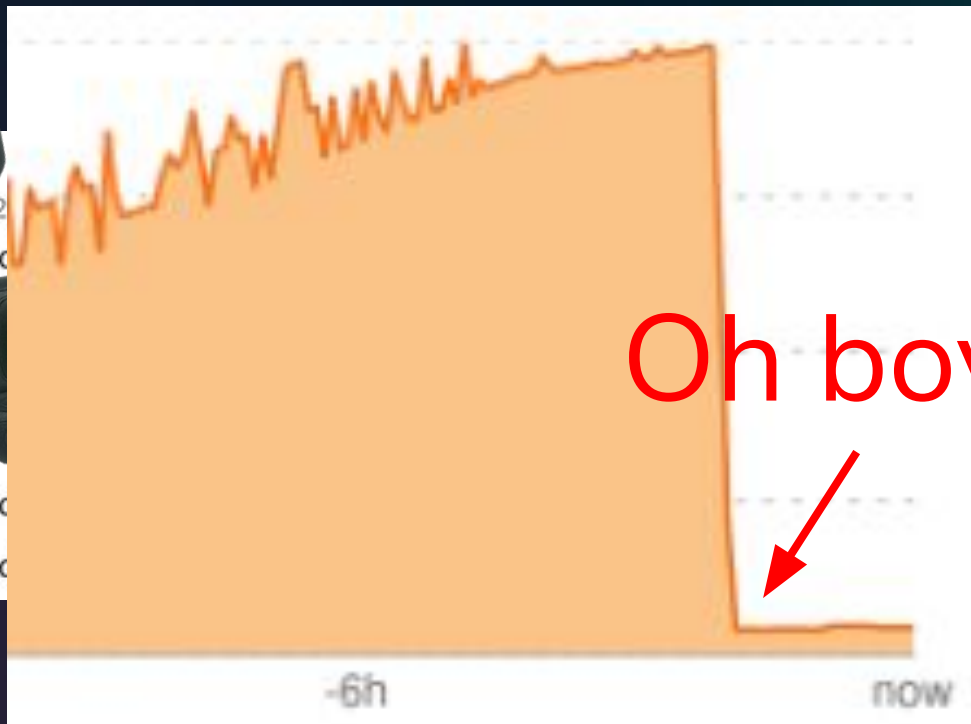
Oz Katz, March 2024



# T-3h

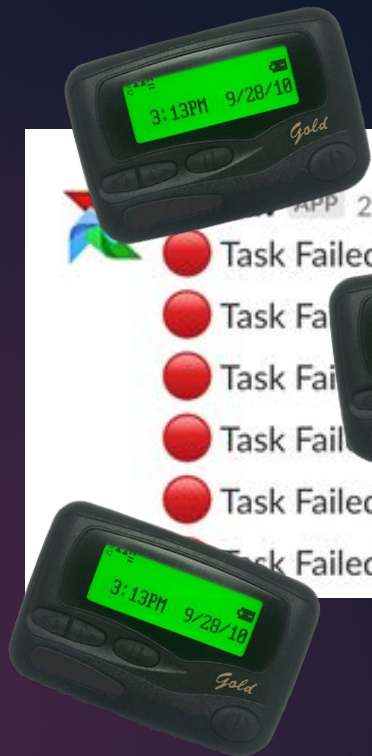


# T+1m



Oh boy

```
166572+00:00 [failed]>  
303724+00:00 [failed]>  
287093+00:00 [failed]>  
148914+00:00 [failed]>  
[failed]>  
[failed]>
```



T+12h





T+2W



Hi 🖐️



Oz Katz  
CTO & Co-Creator of lakeFS



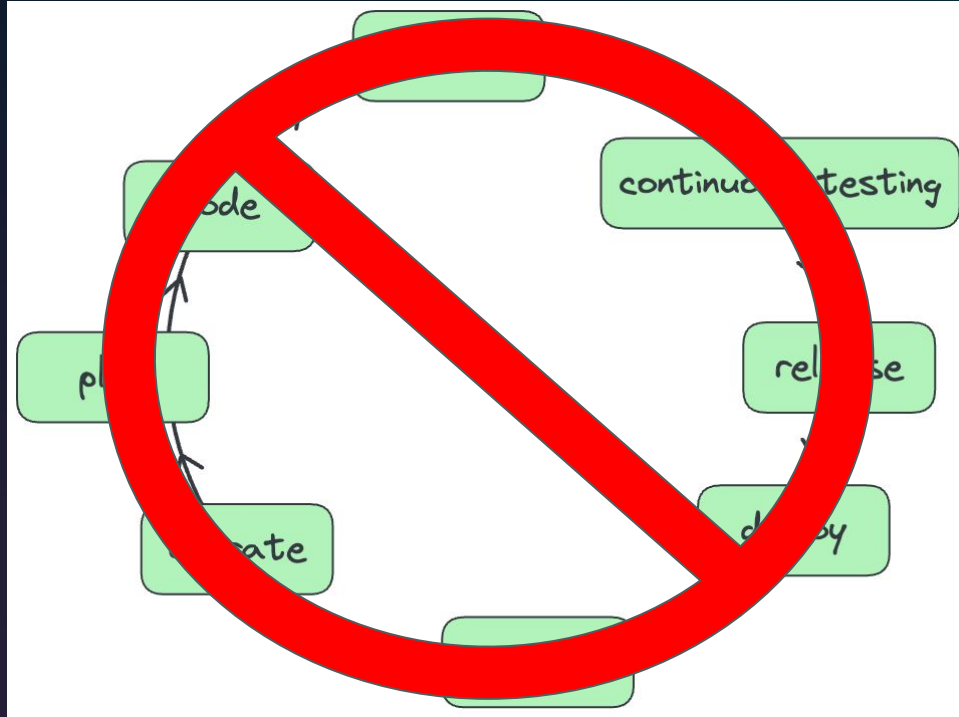
[github.com/ozkatz](https://github.com/ozkatz)



[@ozkatz100](https://twitter.com/ozkatz100)



So how did ~~we~~ I get here?



So how did ~~we~~ I get here?



running my  
untested code  
directly on prod

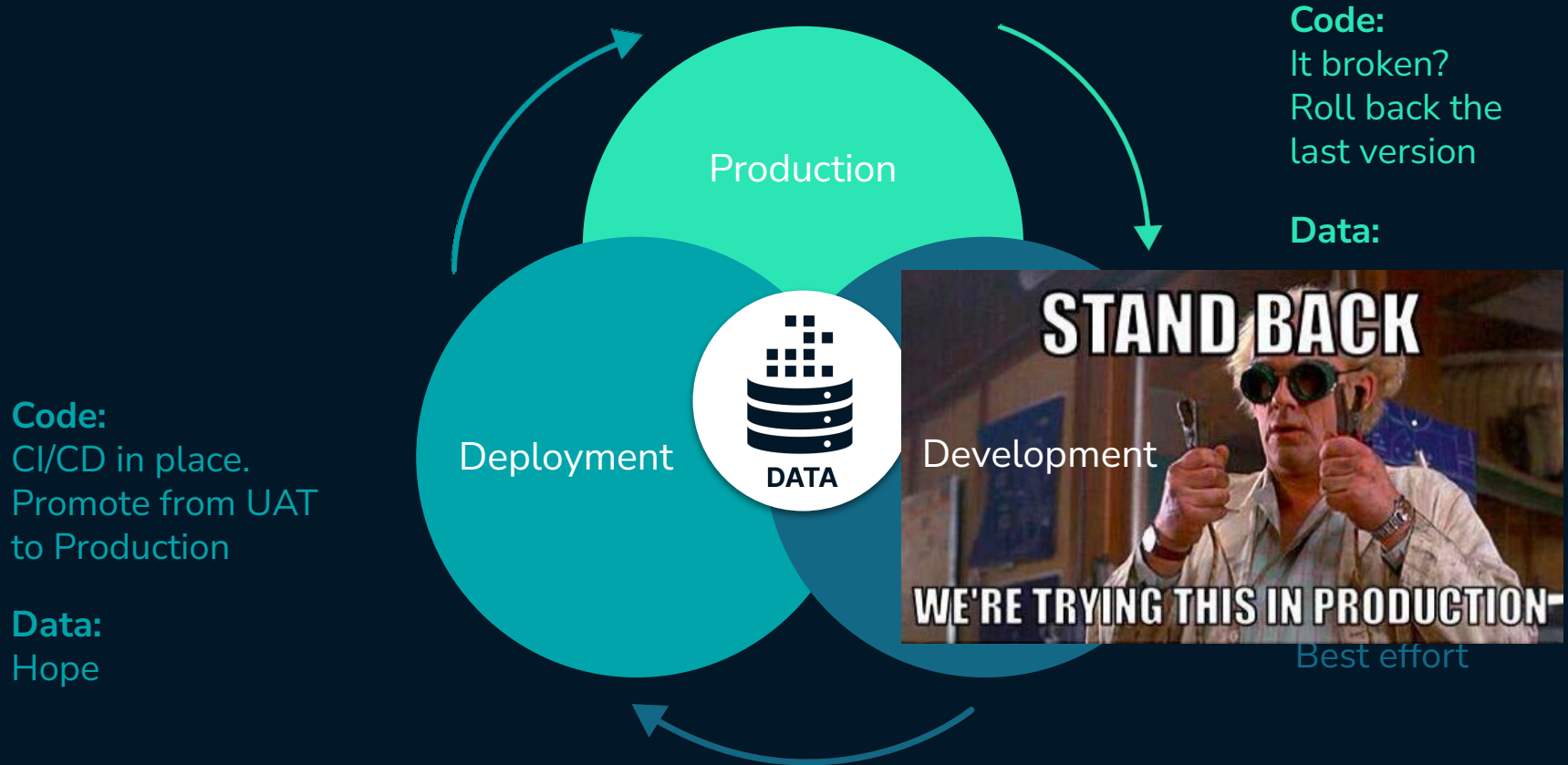


Data  
Lakehouse™

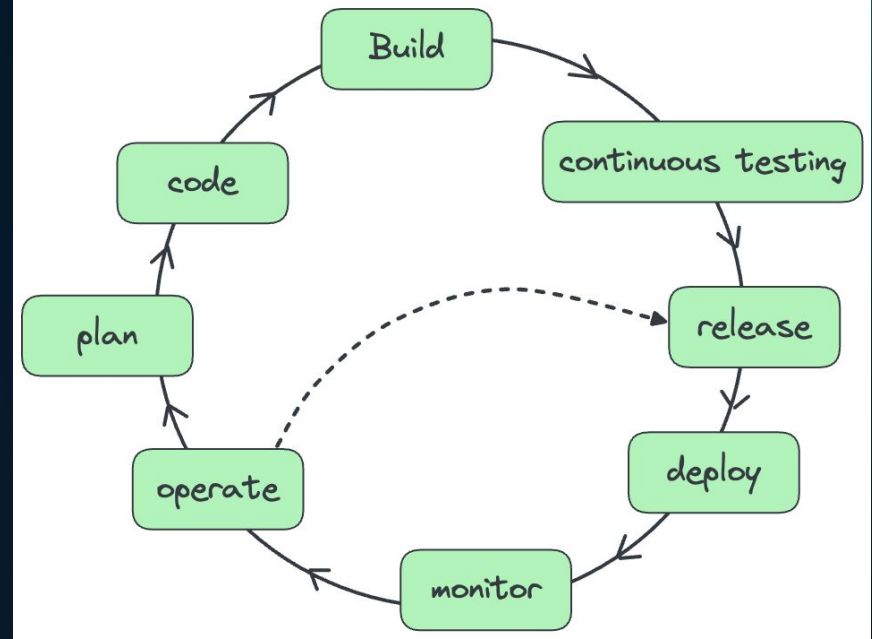




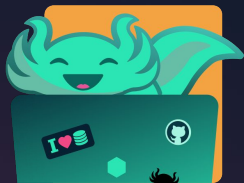
# Code vs. Data



# But data isn't code



Hi 🖐️



Lottie

CCO & spirit animal @ lakeFS

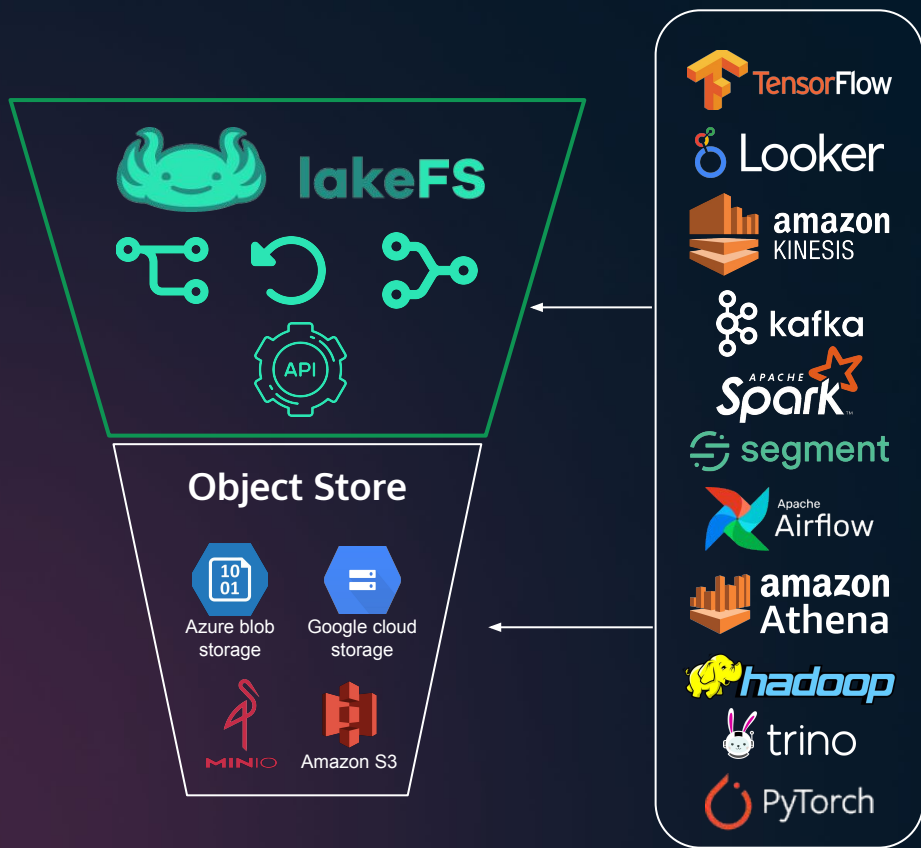


[github.com/treeverse/lakeFS](https://github.com/treeverse/lakeFS)



@lakeFS





s3://data-repo/collections/foo



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

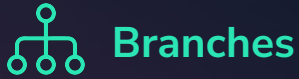
```
lakectl branch create \  
  "lakefs://data-repo@my-experiment" \  
  --source "lakefs://data-repo/main"
```

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

# lakeFS Capabilities

## Development

---



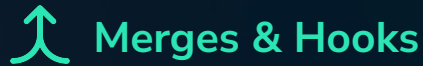
**Experimentation** - Try tools and code in isolation

**Debug** - Create an isolated snapshot of the data at the time of the failure

**Collaborate** - Tools, code or different versions of your data

## Deployment

---



**Version control** - point data consumers to newly deployed data.

**Best Practices & Data Quality** Enforced by pre-merge hooks

## Production

---

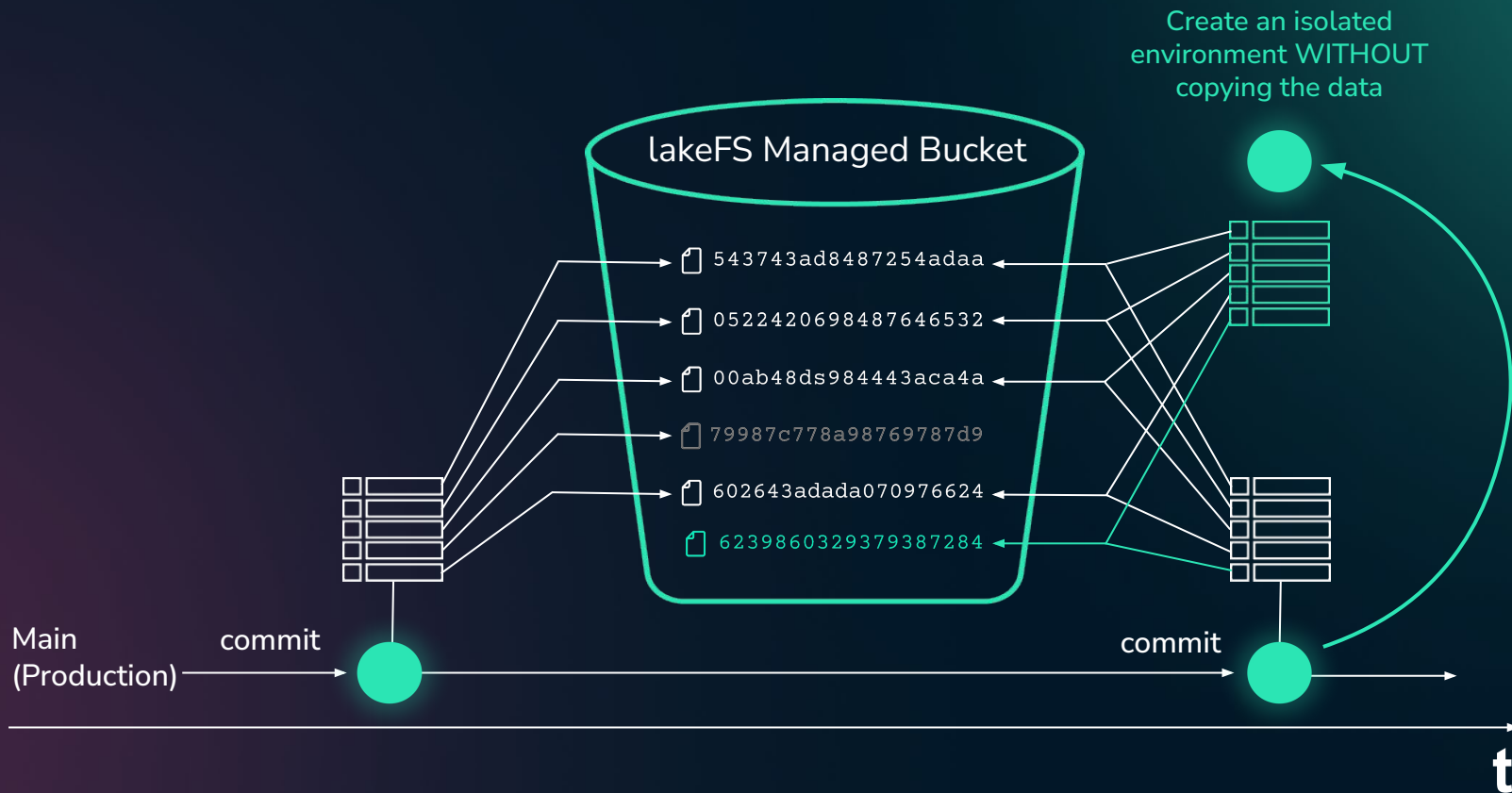


**Roll Back** - Recover from errors by instantly reverting data to a former, consistent snapshot of the data lake.

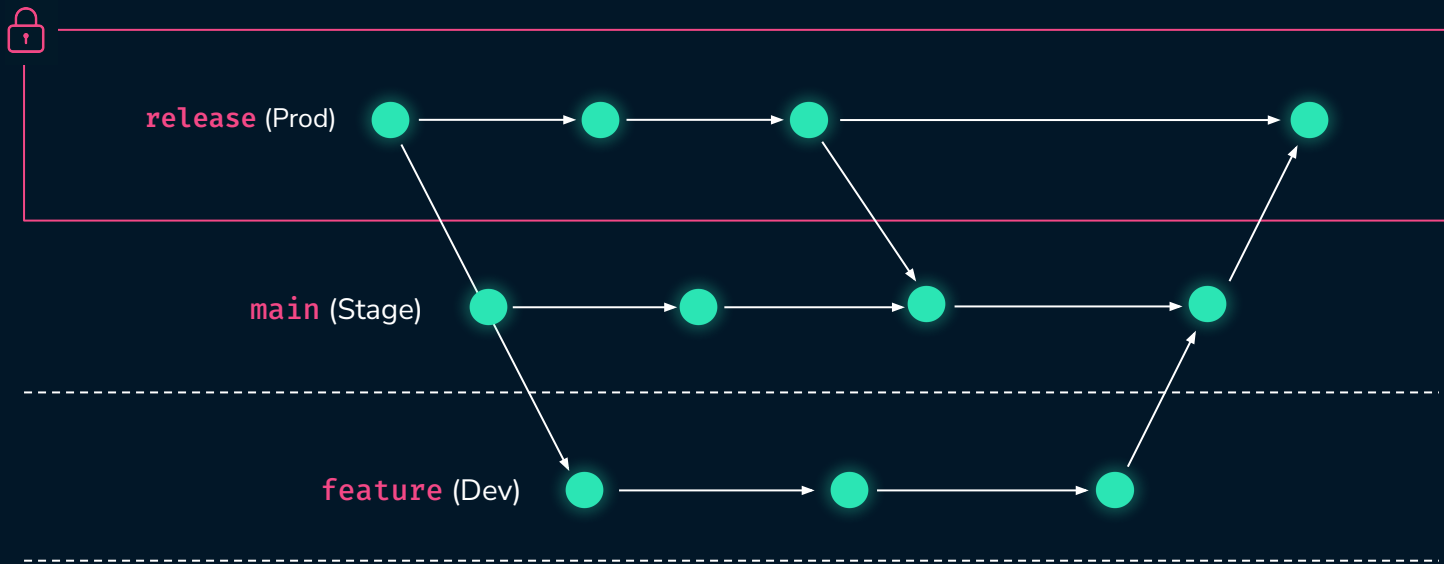
**Troubleshoot** - Investigate production errors by starting with a snapshot of the inputs to the failed process



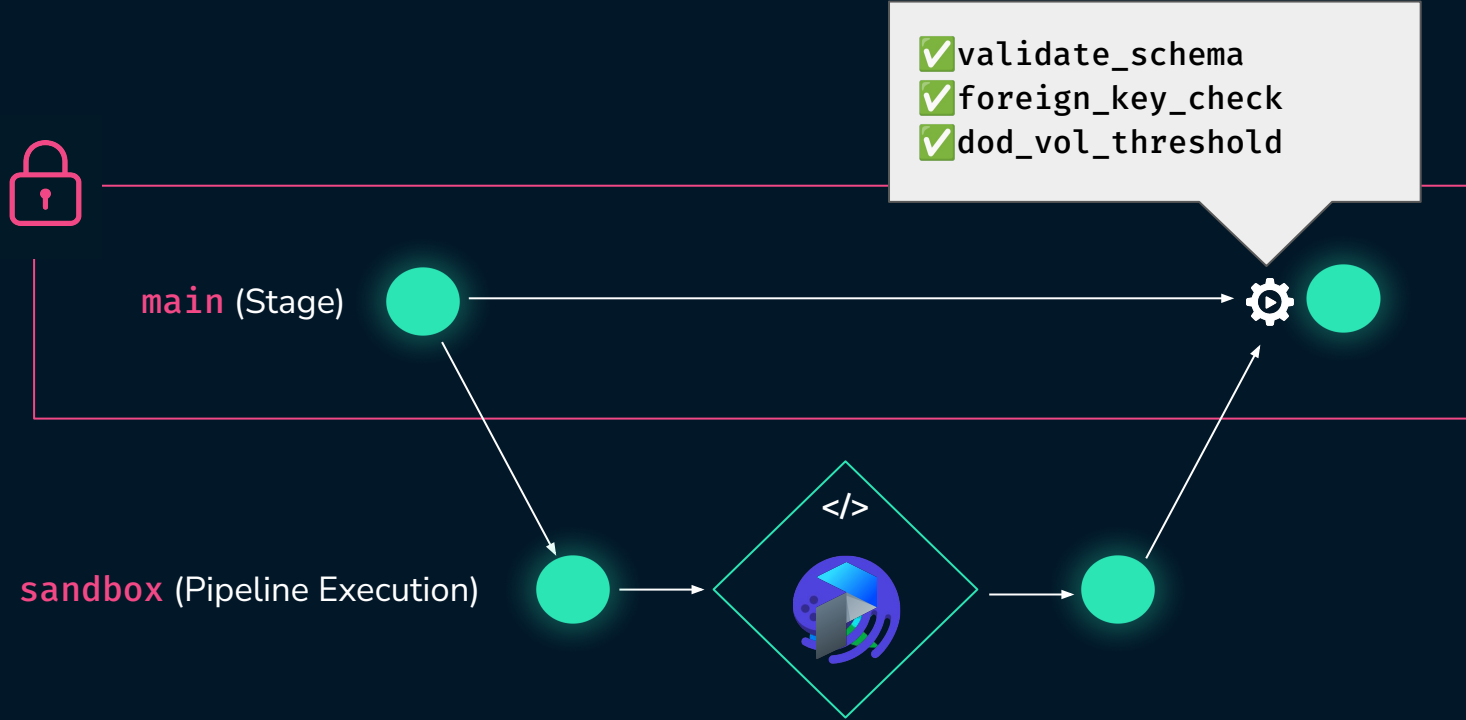
# How Does lakeFS Work?



# So, Dev/Stage/Prod?



# Sandboxed Pipelines



# Sandboxed Pipelines

The screenshot displays the Apache Airflow web interface for a DAG named `lakeFS_workflow`. The interface is in a dark theme and shows the DAG is currently **running**. The top navigation bar includes links for DAGs, Security, Browse, Admin, Docs, and Astronomer, along with the current time (17:38 UTC) and user initials (AU).

The DAG details section shows the DAG is in **Graph View**. Below this, there are several filter and control elements:

- DAG Docs**: A link to view the DAG's documentation.
- Filter Bar**: Includes a date selector (2021-05-24T17:37:17Z), a runs count (25), a run ID dropdown (manual\_\_2021-05-24T17:37:16.931617+00:00), and a search box (Find Task...).
- Layout**: A dropdown menu set to "Left > Right" and an "Update" button.
- Task Status Legend**: A row of colored boxes representing task statuses: queued (orange), running (green), success (green), failed (red), up\_for\_retry (blue), up\_for\_reschedule (blue), upstream\_failed (orange), skipped (orange), scheduled (orange), and no\_status (grey).
- Task Operators**: A row of colored boxes representing task operators: CommitOperator (orange), CreateBranchOperator (orange), FileSensor (grey), and MergeOperator (orange).

The DAG graph shows four tasks in a linear sequence:

```
graph LR; create_branch[create_branch] --> spark_submit[spark_submit]; spark_submit --> commit[commit]; commit --> merge_branches[merge_branches];
```

The `spark_submit` task is highlighted in green, indicating it is the current task being executed. The `create_branch` task is highlighted in orange, indicating it is a CommitOperator. The `commit` and `merge_branches` tasks are also highlighted in orange, indicating they are MergeOperators.

At the bottom right, there is an "Auto-refresh" toggle switch and a refresh icon.

# Sandboxed Pipelines

why

who

when

how

what

The screenshot shows the lakeFS web interface. At the top, there are navigation tabs: **Objects**, **Uncommitted Changes**, **Commits** (selected), **Branches**, **Tags**, **Compare**, **Actions**, and **Settings**. The main content area displays a commit titled "Daily Aggregations pipeline for 2024-03-14".

Commit details:

- ID:** [1e09b09af32333fa18f3d0e9e0370e53b98a776d9080503af80808a4b729dfa3](#)
- Committer:** oz.katz@treeverse.io
- Creation Date:** 03/11/2024 10:46:52 (3 days ago)
- Parents:** [4fbaa06b207efaa3a1ec44f3668c3ee995e923953edef0c6c8fa5926f1968381](#)

There is a green button labeled "Open Airflow UI".

Metadata table:

Metadata Key	Value
::lakefs::Airflow::url[url:ui]	http://127.0.0.1:3000/runs/834a7ba7-b399-4e10-a4a5-c5aa99578cca
airflow_run_id	834a7ba7-b399-4e10-a4a5-c5aa99578cca

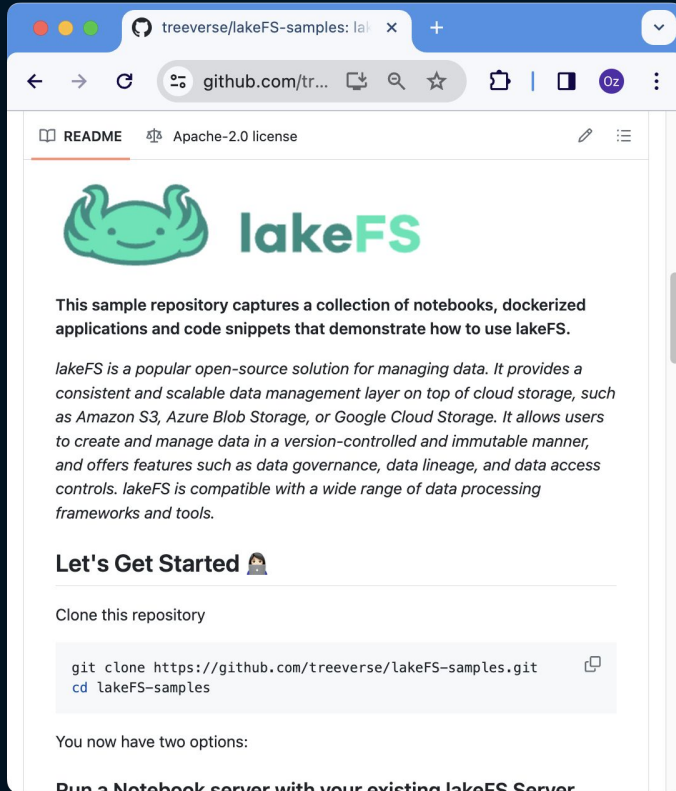
Below the metadata, it says "Showing changes for commit 1e09b09af323".

File tree view for commit 1e09b09af323:

- [daily\\_aggregations/](#) 34 [Hide change summary](#)
- [clean/](#) [Calculate change summary](#)
- [partitioned/](#) [Calculate change summary](#)



# Sandboxed Pipelines




The screenshot shows a web browser displaying the GitHub repository page for 'treeverse/lakeFS-samples'. The page features the lakeFS logo, a description of the repository, and instructions on how to clone it. The browser's address bar shows the URL 'github.com/tr...'. The repository name 'lakeFS' is prominently displayed in a large, green font. Below the logo, there is a paragraph describing the repository's purpose and a section titled 'Let's Get Started' with a 'Clone this repository' button. A code block contains the following commands: 

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

treeverse/lakeFS-samples: lakeFS-samples

github.com/tr...

README Apache-2.0 license



**lakeFS**

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
```

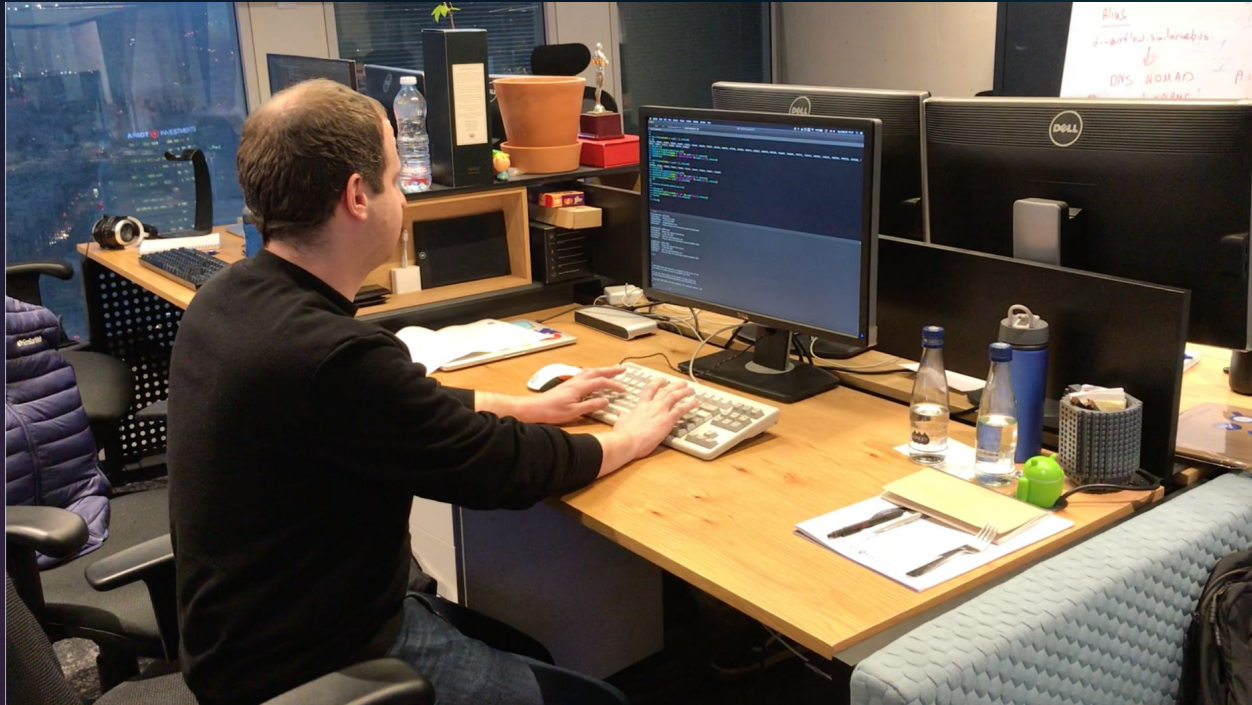
You now have two options:

Run a Notebook server with your existing lakeFS Server

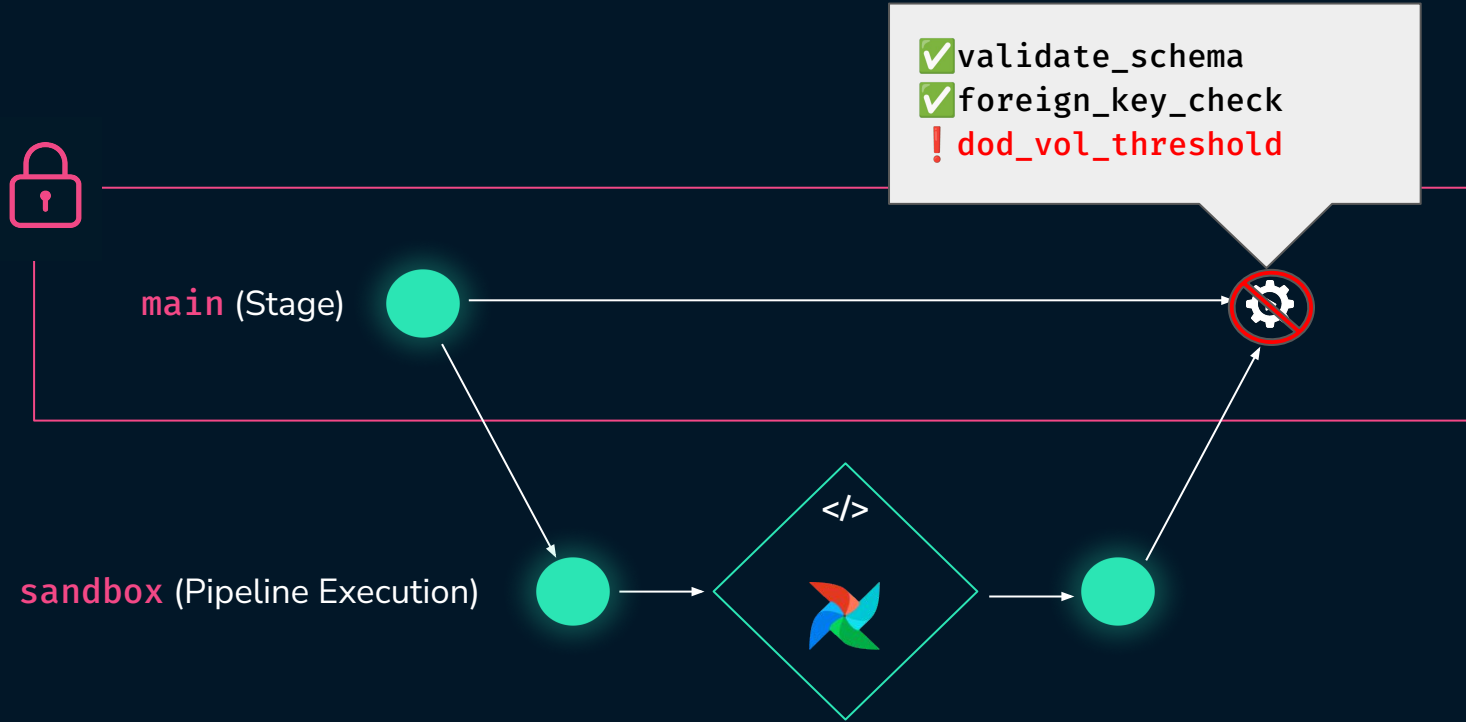


<https://github.com/treeverse/lakeFS-samples/>

# T-3h



# Sandboxed Pipelines



# Sandboxed Pipelines



# lakeFS Community



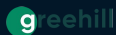
Trusted by more than **1K Companies**



Liked by more than **4K Members**



ER  
BEDROCK



greenhill



Context Labs



AXTRIA

KARIUS



Terramera



VOLVO

BAE SYSTEMS

enigma



N

WINDWARD°



Paige



Proton Mail



PETRONAS

DAIMLER



EPCOR



Tredence  
Connect the Dots

LOCKHEED MARTIN

STATE STREET GLOBAL  
ADVISORS

CONNOR, CLARK & LUNN  
FINANCIAL GROUP



GENERALI

pollinate

lakefs.io/slack

