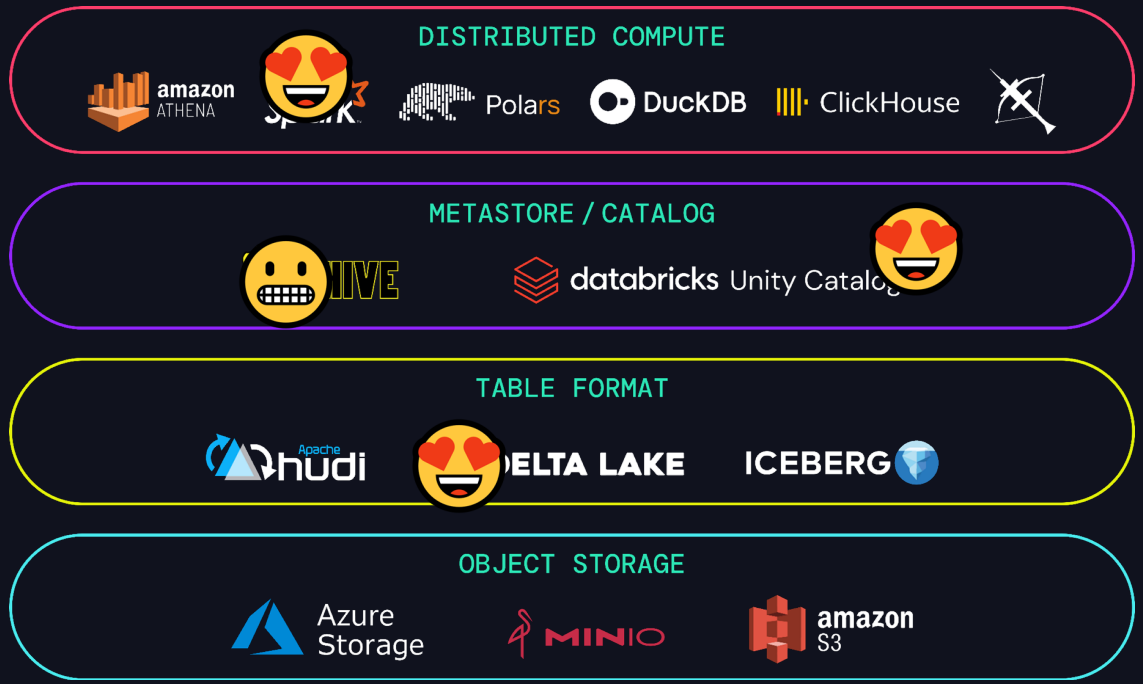
DATA+AI SUMMIT BY databricks

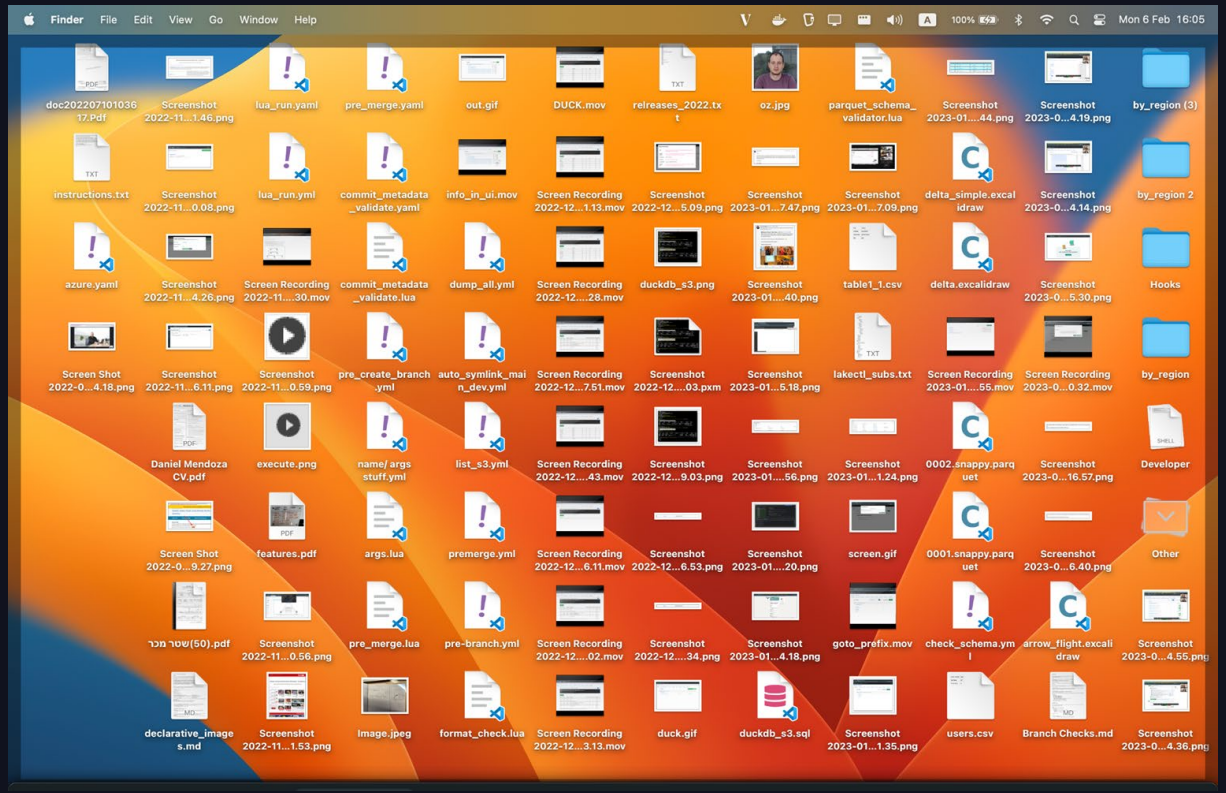
Why Version Control Is Essential For Your Lakehouse Architecture

Oz Katz, June 2024

DATA LAKEHOUSE

Humans, organizing things as humans tend to do.

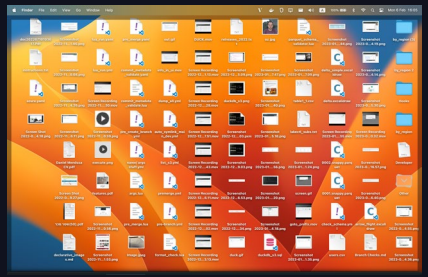




YES,
THIS IS MY
ACTUAL
DESKTOP

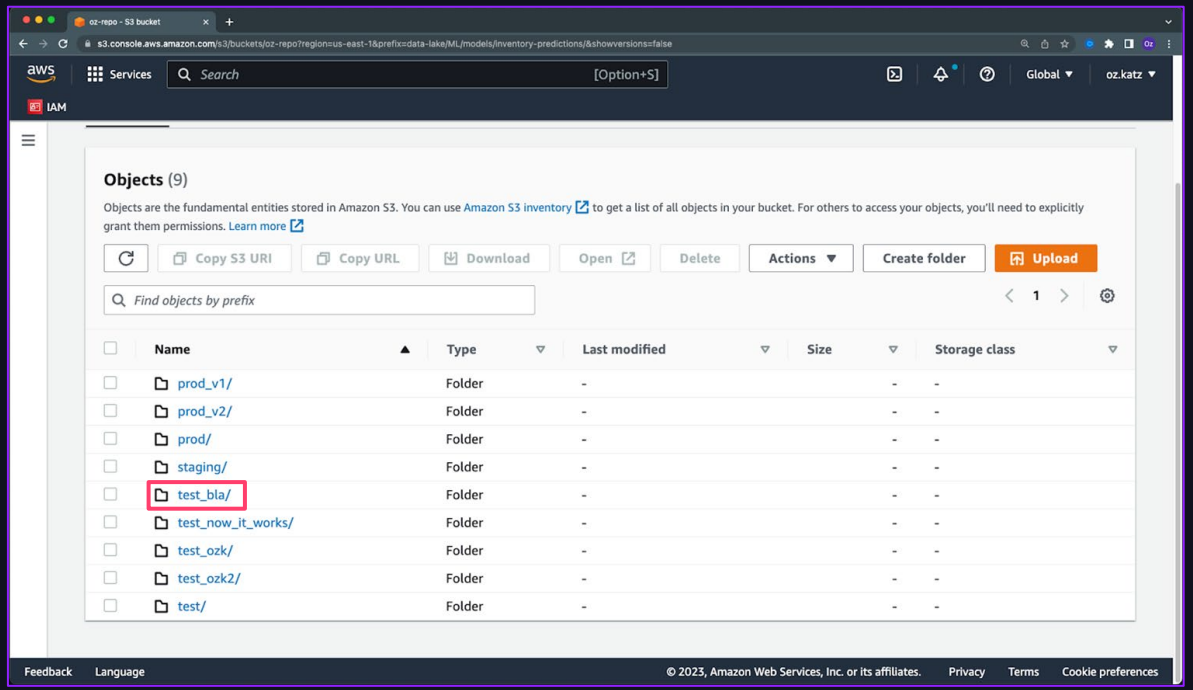
HUMANS

ARE CREATIVE, MESSY

```
mess =  * len(engineers) * len(data_scientists)²
```

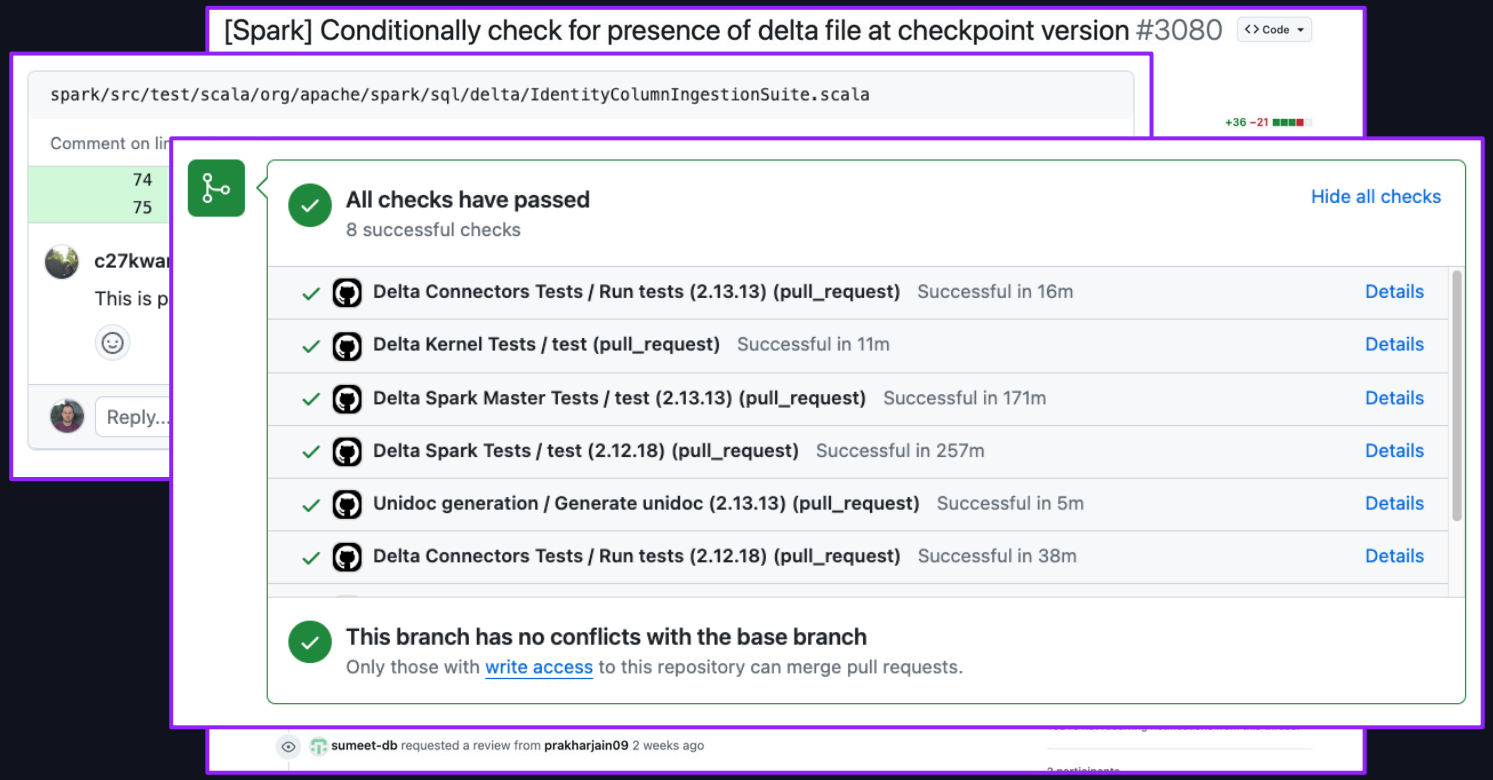
HUMANS

NAMING THINGS



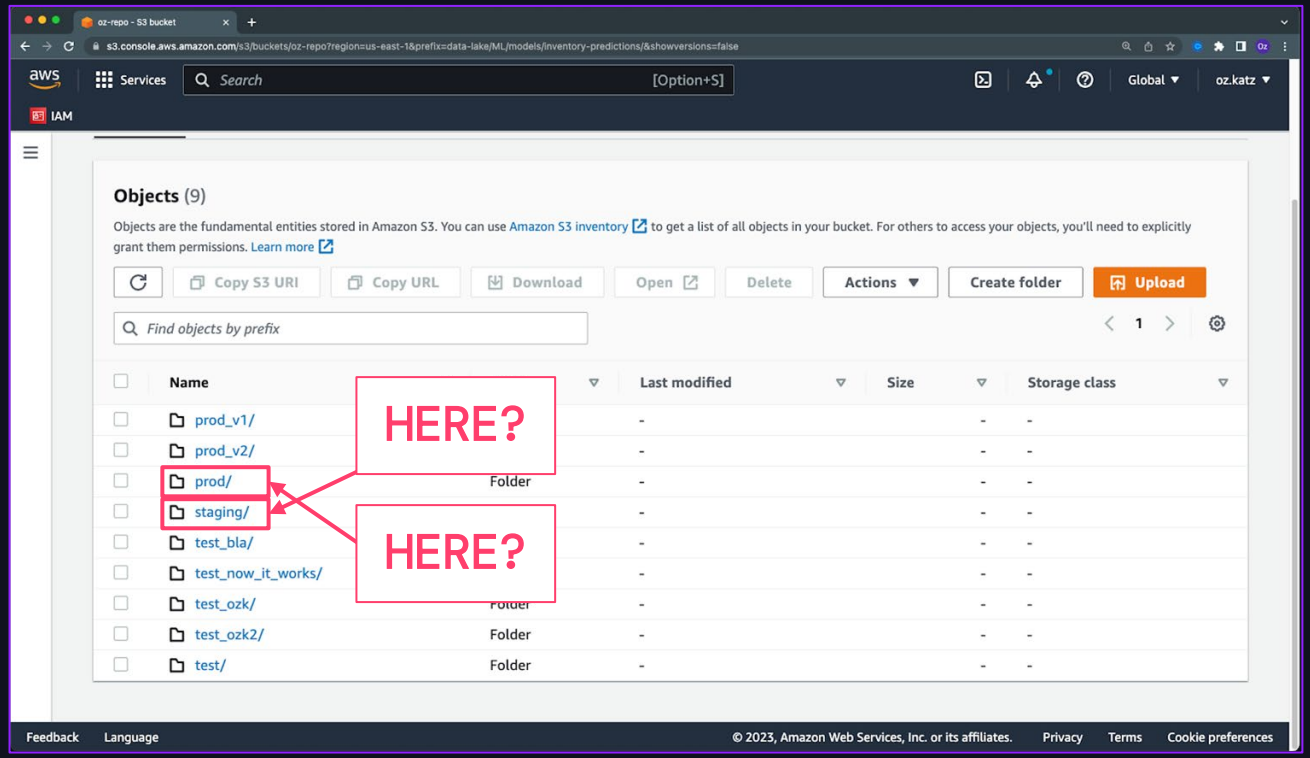
YOUR CODE (ANOTHER HUMAN ARTIFACT)

IS MUCH BETTER, I ASSURE YOU



BUT DATA IS HARDER

WHERE DO I RUN TESTS?



8

WHEN DO I RUN TESTS?

(SPOILER: TOO LATE)

Our ETL starts

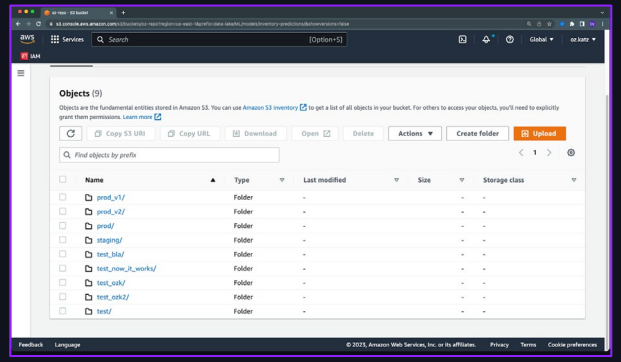
Spark writes
a bunch of data
a bunch of table

find out there's
oblem! The tests
did their job!

But it's too late and someone's jobs have already completed

WHAT IF…

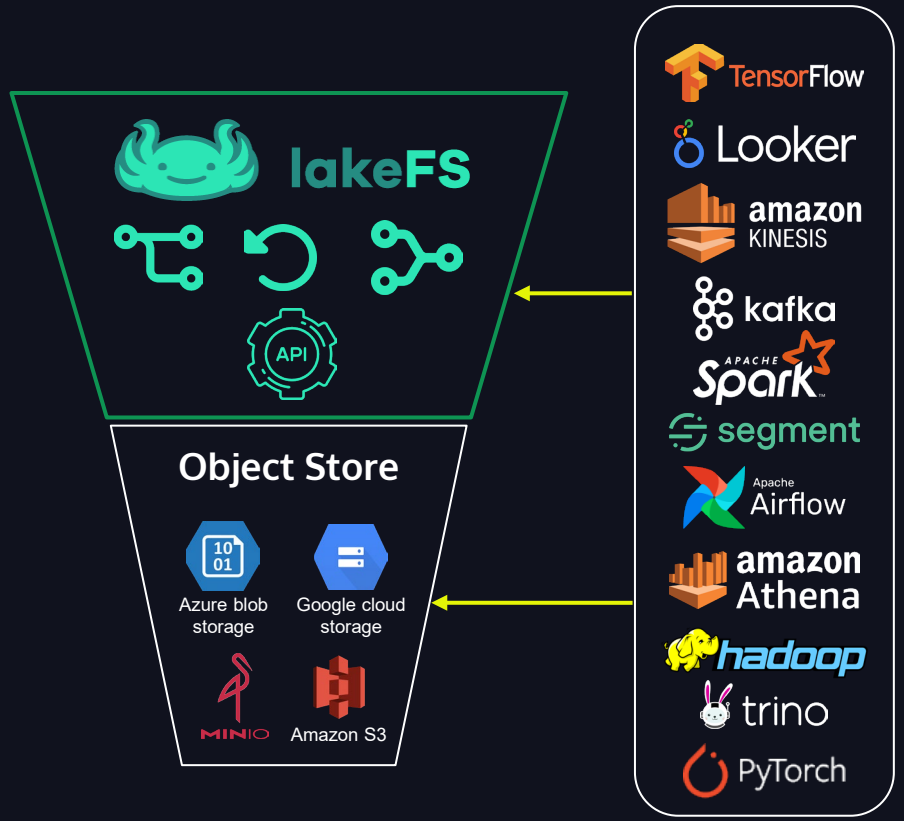
WE COULD TO DO THIS:



+

=

WE CAN!



s3://data-repo/collections/foo

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

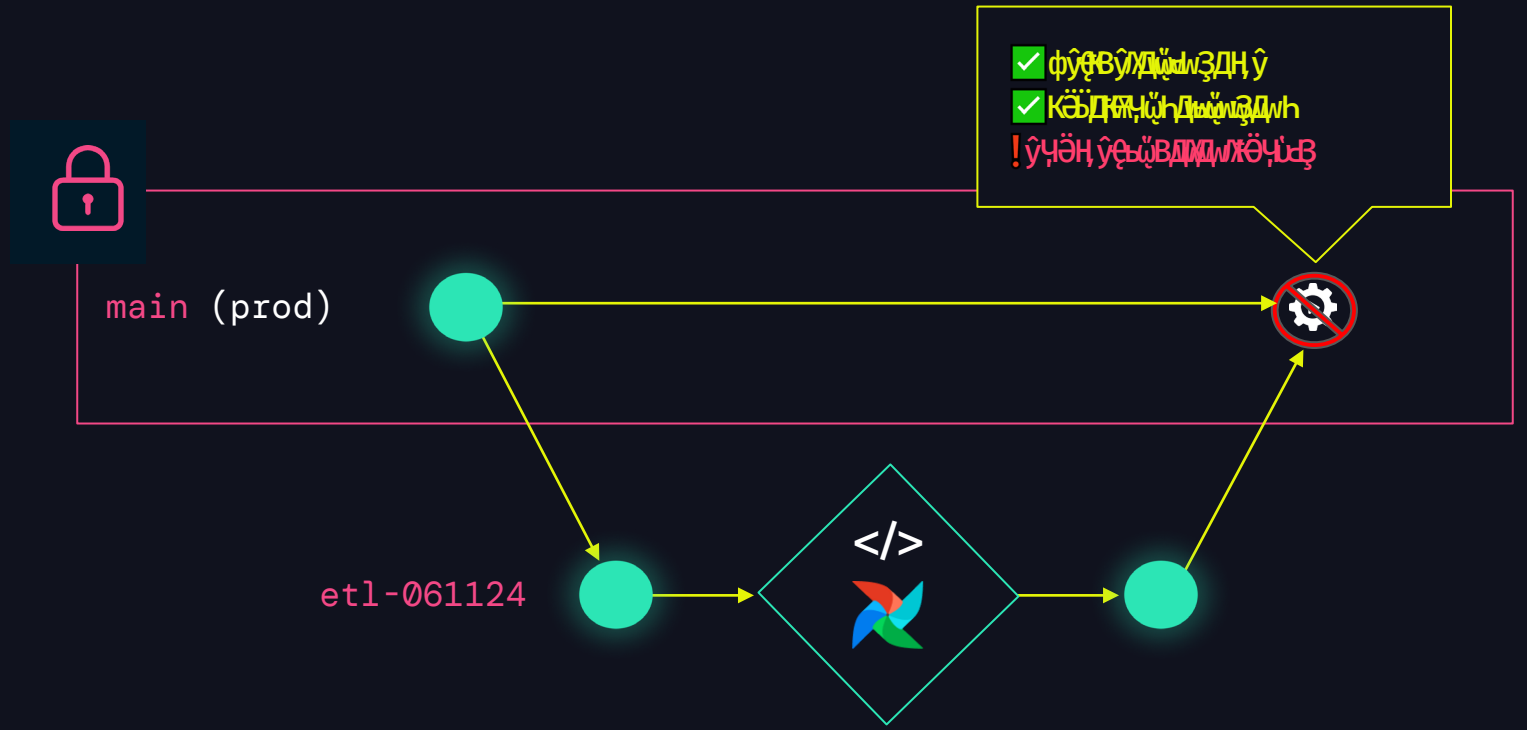
```
lakectl branch create \
      "lakefs://data-repo@my-experiment" \
      --source "lakefs://data-repo/main"


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

WE CAN!

https://docs.lakefs.io/integrations/spark.html

WHERE & WHEN DO I RUN TESTS?

ON A BRANCH, BEFORE MERGING



WHERE DO I RUN TESTS?

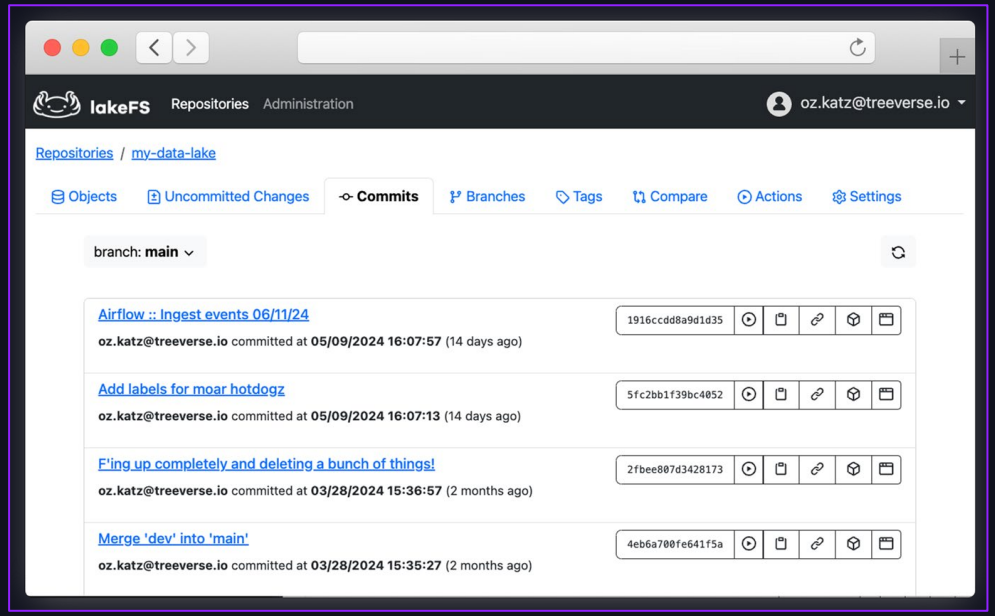
ON A BRANCH, BEFORE MERGING

PYSPARK

```python
repo = lakefs.repository('my-data-lake')
branch = repo.branch('etl-061124').create(source_reference='main')

# write
events_path = f'lakefs://my-data-lake/{branch.id}/bronze/events/'
dt = DeltaTable.forPath(spark, events_path)
dt.update(condition=expr('id > 100'), set={'id': expr('id % 100')})
…

# audit(!!)
run_checks(events_path)

# publish
branch.commit(message='Events Transformed!')
branch.merge_into('main')  # ATOMIC & WILL FAIL IF TESTS DIDN'T PASS
```

COOL! WHAT ELSE CAN IT DO?

LOTS OF GIT-Y STUFF



- **Reproducibility** - using immutable commits
- **Rollback to last known good state** - lowering cost of mistakes in prod
- **Multiple dev environments** - without copying data, using copy-on-write branching

THANK YOU

Oz & your production DAGs

LEARN MORE

lakefs.io/slack

github.com/treeverse/lakeFS

docs.lakefs.io/quickstart