DATA+AI
SUMMIT 2022

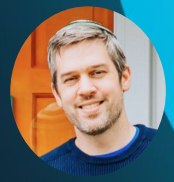
The Databricks Notebook

Front door to the Lakehouse

ORGANIZED BY databricks

Austin Ford

Senior Product Manager, Databricks

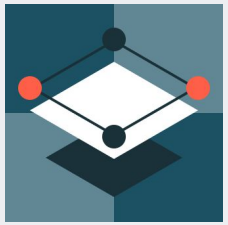


Rafi Kurlansik

Lead Product Specialist, Databricks

Why is becoming data-driven such a challenge for companies?

There are major hurdles in the way..



Data is spread out across many sources

Making data easily available is complex

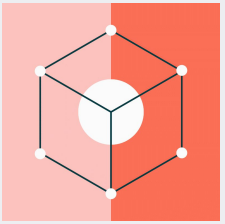
- Can have many data types: tabular data, text blobs, images, streaming logs, etc.
- Can be stored across clouds
- Each source has different semantics, access controls, and governance patterns

Disparate tools limit efficiency, reproducibility, and scale

Data is just the starting point; getting value out is its own challenge

- Users can be siloed, have different skill sets, and use different tools
- Results aren't reproducible and don't reach the right audience
- Getting to production takes forever

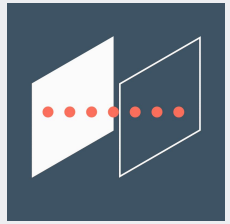
But—hurdles are meant to be jumped!



The Databricks Lakehouse unifies all data in one place, regardless of type or source

Companies can leave data where it is and still..

- Make existing data lakes and warehouses accessible, governable, and secure
- Leverage an open ecosystem
- Unify all use cases (DS, ML, BI, etc.) on top

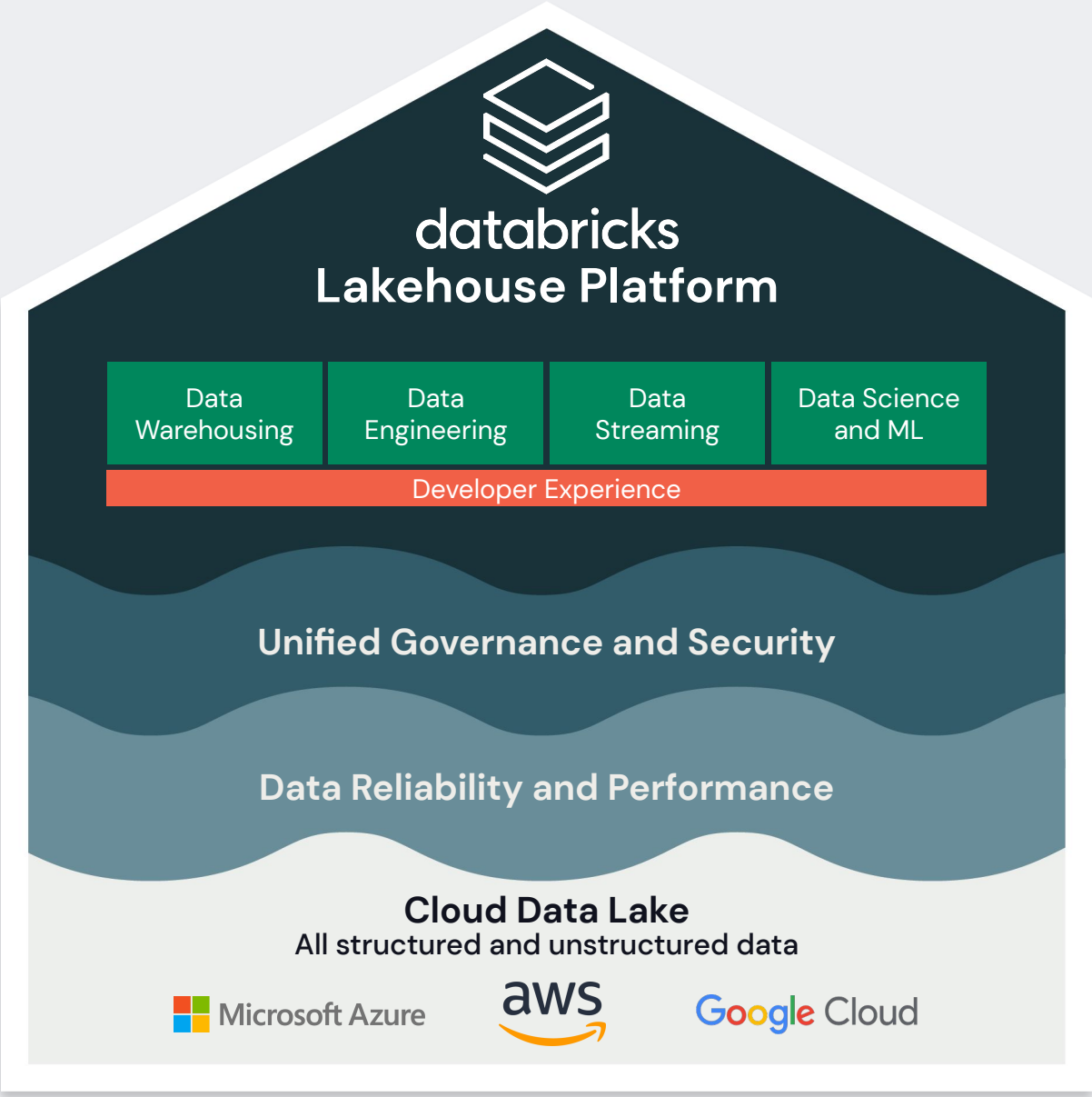


The Databricks Lakehouse developer experience accelerates the journey to insights

Users get a powerful developer platform in which to derive insights and deliver value

- A collaborative **Notebook** that enables efficient data analysis, insights sharing, and faster paths to production
- Support for all popular developer tools

Databricks Lakehouse is the foundation for **all your data analytics needs**



Let’s talk about the Notebook—

the front door to the Lakehouse

Our vision for the Notebook

The front door to the Lakehouse

World-class data native developer experience

- Shortened distance to data insights, especially with Python and SQL
- Accelerated development
- Easy paths to production
- All the usual programming ergonomics

Foundation for sharing and consuming insights

- Effortless sharing of notebooks and results to anyone in your organization
- Optimized experiences for viewing and evaluating data assets

Portal to the entire Databricks Lakehouse

Connections to all the platform's powerful capabilities at the time and in the place users need, for..

- data engineering
- data science
- machine learning
- and more!

The Databricks Notebook

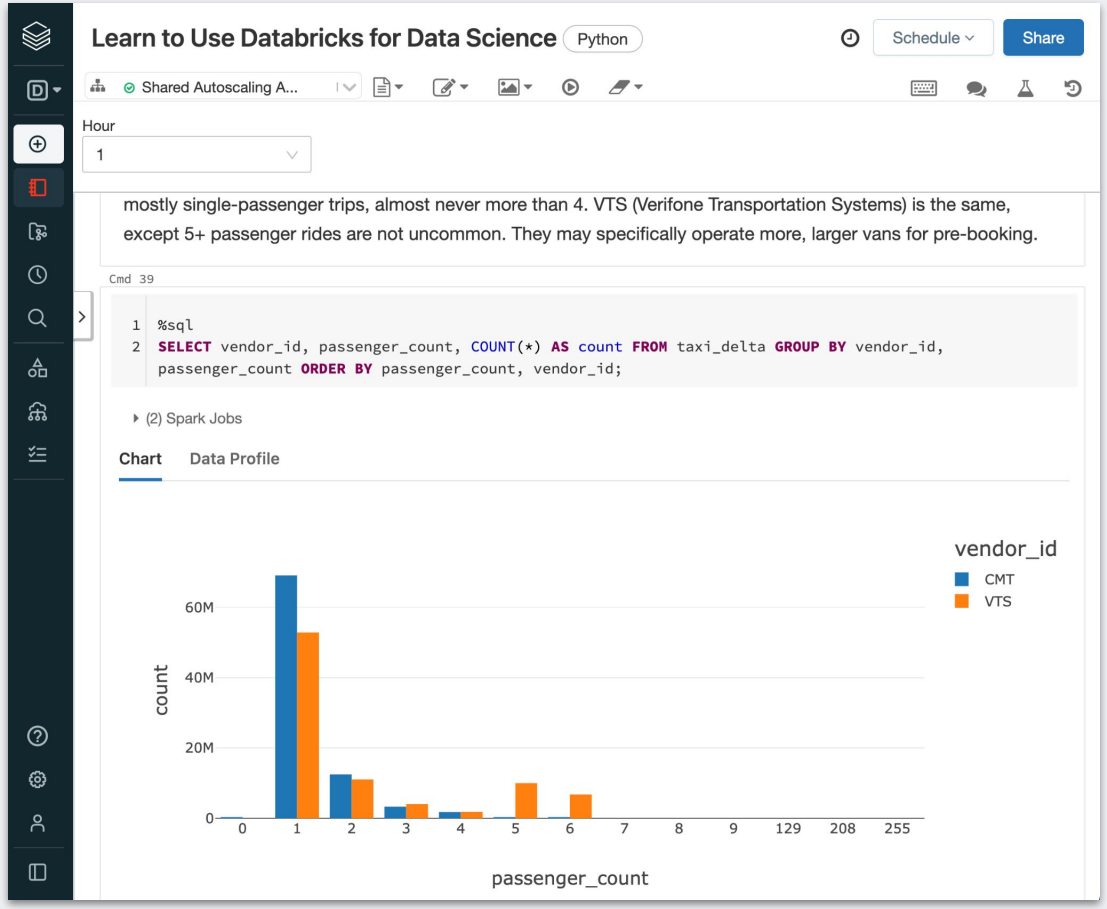
Through the front door of the Lakehouse

Multi-language
Use Python, SQL, Scala, and R, all in one Notebook

Collaborative
Real-time co-presence, co-editing, and commenting

Jupyter-compatible
Use the power of the Jupyter ecosystem in the Notebook

Ideal for exploration
Explore, visualize, and summarize data with built-in charts and data profiles



Adaptable
Install standard libraries and use local modules

Reproducible
Automatically track version history, and use git version control with Repos

Get to production faster
Quickly schedule notebooks as jobs or create dashboards from their results, all in the Notebook

Enterprise-ready
Enterprise-grade access controls, identity management, and auditability

Multi-language notebooks

Use the right tools for the job

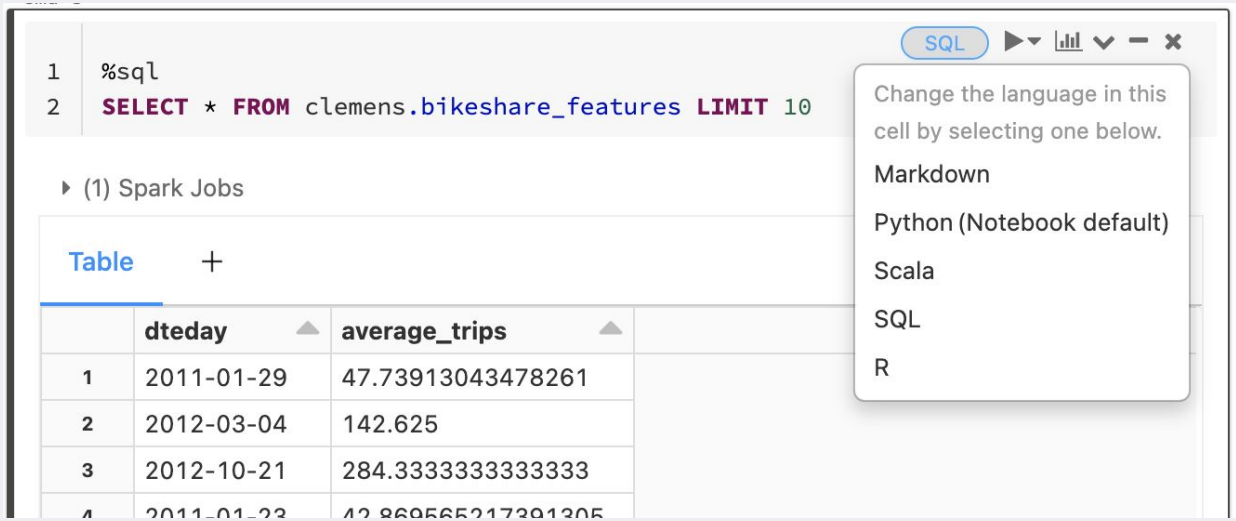
Users can mix and match languages based on their use case and preferred workflow, choosing from **Python**, **SQL**, **Scala**, and **R**

New!

Users can access the output of a SQL cell as a Python dataframe

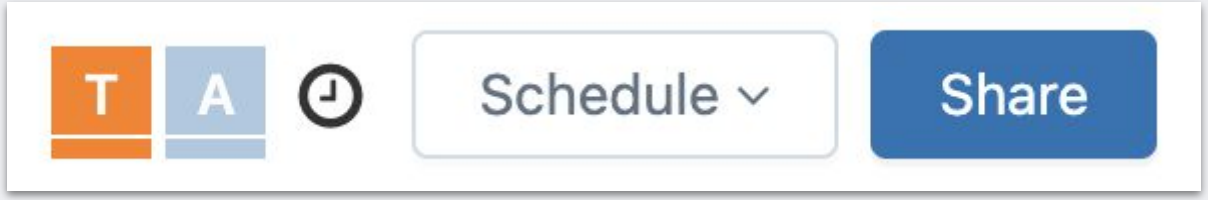
Coming Soon

Users can re-style their Python code using the black formatter



Collaborate in real-time

Data science is a team sport




```
df_new = utils.scrub(df, drop="num_columns")
```

Users can share the same view of a notebook with **co-presence**

Users can **co-edit** in real-time with their colleagues to jointly iterate, debug, and more

Comments enable users to alert their colleagues to action items or interesting findings

Jupyter-compatible

Bringing the power of the Jupyter ecosystem to the Databricks Notebook

Newly GA!

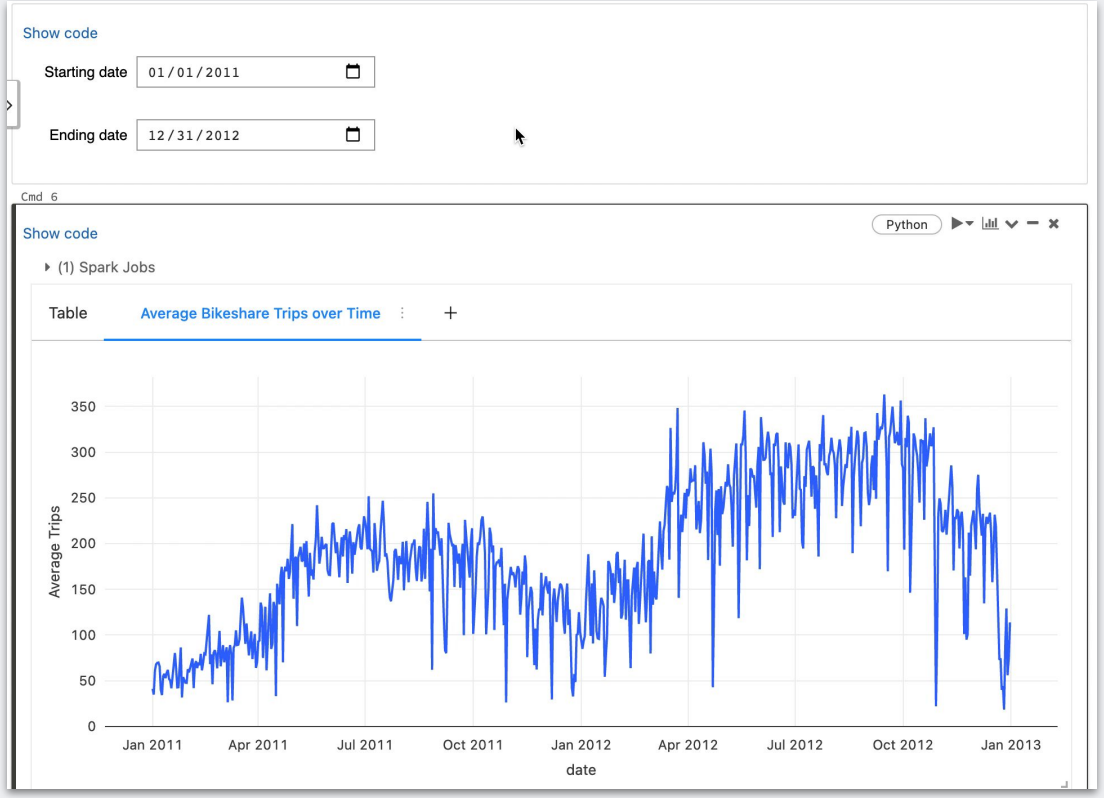
The Databricks Notebook uses the **IPython kernel** to power Python cell execution

Public Preview!

Use **ipywidgets** to turn notebooks into powerful, interactive apps

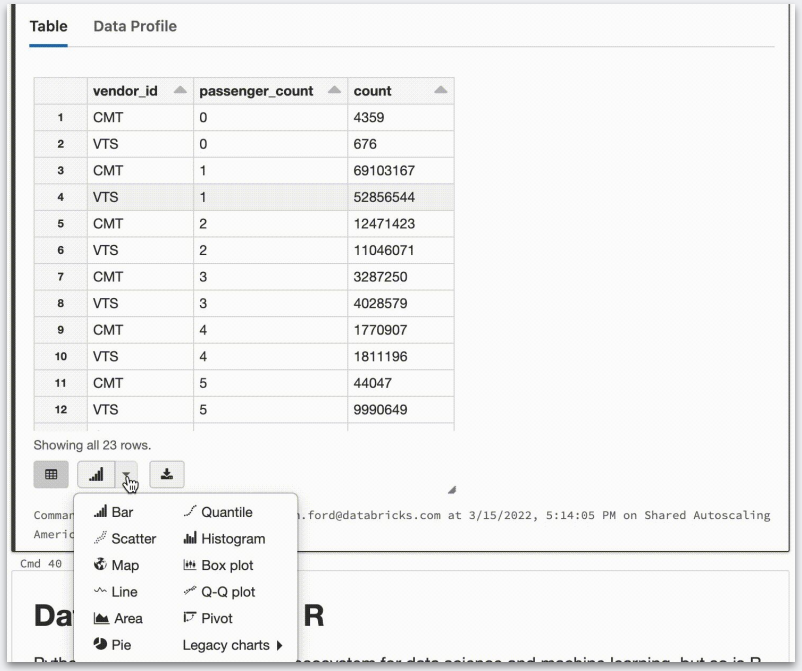
Coming 2022

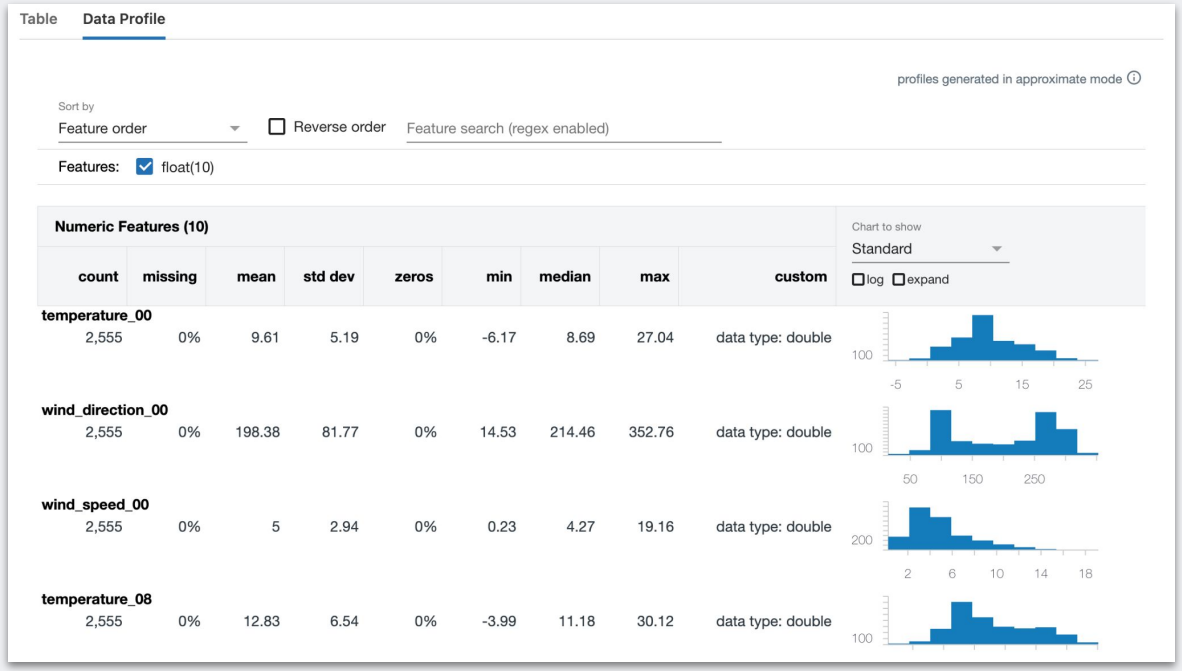
Natively store Databricks notebooks as **.ipynb files** inside of Repos



Explore data efficiently

Native tools for visualizing and understanding data





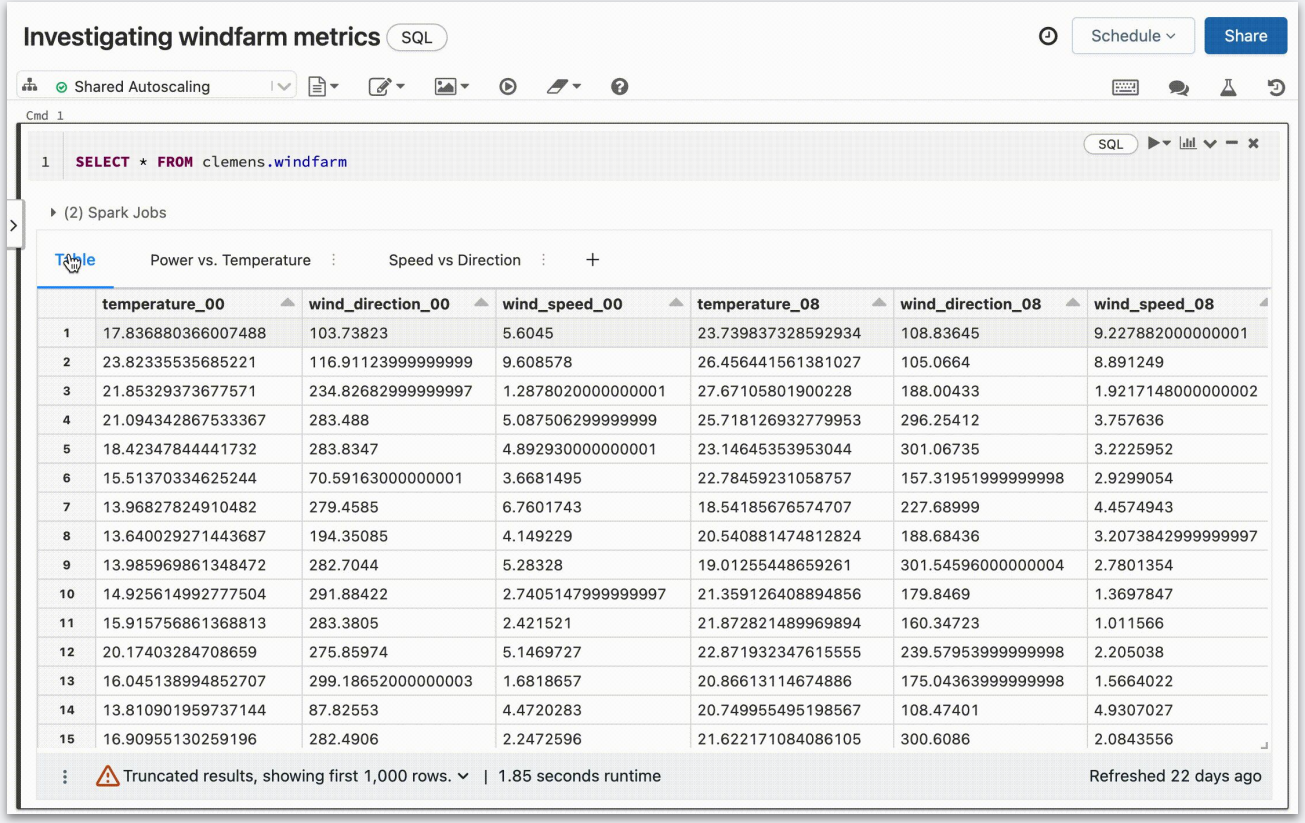
Create **interactive charts** to visualize data in the Notebook with only two clicks

Summarize a data set's essential properties and statistics in a **data profile** with the push of a button

Coming Soon

Richer cell results in the Notebook

Unified visualizations between Databricks SQL and the Notebook



	temperature_00	wind_direction_00	wind_speed_00	temperature_08	wind_direction_08	wind_speed_08
1	17.836880366007488	103.73823	5.6045	23.739837328592934	108.83645	9.227882000000001
2	23.82335535685221	116.91123999999999	9.608578	26.456441561381027	105.0664	8.891249
3	21.85329373677571	234.82682999999997	1.2878020000000001	27.67105801900228	188.00433	1.9217148000000002
4	21.094342867533367	283.488	5.087506299999999	25.718126932779953	296.25412	3.757636
5	18.42347844441732	283.8347	4.892930000000001	23.14645353953044	301.06735	3.2225952
6	15.51370334625244	70.59163000000001	3.6681495	22.78459231058757	157.31951999999998	2.9299054
7	13.96827824910482	279.4585	6.7601743	18.54185676574707	227.68999	4.4574943
8	13.640029271443687	194.35085	4.149229	20.540881474812824	188.68436	3.2073842999999997
9	13.985969861348472	282.7044	5.28328	19.01255448659261	301.54596000000004	2.7801354
10	14.925614992777504	291.88422	2.7405147999999997	21.359126408894856	179.8469	1.3697847
11	15.915756861368813	283.3805	2.421521	21.872821489969894	160.34723	1.011566
12	20.17403284708659	275.85974	5.1469727	22.871932347615555	239.57953999999998	2.205038
13	16.045138994852707	299.18652000000003	1.6818657	20.86613114674886	175.04363999999998	1.5664022
14	13.810901959737144	87.82553	4.4720283	20.749955495198567	108.47401	4.9307027
15	16.90955130259196	282.4906	2.2472596	21.622171084086105	300.6086	2.0843556

- Create multiple charts and data profiles from a single cell data result
- Craft a wider variety of visualizations with more visual appeal
- Flexible chart configuration using the Databricks SQL chart builder

Public Preview!

UI-based data analysis and transformation

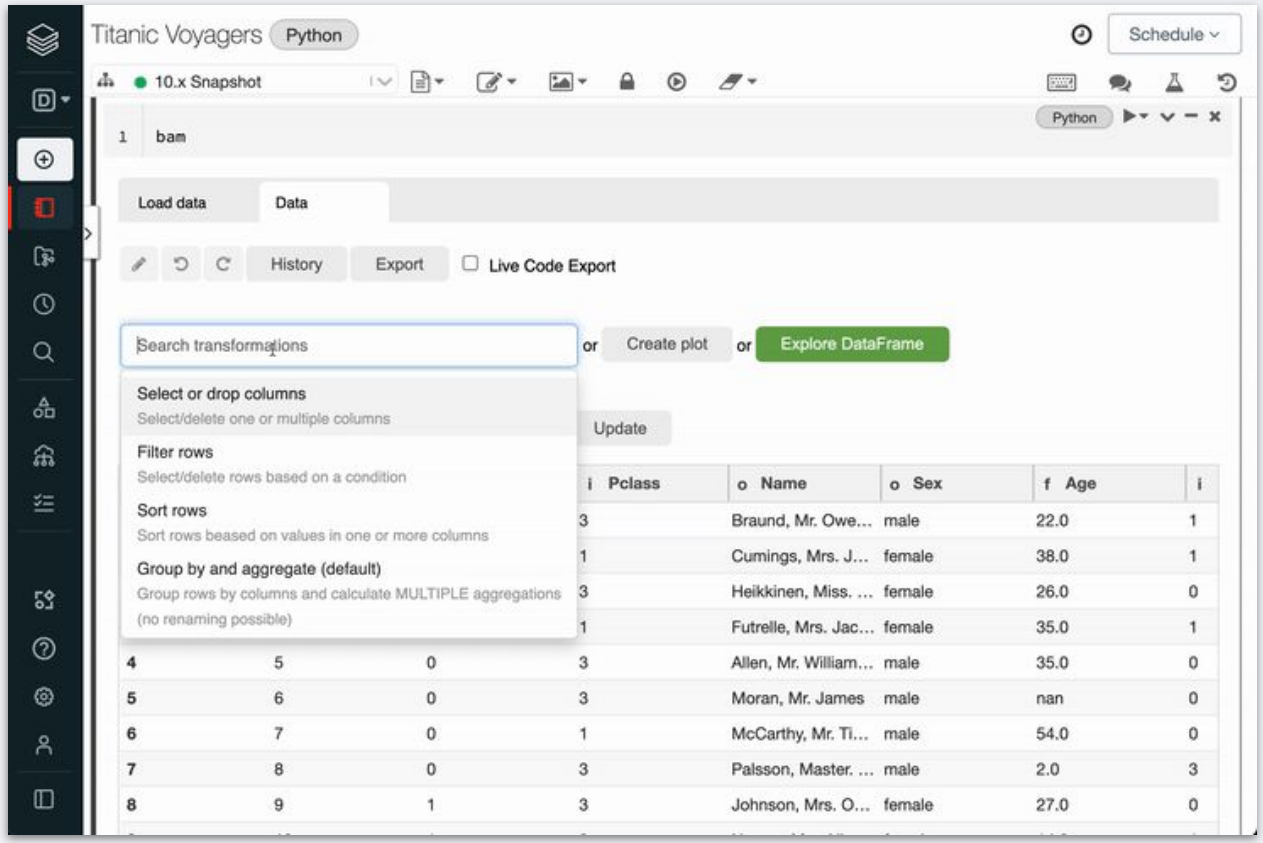
Integrating bamboolib into the Notebook

Prepare, transform, visualize, and explore your data—all through a UI!

Be more efficient by spending less time writing boilerplate code

Operations in bamboolib generate code so users can see, customize, and learn from what happens via the UI

Enable citizen data scientists who know what they want to do to do it in Python



Adaptable environments

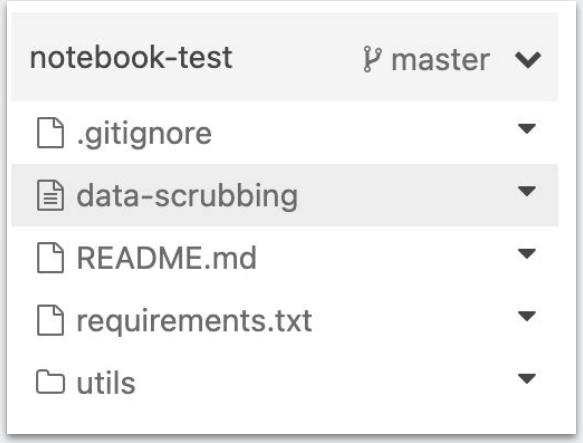
Use standard libraries and custom modules in the Notebook

```
%pip install folium seaborn==0.11.1
```

```
import seaborn as sns

sns.violinplot(data=tips, x="day", y="total_bill", hue="smoker",
               split=True, inner="quart", linewidth=1,
               palette={"Yes": "b", "No": ".85"})
sns.despine(left=True)
```

Install Python libraries for a notebook without affecting other users with **%pip**




```
import utils

df2 = utils.scrub(df1, drop="num_columns")
```

Import local modules using **arbitrary file support** when working in Repos

Enterprise-ready

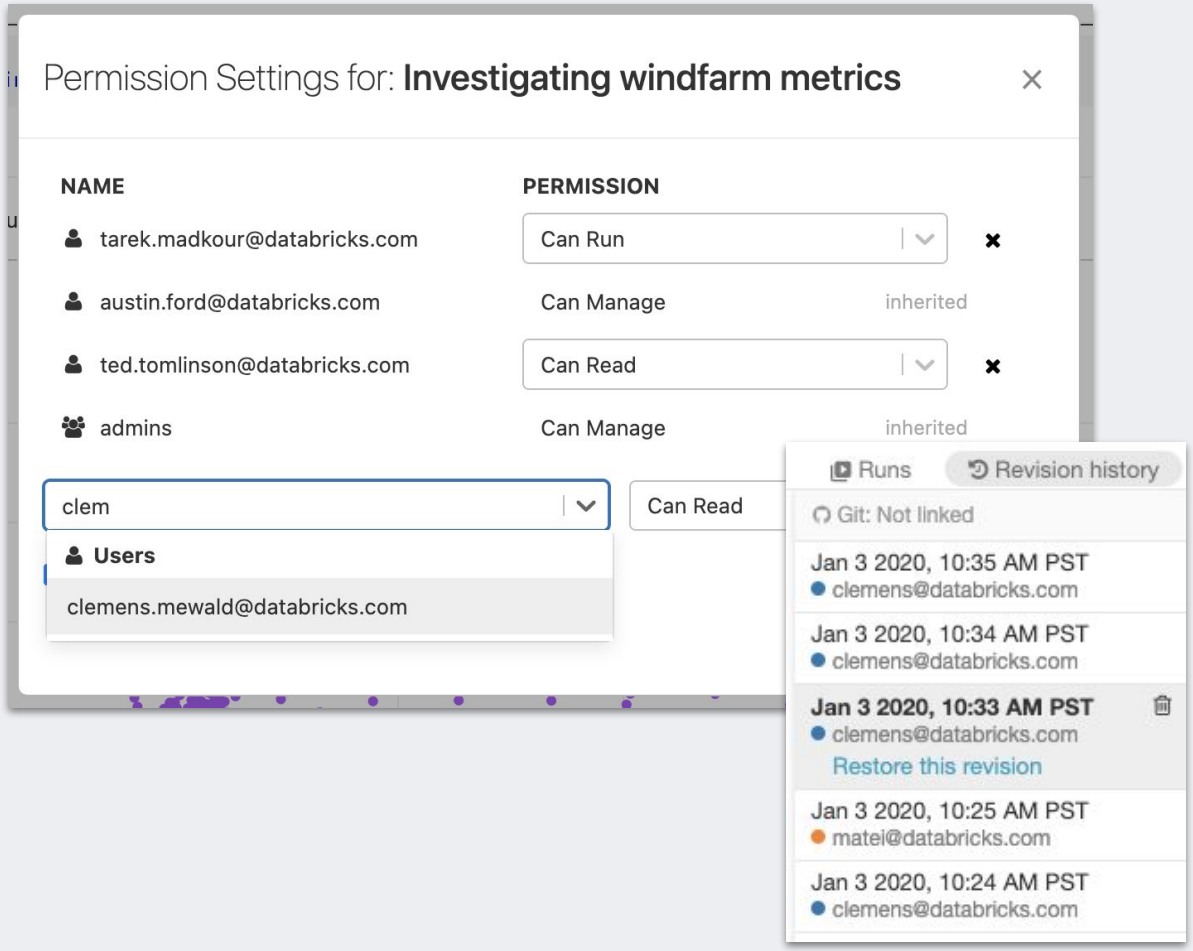
Fulfilling essential governance requirements

Notebooks provide full access controls and identity management and can be shared for reading, execution, editing, or managing

All notebook access and user revisions logged with user identities

New!

All Notebook command executions tracked in **audit logs**



Data exploration complete.

Now, how to get to production?

From exploration to production

Enabling easy maintenance of assets over time

Take your notebook into production natively in Databricks

We've made it easy to take notebooks that power your simple ETL pipelines, reports, and dashboards and put them directly into production when they are ready

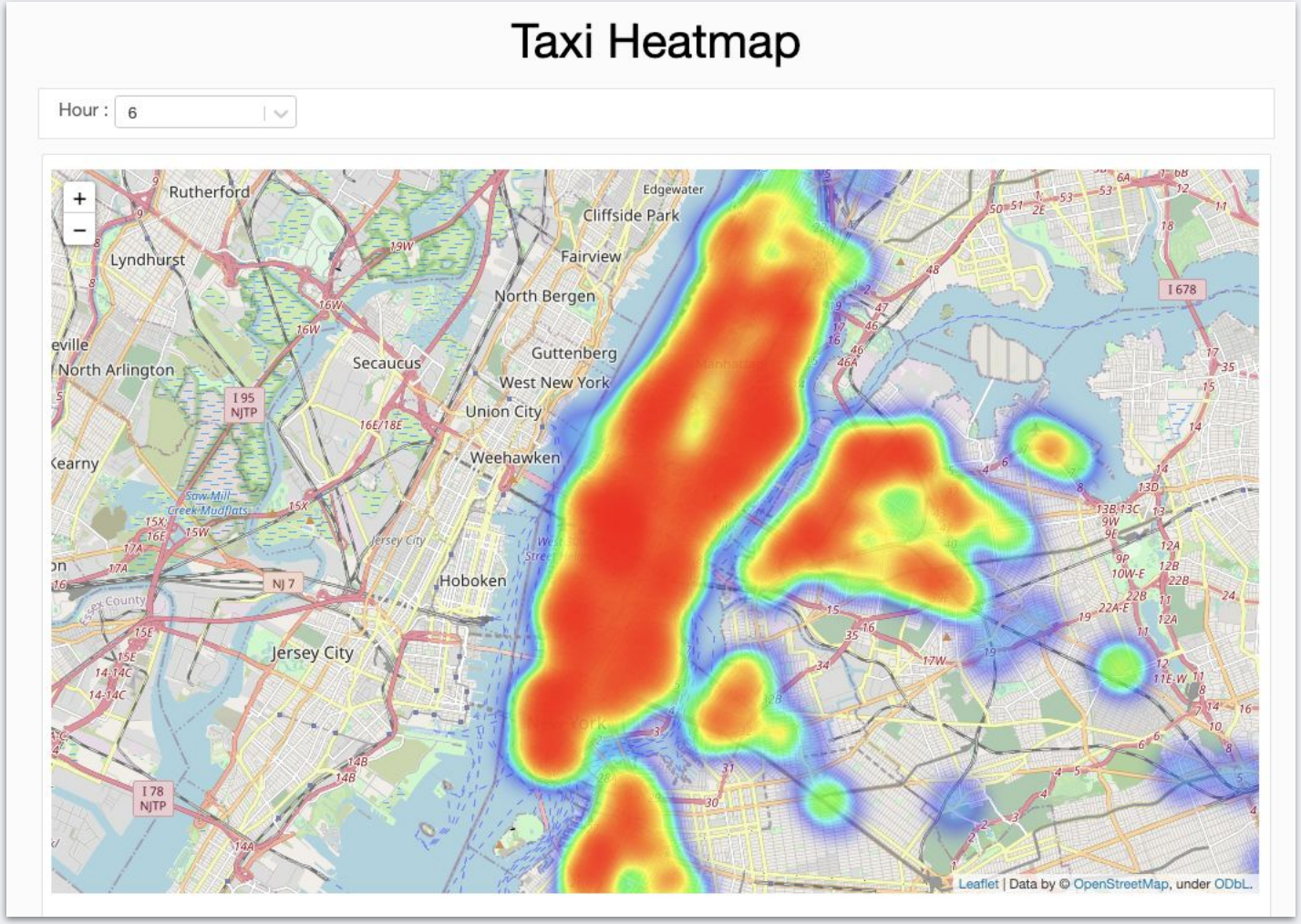
Use your favorite tool to productionize as you see fit

Sometimes you need more than a notebook, especially for complex projects—and Databricks enables this too

Faster paths to production

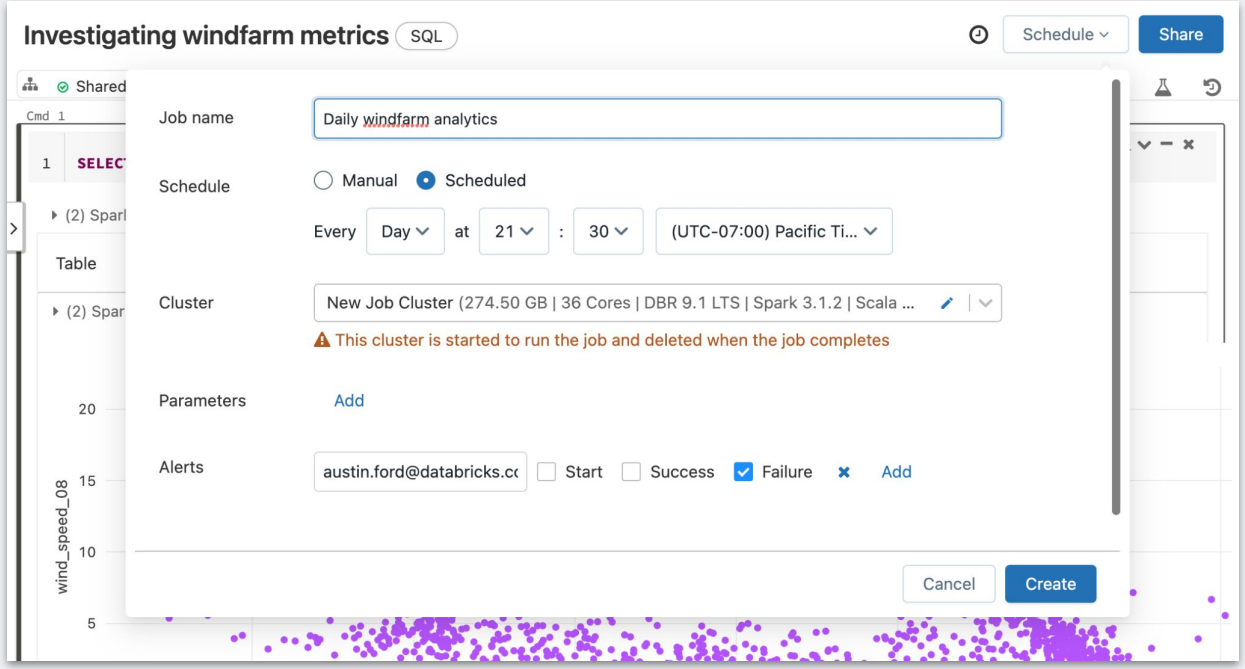
Quickly deliver value from work in the Notebook

Create **interactive dashboards with parameters** directly from the results of a notebook



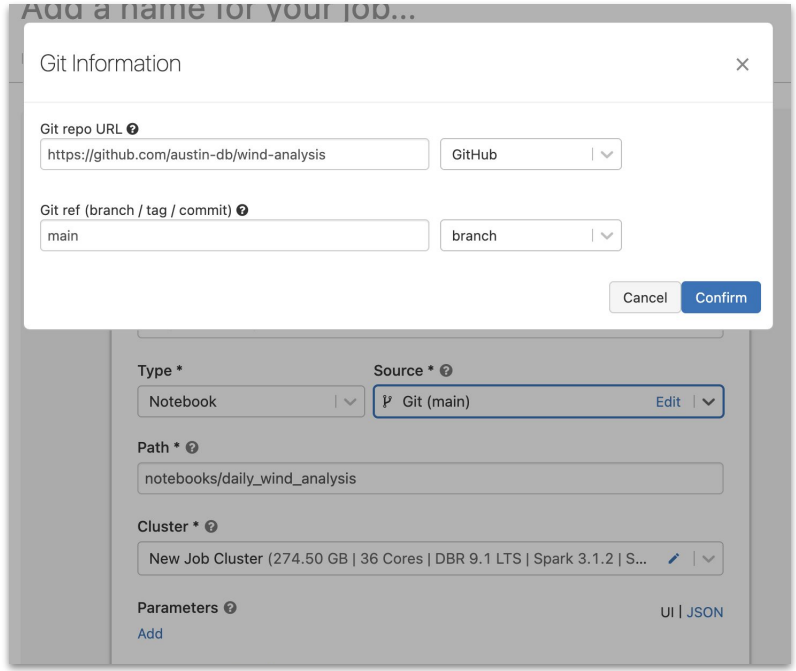
Faster paths to production

Quickly deliver value from work in the Notebook



Schedule a notebook as a production job with the push of a button

Public Preview!



Pin jobs to git branches, tags, or commits using the Workflows integration with Repos

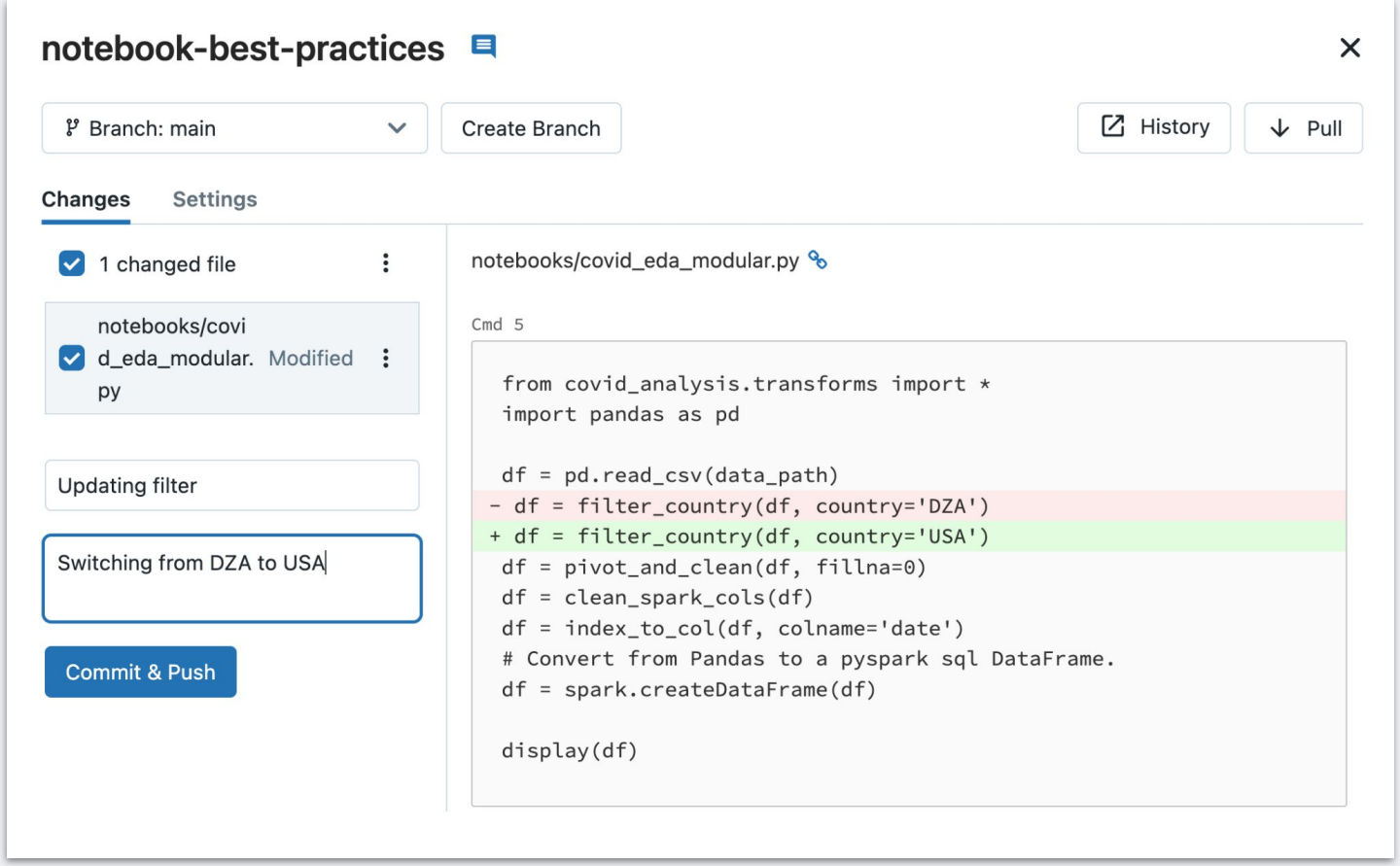
Best practices: Version control

Integrate with the most popular git providers

Databricks Repos support for git lets users collaborate on the same code without interfering with each others' work

Users' committed changes are logged in git history, making them fully **reproducible** and **auditable**

Protected branches and pull requests provide guardrails for what code moves to production



Best practices: Modular code

Simplify complex code projects for maintainability

dza_filter **transforms.py**

File Edit

```python
import pandas as pd

# Filter by country code.
def filter_country(pdf, country="USA"):
    pdf = pdf[pdf.iso_code == country]
    return pdf

# Pivot by indicator, and fill missing values.
def pivot_and_clean(pdf, fillna):
    pdf["value"] = pd.to_numeric(pdf["value"])
    pdf = pdf.fillna(fillna).pivot_table(
        values="value", columns="indicator", index="date"
    )
    return pdf

# Create column names that are compatible with Delta tables.
def clean_spark_cols(pdf):
    pdf.columns = pdf.columns.str.replace(" ", "_")
    return pdf


# Convert index to column (works with pandas API on Spark, too).
def index_to_col(df, colname):
    df[colname] = df.index
```

dza_filter **covid_eda_prod** Python Connected Schedule

Cmd 1

Automatically load modules

Cmd 2

```python
%load_ext autoreload
%autoreload 2
```

Command took 0.38 seconds -- by rafi.kurlansik@databricks.com at 6/22/2022, 3:48:26 PM on 11.x Shared Autoscaling

Cmd 3

Python

Cmd 4

Setup

Cmd 5

File support in Repos and the **File Editor** enable users to refactor large notebooks into modules and common libraries

Modules can be imported and reloaded on demand with the **%autoreload** magic command

Best practices: Testing

Ensure code behaves correctly

Test your Python code with either notebooks or scripts using frameworks like pytest and unittest

Manually run test suites inside notebooks or at the command line with the **Web Terminal**

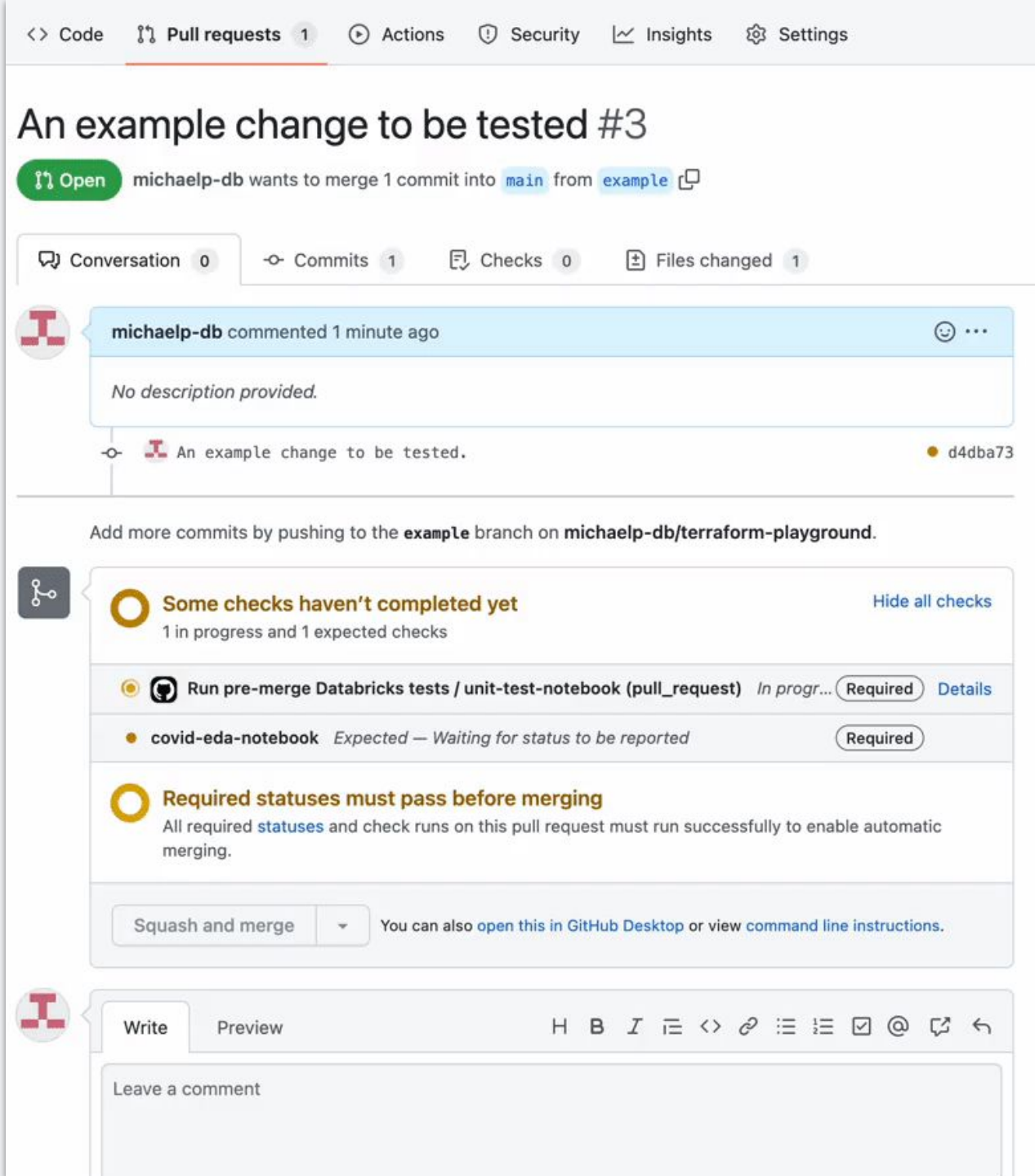
Automate testing by scheduling test notebooks with **Workflows**

Best practices: CI/CD

Automate code testing and deployment

Trigger continuous integration testing using the **Repos API** and support for git provider webhooks

Continuously deploy the latest production code using the **Workflows** integration with **Repos**



Demo

Recap

The Notebook today

What you can do right now

- Collaborate with your colleagues in our multi-language, interactive, and data-native Notebook that unlocks the power of the Databricks Lakehouse
  - Jump from pulling data with SQL to exploring it in Python without having to write any additional code
- Quickly and efficiently explore and visualize your data to derive shareable insights that deliver meaningful business value
  - Explore your data with less code using bamboolib
  - Use ipywidgets to turn insights into interactive applications for your stakeholders

The Notebook today

What you can do right now

- Take your work to production using Repos for version control and Databricks Workflows for scheduled, automated execution
- Alongside version control, employ other software development best practices like modular code, testing, and CI/CD

The Notebook tomorrow

New features coming soon

- Convey insights about your data using the new, easily configurable, and beautiful charts we are bringing to the Notebook from Databricks SQL
- Ipywidgets and bamboolib on GCP (with DBR 11.1)

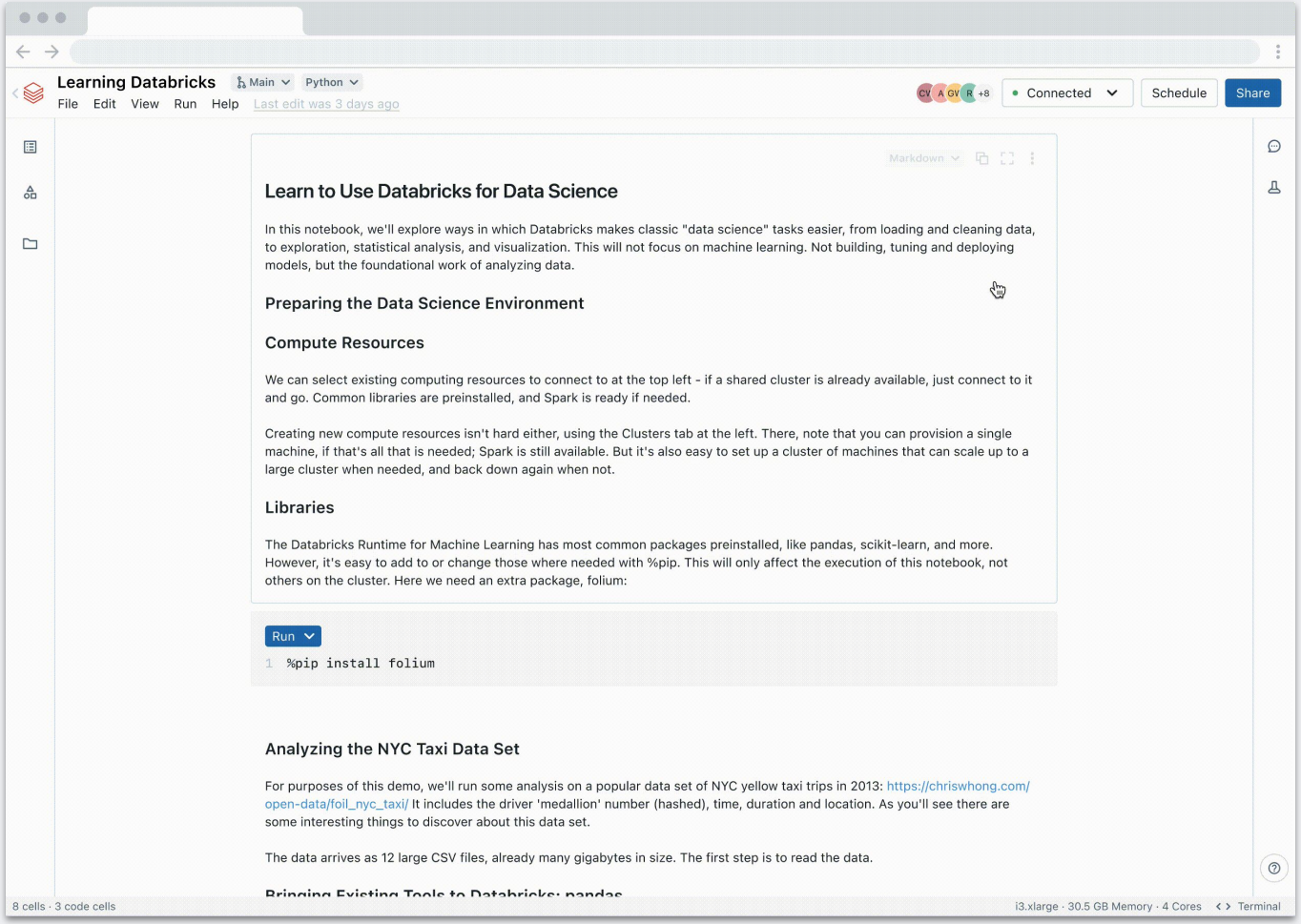
Where we're going

The future of the Notebook

Early ideas!

Focus areas

- EDA in Python and SQL
- Programming ergonomics
- Sharing and viewing
- Modern UI/UX
- Performance, reliability, and stability



DATA+AI
SUMMIT 2022

Thank you

Austin Ford

Senior Product Manager,
Databricks

Rafi Kurlansik

Lead Product Specialist,
Databricks