



Posit + Databricks

Improved productivity for your
data teams

Rafi Kurlansik
James Blair





James Blair
Posit
Product Manager
Cloud Integrations



Rafi Kurlansik
Databricks
Product Specialist
Developer Experience, AI

6000+
global employees

\$1.5B+
in revenue

\$4B
in investment

Inventor of the **lakehouse**
&
Pioneer of **generative AI**



databricks
The data and AI company

Gartner-recognized Leader
Database Management Systems
Data Science and Machine Learning Platforms

Creator of



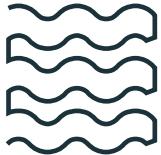
mlflow™



The winners in every industry will be
data + AI companies



Data Lake



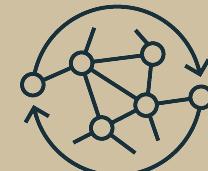
Machine Learning



Streaming



Generative AI



Data Science



Most organizations struggle
to realize this vision

Governance



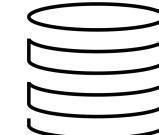
Orchestration & ETL



BI

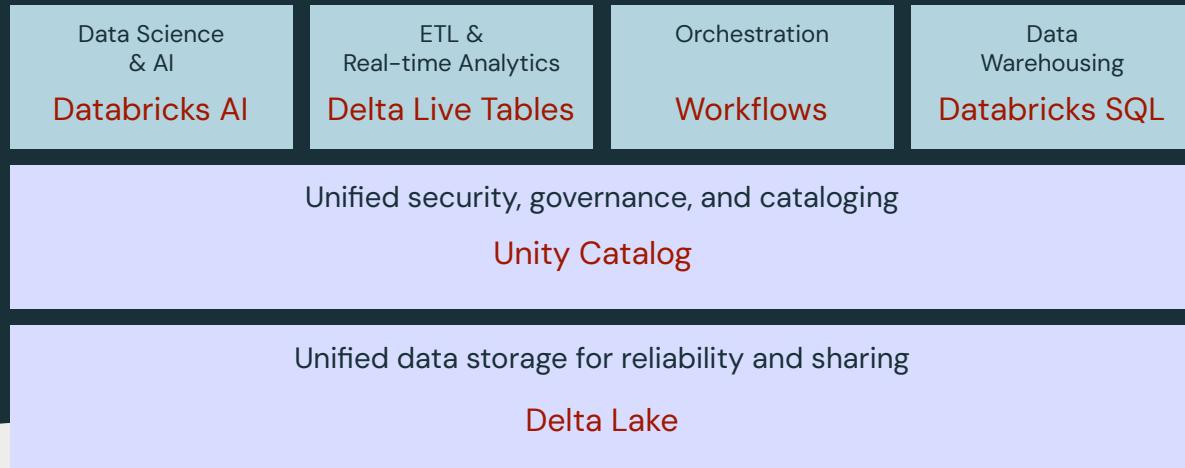


Data Warehouse



The Data Lakehouse

An open, unified foundation for all your data



2020

Databricks pioneered
the lakehouse
architecture



Today

74% of global
enterprises have
adopted lakehouse

MIT Technology Review
Insights, 2023

Data Lakehouse

An open, unified foundation
for all your data



Generative AI

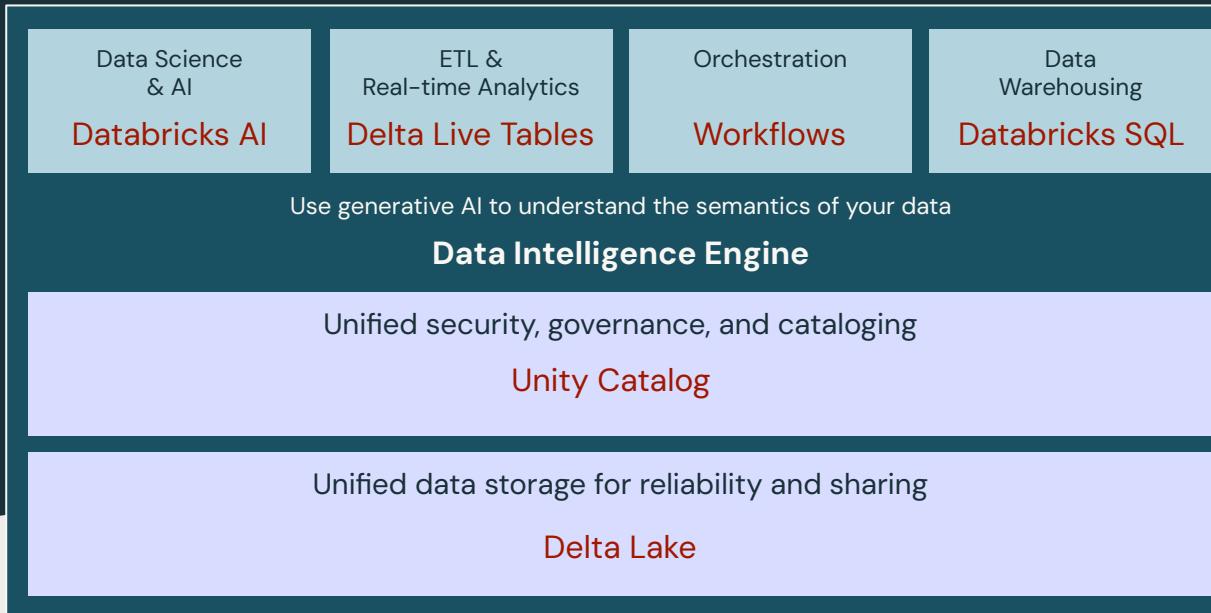
Easily scale and use data and AI



Data Intelligence Platform

Democratize data + AI across
your entire organization

Databricks Data Intelligence Platform



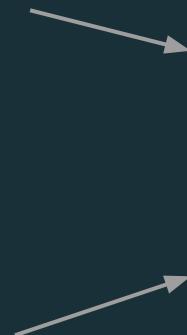
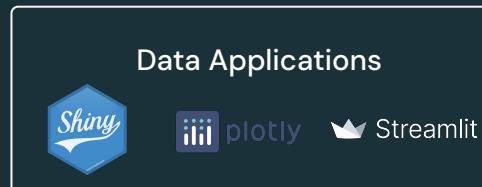
How do we access the Data Intelligence Platform with Posit products?



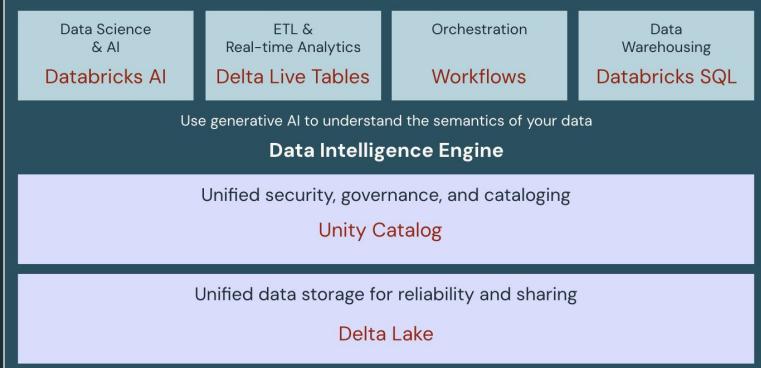
Databricks Connect V2

Python GA, R & Scala in preview

Bringing remote connectivity to Databricks



Databricks Data Intelligence Platform



Databricks Extension for Visual Studio Code

GA

Simple Setup

Find us on the VS Code Marketplace and get connected to compute in minutes

Native Experience

Write code using the productivity features you love from VS Code

Run on Databricks

Execute batch workloads or start interactively debugging directly from your IDE

The screenshot shows the Databricks Extension integrated into the Visual Studio Code interface. The Explorer sidebar on the left displays a project structure for a 'JOB_WITH_FILE_TASK' job, including '.databricks', 'development', 'input.py', 'production', 'typings', 'agg.py', and 'bundle.yml'. The main editor area shows a Python script named 'agg.py' with the following code:

```
agg.py — job_with_file_task
bundle.yml M agg.py 1, M
agg.py > ...
1  from pprint import pprint
2  from pyspark.sql.functions import col, max
3
4  range_max = 100
5
6  # Print maximum of range defined in the input task.
7  df = spark.range(range_max).agg(max(col("id")))
8  pprint(df.collect())
9
```

The terminal pane at the bottom shows a pyenv shell session:

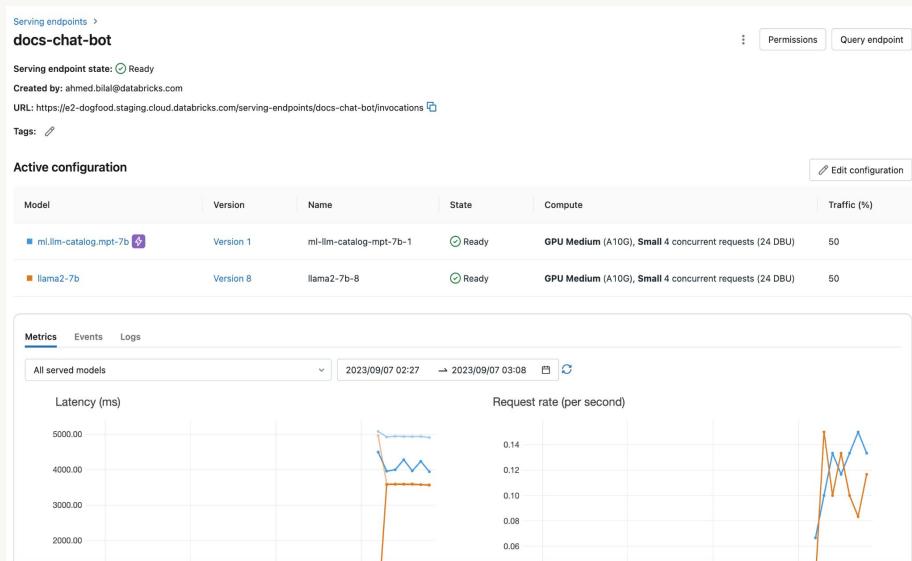
```
pyenv shell 3.8.13
(base) rafik.kurlansik@NY933MH6LM job_with_file_task % pyenv shell 3.8.13
(base) rafik.kurlansik@NY933MH6LM job_with_file_task %
```

The status bar at the bottom indicates the file is 'main*', has 0 errors and 1 warning, and is connected to a Python 3.8.13 64-bit ('3.8.13': pyenv) environment.

Databricks Model Serving

Real-time inference of Models, reducing latency and cost by up to 3-5x

- Fully managed, including GPU support, so you don't waste time managing infrastructure
- Optimized for specific LLMs to provide higher performance
- Automatic logging of Inference Tables into Delta tables and monitoring of performance





RStudio

Project: (None)

Environment History Connections Tutorial

Environment is empty

Quarto Running Code

Source Visual B I Normal Format Insert Table Outline

title: "Quarto Report"
format: html
editor: visual

Quarto

Quarto enables you to weave together content and executable code into a finished document. To learn more about Quarto see <https://quarto.org>.

Running Code

When you click the **Render** button a document will be generated that includes both content and the output of embedded code. You can embed code like this:

```
{r}  
1 + 1
```

You can add options to executable code like this

```
{r}  
#| echo: false  
? + ?
```

(Top Level) ↴ Copilot: No completions available. Quarto ↴

Console Terminal Background Jobs

R 4.3.2 · ~/ ↴ Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |

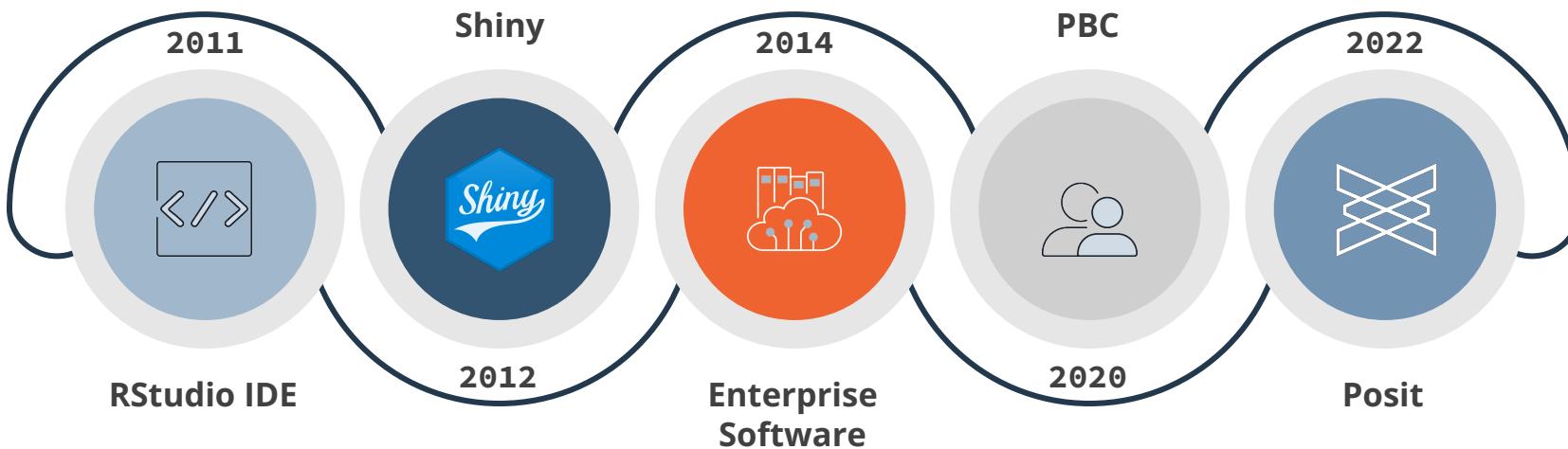
>



<

>

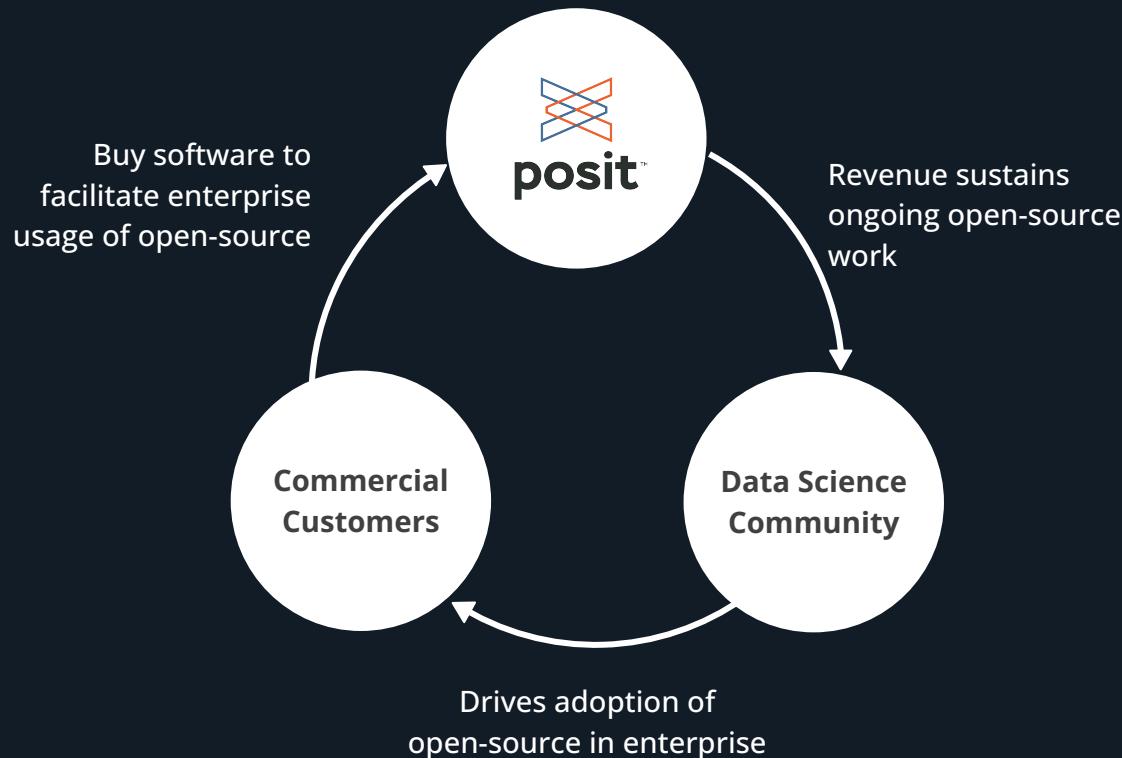
v



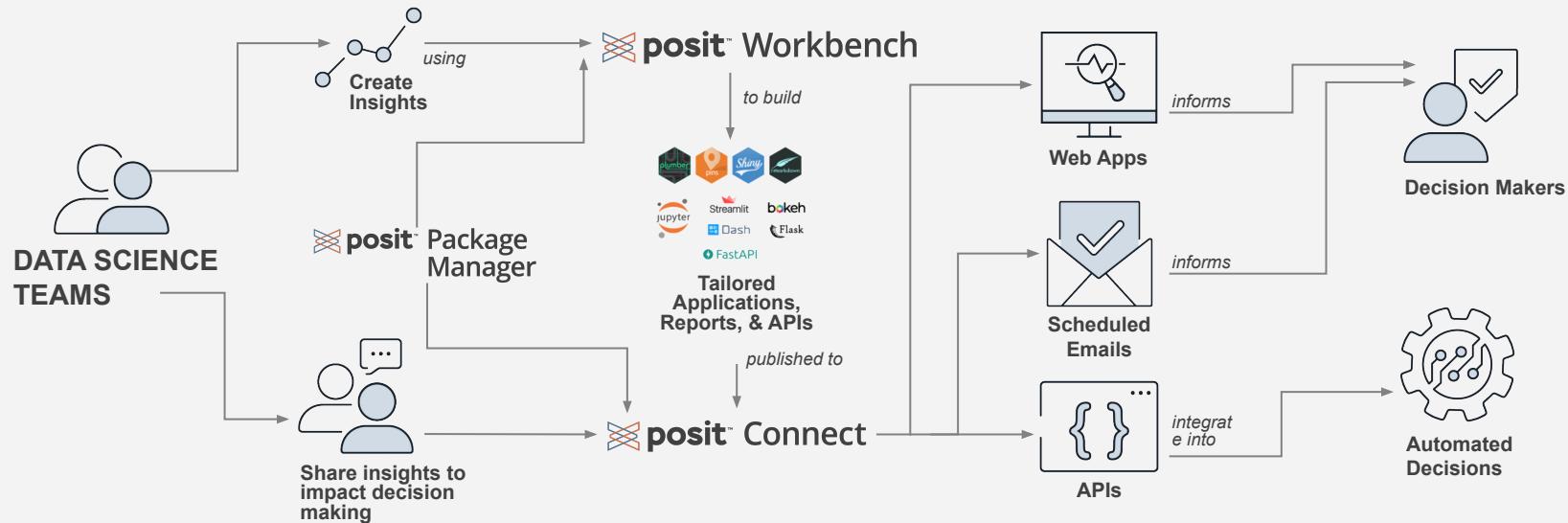
<

>

Contribute ~50% of engineering
resources to open-source projects



<



v



^



posit™ Team

odbc

sparklyr

SDKs

APIs

Data Science
& AI

Databricks AI

ETL &
Real-time Analytics

Delta Live Tables

Orchestration

Workflows

Data
Warehousing

Databricks SQL

Use generative AI to understand the semantics of your data

Data Intelligence Engine

Unified security, governance, and cataloging

Unity Catalog

Unified data storage for reliability and sharing

Delta Lake

Open Data Lake

All Raw Data
(Logs, Texts, Audio, Video, Images)



Demo



Demo recap



VS Code in Posit Workbench

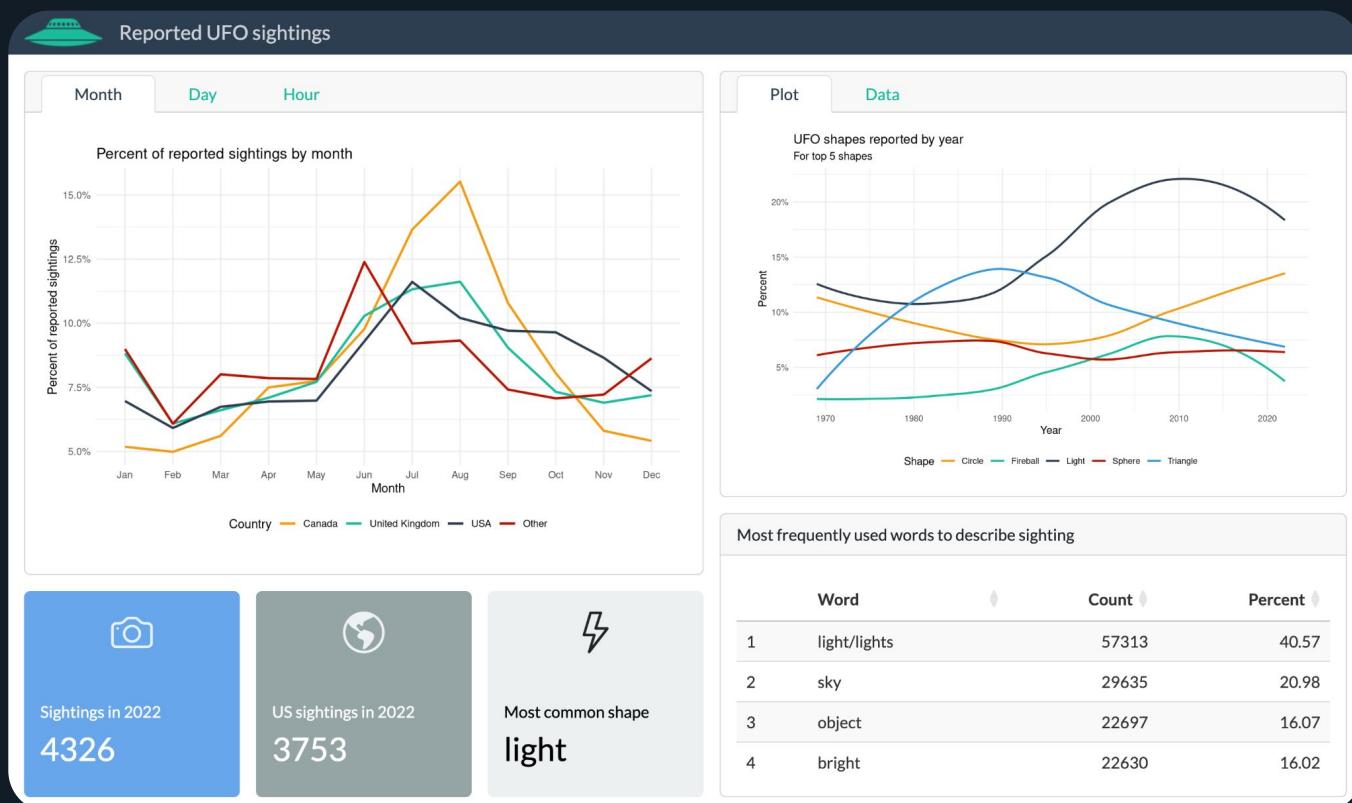
- ◆ Databricks VS Code extension
- ◆ Databricks Connect V2

Databricks Model Serving via:

- ◆ Databricks Python SDK (GA)
- ◆ Foundation Models API (preview)



Demo



>

Databricks x Posit: **Now**





Seamless Databricks development in RStudio and VSCode



The screenshot shows the Posit Workbench interface with the 'Databricks' tab selected. At the top, there's a search bar and several tabs: Environment, History, Connections, Git, Tutorial, and Databricks. Below the tabs, there's a table titled 'Clusters' with columns for Name, Runtime, Unity Catalog, and Creator. Two clusters are listed: 'Secondary Cluster' (14.2 ML, Yes, james@posit.co) and 'Test Cluster' (14.0 ML, Yes, james@posit.co). A detailed configuration panel is open for the 'Secondary Cluster', showing sections for Configuration (Cluster ID: 1120-195830-z4p39osf, Policy: Personal Compute, Creator: james@posit.co, Source: UI, Access: Single User, Unity Catalog: Yes), Performance (Runtime: 14.2 ML (includes Apache Spark 3.5.0, Scala 2.12), Node Type: i3.xlarge, Active Cores: 4, Active Memory: 30.5 GB, Started: 11/28/2023, 9:56:08 PM), and Tags (ResourceClass:SingleNode, rs:owner:james@posit.co, rs:project:solutions). Another cluster, 'db-ui-test' (14.1 ML, Yes, james@posit.co), is shown with a red error icon.



- Posit Workbench managed Databricks credentials - no need to supply PATs
- Manage and connect to Databricks clusters directly from the IDE
- Use managed credentials alongside Databricks tools and SDKs to interact with Databricks from RStudio and VS Code
- Enable Databricks VS Code extension to interact directly with Databricks





Making R and RStudio even sparklyr



New Databricks Connection for "Test Cluster"

Cluster ID: 0907-051241-shqxcbmx
✓ Found - Cluster's DBR is 14.0

Python Env: ✓ Found - Using r-sparklyr-databricks-14.0'

Master: rstudio-partner-posit-default.cloud.databricks.com

Connection: Environment History Connections Git Tutorial Databricks

Test Cluster (0907-051241-shqxcbmx)

Using "Databricks Connection Data" 

Spark

• demos

- default
- cars
- information_schema
- nuforc
- nuforc_reports

summary : chr MADAR Node 100 Steady flash ...
country : chr USA USA USA France USA
city : chr Mountlake Terrace Hamden Ch ...
state : chr WA CT VA MI NA CA
date_time : dttm 2019-06-23 18:53:00 2019-06 ...
shape : chr NA Light circle light cigar ...
duration : chr NA 5 hours 15 seconds 2 min ...
stats : chr Occurred : 6/23/2019 18:53 ...
report_link : chr http://www.nuforc.org/webre ...

- Posit has updated the `sparklyr` package to support Databricks Connect v2
- This provides support for R on the latest Databricks runtimes (v13.0 and beyond!)
- `sparklyr` provides a fully interactive R experience via remote connection to Databricks
- Growing support for additional Spark functions and features





Simplified **odbc** connectivity



The screenshot shows the RStudio interface with a Databricks connection setup. In the top right, a "New Connection" dialog for "Databricks Connection" is open, displaying an "HTTP Path": `sql/protocolv1/o/4425955464597947/0907-051241-shxqcbmx`. Below it, the RStudio environment shows a database browser with tables like `hive_metastore`, `main`, and `samples`. A code editor window contains R code for connecting to Databricks:

```
library(DBI)
con <- dbConnect(
  odbc::databricks(),
  HTTPPath = "sql/protocol",
  timeout = 10)
```

- Posit will begin delivering a Databricks specific ODBC driver
- The `odbc` R package includes a new `databricks()` function to simplify Databricks connections



>

Databricks x Posit: **Future**



<



posit™ Team

odbc

sparklyr

SDKs

APIs

Data Science
& AI

Databricks AI

ETL &
Real-time Analytics

Delta Live Tables

Orchestration

Workflows

Data
Warehousing

Databricks SQL

Use generative AI to understand the semantics of your data

Data Intelligence Engine

Unified security, governance, and cataloging

Unity Catalog

Unified data storage for reliability and sharing

Delta Lake

Open Data Lake

All Raw Data
(Logs, Texts, Audio, Video, Images)



Lakehouse Apps

posit™ Team

Data Science
& AI

Databricks AI

ETL &
Real-time Analytics

Delta Live Tables

Orchestration

Workflows

Data
Warehousing

Databricks SQL

Use generative AI to understand the semantics of your data

Data Intelligence Engine

Unified security, governance, and cataloging

Unity Catalog

Unified data storage for reliability and sharing

Delta Lake

Open Data Lake

All Raw Data
(Logs, Texts, Audio, Video, Images)

Lakehouse Apps: Security Without Compromise

Lakehouse Apps addresses these challenges with a native, secure, no-compromise solution.

Lakehouse Apps run directly on a customer's Databricks instance, where they can integrate with the customer's data, use and extend Databricks services, and enable users to interact with a single sign-on experience – all with the same security, privacy, and compliance controls as Databricks. **Data never needs to leave the customer's instance.**

Lakehouse Apps can be instantly distributed and monetized to over 10,000 Databricks customers through the Databricks Marketplace. Customers can discover, install, secure, manage, and govern them as efficiently as Databricks native features. Compute resources used by apps are billed directly to the customer by Databricks. A customer's internal apps can also be directly installed into an instance.

Lakehouse Apps are secured, sandboxed, and governed. Every app runs within a secure, configurable sandbox. Customers can use Unity Catalog, Databricks' unified governance solution, to select which resources an app can access, control who can interact with the app, and automatically govern an app's activities.

Lakehouse Apps are built with the technology of your choice. Apps run on secure, auto-scale compute that runs containerized code that can be written in virtually any language, so developers are not limited to building in any specific framework. Applications that integrate with Databricks today can be easily converted into Apps.

Lakehouse Apps are fully Lakehouse-native. The Databricks platform includes a powerful set of scalable and cost-effective serverless services, including fast data warehousing, workflows and pipelines, and AI/LLM training and serving. Lakehouse applications can leverage any of these capabilities. Apps can also use catalogs and contribute metadata and lineage in Unity Catalog, integrate with the Lakehouse filesystem, and extend the lakehouse with custom functionality.





Resources



- [GH Repository Link](#)
- [Sparklyr and Databricks](#)
- [Data Intelligence Platforms](#)
 - [Databricks Connect](#)
 - [VS Code extension](#)
 - [Model Serving](#)
- [Lakehouse Apps](#)

Questions