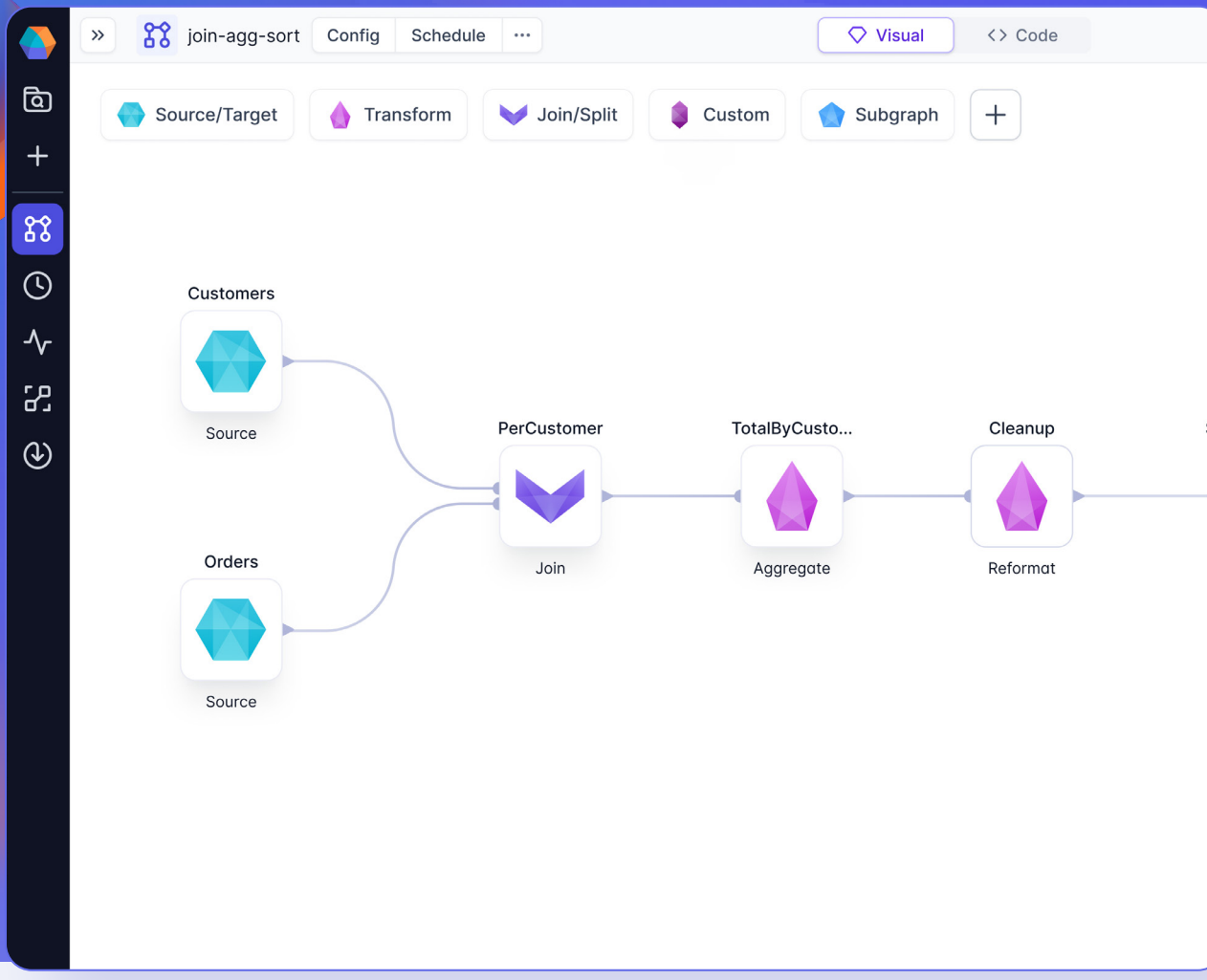
eBook

Build Data Pipelines on Databricks in 5 Easy Steps



Prophecy

Introduction: The rise of Databricks for data and AI

The exponential growth in data volume, complexity, and the insatiable demand for actionable insights is driving the adoption of the **Databricks Data Intelligence Platform** as the underlying infrastructure to support enterprises' data, analytics, and AI needs. Data Intelligence Platform combines Databricks' data lakehouse with technology the company obtained through acquisition of MosaicML. MosaicML is an AI development platform designed to tackle the challenges of training and serving large AI models. Combining the power of a data lakehouse with large AI models running on an open-source foundational model called DBRX offers Databricks users tremendous new capabilities with their data.

Databricks has become the de facto standard in the data intelligence landscape by providing a unified platform that brings together data engineering, data analytics, and machine learning, allowing organizations to build products and innovate with their data, at scale. This comprehensive solution enables enterprises to leverage the power of a lakehouse architecture to combine the economics of a data lake and the performance of a data warehouse.

Every part of the model development life cycle requires good data. Enterprises will differentiate from competitors by using high quality data transformations that allow them to create data models and accurate intelligent data applications. The importance of data transformation is still at the core of data products. It determines whether the enterprise is going to be successful when they make business decisions based on its high quality data. Data preparation like cleaning, featurization, and embedding of data for use in models is more essential than ever.

Raj Bains
Founder CEO, Prophecy

Challenges: A powerful platform built for engineers

Building data pipelines to transform raw data into a format suitable for AI and analytics can be complex and time-consuming. This process involves extracting, transforming, loading, and orchestrating data from various sources using tools like Kafka, Apache Spark, and Airflow. It requires proficiency in programming languages like Python, Java, or Scala. As a result, data engineers are often hired to build and maintain these pipelines, creating a dependence on their skills and availability. The data engineers also need a deep understanding of distributed systems and microservices management. This dependence can lead to bottlenecks, as analysts become locked out of this process and must wait for data engineers to provide them with the necessary data, causing delays in generating valuable insights.

Another challenge arises from the varied skill levels and limited supply of data engineers, which can result in long onboarding and development life cycles due to different coding standards and practices. A data engineer with a strong background in data management might not have strong experience in the cloud. High turnover rates can also lead to a lack of standardization and constantly changing data infrastructure. Poor communication and collaboration between data producers and consumers can further exacerbate these issues, causing data silos and slow turnaround times for data products, ultimately impacting overall team productivity.

To overcome these challenges, organizations need to streamline their data pipelines, improve collaboration between data teams, and ensure consistent data engineering standards are maintained.

Better Together: Databricks + Prophecy

Success lies in simplifying data architectures, maximizing reusability, and empowering teams for independent and rapid insights discovery. Databricks and Prophecy combine the world's most advanced data intelligence platform with powerful self-service data transformation capabilities to provide the ultimate productivity solution for all enterprise data teams.

Prophecy addresses the inherent complexity of Databricks to data consumers — unifying the usability of a self-service, visual interface with the power of code. Prophecy adds value in 4 key areas:

Productivity through self-service	No lock-in with open formats	Your Standards with extensibility	Completeness with a unified platform
• Visual, drag-and-drop interface: empowering a range of users to build and deploy production-ready data pipelines. • Prophecy Data Copilot: accelerating pipeline development through intelligent suggestions, task automation.	• High-quality open source code generation from the visual pipelines in PySpark, Scala, SQL with dbt Core. • Software engineering best practices are enabled via Git, CI/CD, tests.	• Data engineers can build and import existing standards using Spark or SQL. • Your standards are then accessible in the visual layer to all users.	• One platform for all data users. • A unified view across all data platforms. • One platform covering the entire pipeline lifecycle.

Databricks and Prophecy together unlock the full potential of data assets. Whether it's ingesting massive datasets, transforming complex data formats, or performing intricate analytics, this partnership ensures efficiency, scalability, and agility. Data engineers, analysts, data scientists, and business users alike can collaborate seamlessly, driving innovation and accelerating time-to-insights.

Building data pipelines on Databricks in 5 easy steps

Building data pipelines on Databricks using Prophecy is very easy. In this section, we'll walk through five straightforward steps to build robust, production-ready data pipelines using the powerful combination of Databricks and Prophecy.

STEP 1

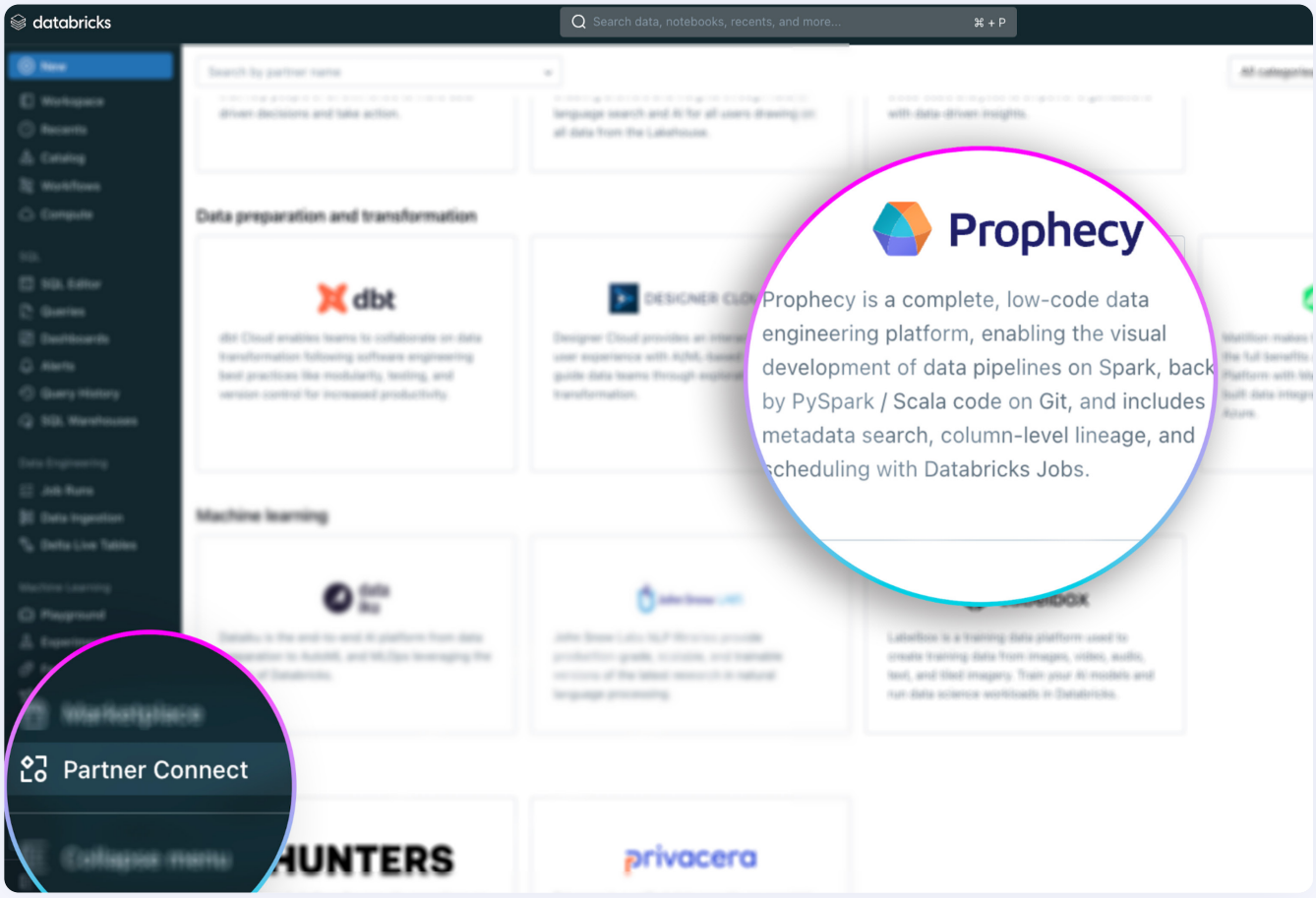
Set up the environment for development and execution

Starting with building pipelines on Databricks can be approached in two ways. It's likely the administrator has already linked Prophecy and Databricks using a Prophecy Fabric. To begin, you'll need the administrator to add you in the correct Team, granting access to the assigned Prophecy Fabric for team members. Each team member will provide their own access token to authenticate with Databricks.

A Prophecy Fabric is a logical execution environment. Teams can organize their domain into multiple environments such as development, staging, and production. In the case of Databricks, the prophecy Fabric uses REST API requests to connect to the user's Databricks workspace. It authenticates as a user based on the unique personal access token.

As a user, we usually run Prophecy pipelines interactively in a development environment. These Pipelines can be productionized with automated workflows after proper testing and QA. It is common to have separate development and production workspaces in Databricks. The best practice is to create separate Fabrics for each workspace. Fabrics aren't limited to Databricks, additional options like EMR, Synapse, data warehouses, and Airflow are also available.

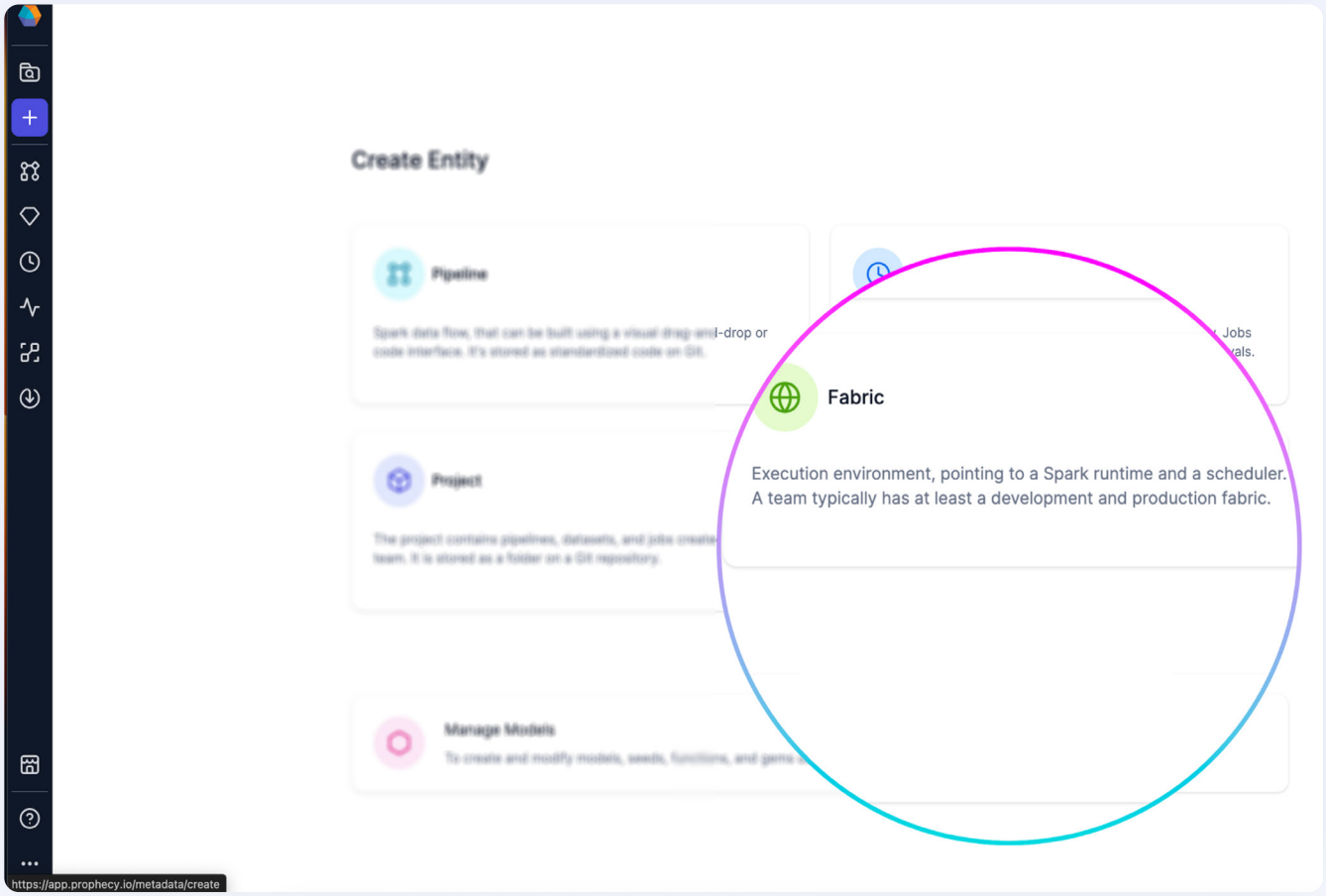
If there isn't already a connection between Prophecy and Databricks, you'll need to create one by setting up a Fabric. You may also sign up for a free Prophecy trial **here**. You may also configure the integration from the Databricks UI by seamlessly accessing Prophecy through the Partner Connect page.



Accessing Prophecy from the Partner Connect page

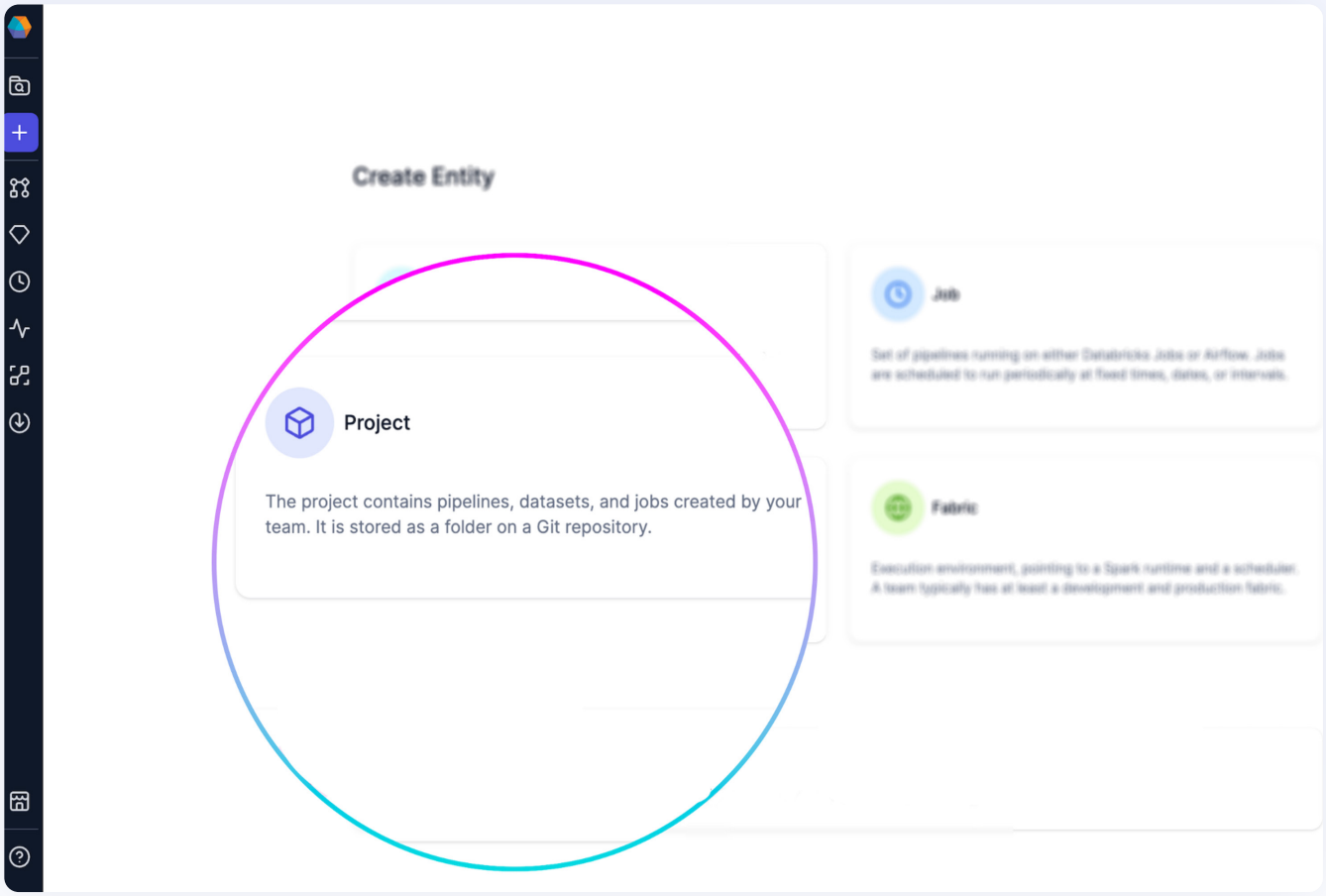
Databricks automatically sets up a secure connection and passes along your login credentials. Simply choose a new password for Prophecy, and you're ready to go. Keep in mind that security-conscious enterprises that use Databricks with limited network access have to additionally add the Prophecy Data Plane IP address (3.133.35.237) to the Databricks allowed access list. For more information, take a look at the **documentation** or click **here** for a self-service demo.

Now that we've established the connection, we can access the Create Entity page by clicking the plus sign on the left-hand side menu. From there, we can create a new Fabric and set up the user access token we mentioned earlier. This ensures successful communication with the Databricks workspace.



Once a Databricks Fabric is set up, you can now create a Project. In Prophecy, the primary unit for both development and production deployment is the “Project.” A project is represented as code stored on Git. This means that the entire business logic associated with assets like Pipelines, Datasets, and Jobs is saved as code within a Git repository. This Project’s repository can be hosted on a Git provider or organized within a specific folder within a repository.

By storing your code and tracking changes on git, you’re ensuring that your organization can benefit from software engineering best practices.

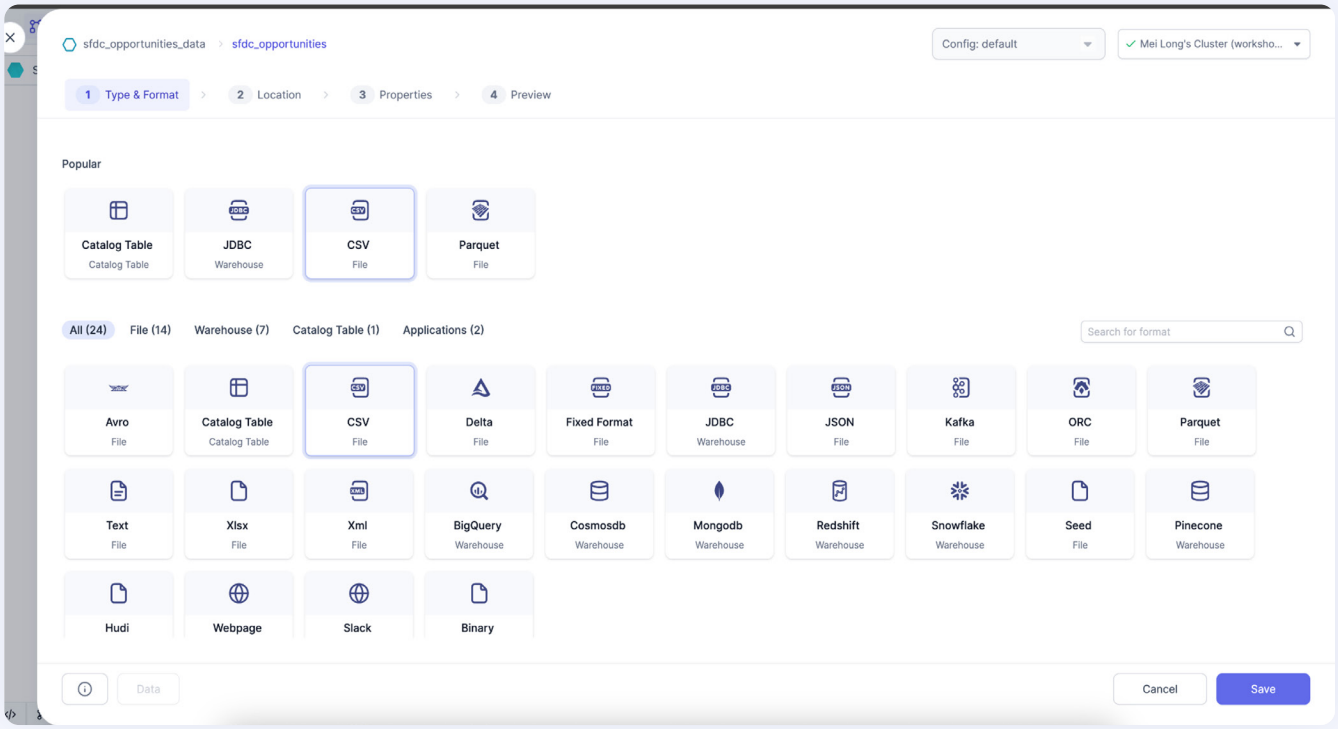


STEP 2

Read and parse raw data from sources

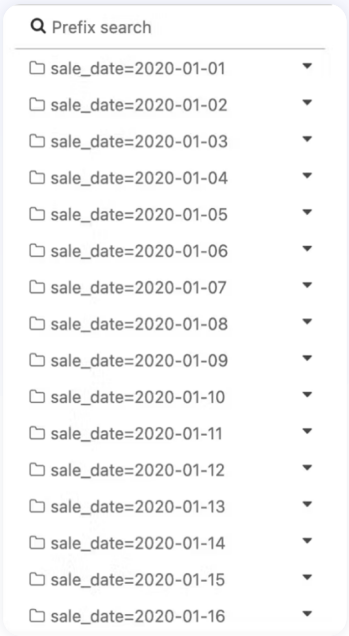
As a data professional building and using the data lakehouse, it's essential to understand the sources of your data. Otherwise, it's impossible to build an architecture that serves your business requirements.

Prophecy supports data from various sources, including object storage, applications such as Salesforce.com, transactional databases, and log files. With Databricks, the data lakehouse architecture differs from traditional data warehouses. It doesn't require a set of predefined business questions. Instead, our goal is to store the raw data as-is in its original form. This data is immutable and serves downstream processing, providing the business with maximum flexibility to meet their specific requirements.



In many cases the default settings for prophecy datasets will allow users to jump in and quickly have things working without getting into too many details. We're going to ingest CSV files from the Salesforce account and opportunity subjects.

You might wonder: what are some considerations and best practices when working with raw data in the object store? Let's start with how to store raw data on S3 - it's a best practice to have files persisted in various partitioned folders rather than having all files in one folder. The folders are usually partitioned by date. This structure allows us to pick and choose the folder which we want to load the data, providing flexibility and minimizing costs. It also makes the processing more efficient.



Another consideration is our file size. As mentioned previously, you may have a few files in each folder within a bucket. Common files have an ideal size of 128MB to 512MB. Having too many small files will cause significant performance degradation.

The choice of file format is a crucial consideration. We advise opting for a splittable file format. Spark, being a distributed system geared for parallel processing of numerous files, can encounter performance imbalances and stragglers if the file format is excessively large and non-splittable. For example, the gzip format, which lacks splitability, leads to a sequential load process and slows down job processing time.

Lastly, we need to contemplate the compression codec to utilize. Prophecy's default setting is Snappy. Snappy is the optimal choice for typical use cases, offering a commendable compression rate and high performance. If better compression rate is desired, we may consider another codec like LZO, but be aware better compression also means more CPU utilization and longer processing time.

STEP 3

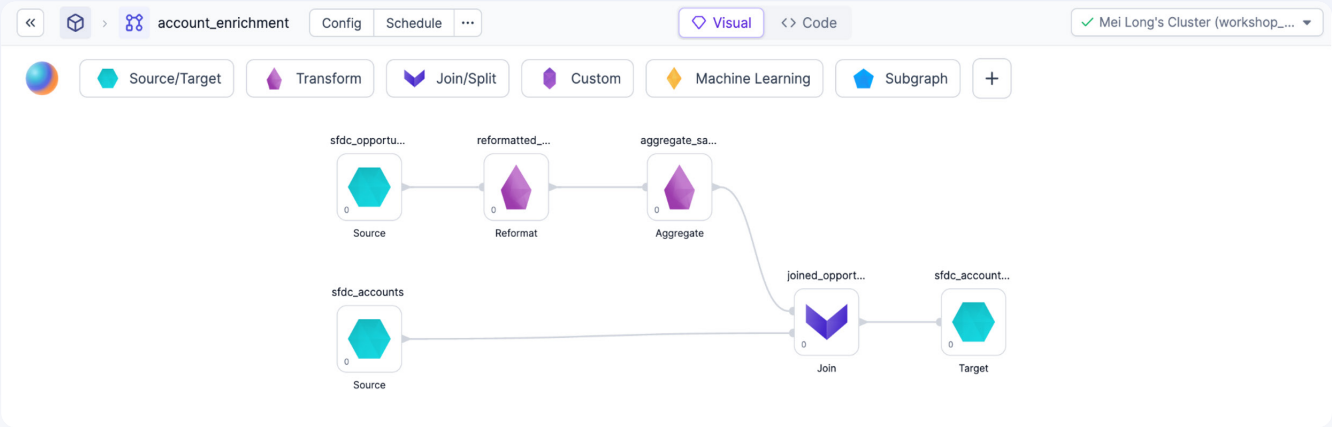
Build a visual pipeline to transform Salesforce data using copilot

Now that we are all set up, it is time to build the pipeline from the sources that were defined previously. Prophecy offers multiple capabilities to help accelerate and simplify this process.

Prophecy's intuitive, visual, drag-and-drop interface is designed to empower a wide range of users, from data analysts, data scientists to data engineers, to build and deploy production-ready data pipelines with ease. The user-friendly Pipeline Canvas serves as the central workspace for creating, modifying, and visualizing data transformations, streamlining the data transformation process. The pipelines are composed using Prophecy Gems. Prophecy offers many types of Gems out-of-the-box. They're source or target Gems for reading and writing data; transform, join, and custom gems for data enrichment and transformations.

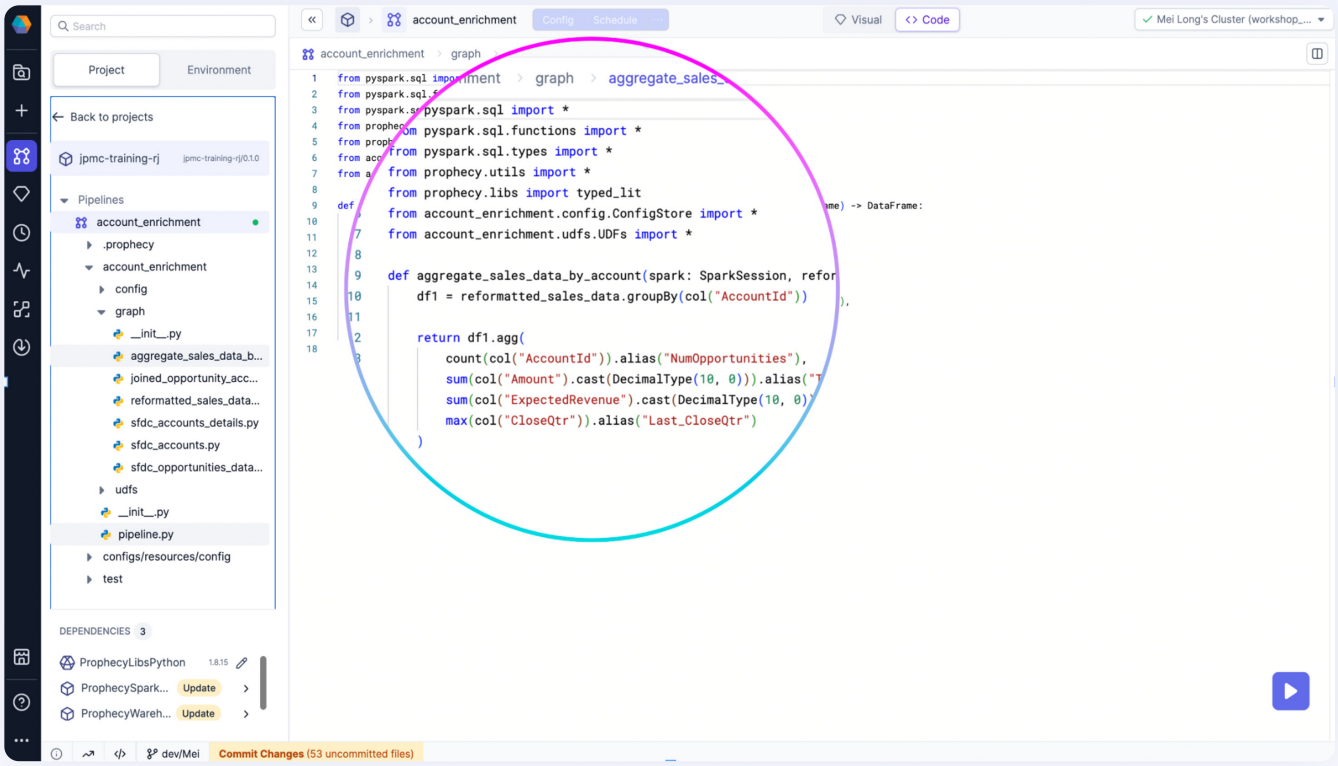
In addition to built-in Gems, data Copilot, an AI-powered assistant, further enhances the user experience by delivering intelligent suggestions on transformation logics and automating data documentation. Users can leverage plain English prompts to build and modify data pipelines, making the process more accessible and efficient.

data is immutable and serves downstream processing, providing the business with maximum flexibility to meet their specific requirements.



Build a Salesforce pipeline that utilizes Copilot for transformations

Behind the scenes, Prophecy translates the visual pipelines into high-quality open source code in Python, Scala, or SQL. This ensures that the data pipelines adhere to industry-standard coding best practices and enables users to view, modify, and reuse the code if necessary



Code view for the Salesforce accounts enrichment pipeline - showing aggregation code

Prophecy enables data engineering best practices, such as version control with Git, continuous integration and delivery (CI/CD), and automated testing. These features help ensure that data pipelines are reliable, maintainable, and scalable, reducing the risk of errors and inconsistencies.

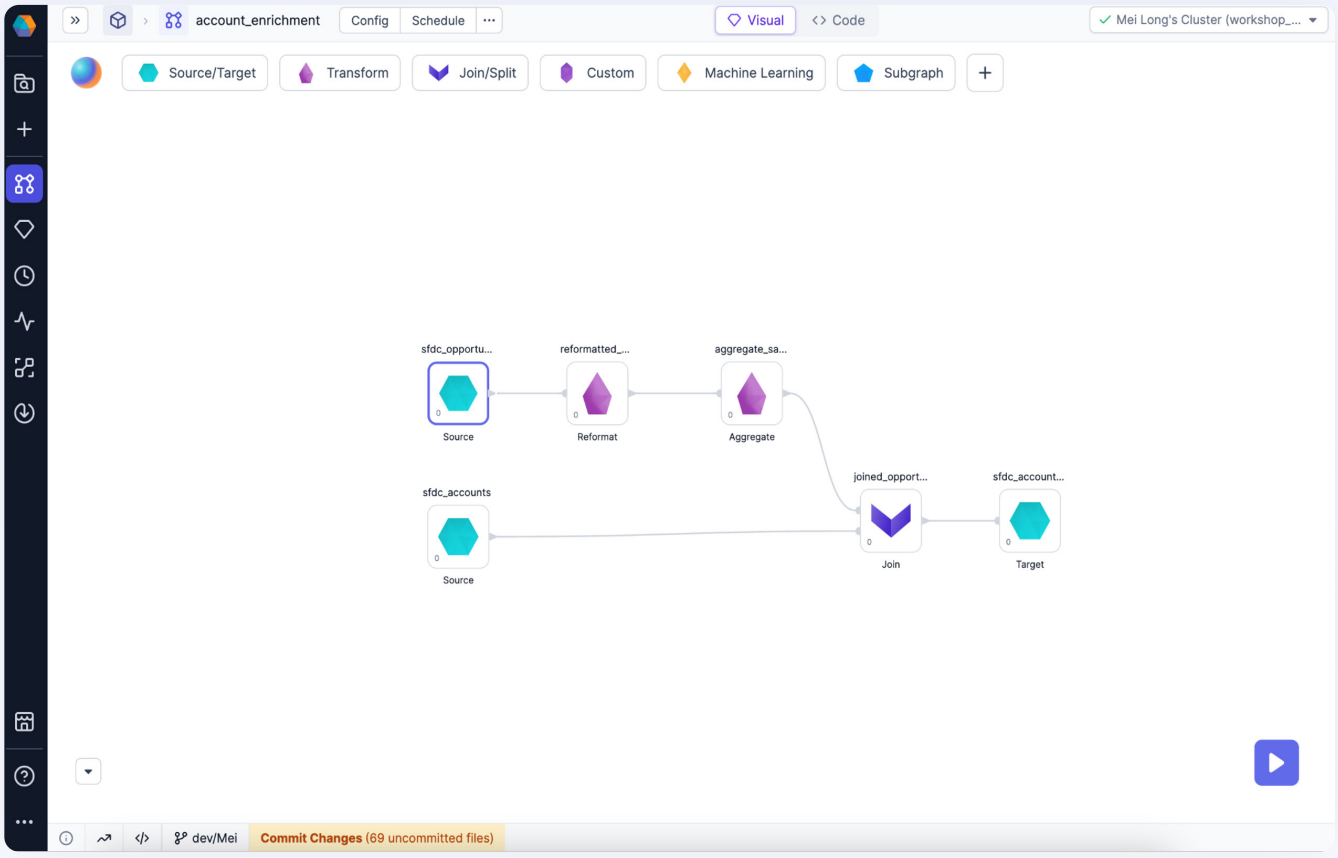
Last but not least, results of the visual pipelines are automatically written to one or more targets such as Delta tables, s3 files, or more. Prophecy provides many target Gems out of the box. Users have the flexibility to write data using these Gems to complete their lakehouse architecture. The gold layer Delta tables enable users to leverage Databricks' powerful features, such as its visualization and dashboarding capabilities.

STEP 4

Schedule and Run Pipeline Workflows on Databricks

After Prophecy translates the visual pipeline into open-source Spark code, it can be executed on the chosen execution engine, such as Databricks, leveraging its scalability and performance for efficient pipeline execution.

To execute your data pipelines interactively on Databricks, simply navigate to the pipeline you want to run within the Prophecy UI and click on the “run” button. Prophecy will automatically handle the deployment and execution of your pipeline on the Databricks platform, taking advantage of its powerful processing capabilities. This integration allows users to benefit from Databricks’ distributed computing, optimized performance, and ability to handle large-scale data processing tasks without the need for manual configuration or complex setup processes.

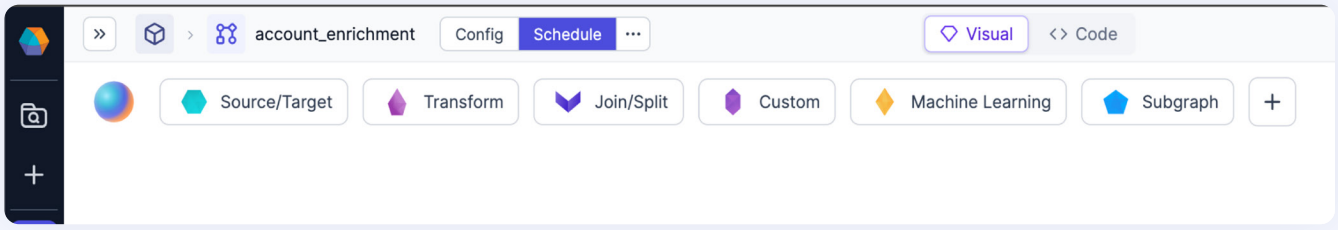


Running a pipeline interactively from the Prophecy UI

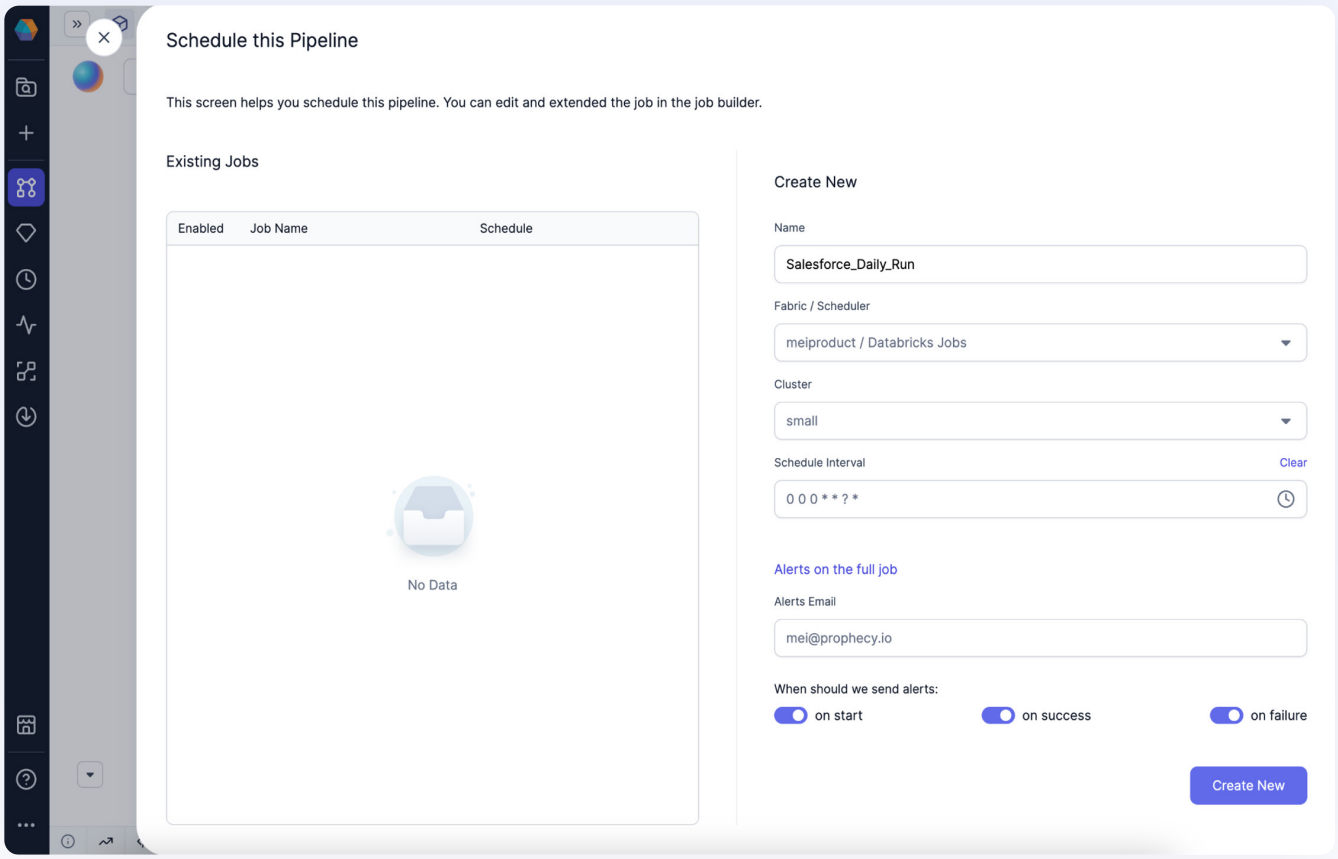
Interactive runs are common when developing pipelines. Once the pipelines are developed, we can use databricks workflows to schedule jobs. Databricks Workflows is a managed orchestration service specifically designed for the Databricks lakehouse platform.

We can implement our data processing and analysis workflow using tasks. A job is composed of one or more tasks.

Workflows lets us define and manage multi-task workflows for ETL pipelines. With a wide range of supported task types data teams can automate and orchestrate pipelines.



We can use a job to run a data processing pipeline on a Databricks cluster with scalable resources. A job can consist of a single task or can be a large, multi-task workflow with complex dependencies. Databricks natively manages the task orchestration, cluster management, and error reporting for all of our jobs. We can run our jobs immediately or periodically through an easy-to-use scheduling system. Remember Prophecy Projects are code on git. This is also true for our jobs. All jobs are translated into code and committed to our github account following engineering best practices.



STEP 5

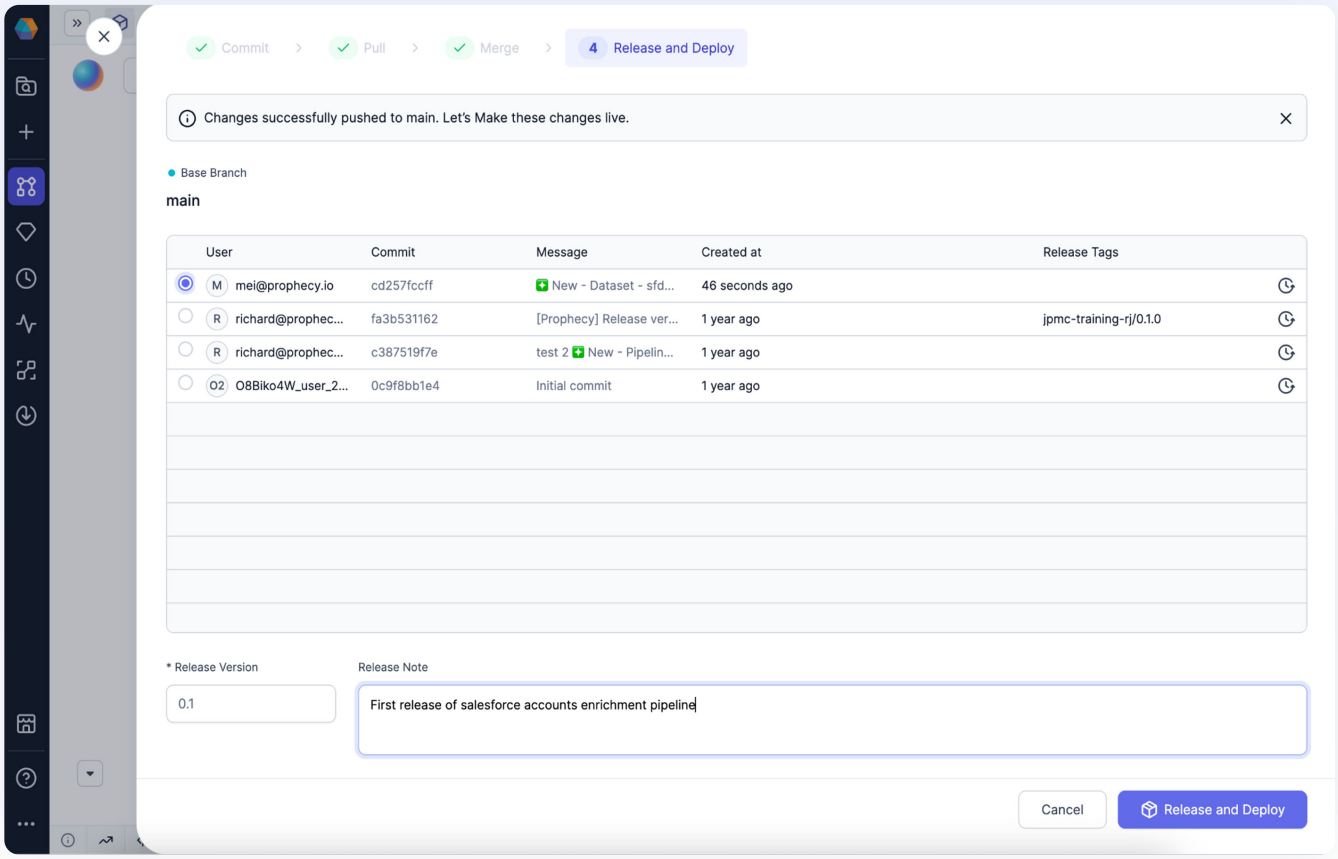
Commit and merge Spark code using Git

As mentioned above, in Prophecy, the primary unit for both development and production deployment is the “Project.” A project is represented as code stored on Git.

This means that everything we built earlier associated with assets like Pipelines, Datasets, and Jobs is saved as code within a Git repository. This Project’s repository can be hosted on a Git provider or organized within a specific folder within a repository.

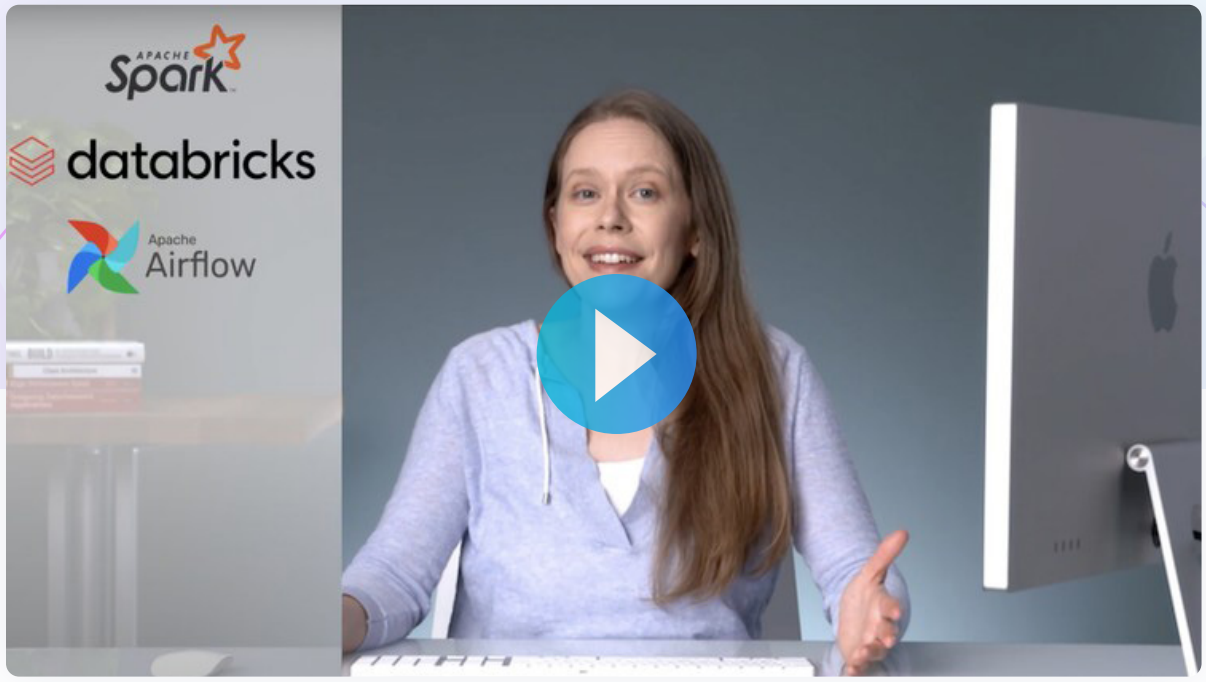
By storing our code and tracking changes on git, we’re ensuring that our organization can benefit from software engineering best practices.

Remember that Prophecy is compatible with various Git providers, with GitHub being one of the widely used platforms. If you’re new to Git, there’s no need to feel overwhelmed. Prophecy offers an intuitive user interface that streamlines the Git process. You don’t have to grasp the command line interface (CLI) commands to incorporate engineering best practices when developing pipelines.



Prophecy University Course

Prophecy for Data Engineering: Low-code Data Transformation



Ready to elevate your data engineering skills and streamline your workflows with Spark and Prophecy?

In our comprehensive course, “Prophecy for Data Engineering: Low-code Data Transformation”, you’ll master the implementation and deployment of a data lakehouse using Prophecy on Databricks, gain a deep understanding of Apache Spark and its best practices, and learn how to share and extend pipeline components with data practitioners and analysts. You’ll also discover how to deploy pipelines to production and CI/CD, effectively utilize version control and change management in data engineering, and deploy data quality checks and unit tests.

Don’t miss this opportunity to enhance your data engineering skills and boost your productivity. Enroll now, and start your journey towards data engineering excellence today.

Prophecy University Courses are Developed and hosted on udemy

Conclusion: Empower all data users to harness the potential of Databricks

In the ever-evolving landscape of data engineering, Prophecy emerges as a powerful ally. By seamlessly integrating with Databricks, Prophecy revolutionizes the way you develop, deploy, and manage data pipelines. Here's a recap of the essential steps:

1. Set up the environment for development and execution:
   - Seamlessly connect to Prophecy within your Databricks environment via Prophecy's Partner Connect page.
   - Leverage Databricks clusters to harness the full potential of Apache Spark.
   - Create a Project for development and deployment
2. Read and parse raw data from sources:
   - Create a data source to read Salesforce data from object storage.
   - Data source best practices for parsing data from object storage.
3. Build a visual pipeline to transform Salesforce data using copilot:
   - Say goodbye to complex code! Use Prophecy's visual, drag-and-drop interface to design data pipelines.
   - Leverage Data Copilot for intelligent suggestions and automated tasks.
   - Execute your pipelines on Databricks clusters with confidence through a single click from the Prophecy UI, knowing that Prophecy leverages its scalability and adheres to best practices.
4. Schedule and Run Pipeline Workflows on Databricks:
   - Schedule pipelines with Databricks Workflows integration.
   - Execute pipelines periodically and choose appropriate cluster for workflow runs.
5. Commit and merge Spark code using Git:
   - Commit and merge pipeline code to designated git repository.
   - Establish data engineering standards across your organization.

Ready to unlock the full potential of Databricks across your enterprise?

Explore how Prophecy can empower your data team: **Learn More**

Prophecy

Prophecy is a low-code data transformation platform that offers an easy-to-use visual interface to build, deploy, and manage data pipelines with software engineering best practices. Prophecy is trusted by enterprises including multiple companies in the Fortune 50 where hundreds of engineers run thousands of ETL workloads every day. Prophecy is backed by some of the top VCs including Insight Partners and SignalFire. Learn how Prophecy can help your data engineering in the cloud at www.prophecy.io.