What does Databricks do?

The $28B private company nobody understands

Justin
May 27, 2021

The TL;DR

Databricks sells a data science and analytics platform – i.e. a place to query and share data – built on top of an open source package called Apache Spark.

- Apache Spark is an open source engine for **running analytics and machine learning** across distributed, giant datasets
- Spark is notoriously **hard to run on your own infrastructure** and companies often don't have the expertise to do that
- Databricks provides a **managed service for running Spark clusters**, as well as notebooks for visualization and exploration, plus the ability to schedule pipelines
- More recently, Databricks has been **expanding the product portfolio** to include ML and data warehousing

This is a pretty big company, all things considered - $28B was their most recent valuation, making it one of the most valuable private companies on the planet. And they're planning on going public in 2021.

Apache Spark, the OG

Since Databricks is built on top of this open source "Spark" thing, understanding Databricks means understanding Spark. So what's Apache Spark exactly?

Spark is a tool for **running distributed data pipelines** (think: query this, move this to that place). As teams started storing more and more data than ever before, it stopped making sense to put all of it on a single server:

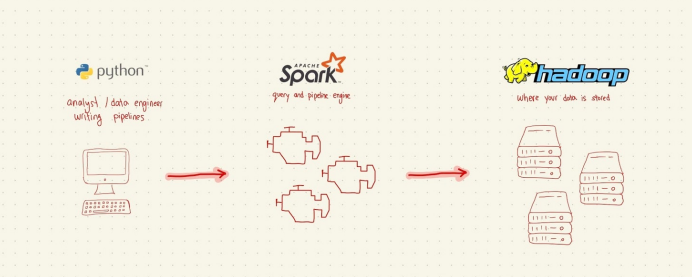
- **Storage**: if you've got 1 petabyte (one million gigabytes), you'd need to get a server with that much storage, which literally doesn't exist. Plenty of teams are working with more data than that,
- **Speed**: writing queries, running pipelines, and building models would be *very very* slow if all of your data is in one place.

In a distributed system, data gets stored on *different servers* (some pieces here, some there) that stay in sync with each other. When you query that data, your query engine figures out where the data you need is and fetches it from there. One of the first such storage and query engines was Hadoop and the HDFS file system, which you've probably heard of.

Spark exists in this universe, but at a higher level of abstraction - it provides APIs for running distributed "jobs" like queries or pipelines. To get concrete, here's something you might write in Spark (from their homepage):

```
df = spark.read.json("logs.json") df.where("age > 21")   .select("name.first").show()
```

This bit of Python code reads some log files, filters them for people with an age over 21, and shows the "name.first" column. And while this might seem simple, Spark is taking care of a lot of complexity on the backend around distributed queries. And it's very popular (almost 30K Github stars) and highly adopted among Data Science teams (we used it at DigitalOcean).



Distributed systems are very, very complicated (and not just in the data realm). This isn't the kind of thing that your typical software engineer is going to be comfortable configuring and setting up from scratch. So setting up a Spark "cluster" (a group of servers) is pretty difficult. And that's where Databricks comes in.

> 🔍 **Deeper Look**🔍
>
> One thing to note is that while Spark itself is a distributed system, it can be used to query data that's not distributed. An example is using an S3 bucket to back your Spark cluster. In that case, the value of Spark is as a distributed query engine. Reminder: **Spark is not a place to store your data**. It's a place to query and analyze your already-stored data.
>
> 🔍 **Deeper Look**🔍

The core Databricks product

Surprisingly, nestled deep within a clandestine FAQ section on their site, Databricks does a half decent job of explaining what the core product does:

> *Apache Spark™ made a big step towards achieving this mission by providing a unified framework for building data pipelines. Databricks takes this further by providing a zero-management cloud platform built around Spark that delivers 1) fully managed Spark clusters, 2) an interactive workspace for exploration and visualization, 3) a production pipeline scheduler, and 4) a platform for powering your favorite Spark-based applications. So instead of tackling data headaches, you can finally focus on finding answers that make an immediate impact on your business.*

Let's break these down one by one:

1. **A fully managed Spark cluster**

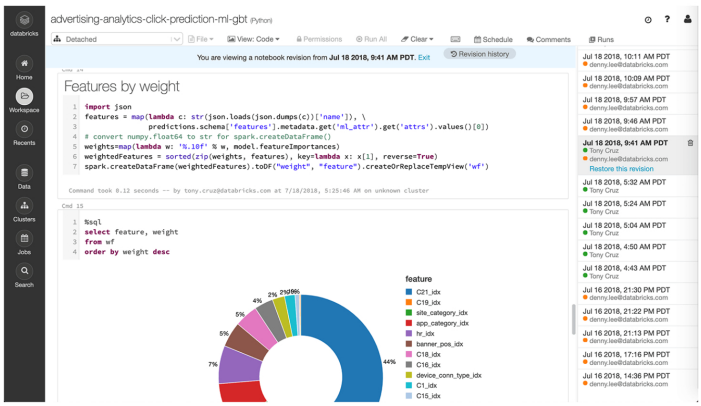
As mentioned, Spark clusters are pretty hard to create and manage. Databricks takes care of your infrastructure for you so you can focus on writing queries and pipelines.

> ⛓ **Related Concepts** ⛓
>
> The "distributed infrastructure is hard" problem isn't unique to Spark - the same narrative is true for MongoDB, Elastic, Redis, Kubernetes, etc. All of these platforms have managed services you can pay for that lets you avoid managing infrastructure for a fee. And we typically call this PaaS (platform as a service).
>
> ⛓ **Related Concepts** ⛓

Currently, Databricks supports deploying on AWS and Azure (which is making a big marketing push). By the way - the founders of Databricks are the same people who originally built Spark.

2. **An interactive workspace for exploration and visualization**

Databricks gives you a notebook-like interface to write Spark jobs (like that Python code we saw above) and make nice graphs.

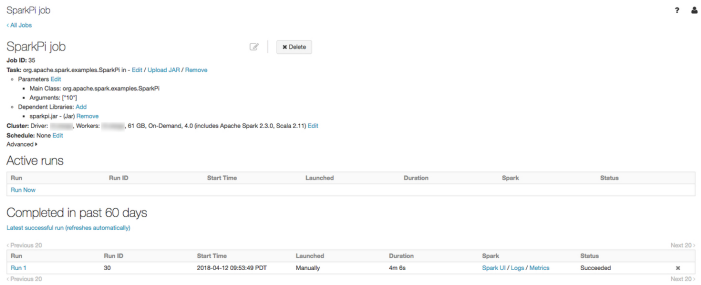


Compare this to a Jupyter notebook (the most popular notebook-like interface for data scientists), but with a lot less configuration overhead and more features

data scientists), but with a lot less configuration overhead and more features around collaboration.

3. **A production pipeline scheduler**

Databricks provides a scheduler for running your data pipelines on a regular...schedule. This feature is directly competitive with ETL engines like Airflow and Prefect. If you have a big job that aggregates your billing data into an analysis-ready format, and you want to run it every day at 6AM, you can use Databricks for that.



4. **A platform for powering your favorite Spark-based applications**

I have no idea what this means beyond what we've already covered, which is managed Spark clusters. It could be a general catch all for other product lines, so let's talk about those!

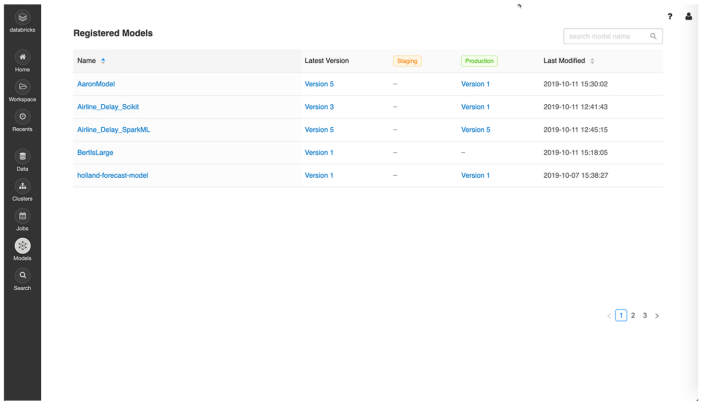
(Yes, even I am often stumped by the careless copywriting of enterprise marketing teams)

New Databricks product lines

New company, old story – as Databricks has grown, they've expanded their product suite to include more use case specific services that usually aren't highly adopted, but help lock customers into their ecosystem. A couple of examples:

1. **MLFlow**

Databricks built and maintains an open source package called MLFlow, which is a full platform for the machine learning lifecycle (a post on that incoming one day). While the package is open source, Databricks offers a paid service that manages MLFlow infrastructure for you called Managed MLFlow.



MLFlow doesn't *necessarily* use Spark, which is why it was a pretty significant move for Databricks (which bills themselves as "the Spark company") when they released it in 2018.

2. **Delta Lake**

Moving more into the analytics and BI realm, Databricks recently released a pretty interesting solution that lets you query your data *lake* as if it were a data *warehouse*. For a quick refresher, data lakes are big, unstructured places for you to store raw data really cheaply, while warehouses are for structured data that needs to be queried quickly. The new Delta Lake product purports to give you data warehouse speeds when querying your data lake, so you can keep storage costs really low.

Logistically, Delta Lake is an open source layer that sits on top of your typical data lakes, like S3 or HDFS. I *think* the open source version is from Databricks

originally, but they seem to be deliberately obscuring that fact under the guise of a shell company called LF Projects. If only I ran a true crime podcast...

Further reading

- The partnership between Microsoft (Azure) and Databricks is a lot more involved than your typical "we deploy on Azure" - Databricks calls it a "first party service" in their release post

`A rant:`

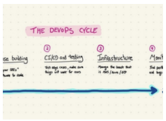
The TechCrunch post announcing the most recent Databricks funding round has a kinda sus opening line that is indicative of the general lack of understanding that pervades tech about, well, tech:

"Databricks is a SaaS business built on top of a bunch of open-source tools, and apparently it's been going pretty well on the business side of things."

Here at Technically, we endeavor to *actually understand* the technology we use, see, and hear about. And so when describing Databricks to your friends and family (don't do this), explain it through the lens of why people use it and what it actually does, not that fact that it's "built on open source tools" like 1,000 other companies.

A basic explainer of what that outage was all about

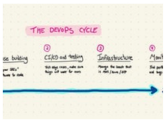
Justin Oct 5, 2021 29 4



What does GitLab do?

The TL;DR GitLab is a somewhat contrarian take on DevOps: it's basically one giant tool for literally anything you'd want to do relating to building and...

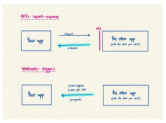
Justin Jan 4 8



What's DevOps?

IT has a cool new name

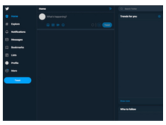
Justin Jan 5, 2021 34



What are webhooks?

Triggered

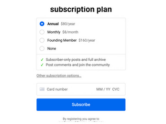
Justin Sep 13, 2021 28 5



What's Headless E-Commerce?

We may be running out of names

Justin Nov 2, 2021 20 7



I built a (basic) Substack clone in a month

This was probably a waste of my time

Justin Sep 2, 2020 50 4

See all ›