

# Presto: Make sense of all your data, any size, anywhere

Get the insights you need with Presto, a fast and flexible open-source SQL query engine.



# Table of contents

03

Introduction

09

Common Presto  
use cases

04

Presto:  
A brief history

11

Getting started with  
Presto and watsonx.data

05

Diving into the  
Presto technology

# Introduction

Data is the lifeblood of today's businesses. It comes in many different shapes and sizes—structured data stored in tables, unstructured data such as objects and files, data at the edge from mobile and IoT devices, data in the cloud and data on premises. And the amount of data is exploding—more data is being created and stored in many different ways and places than ever before.

But storing data is just the beginning. In addition to producing and collecting data, company teams need to combine and analyze it. They need actionable insights on their stored data, often in real time, to make business-critical decisions. To that end, companies must leverage tools that can query data fast, across many different data sources.

While it may not be new, Structured Query Language (SQL), a common query language, is now more popular than ever before to address this need. Developed in the 1970s by IBM researchers Raymond Boyce and Donald Chamberlin, SQL is the standard and is the most widely used programming language for databases. SQL explores data and helps make sense of it. It can provide those real-time, actionable insights that organizations need.

And this is where Presto comes in. Presto is a fast, flexible distributed SQL query engine. It provides ANSI SQL and enables querying of data where it lives, including Hadoop, Hive, AWS S3, Cassandra, MongoDB, Elasticsearch and more. This means you can run standard SQL on any of these data sources.

Presto is designed to be interactive and can handle petabytes of data in your data lake, warehouse or other data source. Most queries can be finished in real time or near-real time. Expect fast insights on data of any type and any size; Presto is also completely open source.

In this white paper, we'll explore the origins of Presto, what it is and how it provides federated queries, some common use cases deployed at companies today, and how to get started with Presto.

# Presto: A brief history

The Presto project came out of Facebook. Back in 2012, Facebook was leveraging Apache Hive to perform extemporized data analysis on massive data sets. As their data sets continued to grow exponentially, the team found that Hive was just too slow because of its foundation—MapReduce—which at the time required intermediate data sets to be persisted to disk. That meant a lot of I/O to disk for transient, intermediate result sets, and this would not work.

So, the Facebook team developed Presto, a new distributed SQL query engine, designed as an in-memory engine without the need to persist intermediate result sets for a single query. Unlike Hive, Presto could process the same query orders much faster, completing many queries with sub-second latency.

In 2013, Facebook made Presto open source under the GitHub repository “prestodb.” Early adopters and collaborators included companies such as Airbnb, Uber, Twitter and Netflix. Today, hundreds of companies use Presto in production. Because of its flexibility and extensibility, Presto can support a variety of SQL use cases. Organizations typically start with a Presto cluster for their interactive, extemporized queries and then expand from there.

Today Presto is under the auspices of The Linux® Foundation’s Presto Foundation. The Presto Foundation’s charter ensures the open governance and transparency of the project with no one vendor controlling it. There’s a whole industry consortium forming around Presto, and current members of the Presto Foundation are Meta, Uber, Alibaba Cloud, Alluxio, Denodo, Hewlett Packard Enterprise, Intel and Ahana, an IBM company. Any company is welcome to join to help drive the Presto technology and community, and to support the development and community processes.

# Diving into the Presto technology

## SQL is the new SQL

Presto is a federated, distributed SQL query engine that runs on a cluster of machines. It enables interactive, extemporized analytics on large amounts of data. Presto makes it possible to query data where it lives, in data sources such as Hive, AWS S3, Hadoop, Cassandra, relational databases, NoSQL databases, or even proprietary data stores. Presto allows users to access data from multiple sources, allowing for analytics across an entire organization.

Being a distributed SQL engine is what makes Presto so appealing. SQL is the common language of data systems. Even after over 40 years, SQL has not only stood the test of time but is gaining popularity because it's the most common way to work with any database. Using the ANSI SQL standard is important for a federated system such as Presto because you can continue to use common commands such as SELECT, UPDATE, DELETE, INSERT and WHERE.

Because it's based on SQL, Presto is easily accessible and doesn't require any advanced or new learning. This SQL compliance immediately enables a large number of use cases. SQL code that works on other databases will likely work on Presto, ready for immediate use, without any changes to the SQL. And dozens of popular business intelligence (BI) tools will work with the Presto engine without any additional integration.

## Presto: The federated query engine

As the amount of data and the types of data formats and data storage have continued to grow, federated databases have become more important. A federated database is a system that connects many different databases into a single database, which can then be queried.

Presto is a federated query engine that queries data across multiple external data sources. It can access data in place. In other words, you don't need to move data to query it with Presto. This is extremely valuable when it comes to cost, time and resources. Instead of moving data or using the extract-transform-load (ETL) process to integrate your data in one place to query it, Presto can provide unified access to—and federate queries across—all of your data sources.

Presto connectors include over a dozen today, a number that continues to grow. These connectors are for databases, object stores, data lakes, streaming data and many more types of data systems. In fact, you can build a Presto connector to almost any data system, including AWS S3, AWS Redshift, Hadoop, Microsoft SQL server, MongoDB, and more.

Presto can query a single data source or a join across multiple data sources, such as AWS S3 and MySQL or Kafka and MongoDB. You can query traditional data sources and non-relational data sources (MongoDB, Elasticsearch) and data lakes (HDFS, AWS S3). This is one of the biggest advantages of Presto—you can query across disparate and different data models without having to move your data.

## Presto architecture and technical concepts

Your Presto cluster sits between your BI tools (such as Superset, Tableau and Looker) and your data sources. As previously discussed, Presto queries across many different data sources and provides that data back to your BI tool for your organization.

A full Presto installation includes a coordinator and multiple workers. Queries are submitted from a client such as the Presto CLI to the coordinator. The coordinator parses, analyzes and plans the query execution, and then distributes the processing to the workers.

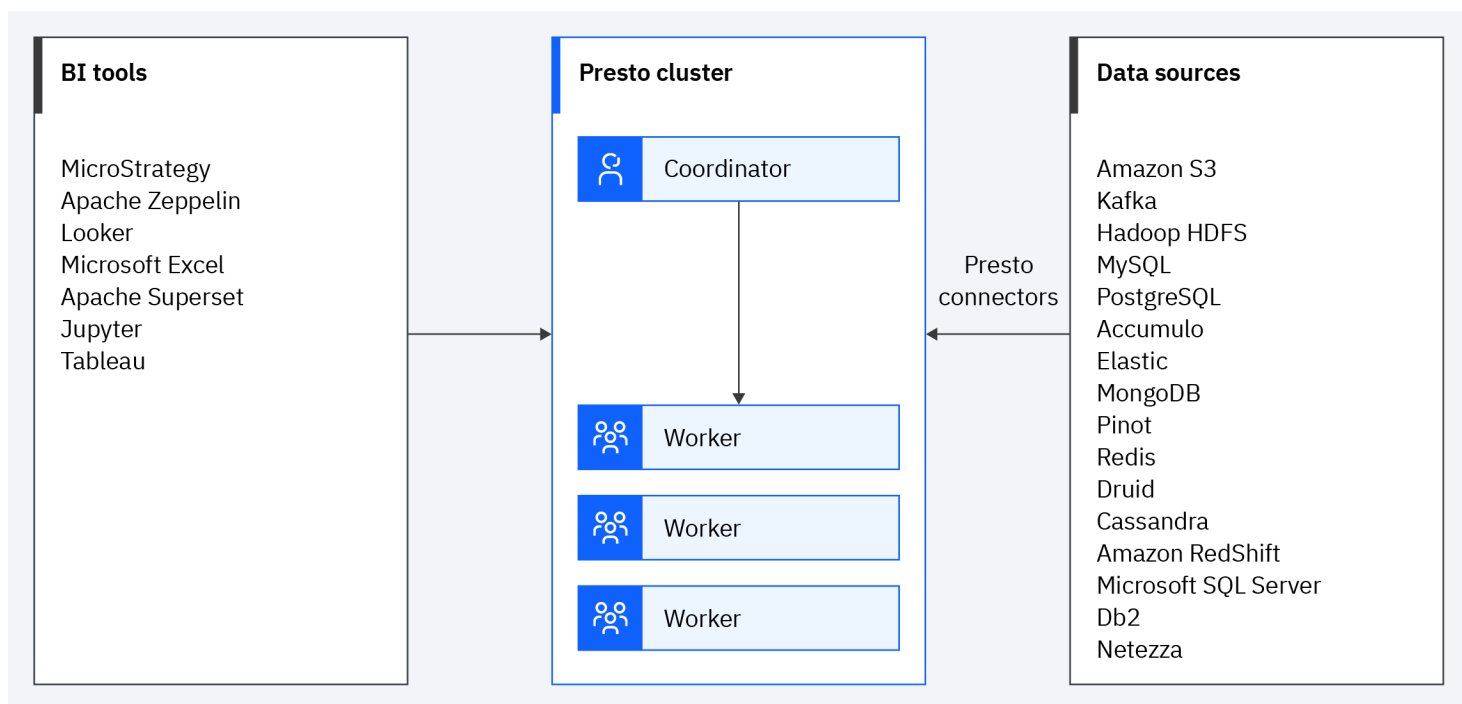


Image: Your Presto cluster sits between your BI tools and your data sources

There are four types of Presto data sources that you need in your deployment:

**Connector**

A connector adapts Presto to a data source such as Hive or to a relational database. Presto contains several built-in connectors including JMX, a Hive connector and a TPCCH connector.

**Catalog**

A Presto catalog contains schemas and references a data source using a connector.

**Schema**

A schema is used to organize tables for querying.

**Table**

A table is a set of unordered rows organized into named columns with types.

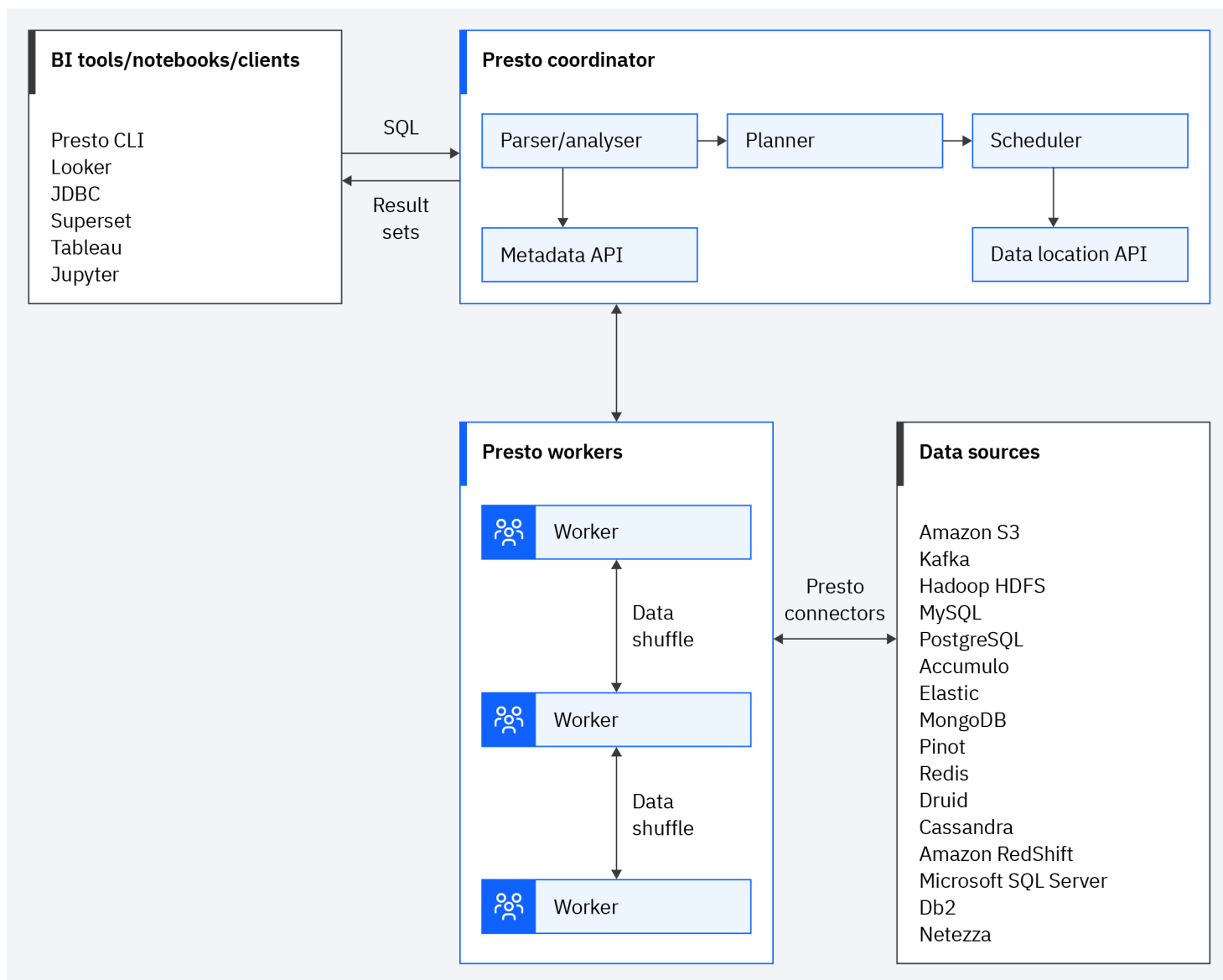


Image: Your Presto cluster sits between your BI tools and your data sources

## Executing a query

Let's look at what happens when you execute a query.

Example: If your query joins together many large tables, it may need multiple stages to execute, aggregating tables together. After each execution stage, there may be intermediate data sets. Unlike distributed query engines such as Hive that were designed to persist intermediate results to disk, Presto saves time by executing queries in the memory of the worker machines. It performs operations on intermediate datasets there, instead of persisting them to disk.

With Presto, data can reside in many different places and Presto performs the executions in memory across your workers, moving data between workers as needed. This process avoids the need to write and read from disk between stages; the result: faster query execution time.

The key pieces of the query execution model are as follows:

### Statement

When Presto parses a statement, it converts the statement into a query and creates a distributed query plan; the query plan is then realized as a series of interconnected stages running on Presto workers.

### Stage

When Presto executes a query, it does so by breaking up the execution into a hierarchy of stages.

### Task

Tasks are the “work horse” in the Presto architecture; a distributed query plan is deconstructed into a series of stages and translated to tasks, which then act upon splits or process splits.

### Split

Tasks operate on splits, which are sections of a larger data set. Stages at the lowest level of a distributed query plan retrieve data using splits from connectors, and intermediate stages at a higher level of a distributed query plan retrieve data from other stages.

### Driver

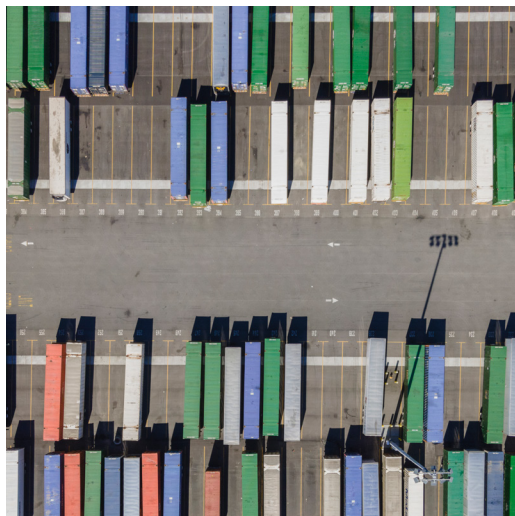
Drivers act upon data and combine operators to produce output that is then aggregated by a task and delivered to another task in another stage.

### Operator

An operator consumes, transforms and produces data.

### Exchange

Exchanges transfer data between Presto nodes for different stages of a query.





# Common Presto use cases

Presto is used for a wide range of analytical use cases and is particularly good for interactive and extemporized querying. Here are five of the most common use cases in use today.

## Extemporized querying

Use SQL to run “one-off” queries whenever you want, wherever your data resides. Presto makes it possible for you to query data where it’s stored, so you don’t have to run ETL to move data into a separate system. With Presto connectors and their in-place execution, platform teams can quickly provide access to datasets that analysts find interesting. Along with that access comes the power of Presto to run queries in seconds instead of hours; analysts can iterate quickly on innovative hypotheses with the interactive exploration of any dataset, residing anywhere.

## Reporting and dashboarding

Query data across multiple sources to build reports and dashboards for self-service business intelligence. Presto gives data scientists and analysts the ability to query data across sources on their own, so they’re not dependent on data platform engineers.

## Transformation using SQL (ETL)

Aggregate terabytes of data across multiple data sources and run efficient ETL queries against that data with Presto. Instead of traditional batch processing systems, use Presto to run resource-efficient and high-throughput queries.

## Data lakehouse analytics

Query data directly on a data lake or lakehouse without the need for transformation. Presto enables you to query any type of data in your data lake, including both structured and unstructured data.

## Federated querying across multiple data sources

Query data across many different data sources including databases, data lakes, lake houses, on premises or in the cloud. Presto allows you to aggregate answers back in the Presto in-memory database.

Who's running Presto in production today? Teams at hundreds of companies. Here are some real-world use cases:

### Adobe

As part of Adobe Advertising, the Adobe Data Processing Platform (ADP) is a behemoth in the world of data analytics that handles an astounding data throughput of billions of events daily. That's in addition to 20 billion user profiles for audience targeting and segmentation and 200 billion auctions per day for the company's real-time ad bidding. Presto plays a central role in powering this platform. For instance, imagine having to sift through data equivalent to the size of New York's population multiple times over, every single day. Presto makes this possible, supporting complex queries, interactive analytics, and dynamic data extraction, powering over 2,000 daily reports and hundreds of pipelines on a 7 PB data lake with over 400 billion records.

### Bolt

Bolt is a transportation company that provides ride hailing, micro-mobility rentals, and food delivery to over 100 million users in 45 countries across Europe and Africa. By leveraging Presto's capabilities, Bolt has been able to address scalability limits, cost efficiency, and workload management challenges. Presto plays a critical role in Bolt's data infrastructure—supporting up to 100,000 daily (1.5 million queries per month) with over 2000 active internal users on a 2 PB data lake.

### Meta

Since 2018, Presto is responsible for supporting much of the SQL analytic workloads at Meta, including interactive and BI queries, and long-running batch ETL jobs. In addition, Presto powers several user-facing analytics tools, serves high-performance dashboards, provides an SQL interface to multiple internal NoSQL systems, and supports Facebook's A/B testing infrastructure. Overall, Presto supports 30,000 queries per day with 1000 daily active users on a 300 PB data lake.

### Uber

In late 2016, Uber had over 2,000 people running more than 100,000 analytic queries daily. To run analytic queries against multiple data sources, the Uber team designed an analytics system that leverages Presto—because of its scalability, high performance and smooth integration with Hadoop—and Parquet, a columnar storage format for Hadoop. Presto now powers over 100 million queries per day with 7,000 weekly active users on a 50 PB data lake.

### ByteDance

Presto has been widely used in ByteDance to power data warehouse applications, BI tools, advertising and more. ByteDance leaders migrated their extemporized workloads from Apache Hive and Apache Spark to Presto to run tens of thousands of compute cores and now serves over 2 million queries per day.

### Twilio

Twilio is a customer engagement platform that makes it possible for businesses to connect with their customers through various APIs and channels of communication. Prior to 2021, Twilio loaded its data into a Redshift data lake for analytical queries and business use cases, using dashboarding BI tools such as Looker. However, as the data grew beyond 5 PB, the company encountered challenges in managing Redshift at scale. To address these issues, Twilio implemented Presto, which allowed teams to decouple the storage and compute layers and scale without affecting performance. Today, Presto supports over 2,700 active users running 1 million queries scanning 40 PB of data per month.



# Getting started with Presto

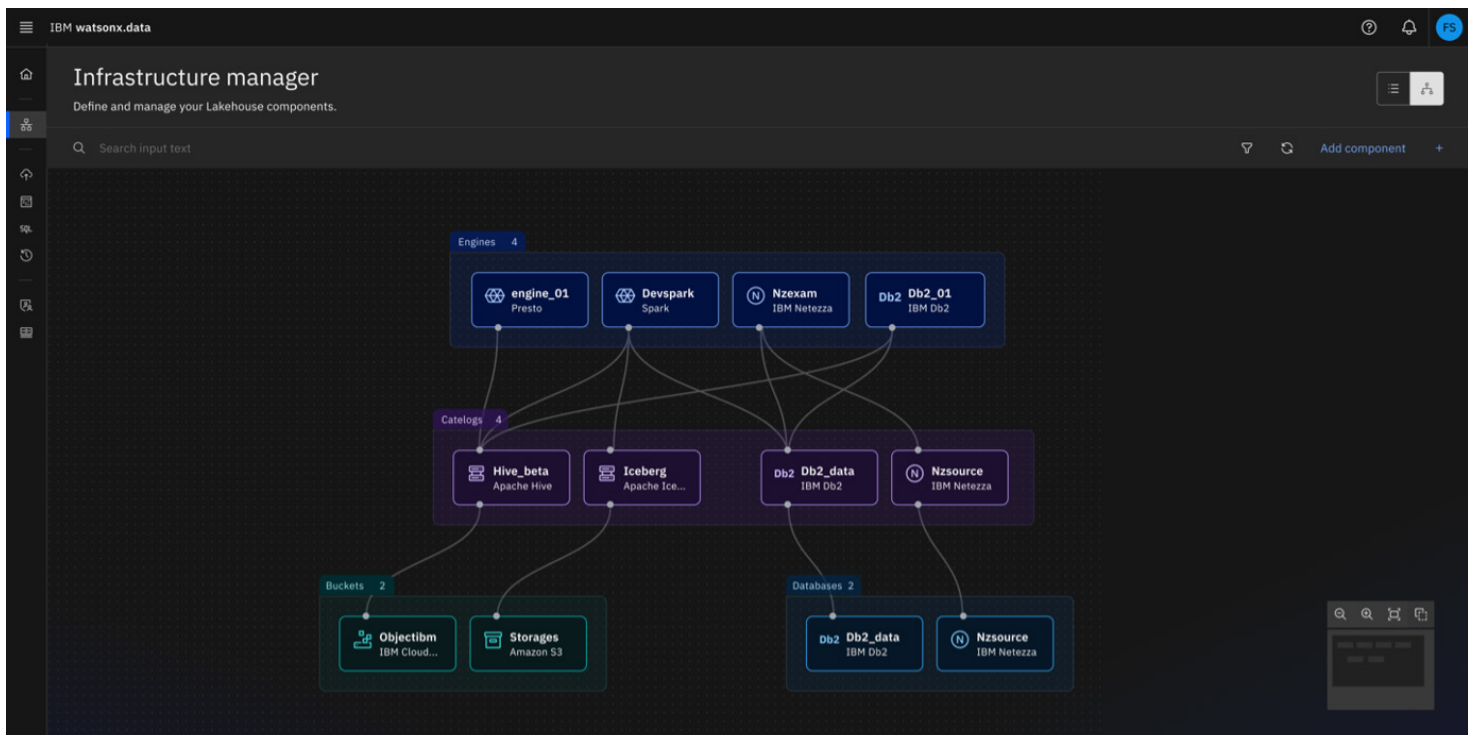


Image: User interface of watsonx.data

Presto is open source; you can install Presto manually and [get started yourself](#). There are a few tutorials to help you get started if you do it this way.

Alternatively, you can use Presto with a data lakehouse solution such as [IBM® watsonx.data™](#). Watsonx.data is a fit-for-purpose data store, built on an open lakehouse architecture, supported by querying, governance and open data formats to access and share data.

Organizational teams can drive analytics costs down with cost-efficient compute and storage and multiple analytics engines—including Presto, Spark, [IBM Db2®](#) and [IBM Netezza®](#)—that dynamically scale up and down. Watsonx.data incorporates the latest performance enhancements to the Presto query engine and simplifies access to Presto, so that your team can get self-service SQL across your business.

With watsonx.data, you can bring your own data and stay in control of your data. In a few clicks, users can connect to existing analytics environments and start deploying fit-for-purpose query engines with integrated metadata and storage through a single point of entry. Seamlessly connect watsonx.data with various types of object storage such as AWS S3, Ceph, or IBM Cloud® object storage, and registered databases such as MongoDB, MySQL, PostgreSQL and more.

Try [watsonx.data](#) for free today or [request a live demo](#) to see Presto and watsonx.data in action.

## More resources

[Join the Presto meetup group →](#)

[Free O'Reilly ebook: Learning and operating Presto →](#)



© Copyright IBM Corporation 2023

IBM Corporation  
New Orchard Road  
Armonk, NY 10504

Produced in the  
United States of America  
August 2023

IBM, the IBM logo, IBM Cloud, watsonx.data, Db2, and Netezza are trademarks or registered trademarks of International Business Machines Corporation, in the United States and/or other countries. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on [ibm.com/trademark](http://ibm.com/trademark).

Intel, Intel logo, Intel Inside, Intel Centrino, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

All client examples cited or described are presented as illustrations of the manner in which some clients have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual client configurations and conditions. Contact IBM to see what we can do for you.

It is the user's responsibility to evaluate and verify the operation of any other products or programs with IBM products and programs.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED "AS IS" WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

Actual available storage capacity may be reported for both uncompressed and compressed data and will vary and may be less than stated.

