Module Overview

- Lesson 1: Introduction to Databricks
- Lesson 2: Azure Databricks and Architecture
- Lesson 3: Price Tiers and Workloads
- Lesson 4: Azure Databricks Artifacts
- Lesson 5: Azure Databricks Clusters
- Lesson 6: Azure Databricks Workspace

Lesson 1: Introduction to Databricks

After completing this lesson, you will be able to:

- Understand the features of Apache Spark
- Learn the basic definition of Databricks and its evolution

What is Big Data?

- Data has become a critical asset for any organization.
- Today's digital age is generating data exponentially.
- A popular definition of big data is the data characterized by 3 Vs.

Volume	Variety	Velocity
Terabytes to Petabytes and more	Structured and Unstructured data: eg, cvs, json, image, IoT …	Accelerating rate of data ingestion and analysis

- https://powerbi.microsoft.com/en-us/blog/what-is-this-thing-called-big-data/
- https://en.wikipedia.org/wiki/Big_data

What is Big Data?

- The challenge is how to get the value out of this data.
- Big data technologies are used to ingest, process and analyze large volume of data at fast pace.
- Big data technologies run on distributed architectures, offering high availability at low cost.

Apache Spark

- Apache Software Foundation (ASF) open-source data processing project built around speed, ease of use, and sophisticated analytics.
- In-memory engine that is up to 100 times faster than Hadoop.
- Largest open-source data project with 1000+ contributors.
- Highly extensible with support for Scala, Java, Python, .NET, and R alongside Spark SQL, GraphX, Streaming and Machine Learning Library (Mllib).

- https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html

When it comes to ease of use, Spark again happens to be a lot better than Hadoop. Spark has APIs for several languages such as Scala, Java and Python, besides having the likes of Spark SQL. It is relatively simple to write user-defined functions. It also happens to boast an interactive mode for running commands. Hadoop, on the other hand, is written in Java and has earned the reputation of being pretty difficult to program, although it does have tools that assist in the process. (To learn more about Spark, see How Apache Spark Helps Rapid Application Development.)

In-Memory Technology

One of the unique aspects of Apache Spark is its unique "in-memory" technology that allows it to be an extremely good data processing system. In this technology, Spark loads all of the data to the internal memory of the system and then unloads it on the disk later. This way, a user can save a part of the processed data on the internal memory and leave the remaining on the disk.

Spark also has an innate ability to load necessary information to its core with the help of its machine learning algorithms. This allows it to be extremely fast.

Spark's Core

Spark's core manages several important functions like setting tasks and interactions as well as producing input/output operations. It can be said to be an RDD, or resilient distributed dataset. Basically, this happens to be a mix of data that is spread across several machines connected via a network. The transformation of this data is created by a four-step method, comprised of mapping the data, sorting it, reducing it and then finally, joining the data.

Following this step is the release of the RDD, which is done with support from an API. This API is a union of three languages: Scala, Java and Python.

Spark's SQL

Apache Spark's SQL has a relatively new data management solution called SchemaRDD. This allows the arrangement of data into many levels and can also query data via a specific language.

Graphx Service

Apache Spark comes with the ability to process graphs or even information that is graphical in nature, thus enabling the easy analysis with a lot of precision.

Streaming

This is a prime part of Spark that allows it to stream large chunks of data with help from the core. It does so by breaking the large data into smaller packets and then transforming them, thereby accelerating the creation of the RDD.

MLib – Machine Learning Library

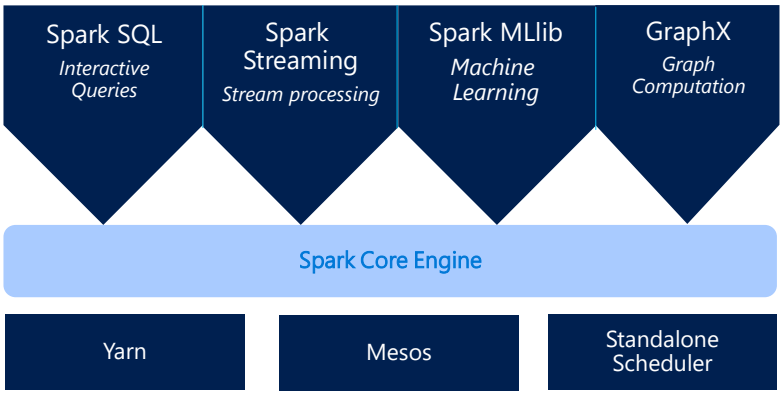
Apache Spark has the MLib, which is a framework meant for structured machine learning. It is also predominantly faster in implementation than Hadoop. MLib is also capable of solving several problems, such as statistical reading, data sampling and premise testing, to name a few.

Apache Spark

A unified, open source, and parallel, and in-memory data processing framework for Big Data Analytics

Spark Unifies:

- Batch Processing
- Interactive SQL
- Stream processing
- Machine Learning
- Deep Learning
- Graph Processing



Apache Spark, a powerful open source data processing engine built around speed, ease of use, and sophisticated analytics, has become the defacto standard for building big data applications.

However, realizing the value and benefits of Spark on its own can be challenging.

Today's data scientists, data engineers and developers need to take Spark and cobble together various complex infrastructure, tools and systems to meet their data needs, severely inhibiting their ability to deliver results.

Some of the major challenges with Apache Spark are:

1. Cannot run multiple versions of Spark
2. There is no built-in file system optimized for cloud storage access
3. There is no Serverless pools offering auto-configuration of resources for SQL and Python workloads
4. Faster reads with Parquet format
5. Automatic Caching isn't possible right now

And many more... For complete list of all the differences between Apache Spark and Databricks: https://databricks.com/product/comparing-databricks-to-apache-spark

What is Databricks

- It's a managed platform for running Apache Spark
- No cluster management
- No tedious maintenance tasks
- Point-and-click platform for developers that prefer a user interface
- Capabilities to automate aspects of data workloads with automated jobs
- Optimized autoscaling to resize a cluster intelligently

Databricks is a managed platform for running Apache Spark - that means that you do not have to learn complex cluster management concepts nor perform tedious maintenance tasks to take advantage of Spark. Databricks also provides a host of features to help its users be more productive with Spark. It's a point and click platform for those that prefer a user interface like data scientists or data analysts. However, this UI is accompanied by a sophisticated API for those that want to automate aspects of their data workloads with automated jobs. To meet the needs of enterprises, Databricks also includes features such as role-based access control and other intelligent optimizations that not only improve usability for users but also reduce costs and complexity for administrators.

The Databricks Unified Analytics Platform accelerates innovation by unifying data science, engineering, and business. Not only does it run an optimized version of Spark, offering 10-40x performance gains, but it also offers interactive notebooks, integrated workflows, and full enterprise security.

Knowledge Check

What is the major problem addressed by Databricks?

What are the capabilities of Spark SQL?

How's Databricks being used?

Answers:

1. Major problems addressed by Databricks?

Databricks provides 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.

2. Capabilities of Spark SQL:

- Spark SQL is a Spark module for structured data processing.
- It provides a programming abstraction called DataFrames and can also act as distributed SQL query engine.
- It executes SQL queries. It can execute unmodified Hadoop Hive queries to run up to 100x faster on existing deployments and data.
- It also provides powerful integration with the rest of the Spark ecosystem (e.g., integrating SQL query processing with machine learning).

3. How's Databricks being used?

Customers utilize Databricks platform for a broad spectrum of use cases including core ETL, data discovery and exploration, data warehousing, data product deployment, and insight publishing using dashboards for internal and external audiences.

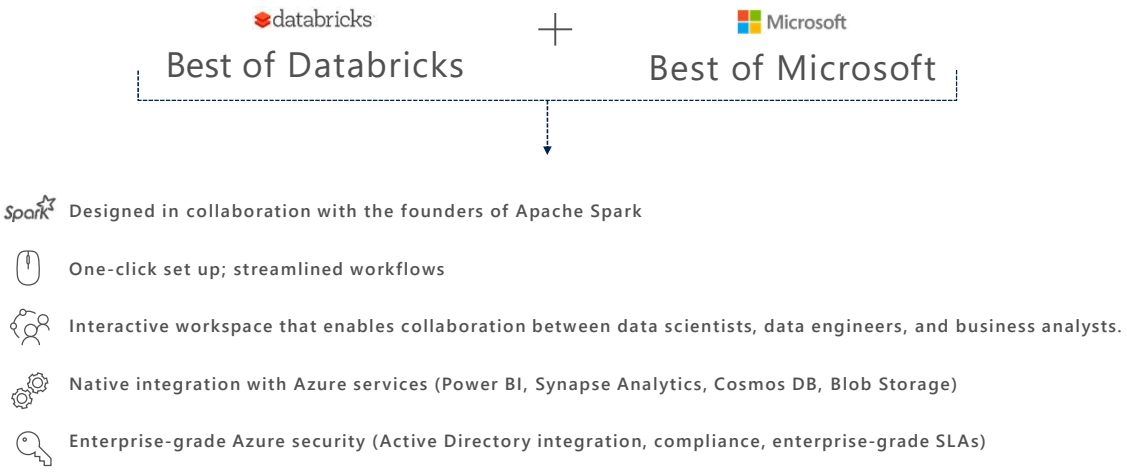
Lesson 2: Azure Databricks and Architecture

After completing this lesson, you will be able to:

- Understand Azure Databricks features and capabilities
- Understand the basics of Azure Databricks architecture

What is Azure Databricks?

A fast, easy and collaborative Apache® Spark™ based analytics platform optimized for Azure



Azure Databricks features –

Enhance your teams' productivity

- **Get started quickly** by launching your new Spark environment with one click.
- **Share your insights in powerful ways** through rich integration with PowerBI.
- **Improve collaboration** amongst your analytics team through a unified workspace.
- **Innovate faster** with native integration with rest of Azure platform.

Build on the most compliant and trusted cloud

- **Simplify security and identity control** with built-in integration with Active Directory.
- **Regulate access** with fine-grained user permissions to Azure Databricks' notebooks, clusters, jobs and data.
- **Build with confidence on the trusted cloud** backed by unmatched support, compliance and SLAs.

Scale without limits

- **Operate at massive scale** without limits globally.
- **Accelerate data processing** with the fastest Spark engine.

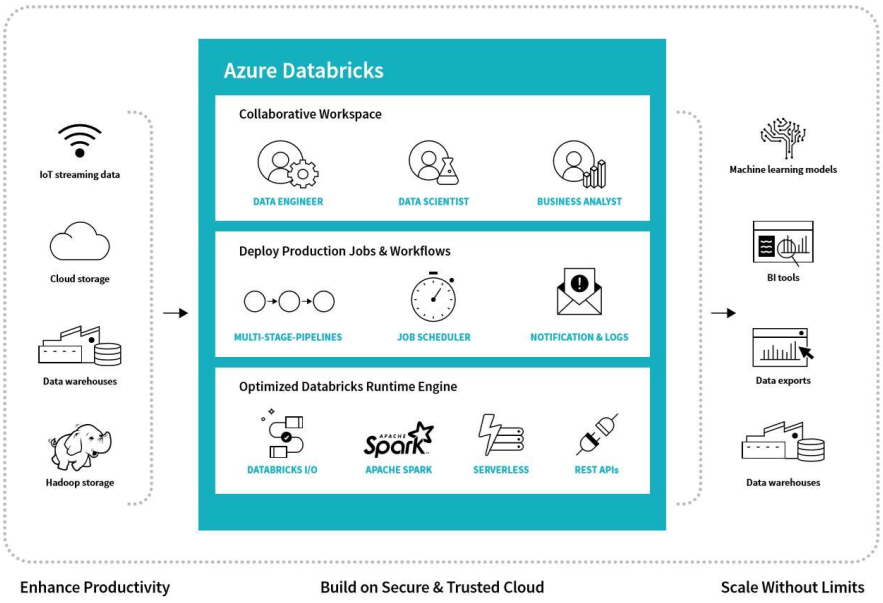
Azure Databricks

- Azure Databricks is a “first party” service on Azure
  - Unlike with other cloud services, it is not an Azure Marketplace, or a 3rd party hosted service
- Azure Databricks is integrated seamlessly with Azure services
  - Azure Portal: Services can be launched directly from Azure Portal
  - Azure Storage Services
  - Azure Active Directory (AAD): For user authentication
  - Azure Synapse and others
- Eliminates the need to create a separate account with Databricks

Azure Databricks is a “first party” Microsoft service, the result of a unique collaboration between the Microsoft and Databricks teams to provide Databricks’ Apache Spark-based analytics service as an integral part of the Microsoft Azure platform. It is natively integrated with Microsoft Azure in a number of ways ranging from a single click start to a unified billing. Azure Databricks leverages Azure’s security and seamlessly integrates with Azure services such as Azure Active Directory, SQL Data Warehouse, and Power BI. It also provides fine-grained user permissions, enabling secure access to Databricks notebooks, clusters, jobs and data.

Azure Databricks

- The Azure Databricks service sits inside the Azure cloud
- Access all your Azure data sources to apply the power of the Azure Databricks analytics engine
- Distribute your results by writing to visual dashboards or back to data warehouses for analytics



Azure Databricks runs on Spark. Data Engineers/Scientists and Business Analysts can all work together in a collaborative space to accomplish batch/stream processing, machine learning, and report authoring.

With Azure Databricks we have the ability to enhance productivity, build upon a secure & trusted Cloud, and Scale without limits.

Scale

- Operate at Massive Scale without Limits, Globally

Databricks enables your analytics processes to **scale up and down automatically**, enabling you to process all your data at once.

- Optimized Performance

Improve performance by as much as 10-100x over traditional Apache Spark deployments with performance optimizations including caching, indexing, and advanced query optimization.

SCALE WITHOUT LIMITS

Globally scale your analytics and data science projects.
Build and innovate faster using machine learning capabilities.

Add capacity instantly. Reduce cost and complexity with a fully-managed, cloud-native platform.

Target any size data or project using a complete set of analytics technologies including SQL, Streaming, MLlib, and Graph.

Security

- Simplify Security and Identity Control
  Built-in integration with Azure Active Directory takes advantage of your existing roles and security settings.
- Build with Confidence
  Azure Databricks is backed by support, compliance and SLAs on the most-trusted cloud platform.
- Regulate Access
  Set fine-grained user permissions to Azure Databricks Notebooks, clusters, jobs, and data with different levels of permissions.

BUILD ON A SECURE, TRUSTED CLOUD

Azure Databricks is uniquely architected to protect your data and business with enterprise-level security that aligns with any compliance requirements your organization may have.

Protect your data and business with Azure Active Directory integration, role-based controls, and enterprise-grade SLAs.

Get peace of mind with fine-grained user permissions, enabling secure access to Databricks notebooks, clusters, jobs and data.

Increase Productivity & Collaboration

- Instant Productivity
  Users can launch a new Spark environment on Azure with a single click using Azure Portal.
- Seamless Collaboration
  A unified workspace provides interactive Notebooks and dashboards for real-time collaboration. Features such as seeing where each other is working in Notebooks, to the ability to add comments, enables users to work synchronously or asynchronously.
- Sharable Insights
  With rich Power BI integration, interactive visualizations can be shared across the organization, allowing for instant feedback and leading quickly to the next business question.

INCREASE PRODUCTIVITY AND COLLABORATION

Azure Databricks delivers the best of Azure and Apache Spark so that data science teams can be immediately productive.

Bring teams together in an interactive workspace. From data gathering to model creation, use Databricks notebooks to unify the process and instantly deploy to production.

Launch your new Spark environment with a single click.

Integrate effortlessly with a wide variety of data stores and services such as Azure SQL Data Warehouse, Azure Cosmos DB, Azure Data Lake Store, Azure Blob storage, and Azure Event Hub. Add artificial intelligence (AI) capabilities instantly and share insights through rich integration with Power BI.

Azure Databricks Environments

Azure Databricks offers three environments for developing data intensive applications

- **Data Science & Engineering**

  Provides an interactive workspace that enables collaboration between data engineers, data scientists, and machine learning engineers. Sometimes called simply "Workspace".

- **Machine Learning**

  An integrated end-to-end machine learning environment, incorporating managed services for experiment tracking, model training, feature development and management, and feature and model serving

- **SQL**

  Provides an easy-to-use platform for analysts who want to run SQL queries on their data lake

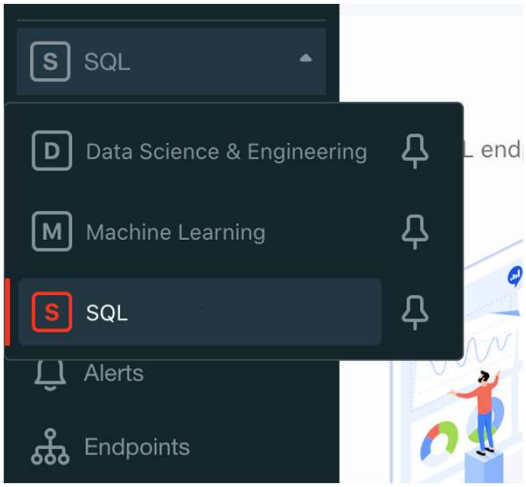
Databricks SQL provides an easy-to-use platform for analysts who want to run SQL queries on their data lake, create multiple visualization types to explore query results from different perspectives, and build and share dashboards.

Databricks Data Science & Engineering provides an interactive workspace that enables collaboration between data engineers, data scientists, and machine learning engineers. For a big data pipeline, the data (raw or structured) is ingested into Azure through Azure Data Factory in batches, or streamed near real-time using Apache Kafka, Event Hub, or IoT Hub. This data lands in a data lake for long term persisted storage, in Azure Blob Storage or Azure Data Lake Storage. As part of your analytics workflow, use Azure Databricks to read data from multiple data sources and turn it into breakthrough insights using Spark.

Databricks Machine Learning is an integrated end-to-end machine learning environment incorporating managed services for experiment tracking, model training, feature development and management, and feature and model serving.

Azure Databricks Environments

To select an environment, launch an Azure Databricks workspace and use the persona switcher in the sidebar



Azure Databricks - Data Science & Engineering

- An analytics platform based on Apache Spark
- Integrated with Azure to provide one-click setup
- Streamlined workflows
- An interactive workspace that enables collaboration between data engineers, data scientists, and machine learning engineers
- Comprises the complete open-source Apache Spark cluster technologies and capabilities

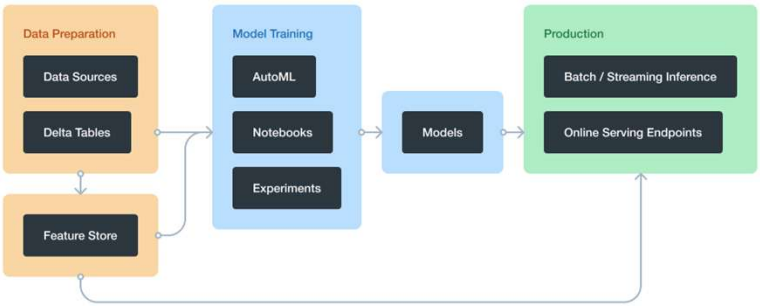
Our labs and demos will use this environment

https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks-ws

Azure Databricks – Machine Learning

An integrated end-to-end machine learning platform incorporating managed services for:

- experiment tracking, model training,
- feature development and management,
- feature and model serving.



https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks-ml

With Databricks Machine Learning, you can:

- Train models either manually or with AutoML.
- Track training parameters and models using experiments with MLflow tracking
- Create feature tables and access them for model training and inference.
- Share, manage, and serve models using Model Registry

Azure Databricks – SQL

- Allows you to run quick ad-hoc SQL queries
  - Queries support multiple visualization types
- Fully managed SQL endpoints in the cloud
- Dashboards for sharing insights
- Alerts help you monitor and integrate
- Enterprise security
  - Integration with Azure Active Directory
  - Role based access (alerts, dashboards, SQL endpoints, queries and data)
  - Enterprise-grade SLAs
- Integration with Azure Services and Power BI

https://docs.microsoft.com/en-us/azure/databricks/scenarios/what-is-azure-databricks-sqla

Fully managed SQL endpoints in the cloud

SQL queries run on fully managed SQL endpoints sized according to query latency and number of concurrent users.

Dashboards for sharing insights

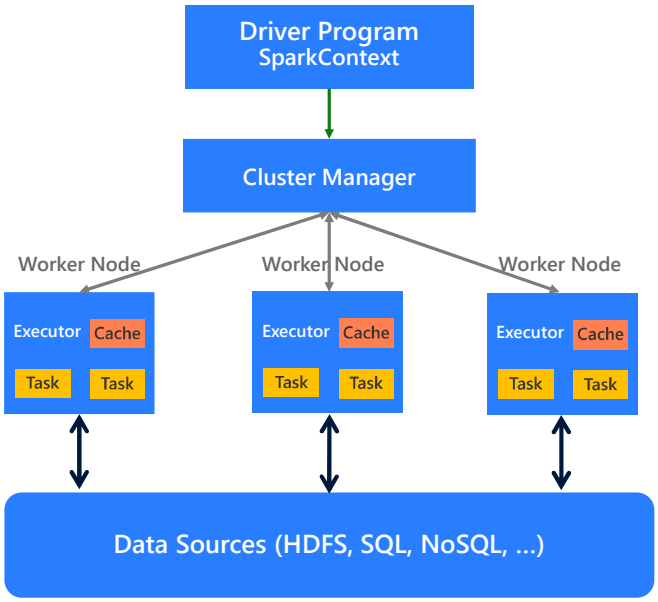
Dashboards let you combine visualizations and text to share insights drawn from your queries.

Alerts help you monitor and integrate

Alerts notify you when a field returned by a query meets a threshold. Use alerts to monitor your business or integrate them with tools to start workflows such as user onboarding or support tickets.

General Spark Architecture

- 'Driver' runs the user's 'main' function and executes the various parallel operations on the worker nodes.
- The results of the operations are collected by the driver
- The worker nodes read and write data from/to Data Sources including HDFS, ADLS, others.
- Worker node also cache transformed data in memory.
- Worker nodes and the Driver Node execute as VMs in public clouds



Spark applications run as independent sets of processes on a cluster, coordinated by the SparkContext object in your main program (called the *driver program*).

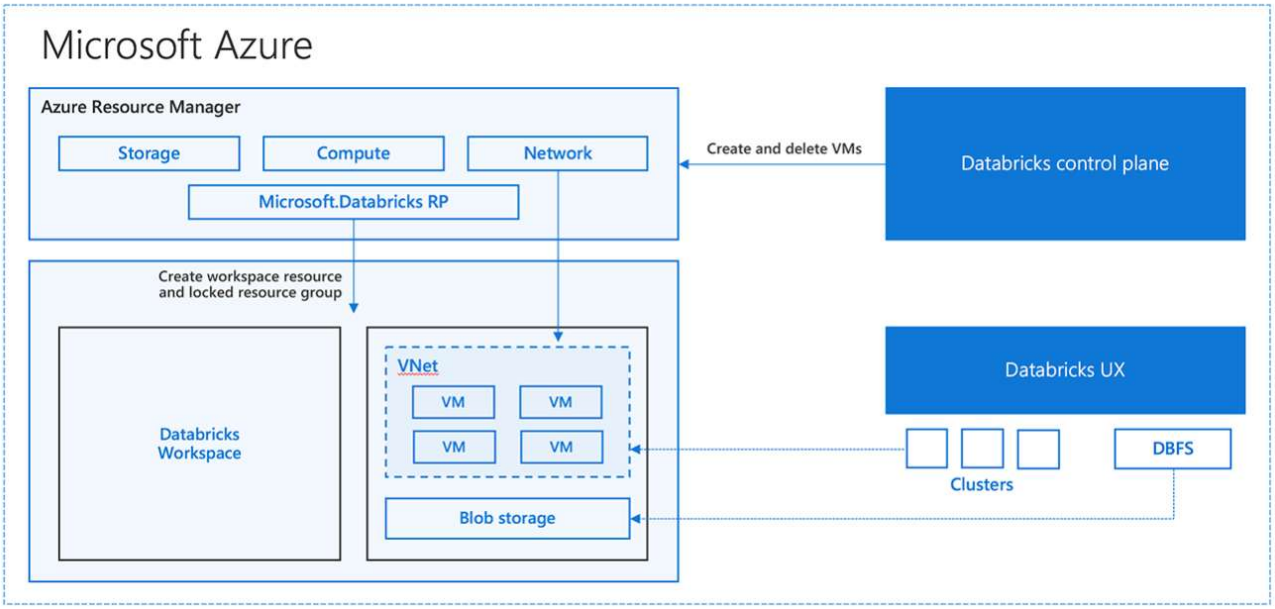
Specifically, to run on a cluster, the SparkContext can connect to several types of *cluster managers* (either Spark's own standalone cluster manager, Mesos or YARN), which allocate resources across applications. Once connected, Spark acquires *executors* on nodes in the cluster, which are processes that run computations and store data for your application. Next, it sends your application code (defined by JAR or Python files passed to SparkContext) to the executors. Finally, SparkContext sends *tasks* to the executors to run.

There are several useful things to note about this architecture:

1. Each application gets its own executor processes, which stay up for the duration of the whole application and run tasks in multiple threads. This has the benefit of isolating applications from each other, on both the scheduling side (each driver schedules its own tasks) and executor side (tasks from different applications run in different JVMs). However, it also means that data cannot be shared across different Spark applications (instances of SparkContext) without writing it to an external storage system.
2. Spark is agnostic to the underlying cluster manager. As long as it can acquire executor processes, and these communicate with each other, it is relatively easy to run it even on a cluster manager that also supports other applications (e.g. Mesos/YARN).
3. The driver program must listen for and accept incoming connections from its executors throughout its lifetime (e.g., see spark.driver.port in the network config section). As such, the driver program must be network addressable from the worker nodes.
4. Because the driver schedules tasks on the cluster, it should be run close to the worker nodes, preferably on the same local area network. If you'd like to send requests to the cluster remotely, it's better to open an RPC to the driver and have it submit operations from nearby than to run a driver far away from the worker nodes.

More information: https://spark.apache.org/docs/latest/cluster-overview.html

Azure Databricks Cluster Architecture

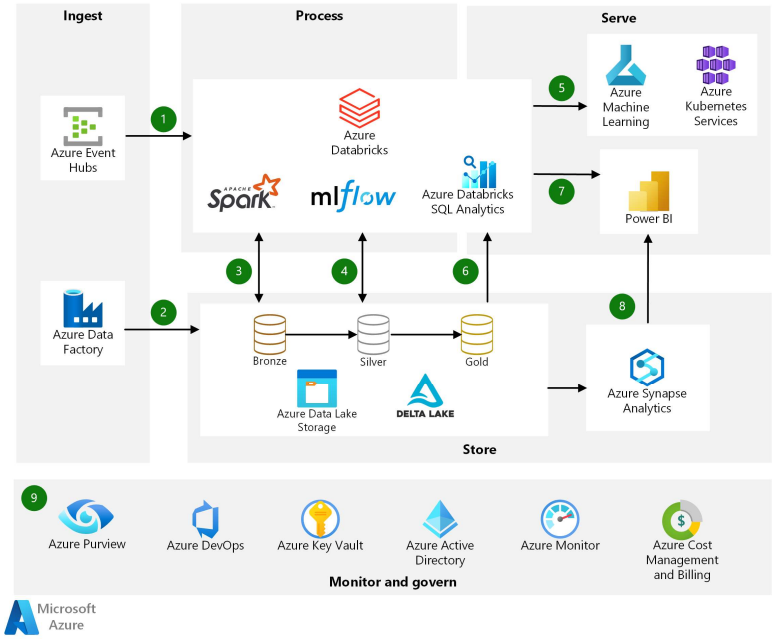


At a high level, the service launches and manages worker nodes in each Azure customer's subscription, letting customers leverage existing management tools within their account.

Specifically, when a customer launches a cluster via Databricks, a "Databricks appliance" is deployed as an Azure resource in the customer's subscription. The customer specifies the types of VMs to use and how many, but Databricks manages all other aspects. In addition to this appliance, a *managed resource group* is deployed into the customer's subscription that we populate with a VNet, a security group, and a storage account. These are concepts Azure users are familiar with. Once these services are ready, users can manage the Databricks cluster through the Azure Databricks UI or through features such as autoscaling. All metadata (such as scheduled jobs) is stored in an Azure Database with geo-replication for fault tolerance.

For users, this design means two things. First, they can easily connect Azure Databricks to any storage resource in their account, e.g., an existing Blob Store subscription or Data Lake. Second, Databricks is managed centrally from the Azure control center, requiring no additional setup.

Modern analytics architecture



https://docs.microsoft.com/en-us/azure/architecture/solution-ideas/articles/azure-databricks-modern-analytics-architecture

This solution outlines a modern data architecture that achieves these goals. Azure Databricks forms the core of the solution. This platform works seamlessly with other services such as Azure Data Lake Storage, Azure Data Factory, Azure Synapse Analytics, and Power BI. Together, these services provide a solution with these qualities:

•Simple: Unified analytics, data science, and machine learning simplify the data architecture.

•Open: The solution supports open-source code, open standards, and open frameworks. It also works with popular integrated development environments (IDEs), libraries, and programming languages. Through native connectors and APIs, the solution works with a broad range of other services, too.

•Collaborative: Data engineers, data scientists, and analysts work together with this solution. They can use collaborative notebooks, IDEs, dashboards, and other tools to access and analyze common underlying data.

NOTES ABOUT GOLD, SILVER and BRONZE layers

1. Data Lake Storage Gen2 houses data of all types, such as structured, unstructured, and semi-structured. It also stores batch and streaming data.

- We organize our data into layers as defined as bronze, silver, and gold:
  - Bronze tables have raw data (JSON,

CSV, Events/IoT, RDBMS data, etc.).

- Silver tables will give a more refined view of our data. We can join fields from various bronze tables to improve streaming records or update account statuses based on recent activity.
- Gold tables give business-level aggregates often used for dashboarding and reporting. This would include aggregations such as weekly sales per store, daily active website users, or gross revenue per quarter by the department.

1.Azure Databricks ingests raw streaming data from Azure Event Hubs.
2.Data Factory loads raw batch data into Data Lake Storage Gen2.
3.For data storage:
    1. Delta Lake forms the curated layer of the data lake. It stores the refined data in an open-source format.
    2. Azure Databricks works well with a medallion architecture that organizes data into layers:
        1. Bronze: Holds raw data.
        2. Silver: Contains cleaned, filtered data.
        3. Gold: Stores aggregated data that's useful for business analytics.
4.The analytical platform ingests data from the disparate batch and streaming sources. Data scientists use this data for these tasks:
    1. Data preparation.
    2. Data exploration.
    3. Model preparation.
    4. Model training.
5.MLflow manages parameter, metric, and model tracking in data science code runs. The coding

possibilities are flexible:

1. Code can be in SQL, Python, R, and Scala.
2. Code can use popular open-source libraries and frameworks such as Koalas, Pandas, and scikit-learn, which are pre-installed and optimized.
3. Practitioners can optimize for performance and cost with single-node and multi-node compute options.

1.Machine learning models are available in several formats:

1. Azure Databricks stores information about models in the MLflow Model Registry. The registry makes models available through batch, streaming, and REST APIs.
2. The solution can also deploy models to Azure Machine Learning web services or Azure Kubernetes Service (AKS).

2.Services that work with the data connect to a single underlying data source to ensure consistency. For instance, users can run SQL queries on the data lake with Azure Databricks SQL Analytics. This service:

1. Provides a query editor and catalog, the query history, basic dashboarding, and alerting.
2. Uses integrated security that includes row-level and column-level permissions.
3. Uses a Photon-powered Delta Engine to accelerate performance

3.Power BI generates analytical and historical reports and dashboards from the unified data platform. This service uses these features when working with Azure Databricks:

1. A built-in Azure Databricks connector for visualizing the underlying data.
2. Optimized Java Database Connectivity (JDBC) and Open Database Connectivity (ODBC) drivers.

4.Users can export gold data sets out of the data lake into Azure Synapse via the optimized Synapse connector. SQL pools in Azure Synapse provide a data warehousing and compute environment.

5.The solution uses Azure services for collaboration, performance, reliability, governance, and security:

1. Microsoft Purview provides data discovery services, sensitive data classification, and governance insights across the data estate.
2. Azure DevOps offers continuous integration and continuous deployment (CI/CD) and other integrated version control features.
3. Azure Key Vault securely manages secrets, keys, and certificates.
4. Azure Active Directory (Azure AD) provides single sign-on (SSO) for Azure Databricks users. Azure Databricks supports automated user provisioning with Azure AD for these tasks:
    1. Creating new users.
    2. Assigning each user an access level.
    3. Removing users and denying them access.
5. Azure Monitor collects and analyzes Azure resource telemetry. By proactively identifying problems, this service maximizes performance and reliability.
6. Azure Cost Management and Billing provide financial governance services for Azure workloads.

Knowledge Check

What are the advantages of using Databricks on Azure?

1. What are the advantages of using Databricks on Azure?

- Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.
- Azure Databricks builds on the capabilities of Spark by providing a zero-management cloud platform that includes:
  1. Fully managed Spark clusters
  2. An interactive workspace for exploration and visualization
  3. A platform for powering your favorite Spark-based applications
- Azure Databricks has a secure and reliable production environment in the cloud, managed and supported by Spark experts.
- Azure Databricks provides enterprise-grade Azure security, including Azure Active Directory integration, role-based controls, and SLAs
- Azure Databricks integrates deeply with Azure databases and stores: SQL Data Warehouse, Cosmos DB, Data Lake Store, and Blob Storage.

Lesson 4: Azure Databricks Artifacts

After completing this lesson, you will be able to:

- Understand the major components of Databricks

Core Artifacts

Clusters / Compute	Set of Azure Linux VMs that host the Spark Driver and Worker Nodes
Workspaces	Enables users to organize, and share, their Notebooks, Libraries and Dashboards
Notebooks	A popular way to develop, and run, Spark Applications
Libraries	Enables external code to be imported and stored into a Workspace
Jobs	Schedule mechanism to submit Spark application code for execution on the Databricks clusters
Secrets	A key-value pair that stores secret material, with a key name unique within a secret scope

Clusters

- Azure Databricks clusters are the set of Azure Linux VMs that host the Spark Driver and Worker Nodes
- Your Spark application code (i.e. Jobs) runs on the provisioned clusters.
- Azure Databricks clusters are launched in your subscription—but are managed through the Azure Databricks portal.
- Azure Databricks provides a comprehensive set of graphical wizards to manage the complete lifecycle of clusters—from creation to termination.

Workspaces

Workspaces enables users to organize—and share—their Notebooks, Libraries and Dashboards

- Workspaces—sort of like Directories— are a convenient way to organize an user's Notebook, Libraries and Dashboards.
- Everything in a workspace is organized into hierarchical folders. Folders can hold Libraries, Notebooks, Dashboard or more (sub) folders.
  - Icons indicate the type of the object contained in a folder
- Every user has one directory that is private and unshared.
  - By default, the workspace and all its contents are available to users.
- Fine grained access control can be defined on workspaces (next slide) to enable *secure collaboration with colleagues*

Notebooks

Notebooks are a popular way to develop, and run, Spark Applications

- Notebooks are not only for authoring Spark applications but can be *run/executed directly* on clusters
  - Shift+Enter
  - click the play button at the top right of the cell in a notebook
  - Submit via Job
- Notebooks support fine grained permissions—so they can be *securely shared* with colleagues for collaboration (see following slide for details on permissions and abilities)
- Notebooks are well-suited for prototyping, rapid

development, exploration, discovery and iterative development

Jobs

Jobs are the mechanism to submit Spark application code for execution on the Databricks clusters

- Spark application code is submitted as a 'Job' for execution on Azure Databricks clusters
- Jobs execute either 'Notebooks' or 'Jars'
- Azure Databricks provide a comprehensive set of graphical tools to create, manage and monitor Jobs.

Libraries

Enables external code to be imported and stored into a Workspace

- Libraries are containers to hold all your *Python, R, Java/Scala* libraries.
- Libraries resides within workspaces or folders.
- Libraries are created by importing the source code
- After importing libraries are immutable—can be deleted or overwritten only.
- You can customize installation of libraries via Init Scripts by writing custom UNIX scripts
- Libraries can also be managed via the Library API

Secrets:

A secret is a key-value pair that stores secret material, with a key name unique within a secret scope.

- There are two types of secret:
- Azure Key Vault-backed
- Databricks-backed
- Libraries resides within workspaces
- Each scope is limited to 1000 secrets. The maximum allowed secret value size is 128 KB.
- Secrets can be created and managed through the databricks-cli
- Secrets can also be managed via the Secrets API

Knowledge Check

What are the core artifacts in Azure Databricks?

What are the core artifacts in Azure Databricks?

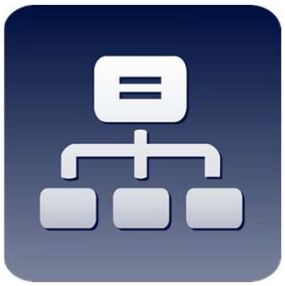
- Clusters
- Workspaces
- Notebooks
- Jobs
- Libraries

Lesson 5: Azure Databricks Clusters

After completing this lesson, you will be able to:

- Understand the Databricks Cluster and how it works?
- Learn about Access Control on Databricks Cluster

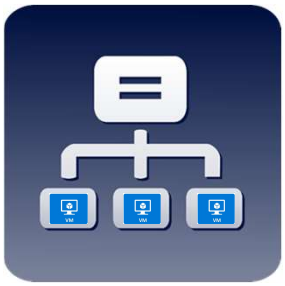
Compute / Cluster



- Clusters are the set of Azure Linux VMs that host the Spark Worker and Driver Nodes
- Spark application code (i.e. Jobs) runs on the provisioned clusters.
- Clusters are launched in your subscription but are managed through the Azure Databricks portal.
  - You can also manage clusters using CLI or API

When provisioning clusters, we can set the type of Azure linux VM's. The default VM for Worker nodes is 14GB memory and 4 cores. The default for Driver nodes is 14GB memory and 4 cores. If more computing power is needed, we can select a different VM type during configuration.

Compute / Cluster



- Azure Databricks provides a comprehensive set of graphical wizards to manage the complete lifecycle of clusters - from creation to termination.
- Types of Cluster:
  - Interactive Clusters: are used to analyze data collaboratively with interactive notebooks.
  - Job Clusters: are used to run fast and robust automated workloads using the UI or API.

Two available types of Clusters. Interactive and Job clusters.

In a nutshell, they perform identically, with the main difference being Job Clusters can be scheduled to spin up to complete a run then spun down once the run is complete and wait for the next scheduled run.

Interactive clusters are great for data scientists and for test environments before moving to production.

Job Clusters are optimal for running data pipelines on a scheduled basis, and are much more cost efficient then having a constantly persisted interactive cluster always incurring costs.

Cluster Access Modes

Azure Databricks supports three access mode

Single User :

- Can be assigned to and used by a single user
- Can run Python, SQL, Scala and R
- Credential Passthrough is not supported
- Support for Unity Catalog

Shared:

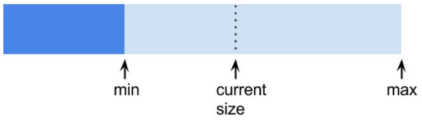
- Can be used by multiple users with data isolation among users
- Can run Python (on DBR version >= 11.1) and SQL workloads.
- Support for Unity Catalog
- Not available in single node

No isolation shared:

- Can be used by multiple users with no data isolation
- No support for Unity Catalog

Cluster Auto Scaling

- Autoscaling allow Databricks to automatically resize your cluster by providing the min and max range of workers.
- When you select autoscaling, the cluster size is automatically adjusted between the minimum and a maximum number of worker limits during the cluster's lifetime.



Cluster Size and Autoscaling

When creating a cluster, you can either provide a fixed number of workers for the cluster or provide a min and max range for the number of workers for the cluster.

When you provide a fixed size cluster, Azure Databricks ensures that your cluster has the specified number of workers. With you provide a range for the number of workers, Databricks chooses the appropriate number of workers required to run your job.

Autoscaling

To allow Databricks to automatically resize your cluster you specify that the cluster is autoscaling and provide the min and max range of workers.

Note

Autoscaling works best with Databricks Runtime 3.4 and above.

When you select autoscaling, the cluster size is automatically adjusted between the minimum and a maximum number of worker limits during the cluster's lifetime.

During runtime Databricks will dynamically reallocate workers to account for the characteristics of your job. Certain parts of your pipeline may be more computationally demanding than others, and Databricks automatically adds additional workers during these phases of your job (and removes when they're no longer needed).

Autoscaling makes it easier to achieve high cluster utilization, because you don't need to provision the cluster to match a workload. This applies especially to workloads whose requirements change over time (like exploring a dataset during the course of a day), but it can also apply to a one-time shorter workload whose provisioning requirements are unknown. Autoscaling thus offers two advantages:

- Workloads can run faster compared to a constant-sized under-provisioned cluster.

- Autoscaling clusters can reduce overall costs compared to a statically-sized cluster.

Depending on the constant size of the cluster and on the workload, autoscaling gives you one or both of these benefits at the same time. Note that the cluster size can go below the minimum number of workers selected when the cloud provider terminates instances. In this case, Databricks continuously retries to increase the cluster size.

You enable autoscaling by selecting the Enable Autoscaling checkbox at the time of cluster creation and entering the number of min and max workers.

Cluster Auto Scaling

- During runtime Databricks will dynamically reallocate workers to account for the characteristics of your job.
- During computationally demanding phases, Databricks automatically adds additional workers and removes when they're no longer needed.

Cluster Size and Autoscaling

When creating a cluster, you can either provide a fixed number of workers for the cluster or provide a min and max range for the number of workers for the cluster.

When you provide a fixed size cluster, Azure Databricks ensures that your cluster has the specified number of workers. With you provide a range for the number of workers, Databricks chooses the appropriate number of workers required to run your job.

Autoscaling

To allow Databricks to automatically resize your cluster you specify that the cluster is autoscaling and provide the min and max range of workers.

Note

Autoscaling works best with Databricks Runtime 3.4 and above.

When you select autoscaling, the cluster size is automatically adjusted between the minimum and a maximum number of worker limits during the cluster's lifetime.

During runtime Databricks will dynamically reallocate workers to account for the characteristics of your job. Certain parts of your pipeline may be more computationally demanding than others, and Databricks automatically adds additional workers during these phases of your job (and removes when they're no longer needed).

Autoscaling makes it easier to achieve high cluster utilization, because you don't need to provision the cluster to match a workload. This applies especially to workloads whose requirements change over time (like exploring a dataset during the course of a day), but it can also apply to a one-time shorter workload whose provisioning requirements are unknown. Autoscaling thus offers two advantages:

- Workloads can run faster compared to a constant-sized under-provisioned cluster.

- Autoscaling clusters can reduce overall costs compared to a statically-sized cluster.

Depending on the constant size of the cluster and on the workload, autoscaling gives you one or both of these benefits at the same time. Note that the cluster size can go below the minimum number of workers selected when the cloud provider terminates instances. In this case, Databricks continuously retries to increase the cluster size.

You enable autoscaling by selecting the Enable Autoscaling checkbox at the time of cluster creation and entering the number of min and max workers.

How Auto Scaling Works?

- Databricks monitors load on Spark clusters and decides whether to scale a cluster up or down and by how much.
- If a cluster has pending Spark tasks, the cluster scales up. If a cluster does not have any pending Spark tasks, the cluster scales down.
- The autoscaling algorithm ensure that users experience fast workloads while maintaining efficient cluster utilization.

Reconfigure autoscaling clusters

If you reconfigure a static cluster to be an autoscaling cluster, Databricks immediately resizes the cluster within the minimum and maximum bounds and then starts autoscaling. As an example, the table below demonstrates what happens to clusters with a certain initial size if you reconfigure a cluster to autoscale between 5 and 10 nodes.

Initial size	Size after reconfiguration
6	6
12	10
3	5

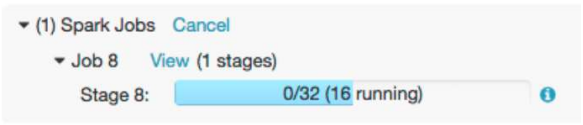
Autoscaling for jobs

Autoscaling for jobs is different from standard autoscaling, and is recommended only with runtime versions 3.4 and above. This feature allows a jobs cluster to scale up and down more aggressively in response to load and is designed to improve resource utilization. In particular, a cluster can scale down idle VMs even when there are tasks running on other VMs. This autoscaling algorithm is different than the one used for standard interactive clusters. To enable this feature for a job running Databricks Runtime 3.4 or higher, select the **Enable Autoscaling** option on the **Configure Cluster** page. For a demonstration of the benefits of job autoscaling, see the blog post on Optimized Autoscaling.

How Auto Scaling Works?

- Clusters with no *pending* tasks *do not* scale up. This usually indicates that the cluster is fully utilized and adding more nodes will not make the processing faster.

For example, this cluster currently has 16 running tasks and 16 pending tasks (total tasks - running tasks) and will be scaled up.



Reconfigure autoscaling clusters

If you reconfigure a static cluster to be an autoscaling cluster, Databricks immediately resizes the cluster within the minimum and maximum bounds and then starts autoscaling. As an example, the table below demonstrates what happens to clusters with a certain initial size if you reconfigure a cluster to autoscale between 5 and 10 nodes.

Initial size	Size after reconfiguration
6	6
12	10
3	5

Autoscaling for jobs

Autoscaling for jobs is different from standard autoscaling, and is recommended only with runtime versions 3.4 and above. This feature allows a jobs cluster to scale up and down more aggressively in response to load and is designed to improve resource utilization. In particular, a cluster can scale down idle VMs even when there are tasks running on other VMs. This autoscaling algorithm is different than the one used for standard interactive clusters. To enable this feature for a job running Databricks Runtime 3.4 or higher, select the **Enable Autoscaling** option on the **Configure Cluster** page. For a demonstration of the benefits of job autoscaling, see the blog post on Optimized Autoscaling.

Cluster Auto Termination

- You can set auto termination for a cluster.
- During cluster creation, you can specify an inactivity period in minutes after which you want the cluster to be terminated.
- If the time difference between the current time and the last command run on the cluster is more than the inactivity period specified, Azure Databricks automatically terminates that cluster.

Automatic Termination

You can also set auto termination for a cluster. During cluster creation, you can specify an inactivity period in minutes after which you want the cluster to terminate. If the difference between the current time and the last command run on the cluster is more than the inactivity period specified, Azure Databricks automatically terminates that cluster.

A cluster is considered inactive when all commands on the cluster, including Spark jobs, Structured Streaming, and JDBC calls, have finished executing.

Warning

Clusters do not report activity resulting from the use of DStreams. This means that an autoterminating cluster may be terminated while it is running DStreams. Turn off autotermination for clusters running DStreams or consider using Structured Streaming.

Configuration

You configure automatic termination in the Auto Termination field on the cluster create dialog.

The default value of the auto terminate setting depends on whether you choose to create a standard or high concurrency cluster:

- Standard clusters are configured to automatically terminate after 120 minutes.
- High concurrency clusters are configured to *not terminate* automatically.

You can opt out of auto termination by clearing the Auto Termination checkbox or by specifying an inactivity period of 0.

A cluster is considered inactive when all commands on the cluster, including Spark jobs, Structured Streaming, and JDBC calls, have finished executing.

Note

Auto termination is best supported in the latest Spark versions. Older Spark versions have known limitations which may result in inaccurate reporting of cluster activity. For example, clusters running JDBC, R, or streaming commands may report a stale activity time which will lead to premature cluster termination. You are strongly recommended to upgrade to the most recent Spark version to benefit from bug fixes and improvements to auto termination.

AutoScaling and AutoTermination Benefits

- Need not worry about # of nodes
  - You don't need to guess or determine by trial and error, the correct number of nodes for the cluster.
- Dynamic Scaling
  - As the workload changes you do not have to manually tweak the number of nodes
- It's pay-per-use!
  - You do not have to worry about wasting resources when the cluster is idle.
- Easy management
  - You do not have to wait and watch for jobs to complete just so you can shutdown the clusters.

Demo:
Azure Databricks Workspace Creation and Cluster Configuration

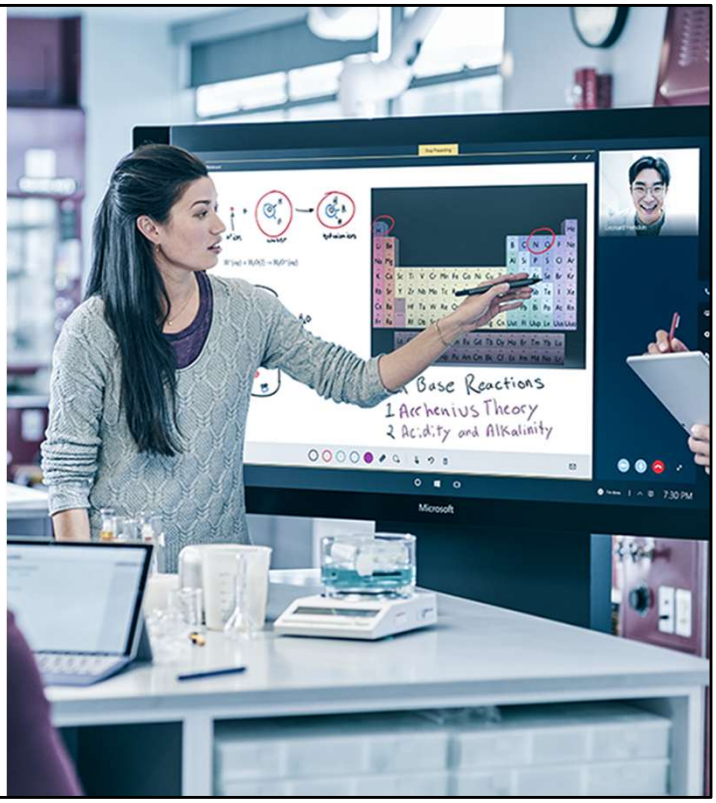
Creating an Azure Databricks Workspace and Configure a Databricks Cluster

Location of Demo Instructions: C:\Demos\M01_L05_Demo01\M01_L05_Demo01.docx

Instructions:
Please follow the demo document to showcase databricks workspace and cluster configuration.

Lab: Workspace and Cluster Configuration

Learn about Workspace Creation and Cluster Configuration



Location of Lab Instructions: C:\LabManuals\M01_L05_Lab01.docx

Instructions:
Please follow the lab document to learn databricks workspace creation and cluster configuration.

How Sharing Happens?

Task sharing is the ability to have different tasks running at the same time on the cluster (Shared), enforcing fair sharing. It is accomplished via:

Preemption

- Proactively preempts Spark tasks from over-committed users to ensure all users get their fair share of cluster time
- Jobs complete in a timely manner even when contending with dozens of other users.

Fault isolation

- Creates an environment for each notebook, effectively isolating them from one another.

Task Preemption for High Concurrency

This functionality is supported on Databricks Runtime 2.2.0-db1 and above.

For Spark 2.2 and above, the Spark scheduler in Azure Databricks automatically preempts tasks to enforce fair sharing between users. This guarantees interactive response times to users on clusters with many concurrently running jobs.

When tasks are preempted by the scheduler, their kill reason will be set to preempted by scheduler. This reason is visible in the Spark UI and can be used to debug preemption behavior.

Preemption options

By default, preemption is conservative: jobs can be starved of resources for up to 30 seconds before the scheduler intervenes. Preemption can be tuned by setting the following Spark configurations *at cluster launch time*:

Whether preemption should be enabled. This can only be set in Spark 2.2 and above.

spark.databricks.preemption.enabled true

The fair share fraction to guarantee per user. Setting this to 1.0 means the scheduler will aggressively attempt to guarantee perfect fair sharing. Setting this to 0.0 effectively disables preemption. The default setting is 0.5, which means at worst a user will get half of their fair share.

spark.databricks.preemption.threshold 0.5

How long a user must remain starved before preemption kicks in. Setting this to lower values will provide more interactive response times, at the cost of cluster efficiency. Recommended values are from 1-100 seconds.

spark.databricks.preemption.timeout 30s

How often the scheduler will check for task preemption. This should be set to less than the preemption

timeout.
spark.databricks.preemption.interval 5s

https://docs.databricks.com/clusters/preemption.html

Demo:
Azure Databricks Cluster Management

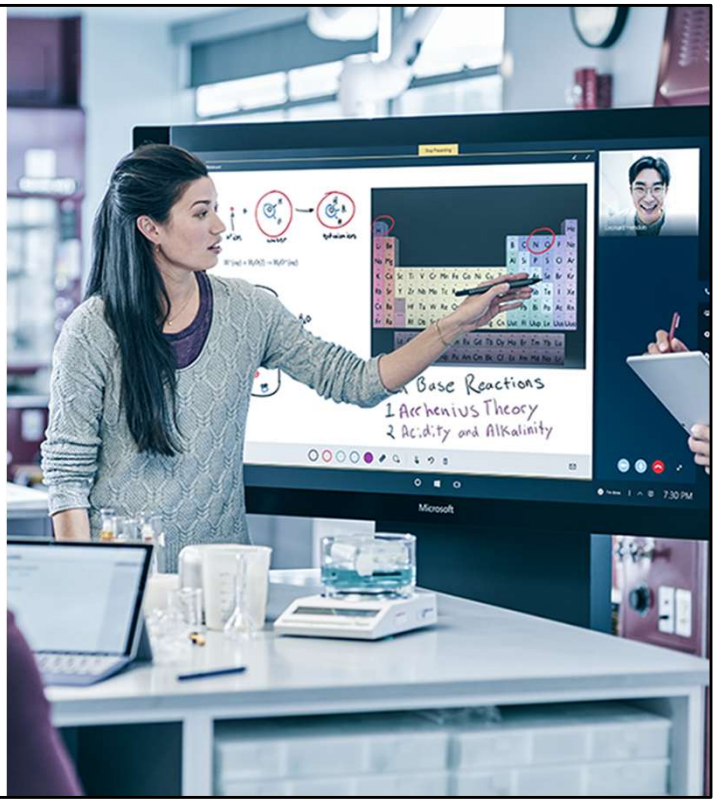
How to manage Azure Databricks Cluster

Location of Demo Instructions: C:\Demos\M01_L05_Demo02\M01_L05_Demo02.docx

Instructions:
Please follow the demo document to showcase cluster management.

Lab: Cluster Management

Learn about managing Azure Databricks Clusters



Location of Lab Instructions: C:\LabManuals\M01_L05_Lab02.docx

Instructions:
Please follow the lab document to learn about managing databricks cluster.

Cluster Access Control

- Types of Permissions:
  - Cluster creation permission controls your ability to create clusters.
  - Individual cluster permissions control your ability to use and modify a specific cluster.

- When cluster access control is enabled:
  - An administrator can configure whether a user can create clusters.
  - Any user with **Can Manage** permission can configure whether a user can attach to, restart, resize, and manage existing clusters.

You have **Can Manage** permission for any cluster that you create.

Cluster access control is only available in in Premium pricing tier

https://docs.azuredatabricks.net/administration-guide/admin-settings/cluster-acl.html#id3

Demo:
Azure Databricks Cluster Access Control

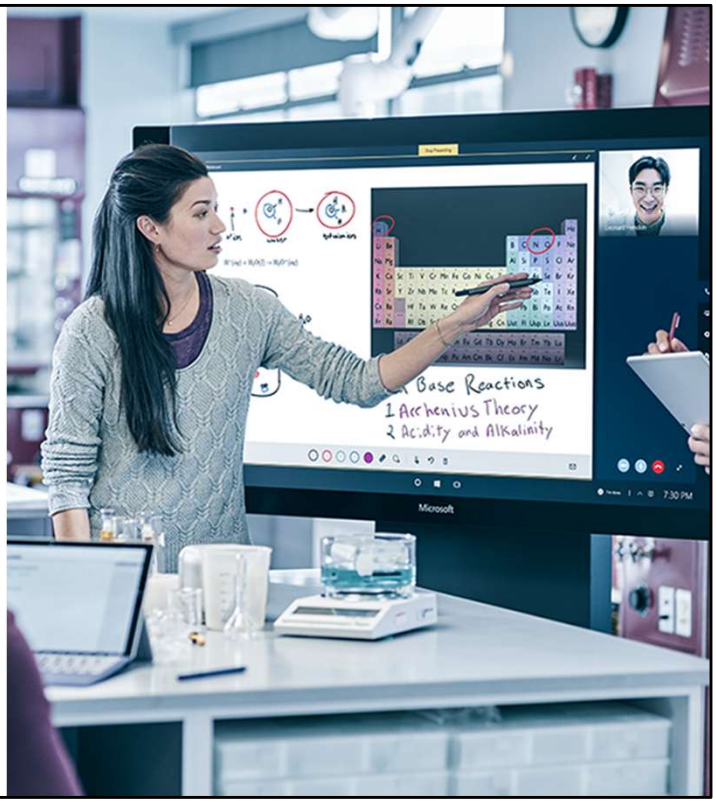
How to manage access to Azure Databricks

Location of Demo Instructions: C:\Demos\M01_L05_Demo03\M01_L05_Demo03.docx

Instructions:
Please follow the demo document to showcase access management in Azure Databricks.

Lab: Cluster Access Control

Learn about Azure Databricks Cluster Access Control



Location of Lab Instructions: C:\LabManuals\M01_L05_Lab03.docx

Instructions:
Please follow the lab document to learn access management in Azure databricks cluster.

Knowledge Check

What are the advantages of Auto Scaling?

What is the difference between Interactive Cluster and Job Cluster?

What is Auto Scaling for Jobs?

1. What are the advantages of Auto Scaling?
   - Workloads can run faster compared to a constant-sized under-provisioned cluster.
   - Autoscaling clusters can reduce overall costs compared to a statically-sized cluster.

2. What is the difference between Interactive Cluster and Job Cluster?

Interactive clusters are used to analyze data collaboratively with interactive notebooks. Job clusters are used to run fast and robust automated workflows using the UI or API.

So, while in development phase, you will mostly use interactive cluster. On the other hand once you move to production where things are more in a automated way, interactive clusters comes in picture.

3. Autoscaling for jobs

Autoscaling for jobs is different from standard autoscaling, and is recommended only with runtime versions 3.4 and above. This feature allows a jobs cluster to scale up and down more aggressively in response to load and is designed to improve resource utilization. In particular, a cluster can scale down idle VMs even when there are tasks running on other VMs. This autoscaling algorithm is different than the one used for standard interactive clusters. To enable this feature for a job running Databricks Runtime 3.4 or higher, select the **Enable Autoscaling** option on the **Configure Cluster** page. For a demonstration of the benefits of job autoscaling, see the blog post on Optimized Autoscaling.

Lesson 6: Azure Databricks Workspace

After completing this lesson, you will be able to:

- Understand Workspace and its importance
- Manage Workspace Access Control
- Manage Azure Databricks Libraries

Workspace

- Workspaces enable users to organize—and share—their Notebooks, Libraries and Dashboards.
- Everything in a workspace is organized into hierarchical folders. Folders can hold Libraries, Notebooks, Dashboard or more (sub) folders.
  - Icons indicate the type of the object contained in a folder
- Every user has one directory that is private and unshared.
  - By default, the workspace and all its contents are available to users.

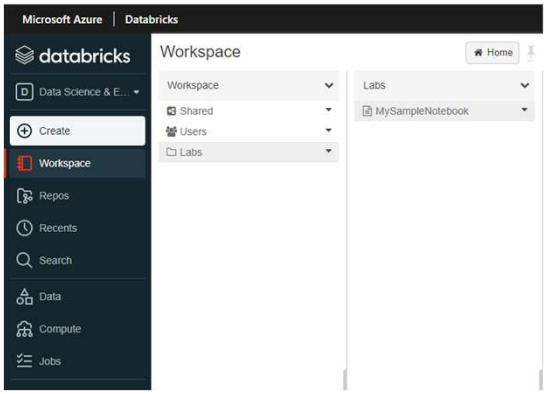
The Workspace is the special root folder for all of your organization's Azure Databricks assets. The Workspace stores all your notebooks, libraries, and dashboards. By default, the Workspace and all its contents are available to users, but each user also has a private home folder that is not shared. You can control who can view, edit, and run objects in the Workspace by enabling Workspace access control.

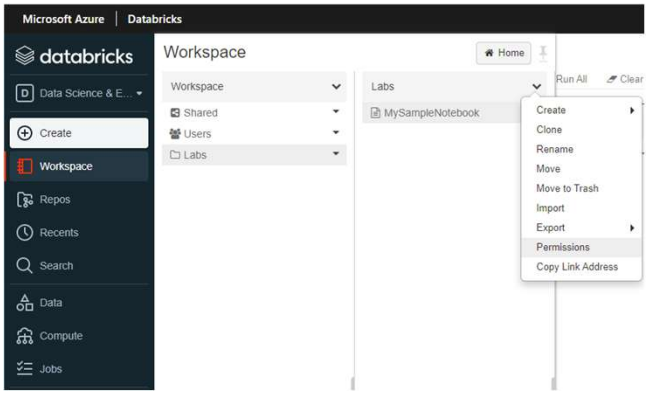
You can create and manage the Workspace using the UI, the CLI, and by invoking the Workspace API. This topic focuses on performing Workspace tasks using the UI. For the other methods, see Databricks CLI and Workspace API.

More Information: https://docs.azuredatabricks.net/user-guide/workspace.html#id1

Workspace

- Fine grained access control can be defined on workspaces to enable *secure collaboration with colleagues*.

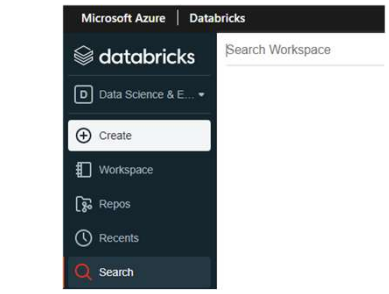


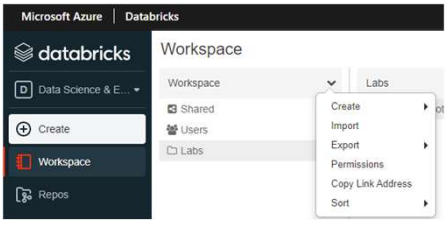


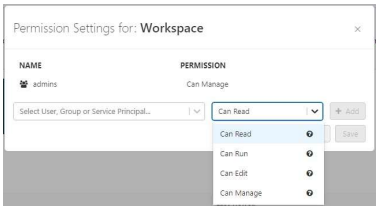
For Files, Folders, or Workspaces click the drop-down arrow next to the name. From there navigate to permissions to set fine grained access control.

Workspace Operations

- You can search the entire workspace.
- In the Azure Databricks Portal, via the Workspaces drop down menu, you can:
  - Create Folders, Notebooks and Libraries.
  - Import Notebooks into the Workspace.
  - Export the Workspace to a database archive.
  - Set Permissions. You can grant 4 levels of permissions
    - Can Manage
    - Can Read
    - Can Edit
    - Can Run







To search the workspace, click the Search Icon in the sidebar and type a search string in the Search Workspace field. As you type, objects (folders, notebooks, and libraries) whose name contains the search string are listed.

https://docs.azuredatabricks.net/user-guide/workspace.html#id1

Folder Operations and Access Control

- In the Azure Databricks Portal, via the Folder drop down menu, you can:
  - Create Folders, Notebooks and Libraries within the folder.
  - Clone the folder to create a deep copy of the folder.
  - Rename or delete the folder.
  - Move the folder to another location.
  - Export a folder to save it and its contents as a Databricks archive.
  - Import a saved Databricks archive into the selected folder.
  - Set Permissions for the folder. As with Workspaces you can set 5 levels of permissions: *No Permissions, Can Manage, Can Read, Can Edit, Can Run*.

To search the workspace, click the Search Icon in the sidebar and type a search string in the Search Workspace field. As you type, objects (folders, notebooks, and libraries) whose name contains the search string are listed.

https://docs.azuredatabricks.net/user-guide/workspace.html#id1

Libraries

- Enables external code to be imported and stored into a Workspace.
- Libraries are containers to hold all your *Python*, *R*, *Java/Scala* libraries.
- Libraries resides within workspaces or folders.
- Libraries are created by importing the source code.
- Imported libraries are immutable—can be deleted or overwritten only.
- You can customize installation of libraries via Init Scripts by writing custom UNIX scripts (not available on Shared Cluster!).
- Libraries can also be managed via the Library API or Library CLI.

To make third-party or locally-built code available to execution environments running on your clusters, you create a library. Libraries can be written in Python, Java, Scala, and R.

To allow a library to be shared by all users in a Workspace, create the library in the **Shared** folder. To make it available to a single user, create the library in the user folder.

You can create and manage libraries using the UI, the CLI, and by invoking the Libraries API. This topic focuses on performing library tasks using the UI. For the other methods, see Databricks CLI and Libraries API.

Some libraries require lower level configuration and cannot be uploaded using the methods described in this topic. To install these libraries you can write a custom UNIX script that runs at cluster creation time, following the instructions in Cluster Node Initialization Scripts or SSH Access to Clusters.

Libraries can be created, attached to a cluster, detached from a cluster, and deleted.

When you create a library, you either upload or install the library package. Packages that you upload or install using Maven are stored in the FileStore in FileStore/jars. Databricks installs Python packages in the Spark container using pip install.

To use a library, you first attach it to a cluster. To use the library in a notebook that was attached to the cluster before the library was attached, you must reattach the cluster to the notebook.

There are two steps to permanently delete a library:

1. Move the library to the Trash folder.
2. Either permanently delete the library in the Trash folder or empty the Trash folder.

When you move a library to the Trash folder, the library is *not* marked for deletion, which means that it remains available on any clusters that it is attached to. When you permanently delete a library, the cluster to which the library is attached identifies the library as marked for deletion. When you detach a library from a

cluster or permanently delete a library previously attached to the cluster, you must restart the cluster.

Secrets

- A secret is a key-value pair that stores secret material, with a key name unique within a secret scope
- Secrets can be read from a notebook or job using the Secrets Utility (dbutils.secrets)
- A scope is limited to **1000 secrets**. The **maximum** allowed **secret value size** is **128 KB**
- A scope can be Databricks-backed or Azure Key Vault based
- Secret can be created and managed using the Secrets CLI.
- Secrets can also be managed via the Secrets API

Demo:
Azure Databricks Workspace Operations

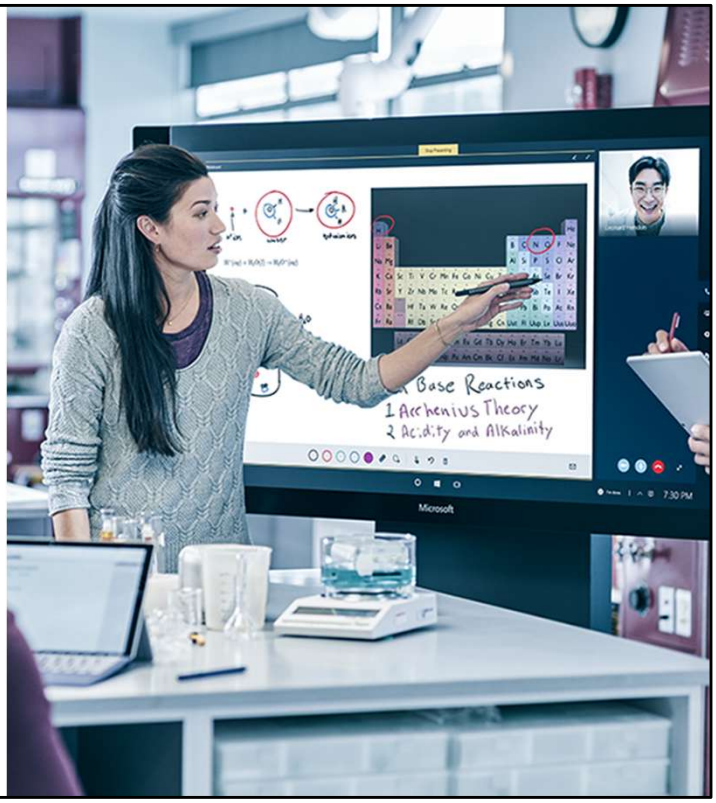
How to manage Azure Databricks Workspaces

Location of Demo Instructions: C:\Demos\M01_L06_Demo01\M01_L06_Demo01.docx

Instructions:
Please follow the demo document to showcase management of Azure Databricks Workspaces.

Lab: Workspace Management

Managing Azure Databricks Workspaces



Location of Lab Instructions: C:\LabManuals\M01_L06_Lab01.docx

Instructions:
Please follow the lab document to learn management of Workspaces in Azure databricks.

Knowledge Check

What does Workspace help with?

How many levels of permissions are available for Workspaces?

How many levels of permissions are available for Folders?

How does a library permanently deleted?

1. What does Workspace help with?
- Organization of Azure Databricks assets.
- Provides privacy to users on their own folders.

2. How many levels of permissions are available for Workspaces?
- Can Manage
- Can Read
- Can Edit
- Can Run

3. How many levels of permissions are available for Folders?

5 levels of permissions:
- No Permissions
- Can Manage
- Can Read
- Can Edit
- Can Run

4. How does a library is permanently deleted?

There are two steps to permanently delete a library:

1. Move the library to the Trash folder.
2. Either permanently delete the library in the Trash folder or empty the Trash folder.

When you move a library to the Trash folder, the library is *not* marked for deletion, which means that it remains available on any clusters that it is attached to. When you permanently delete a library, the cluster to

which the library is attached identifies the library as marked for deletion. When you detach a library from a cluster or permanently delete a library previously attached to the cluster, you must restart the cluster.

Module Summary

- Azure Databricks and it's capabilities
- Azure Databricks Architecture
- Azure Databricks Clusters concepts
- Working with Workspace and Libraries