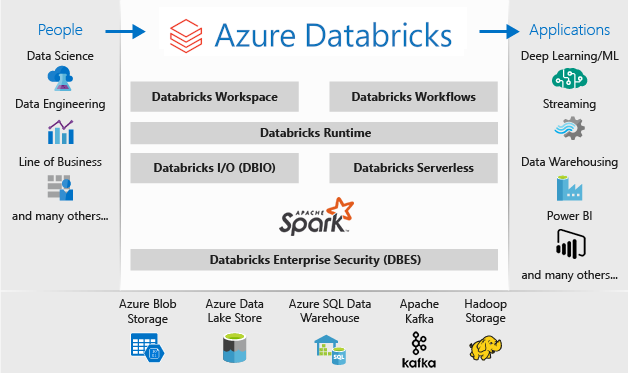
Azure Databricks, an Apache Spark-based analytics platform with one-click setup, streamlined workflows, and an interactive workspace for collaboration between data scientists, engineers, and business analysts.

Designed with the founders of Apache Spark, 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.



**Workload**

Azure Databricks identifies two types of workloads subject to different [pricing](https://databricks.com/product/pricing) schemes: data engineering (automated) and data analytics (interactive).

* **Data engineering** An (automated) workload runs on *an automated cluster* which the Azure Databricks job scheduler creates for each workload.
* **Data analytics** An (interactive) workload runs on an *interactive cluster*. Interactive workloads typically run commands within an Azure Databricks [notebook](https://docs.microsoft.com/en-us/azure/databricks/notebooks/). However, running a *job* on an *existing automated* cluster is also treated as an interactive workload.

**Execution context**

Databricks runtimes are the set of core components that run on Azure Databricks [clusters](https://docs.microsoft.com/en-us/azure/databricks/clusters/). Azure Databricks offers several types of runtimes.

* [Databricks Runtime](https://docs.microsoft.com/en-us/azure/databricks/runtime/dbr)

Databricks Runtime includes Apache Spark but also adds a number of components and updates that substantially improve the usability, performance, and security of big data analytics.

* [Databricks Runtime for Machine Learning](https://docs.microsoft.com/en-us/azure/databricks/runtime/mlruntime)

Databricks Runtime ML is a variant of Databricks Runtime that adds multiple popular machine learning libraries, including TensorFlow, Keras, PyTorch, and XGBoost.

* [Databricks Runtime for Genomics](https://docs.microsoft.com/en-us/azure/databricks/runtime/genomicsruntime)

Databricks Runtime for Genomics is a variant of Databricks Runtime optimized for working with genomic and biomedical data.

* [Databricks Light](https://docs.microsoft.com/en-us/azure/databricks/runtime/light)

Databricks Light provides a runtime option for jobs that don’t need the advanced performance, reliability, or autoscaling benefits provided by Databricks Runtime.

**Workspace assets**

* 07/15/2020
* 2 minutes to read
  + [[https://github.com/mamccrea.png?size=32](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/workspace/workspace-assets.md)](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/workspace/workspace-assets.md" \o "1 Contributor)

This article provides a high-level introduction to Azure Databricks workspace assets.

**Clusters**

Azure Databricks clusters provide a unified platform for various use cases such as running production ETL pipelines, streaming analytics, ad-hoc analytics, and machine learning.

For detailed information on managing and using clusters, see [Clusters](https://docs.microsoft.com/en-us/azure/databricks/clusters/).

**Notebooks**

A notebook is a web-based interface to documents containing a series of runnable cells (commands) that operate on files and [tables](https://docs.microsoft.com/en-us/azure/databricks/data/tables#tables), [visualizations](https://docs.microsoft.com/en-us/azure/databricks/notebooks/visualizations/), and narrative text. Commands can be run in sequence, referring to the output of one or more previously run commands.

Notebooks are one mechanism for running code in Azure Databricks. The other mechanism is [jobs](https://docs.microsoft.com/en-us/azure/databricks/jobs).

For detailed information on managing and using notebooks, see [Notebooks](https://docs.microsoft.com/en-us/azure/databricks/notebooks/).

**Jobs**

Jobs are one mechanism for running code in Azure Databricks. The other mechanism is [notebooks](https://docs.microsoft.com/en-us/azure/databricks/workspace/workspace-assets#ws-notebooks).

For detailed information on managing and using jobs, see [Jobs](https://docs.microsoft.com/en-us/azure/databricks/jobs).

**Libraries**

A library makes third-party or locally-built code available to notebooks and jobs running on your clusters.

For detailed information on managing and using libraries, see [Libraries](https://docs.microsoft.com/en-us/azure/databricks/libraries/).

**Data**

You can import data into a distributed file system mounted into an Azure Databricks workspace and work with it in Azure Databricks notebooks and clusters. You can also use a wide variety of Apache Spark data sources to access data.

For detailed information on managing and using data, see [Data](https://docs.microsoft.com/en-us/azure/databricks/data/#data).

**Models**

The Model Registry is a centralized model store that enables you to manage the full lifecycle of MLflow Models. It provides chronological model lineage, model versioning, stage transitions, and model and model version annotations and descriptions.

For more detailed information, see [Manage the Lifecycle of MLflow Models in MLflow Model Registry](https://docs.microsoft.com/en-us/azure/databricks/applications/mlflow/model-registry).

**Experiments**

An MLflow experiment is the primary unit of organization and access control for MLflow machine learning model training runs; all MLflow runs belong to an experiment. Each experiment lets you visualize, search, and compare runs, as well as download run artifacts or metadata for analysis in other tools.

For detailed information on managing and using experiments, see [Experiments](https://docs.microsoft.com/en-us/azure/databricks/applications/mlflow/tracking#mlflow-experiments).

**Models**

A model refers to an MLflow registered model, which lets you manage MLflow Models in production through stage transitions and versioning. A registered model has a unique name, versions, model lineage, and other metadata.

For detailed information on managing and using models, see [Manage the Lifecycle of MLflow Models in MLflow Model Registry](https://docs.microsoft.com/en-us/azure/databricks/applications/mlflow/model-registry).

### Special folders

An Azure Databricks workspace has three special folders: Workspace, Shared, and Users. You cannot rename or move a special folder.

[Delta Lake](https://delta.io/) is an [open source storage layer](https://github.com/delta-io/delta) that brings reliability to [data lakes](https://databricks.com/discover/data-lakes/introduction). Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing. Delta Lake runs on top of your existing data lake and is fully compatible with Apache Spark APIs.

Specifically, Delta Lake offers:

* ACID transactions on Spark: Serializable isolation levels ensure that readers never see inconsistent data.
* Scalable metadata handling: Leverages Spark’s distributed processing power to handle all the metadata for petabyte-scale tables with billions of files at ease.
* Streaming and batch unification: A table in Delta Lake is a batch table as well as a streaming source and sink. Streaming data ingest, batch historic backfill, interactive queries all just work out of the box.
* Schema enforcement: Automatically handles schema variations to prevent insertion of bad records during ingestion.
* Time travel: Data versioning enables rollbacks, full historical audit trails, and reproducible machine learning experiments.
* Upserts and deletes: Supports merge, update and delete operations to enable complex use cases like change-data-capture, slowly-changing-dimension (SCD) operations, streaming upserts, and so on.

## What is Delta Lake?

[Delta Lake](https://delta.io/) is an [open source storage layer](https://github.com/delta-io/delta) that brings reliability to [data lakes](https://databricks.com/discover/data-lakes/introduction). Delta Lake provides ACID transactions, scalable metadata handling, and unifies streaming and batch data processing. Delta Lake runs on top of your existing data lake and is fully compatible with Apache Spark APIs.

Delta Lake on Azure Databricks allows you to configure Delta Lake based on your workload patterns and provides optimized layouts and indexes for fast interactive queries.

## How is Delta Lake related to Apache Spark?

Delta Lake sits on top of Apache Spark. The format and the compute layer helps to simplify building big data pipelines and increase the overall efficiency of your pipelines.

## What format does Delta Lake use to store data?

Delta Lake uses versioned Parquet files to store your data in your cloud storage. Apart from the versions, Delta Lake also stores a transaction log to keep track of all the commits made to the table or blob store directory to provide ACID transactions.

## How can I read and write data with Delta Lake?

You can use your favorite Apache Spark APIs to read and write data with Delta Lake. See [Read a table](https://docs.microsoft.com/en-us/azure/databricks/delta/delta-batch#deltadataframereads) and [Write to a table](https://docs.microsoft.com/en-us/azure/databricks/delta/delta-batch#deltadataframewrites).

## Where does Delta Lake store the data?

When writing data, you can specify the location in your cloud storage. Delta Lake stores the data in that location in Parquet format.

## Can I stream data directly into and from Delta tables?

Yes, you can use Structured Streaming to directly write data into Delta tables and read from Delta tables. See [Stream data into Delta tables](https://docs.microsoft.com/en-us/azure/databricks/delta/delta-streaming#stream-sink) and [Stream data from Delta tables](https://docs.microsoft.com/en-us/azure/databricks/delta/delta-streaming#stream-source).

## Does Delta Lake support writes or reads using the Spark Streaming DStream API?

Delta does not support the DStream API. We recommend [Table streaming reads and writes](https://docs.microsoft.com/en-us/azure/databricks/delta/delta-streaming).

## When I use Delta Lake, will I be able to port my code to other Spark platforms easily?

Yes. When you use Delta Lake, you are using open Apache Spark APIs so you can easily port your code to other Spark platforms. To port your code, replace delta format with parquet format.

## How do Delta tables compare to Hive SerDe tables?

Delta tables are managed to a greater degree. In particular, there are several Hive SerDe parameters that Delta Lake manages on your behalf that you should never specify manually:

* ROWFORMAT
* SERDE
* OUTPUTFORMAT AND INPUTFORMAT
* COMPRESSION
* STORED AS

## What DDL and DML features does Delta Lake not support?

* Unsupported DDL features:
  + ANALYZE TABLE PARTITION
  + ALTER TABLE [ADD|DROP] PARTITION
  + ALTER TABLE RECOVER PARTITIONS
  + ALTER TABLE SET SERDEPROPERTIES
  + CREATE TABLE LIKE
  + INSERT OVERWRITE DIRECTORY
  + LOAD DATA
* Unsupported DML features:
  + INSERT INTO [OVERWRITE] table with static partitions
  + INSERT OVERWRITE TABLE for table with dynamic partitions
  + Bucketing
  + Specifying a schema when reading from a table
  + Specifying target partitions using PARTITION (part\_spec) in TRUNCATE TABLE

## Does Delta Lake support multi-table transactions?

Delta Lake does not support multi-table transactions and foreign keys. Delta Lake supports transactions at the table level.

## How can I change the type of a column?

Changing a column’s type or dropping a column requires rewriting the table. For an example, see [Change column type](https://docs.microsoft.com/en-us/azure/databricks/delta/delta-batch#change-column-type).

## What does it mean that Delta Lake supports multi-cluster writes?

It means that Delta Lake does locking to make sure that queries writing to a table from multiple clusters at the same time won’t corrupt the table. However, it does not mean that if there is a write conflict (for example, update and delete the same thing) that they will both succeed. Instead, one of writes will fail atomically and the error will tell you to retry the operation.

## Can I modify a Delta table from different workspaces?

Yes, you can concurrently modify the same Delta table from different workspaces. Moreover, if one process is writing from a workspace, readers in other workspaces will see a consistent view.

## Can I access Delta tables outside of Databricks Runtime?

There are two cases to consider: external writes and external reads.

* External writes: Delta Lake maintains additional metadata in the form of a transaction log to enable ACID transactions and snapshot isolation for readers. In order to ensure the transaction log is updated correctly and the proper validations are performed, writes must go through Databricks Runtime.
* External reads: Delta tables store data encoded in an open format (Parquet), allowing other tools that understand this format to read the data. However, since other tools do not support the Delta Lake transaction log, it is likely that they will incorrectly read stale deleted data, uncommitted data, or the partial results of failed transactions.

In cases where the data is static (that is, there are no active jobs writing to the table), you can use VACUUM with a retention of ZERO HOURS to clean up any stale Parquet files that are not currently part of the table. This operation puts the Parquet files present in DBFS into a consistent state such that they can now be read by external tools.

However, Delta Lake relies on stale snapshots for the following functionality, which will break when using VACUUM with zero retention allowance:

* + Snapshot isolation for readers - Long running jobs will continue to read a consistent snapshot from the moment the jobs started, even if the table is modified concurrently. Running VACUUM with a retention less than length of these jobs can cause them to fail with a FileNotFoundException.
  + Streaming from Delta tables - Streams read from the original files written into a table in order to ensure exactly once processing. When combined with OPTIMIZE, VACUUM with zero retention can remove these files before the stream has time to processes them, causing it to fail.

For these reasons we recommend the above technique only on static data sets that must be read by external tools.

**Apache Spark-based analytics platform**

Azure Databricks comprises the complete open-source Apache Spark cluster technologies and capabilities. Spark in Azure Databricks includes the following components:

* **Spark SQL and DataFrames**: Spark SQL is the Spark module for working with structured data. A DataFrame is a distributed collection of data organized into named columns. It is conceptually equivalent to a table in a relational database or a data frame in R/Python.
* **Streaming**: Real-time data processing and analysis for analytical and interactive applications. Integrates with HDFS, Flume, and Kafka.
* **MLlib**: Machine Learning library consisting of common learning algorithms and utilities, including classification, regression, clustering, collaborative filtering, dimensionality reduction, as well as underlying optimization primitives.
* **GraphX**: Graphs and graph computation for a broad scope of use cases from cognitive analytics to data exploration.
* **Spark Core API**: Includes support for R, SQL, Python, Scala, and Java.

**Apache Spark in Azure Databricks**

Azure Databricks builds on the capabilities of Spark by providing a zero-management cloud platform that includes:

* Fully managed Spark clusters
* An interactive workspace for exploration and visualization
* A platform for powering your favorite Spark-based applications

[**Cluster**](https://docs.microsoft.com/en-us/azure/databricks/clusters/)

A set of computation resources and configurations on which you run notebooks and jobs. There are two types of clusters: interactive and automated.

* You create an interactive cluster using the UI, CLI, or REST API. You can manually terminate and restart an interactive cluster. Multiple users can share such clusters to do collaborative interactive analysis.
* The Azure Databricks job scheduler creates an automated cluster when you run a [job](https://docs.microsoft.com/en-us/azure/databricks/jobs) on a new automated cluster and terminates the cluster when the job is complete. You cannot restart an automated cluster.

[**Pool**](https://docs.microsoft.com/en-us/azure/databricks/clusters/instance-pools/)

* A set of idle, ready-to-use instances that reduce cluster start and auto-scaling times. When attached to a pool, a cluster allocates its driver and worker nodes from the pool. If the pool does not have sufficient idle resources to accommodate the cluster’s request, the pool expands by allocating new instances from the instance provider. When an attached cluster is terminated, the instances it used are returned to the pool and can be reused by a different cluster.

# Databricks runtimes

* 05/05/2020
* 2 minutes to read
  + [[https://github.com/mamccrea.png?size=32](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/runtime/index.md)](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/runtime/index.md" \o "2 Contributors)

* + [[https://github.com/JasonWHowell.png?size=32](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/runtime/index.md)](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/runtime/index.md" \o "2 Contributors)

Databricks runtimes are the set of core components that run on Azure Databricks [clusters](https://docs.microsoft.com/en-us/azure/databricks/clusters/). Azure Databricks offers several types of runtimes.

* [Databricks Runtime](https://docs.microsoft.com/en-us/azure/databricks/runtime/dbr)

Databricks Runtime includes Apache Spark but also adds a number of components and updates that substantially improve the usability, performance, and security of big data analytics.

* [Databricks Runtime for Machine Learning](https://docs.microsoft.com/en-us/azure/databricks/runtime/mlruntime)

Databricks Runtime ML is a variant of Databricks Runtime that adds multiple popular machine learning libraries, including TensorFlow, Keras, PyTorch, and XGBoost.

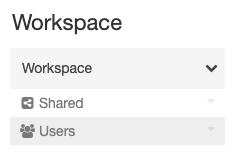
* [Databricks Runtime for Genomics](https://docs.microsoft.com/en-us/azure/databricks/runtime/genomicsruntime)

Databricks Runtime for Genomics is a variant of Databricks Runtime optimized for working with genomic and biomedical data.

* [Databricks Light](https://docs.microsoft.com/en-us/azure/databricks/runtime/light)

Databricks Light provides a runtime option for jobs that don’t need the advanced performance, reliability, or autoscaling benefits provided by Databricks Runtime.

The Workspace root folder is a container for all of your organization’s Azure Databricks static assets.



Within the Workspace root folder:

* Shared Icon **Shared** is for sharing objects across your organization. All users have full permissions for all objects in Shared.
* Users Icon **Users** contains a folder for each user.

## Cluster URL and ID

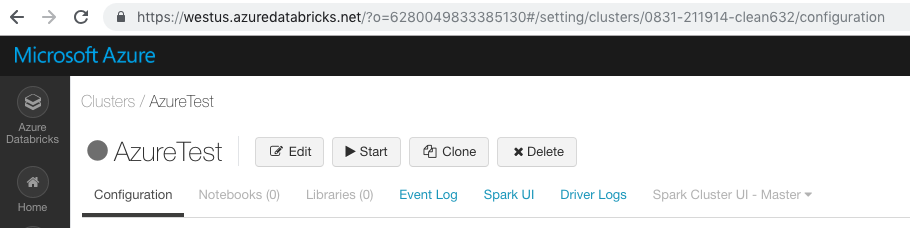
An Azure Databricks [*cluster*](https://docs.microsoft.com/en-us/azure/databricks/clusters/) provides a unified platform for various use cases such as running production ETL pipelines, streaming analytics, ad-hoc analytics, and machine learning. Each cluster has a unique ID called the cluster ID. This applies to both interactive and automated clusters. To get the details of a cluster using the REST API, the cluster ID is essential.

To get the cluster ID, click the **Clusters** tab in sidebar and then select a cluster name. The cluster ID is the number after the /clusters/ component in the URL of this page

Copy

https://<databricks-instance>/#/settings/clusters/<cluster-id>

In the following screenshot, the cluster ID is 0831-211914-clean632.



## Notebook URL and ID

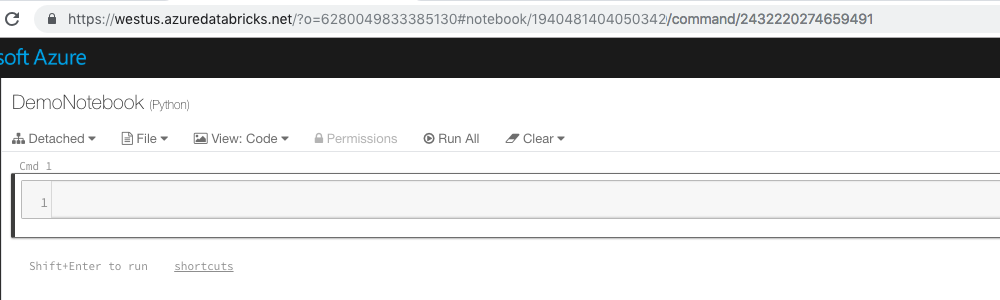
A [*notebook*](https://docs.microsoft.com/en-us/azure/databricks/notebooks/) is a web-based interface to a document that contains runnable code, visualizations, and narrative text. Notebooks are one interface for interacting with Azure Databricks. Each notebook has a unique ID. The notebook URL has the notebook ID, hence the notebook URL is unique to a notebook. It can be shared with anyone on Azure Databricks platform with permission to view and edit the notebook. In addition, each notebook command (cell) has a different URL.

To get to a notebook URL, open a notebook.

In the following notebook, the notebook URL is https://westus.azuredatabricks.net/?o=6280049833385130#notebook/1940481404050342, the notebook ID is 1940481404050342, and the command (cell) URL is

Copy

https://westus.azuredatabricks.net/?o=6280049833385130#notebook/1940481404050342/command/2432220274659491`



## Model ID

A model refers to an MLflow [*registered model*](https://docs.microsoft.com/en-us/azure/databricks/applications/mlflow/model-registry), which lets you manage MLflow Models in production through stage transitions and versioning. The registered model ID is required for changing the permissions on the model programmatically through the Permissions API.

To get the ID of a registered model, you can use the [REST API 2.0](https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/) endpoint mlflow/databricks/registered-models/get. For example, the following code returns the registered model object with its properties, including its ID:

BashCopy

curl -n -X GET -H 'Content-Type: application/json' -d '{"name": "model\_name"}' \

https://<databricks-instance>/api/2.0/mlflow/databricks/registered-models/get

The returned value has the format:

JSONCopy

{

"registered\_model\_databricks": {

"name":"model\_name",

"id":"ceb0477eba94418e973f170e626f4471"

}

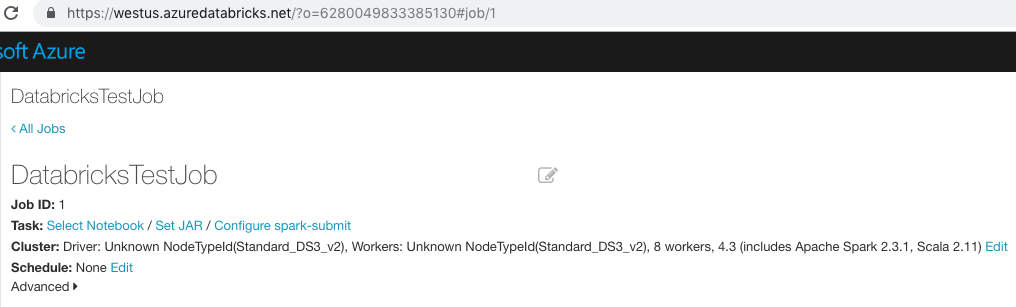
}

## Job URL and ID

A [*job*](https://docs.microsoft.com/en-us/azure/databricks/jobs) is a way of running a notebook or JAR either immediately or on a scheduled basis.

To get to a job URL, click the **Jobs** tab in sidebar and click a job name. This job URL is critical piece of information needed to troubleshoot job runs that have failed and investigate the root cause.

In the following screenshot, the job URL is https://westus.azuredatabricks.net/?o=6280049833385130#job/1, and the job ID is 1.



# Clusters

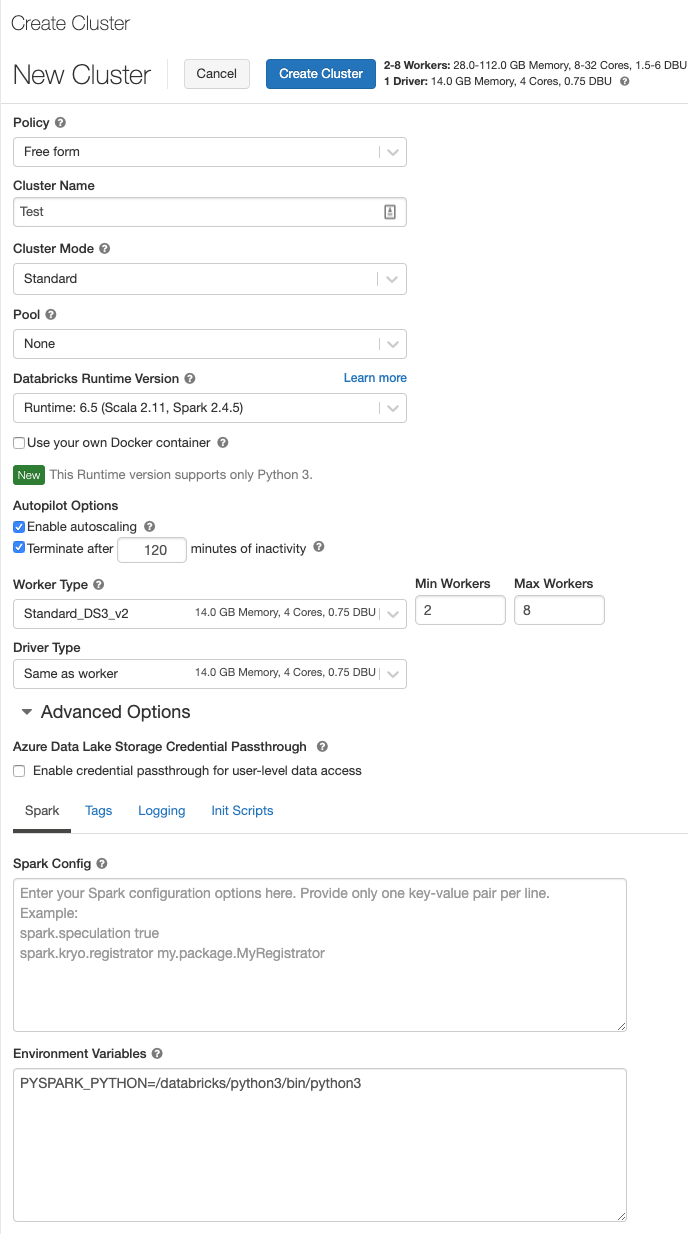
An Azure Databricks cluster is a set of computation resources and configurations on which you run data engineering, data science, and data analytics workloads, such as production ETL pipelines, streaming analytics, ad-hoc analytics, and machine learning.

You run these workloads as a set of commands in a [notebook](https://docs.microsoft.com/en-us/azure/databricks/notebooks/) or as an automated [job](https://docs.microsoft.com/en-us/azure/databricks/jobs). Azure Databricks makes a distinction between interactive clusters and automated clusters. You use interactive clusters to analyze data collaboratively using interactive notebooks. You use automated clusters to run fast and robust automated jobs.

* You can create an interactive cluster using the UI, CLI, or REST API. You can manually terminate and restart an interactive cluster. Multiple users can share such clusters to do collaborative interactive analysis.
* The Azure Databricks job scheduler creates an automated cluster when you run a [job](https://docs.microsoft.com/en-us/azure/databricks/jobs) on a new automated cluster and terminates the cluster when the job is complete. You cannot restart a job cluster.

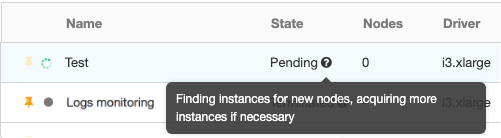
1. Name and configure the cluster.

There are many cluster configuration options, which are described in detail in [cluster configuration](https://docs.microsoft.com/en-us/azure/databricks/clusters/configure#cluster-configurations).



1. Click the **Create** button.

Initially, the cluster list page displays the status of the new cluster as Pending.



Once the cluster state changes

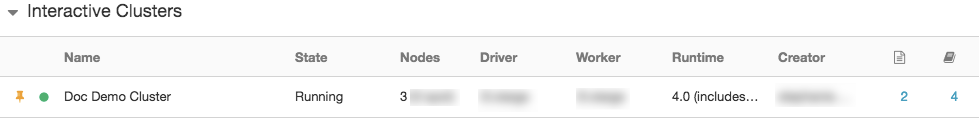
## Display clusters

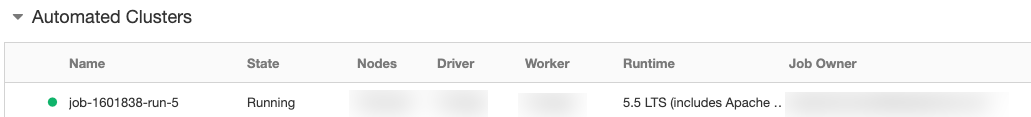
To display the clusters in your workspace, click the clusters icon Clusters Icon in the sidebar.

The Clusters page displays two lists: **Interactive Clusters** and **Automated Clusters**. Each list includes:

* Cluster name
* [State](https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/clusters#clusterclusterstate)
* Number of nodes
* Type of driver and worker nodes
* Databricks Runtime version
* Cluster creator or job owner

In addition to the common cluster information, the **Interactive Clusters** list shows the numbers of notebooks Attached Notebooks attached to the cluster. Above the list is the number of [pinned](https://docs.microsoft.com/en-us/azure/databricks/clusters/clusters-manage#cluster-pin) clusters.





An icon to the left of an interactive cluster name indicates whether the cluster is pinned, whether the cluster offers a [high concurrency](https://docs.microsoft.com/en-us/azure/databricks/clusters/configure#high-concurrency) cluster, and whether [table access control](https://docs.microsoft.com/en-us/azure/databricks/security/access-control/table-acls/object-privileges) is enabled:

* Pinned Pinned
* Starting Starting , Terminating Terminating
* Standard cluster
  + Running Running
  + Terminated Terminated
* High concurrency cluster
  + Running Serverless
  + Terminated Serverless Terminated
* Access Denied
  + Running Locked
  + Terminated Locked Terminated
* Table ACLs enabled
  + Running Table ACLs
  + Terminated Table ACLs Terminated

## View a cluster configuration as a JSON file

Sometimes it can be helpful to view your cluster configuration as JSON. This is especially useful when you want to create similar clusters using the [Clusters API](https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/clusters). When you view an existing cluster, simply go to the **Configuration** tab, click **JSON** in the top right of the tab, copy the JSON, and paste it into your API call. JSON view is ready-only.

## Delete a cluster

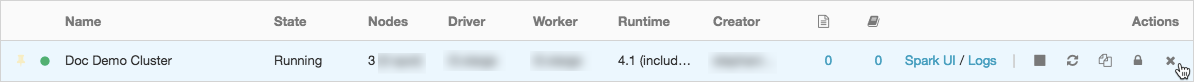
Deleting a cluster terminates the cluster and removes its configuration.

**Warning**

You cannot undo this action.

You cannot delete a [pinned](https://docs.microsoft.com/en-us/azure/databricks/clusters/clusters-manage#cluster-pin) cluster. In order to delete a pinned cluster, it must first be unpinned by an administrator.

To delete a cluster, click the Delete Icon icon in the cluster actions on the [Clusters](https://docs.microsoft.com/en-us/azure/databricks/clusters/clusters-manage#display-clusters) page.



You can also invoke the [Permanent delete](https://docs.microsoft.com/en-us/azure/databricks/dev-tools/api/latest/clusters#clusterclusterservicepermanentdeletecluster) API endpoint to programmatically delete a cluster.

## Cluster mode

Azure Databricks supports two cluster modes: standard and high concurrency. The default cluster mode is standard.

**Note**

The cluster configuration includes an [**auto terminate**](https://docs.microsoft.com/en-us/azure/databricks/clusters/clusters-manage#automatic-termination) setting whose default value depends on whether you are creating a standard or high concurrency cluster:

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

### Standard clusters

Standard clusters are recommended for a single user. Standard can run workloads developed in any language: Python, R, Scala, and SQL.

### High concurrency clusters

A high concurrency cluster is a managed cloud resource. The key benefits of high concurrency clusters are that they provide Apache Spark-native fine-grained sharing for maximum resource utilization and minimum query latencies. This sharing is accomplished with:

* **Preemption:** Proactively preempts Spark tasks from over-committed users to ensure all users get their fair share of cluster time and their jobs complete in a timely manner even when contending with dozens of other users. This uses Spark [Task preemption for high concurrency](https://docs.microsoft.com/en-us/azure/databricks/spark/latest/spark-sql/preemption#preemption).
* **Fault isolation:** Creates an environment for each notebook, effectively isolating them from one another.

## Pool

**Important**

This feature is in [**Public Preview**](https://docs.microsoft.com/en-us/azure/databricks/release-notes/release-types).

To reduce cluster start time, you can attach a cluster to a predefined [pool](https://docs.microsoft.com/en-us/azure/databricks/clusters/instance-pools/) of idle instances. When attached to a pool, a cluster allocates its driver and worker nodes from the pool. If the pool does not have sufficient idle resources to accommodate the cluster’s request, the pool expands by allocating new instances from the instance provider. When an attached cluster is terminated, the instances it used are returned to the pool and can be reused by a different cluster.

See [Use a pool](https://docs.microsoft.com/en-us/azure/databricks/clusters/instance-pools/cluster-instance-pool#cluster-instance-pool) to learn more about working with pools in Azure Databricks.

## Cluster node type

A cluster consists of one driver node and worker nodes. You can pick separate cloud provider instance types for the driver and worker nodes, although by default the driver node uses the same instance type as the worker node. Different families of instance types fit different use cases, such as memory-intensive or compute-intensive workloads.

### Driver node

The driver maintains state information of all notebooks attached to the cluster. The driver node is also responsible for maintaining the SparkContext and interpreting all the commands you run from a notebook or a library on the cluster. The driver node also runs the Apache Spark master that coordinates with the Spark executors.

The default value of the driver node type is the same as the worker node type. You can choose a larger driver node type with more memory if you are planning to collect() a lot of data from Spark workers and analyze them in the notebook.

**Tip**

Since the driver node maintains all of the state information of the notebooks attached, make sure to detach unused notebooks from the driver.

### Worker node

Azure Databricks workers run the Spark executors and other services required for the proper functioning of the clusters. When you distribute your workload with Spark, all of the distributed processing happens on workers. Azure Databricks runs one executor per worker node; therefore the terms executor and worker are used interchangeably in the context of the Azure Databricks architecture.

### Autoscaling types

Azure Databricks offers two types of cluster node autoscaling: standard and optimized. For a discussion of the benefits of optimized autoscaling, see the blog post on [Optimized Autoscaling](https://databricks.com/blog/2018/05/02/introducing-databricks-optimized-auto-scaling.html).

Automated (job) clusters always use optimized autoscaling. The type of autoscaling performed on interactive clusters depends on the workspace configuration.

Standard autoscaling is used by interactive clusters in workspaces in the Standard pricing tier. Optimized autoscaling is used by interactive clusters in the [Azure Databricks Premium Plan](https://databricks.com/product/azure-pricing).

### How autoscaling behaves

Autoscaling behaves differently depending on whether it is optimized or standard and whether applied to an interactive or an automated cluster.

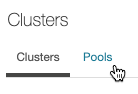
#### Optimized autoscaling

* Scales up from min to max in 2 steps.
* Can scale down even if the cluster is not idle by looking at shuffle file state.
* Scales down based on a percentage of current nodes.
* On automated clusters, scales down if the cluster is underutilized over the last 40 seconds.
* On interactive clusters, scales down if the cluster is underutilized over the last 150 seconds.

#### Standard autoscaling

* Starts with adding 8 nodes. Thereafter, scales up exponentially, but can take many steps to reach the max. You can customize the first step by setting the spark.databricks.autoscaling.standardFirstStepUp Spark configuration property.
* Scales down only when the cluster is completely idle and it has been underutilized for the last 10 minutes.
* Scales down exponentially, starting with 1 node.

o display the pools in your workspace, click the clusters icon Clusters Icon in the sidebar and click the **Pools** tab:



For each pool, the page displays:

* Pool name
* Configured instance type
* Minimum idle instances
* Maximum instance capacity
* Current number of idle instances
* Current number of used instances

## Inherited configuration

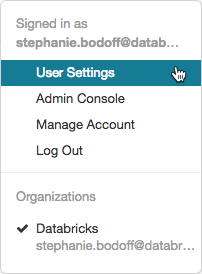
When a cluster is attached to a pool, the following configuration properties are inherited from the pool:

* [Cluster node type](https://docs.microsoft.com/en-us/azure/databricks/clusters/configure#node-types): You cannot select separate driver and worker node types.
* [Custom Cluster Tags](https://docs.microsoft.com/en-us/azure/databricks/clusters/configure#cluster-tags): You can add additional custom tags for the cluster, and both the cluster-level tags and those inherited from the pool will be applied. You cannot add a cluster-specific custom tag with the same key name as a custom tag inherited from a pool (that is, you cannot override a custom tag that is inherited from the pool).

Authentication with [Azure DevOps Services](https://docs.microsoft.com/en-us/azure/devops/?view=vsts) is done automatically when you authenticate using Azure Active Directory (Azure AD). The Azure DevOps Services organization must be linked to the same Azure AD tenant as Databricks.

In Azure Databricks, set your Git provider to Azure DevOps Services on the User Settings page:

1. Click the User icon Account Icon at the top right of your screen and select **User Settings**.



# Widgets

* 07/15/2020
* 5 minutes to read
  + [[https://github.com/mamccrea.png?size=32](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/notebooks/widgets.md)](https://github.com/MicrosoftDocs/databricks-pr/blob/live/databricks/notebooks/widgets.md" \o "1 Contributor)

Input widgets allow you to add parameters to your notebooks and dashboards. The widget API consists of calls to create various types of input widgets, remove them, and get bound values.

Widgets are best for:

* Building a notebook or dashboard that is re-executed with different parameters
* Quickly exploring results of a single query with different parameters

ScalaCopy

dbutils.widgets.help()

**Widget types**

There are 4 types of widgets:

* text: Input a value in a text box.
* dropdown: Select a value from a list of provided values.
* combobox: Combination of text and dropdown. Select a value from a provided list or input one in the text box.
* multiselect: Select one or more values from a list of provided values.

Widget dropdowns and text boxes appear immediately below the toolbar in your notebook.

he Create Library dialog displays.

