IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

BYTEWEAVR, LLC,	§	
	§	
Plaintiff,	§	
	§	
v.	§	**JURY TRIAL DEMANDED**
	§	
DATABRICKS, INC.,	§	
	§	**CIVIL ACTION NO. _____**
Defendant.	§	
	§	
	§	

PLAINTIFF'S ORIGINAL COMPLAINT FOR PATENT INFRINGEMENT

Plaintiff BYTEWEAVR, LLC files this Complaint in this Eastern District of Texas (the "District") against Defendant Databricks, Inc. for infringement of U.S. Patent No. 6,839,733 (the "'733 patent"), U.S. Patent No. 7,949,752 (the "'752 patent"), U.S. Patent No. 6,965,897 (the "'897 patent"), U.S. Patent No. 7,082,474 (the "'474 patent"), U.S. Patent No. 8,275,827 (the "'827 patent"), U.S. Patent No. 6,862,488 (the "'488 patent"), and U.S. Reissued Patent No. RE42153 (the "'153 patent") (collectively referred to as the "Asserted Patents").

THE PARTIES

1. BYTEWEAVR, LLC ("BYTEWEAVR" or "Plaintiff") is a Texas limited liability company, with registered address at 17350 State Hwy 249, Suite 220, Houston, Texas 77064.

2. On information and belief, Defendant Databricks, Inc. ("Databricks" or "Defendant") is a corporation formed and organized under the laws of Delaware with its principal executive offices and corporate headquarters located at 160 Spear Street, San Francisco, CA 94105. Databricks is registered to do business in Texas. *See* TEXAS SECRETARY OF STATE, https://direct.sos.state.tx.us/ at Filing No. 804532217 (showing that Databricks has been registered

since 2022 as a foreign corporation in Texas) (last visited Oct. 11, 2023). Databricks' registered agent in Texas is United Agent Group Inc. located at 5444 Westheimer #1000, Houston, TX 77056.

3. Databricks was founded in 2013, and in September of 2023, announced a valuation of $43 billion dollars *See Databricks Raises Series I Investment at $43B Valuation*, DATABRICKS, https://www.databricks.com/company/newsroom/press-releases/databricks-raises-series-i-investment-43b-valuation. Over 50% of Fortune 500 companies use Databricks' platforms.

4. On information and belief, Databricks provides data management and analytics via "combin[ing] the best elements of data lakes and data warehouses to help you reduce costs and deliver on your data and AI initiatives faster," referred to as the "Databricks Lakehouse." See *The Databricks Lakehouse Platform*, DATABRICKS, https://www.databricks.com/product/data-lakehouse (last visited Oct. 12, 2023). Databricks is "a unified, open analytics platform for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale." *See What is Databricks?*, DATABRICKS, https://docs.databricks.com/en/introduction/index.html (last visited January 23, 2024).

5. In 2020, Databricks introduced the "Lakehouse," which was based on open source data formats such as Apache Parquet, and Hadoop. *See What is a Lakehouse?*, DATABRICKS, available at https://www.databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html (Last visited on December 15, 2023). The Databricks Lakehouse is based on a variety of open-source data lake and data warehouse technologies such as Hadoop and Apache Parquet. *See Lakehouse: A New Generation of Open Platforms that Unify Data Warehousing and Advanced Analytics?*, MICHAEL ARMBRUST, available at https://www.cidrdb.org/cidr2021/papers/cidr2021_paper17.pdf (January 2021).

6. On information and belief, the Databricks Lakehouse Platform includes several different products including at least Databricks SQL (a serverless data warehouse), Delta Lake, Unity Catalog, Databricks Marketplace, and Data Intelligence Platform. *See Databricks SQL*, DATABRICKS, https://www.databricks.com/product/databricks-sql (last visited on December 15, 2023); *Delta Lake on Databricks*, DATABRICKS, https://www.databricks.com/product/delta-lake-on-databricks (last visited on December 15, 2023); *Unity Catalog*, DATABRICKS, https://www.databricks.com/product/unity-catalog (last visited on December 15, 2023); *Databricks Marketplace*, DATABRICKS, https://www.databricks.com/product/marketplace (last visited on December 15, 2023); *Data Streaming*, DATABRICKS, https://www.databricks.com/product/data-streaming (last visited on December 15, 2023). The Data Intelligence Platform "integrates with cloud storage and security in [the customer's] cloud account, and manages and deploys cloud infrastructure on" behalf of the customer. *See What is Databricks?*, DATABRICKS, https://docs.databricks.com/en/introduction/index.html (last visited January 23, 2024).

7. The Databricks Lakehouse Platform and their components are utilized by customers of Databricks across industries, including Energy, Financial Services, Telecommunications, Technology, Advertising, and Healthcare and Life Sciences, among many others. *See Databricks for Industry*, DATABRICKS, https://www.databricks.com/solutions (last visited December 15, 2023); *Industry Solutions*, DATABRICKS, https://www.databricks.com/solutions/accelerators (last visited December 22, 2023). On information and belief, Databricks collects revenues and profits from the installation, licensing, and use of the Databricks Lakehouse Platform. *See Databricks Pricing,* DATABRICKS, https://www.databricks.com/product/pricing (last visited December 15, 2023).

Databricks, for example, offers a "Pay as you go" pricing model where the price is based on the Databricks Unit or "DBU". *See id*.

8. On information and belief, Defendant Databricks on its own and/or via subsidiaries, distributors, and affiliates maintains a corporate and commercial presence in the United States, including in Texas and this District. Defendant maintains its business presence in the U.S. and Texas via at least the following activities: 1) distributing and providing its Databricks Platforms, among other products and services of Databricks, to customers; 2) maintaining an online presence (https://www.databricks.com) that solicits sales and sales inquiries of and provides customer support for Databricks products and services; 3) registering to do business in Texas; 4) employing persons across the world who support the development of products and services and provide customer support to U.S. residents and companies, and 5) employing persons in the United States, including residents of Texas and this District. For example, Defendant employs Texas residents in at least one location in the Plano, Texas area at 6900 Dallas Pkwy, Suite 02-106, Plano, Texas 75024. *See, e.g.*, *Worldwide Locations*, DATABRICKS, https://www.databricks.com/company/contact/office-locations (showing Databricks locations in the U.S. and Texas). Thus, Defendant Databricks does business in the United States, the state of Texas, and in the Eastern District of Texas.

JURISDICTION AND VENUE

9. This action arises under the patent laws of the United States, namely 35 U.S.C. §§ 271, 281, and 284-285, among others.

10. This Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

11. On information and belief, Defendant Databricks is subject to this Court's specific and general personal jurisdiction pursuant to due process and/or the Texas Long Arm Statute, due at least to its substantial business in this State and this District, including: (A) at least part of its infringing activities alleged herein, including its registration to do business in Texas, which purposefully avail the Defendant of the privilege of conducting those activities in this state and this District and, thus, submits itself to the jurisdiction of this Court; and (B) regularly doing or soliciting business, engaging in other persistent conduct targeting residents of Texas and this District, and/or deriving substantial revenue from infringing goods offered for sale, sold, and imported and services provided to and targeting Texas residents and residents of this District vicariously through and/or in concert with its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

12. For example, Databricks has corporate offices in the United States, including in Texas. Databricks owns or leases a corporate office in this district at 6900 Dallas Pkwy, Suite 02-106, Plano, Texas 75024. *See Worldwide Locations*, DATABRICKS, https://www.databricks.com/company/contact/office-locations (last visited Oct. 12, 2023).

13. Such a corporate and commercial presence by Defendant Databricks furthers the development, design, manufacture, importation, distribution, sale, offering for sale, and use of Defendant's infringing data management and analytics products and services in Texas, including in this District.

14. Databricks utilizes a Partner Connect program to allow for the easy and quick integration of the Databricks Lakehouse platform with companies all over the country. See *Partner Connect*, DATABRICKS, https://www.databricks.com/partnerconnect (last visited Dec. 22, 2023). Through utilization of its business segments and partners Databricks has committed acts of direct

and/or indirect patent infringement within Texas, this District, and elsewhere in the United States, giving rise to this action and/or has established minimum contacts with Texas such that personal jurisdiction over Databricks would not offend traditional notions of fair play and substantial justice.

15. On information and belief, Databricks has placed and continues to place infringing data management and analytics products and services, including the Databricks Platforms and their components into the U.S. stream of commerce. *See Databricks Pricing*, DATABRICKS, https://www.databricks.com/product/pricing (last visited Oct. 12, 2023). Databricks has placed such products and services into the stream of commerce with the knowledge and understanding that such products and services are, will be, and continue to be sold, offered for sale, and/or imported into the State of Texas and this District. *See Litecubes, LLC v. Northern Light Products, Inc.*, 523 F.3d 1353, 1369-70 (Fed. Cir. 2008) ("[T]he sale [for purposes of § 271] occurred at the location of the buyer."); *see also Semcon IP Inc. v. Kyocera Corporation*, No. 2:18-cv-00197-JRG, 2019 WL 1979930, at *3 (E.D. Tex. May 3, 2019) (denying accused infringer's motion to dismiss because plaintiff sufficiently plead that purchases of infringing products outside of the United States for importation into and sales to end users in the U.S. may constitute an offer to sell under § 271(a)).

16. Venue is proper in this District pursuant to 28 U.S.C. §§ 1391(c) and 1400(b). As alleged herein, Defendant Databricks has committed acts of infringement in this District. As further alleged herein, Defendant Databricks, via its own operations and employees located there, has a regular and established place of business in this District. Databricks' regular and established place of business is at least at 6900 Dallas Pkwy, Suite 02-106, Plano, Texas 75024, which according to publicly available records is located in Collin County. Accordingly, Databricks may be sued in this district under 28 U.S.C. § 1400(b).

17. On information and belief, Defendant Databricks has significant ties to, and presence in, the State of Texas and the Eastern District of Texas, making venue in this District both proper and convenient for this action.

THE ASSERTED PATENTS AND TECHNOLOGY

18. The Asserted Patents cover various aspects of network systems extensible by users as subscribers to a network service. Such extensibility by users of network services includes interaction with the network by creating, copying, modifying, editing, and deleting agents. Such agents are invoked by users to consume service resources. Such network systems further include automation of validation of equipment and/or processes via a user interface and validation processing engine.

19. The ’733 patent involves at least admitting a user to a network system wherein at least one agent is operable to consume a service resource (e.g., CPU, memory resource, etc.) while utilizing a service to perform a task for the user. The user is allowed to create, modify, or delete the agent within the network system.

20. The ’752 patent involves at least receiving, using a computing device, data for creating a network-based agent. An execution of the network-based agent is invoked in response to receiving a URL that defines a type of event and identifies the agent. Invoking execution of the network-based agent uses a service and a service resource that is consumed by the network-based agent for performing the invoking operation. The result of the operation is communicated over a network communication link.

21. The ’488 patent involves at least automating, in a computing environment, the validation of equipment and/or processes for use, for example, in a pharmaceutical and/or bio-technology manufacturing facility. A user interface is provided that accepts and/or displays data

representative of validation processing and/or validation workflow management information. A validation processing engine is provided that comprises a processing rule that operates to produce validation protocol information.

22. The ’897 patent involves at least arranging data in a data file on a mixed format physical layout. This layout has a plurality of fixed-sized fields, a plurality of variable-sized fields, and a plurality of offset slots. The fixed-sized fields are of a first size and the offset slots are of a second size. The data on the mixed format physical layout is divided into the fixed-sized fields and the variable sized fields. The data of the variable sized fields and the fixed-sized fields is compressed.

23. The ’474 patent involves at least receiving client requests from server systems to use a distributed processing system to process a workload. The first workload is sent to a host distributed device. An index defining a location of data required to process the first workload is sent to the host distributed device. The data is accessed from a first data address in the index. And the index is updated to include a storage address of storage coupled to the host distributed device as a location of the data.

24. The ’827 patent involves at least configuring a distributed processing system with distributed devices coupled to a network. The devices include client agents that process workloads for the system. The client agents have software-based network attached storage (NAS) components that assess unused or underutilized storage resources in distributed devices. The NAS devices have storage resources related to the unused or underutilized storage resources. The system processes data storage or access workloads and enables the distributed devices to store location information associated with data stored by the distributed devices through the use of client agents. At least one

of the distributed devices is enabled to function as a stand-alone dedicated NAS device through the use of the client agents.

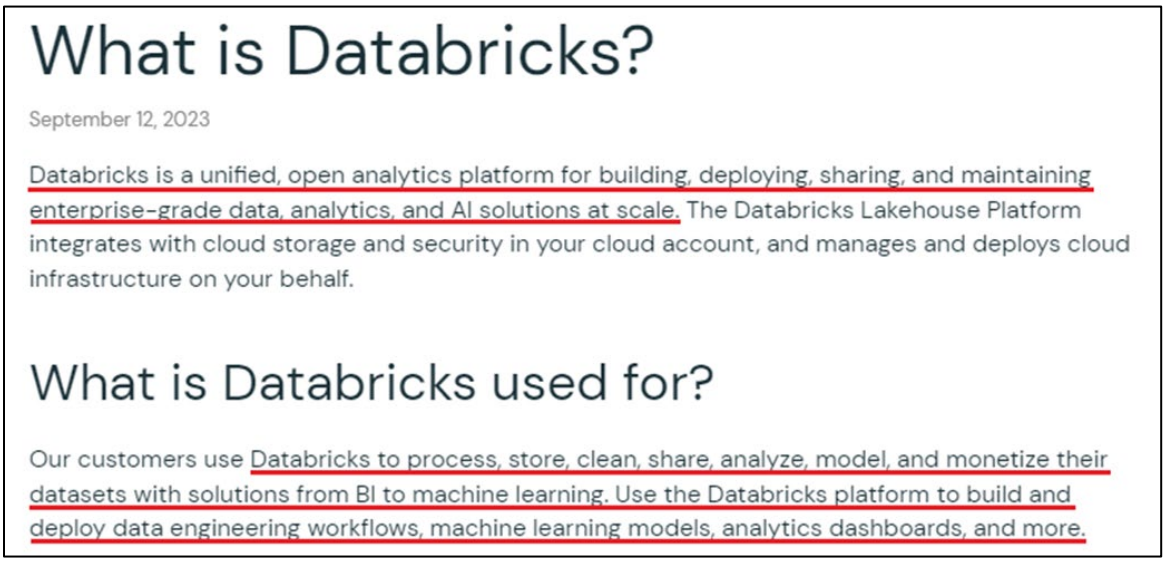
25. The '153 patent involves at least a server system coupled to a network with network-connected distributed client systems having under-utilized capabilities. The client systems run a client agent program to provide workload processing for a project of a distributed computing platform. The server system distributes project workloads to the client systems and distributes initial project and poll parameters to the client systems. Poll communications are received from the client systems during the processing of project workloads and a dynamic snapshot information of a current project status is provided based on the poll communications. The poll communications are analyzed to determine whether to modify the initial project and poll parameters, which indicate how many client systems are active in the project. If fewer client systems are desired, including within a polling response communications, the number of actively participating client systems is reduced. And if a greater number of client systems is desired, then client systems are added to active participation in the project. The poll response communications are sent to the client systems to modify the initial project and poll parameters, depending on the analysis of the poll communications. The steps of receiving and analyzing poll communications and sending poll response communications are repeated to dynamically coordinate project activities of the client systems during project operations.

26. On information and belief, a significant portion of the operating revenue of Defendant is derived from the development, design, manufacture, distribution, licensing, sale, offering for sale, and use of Databricks' data management and analytics products and services, including the Databricks Lakehouse Platforms and their components. *See, Databricks Raises Series I Investment at $43B Valuation,* DATABRICKS, https://www.databricks.com/company/newsroom/press-releases/databricks-raises-series-i-investment-43b-valuation (last visited

December 15, 2023). For example, Defendant Databricks utilizes its subsidiaries, distributors, resellers, vendors, contractors, dealers, installers, trainers, and other partners to provide data management and analytics products and services and related products and services to consumers. For the year 2023, Defendant reported a $1.5 billion revenue run rate. I*d*. In 2023, Defendant reported over 10,000 global customers and over 50% of the Fortune 500 utilizing the Databricks Lakehouse Platform. *Id*. Thus, the majority of Databricks' revenue derives from Databricks' data management and analytics products and services distributed, licensed, sold and offered for sale by consumers in the United States.

27. The Asserted Patents cover Defendant's data management and analytics products and components, software, services, and processes related to same that cover various aspects of network systems extensible by users as subscribers to a network service, including such network systems that allow a user to interact with the network by creating, copying, modifying, editing, and deleting agents to support consumption of network services and/or allow a user to provide for automation of validation of equipment and/or processes via a user interface and validation processing engine (collectively referred to herein as the "Accused Instrumentalities"). *See, e.g., The Databricks Data Intelligence Platform,* DATABRICKS, https://www.databricks.com/product/data-intelligence-platform (last visited Dec. 18, 2023). Defendant's infringing Accused Instrumentalities include, but are not limited to, components of Databricks Lakehouse Platform, including, but not limited to networks, methods, processes, software, firmware, distributions, infrastructure, environments, interfaces, hosts, tools, data connections, databases, resources, and related services provided to partners, users, customers, clients, and consumers via at least Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, and Databricks Spark Applications.

28. The Asserted Patents, including claim 37 of the '733 patent, cover Accused Instrumentalities of Defendant that utilize Databricks' Lakehouse, which, as described below, provides a managed machine learning lifecycle system including and interfacing with MLflow. Databricks provides a web-based interface for interacting with MLflow to load, train, run, and track using Databricks clusters. The MLflow model utilizes different Databricks services such as Databricks' file system for defining the path of the MLflow model.



What is Databricks?

September 12, 2023

Databricks is a unified, open analytics platform for building, deploying, sharing, and maintaining enterprise-grade data, analytics, and AI solutions at scale. The Databricks Lakehouse Platform integrates with cloud storage and security in your cloud account, and manages and deploys cloud infrastructure on your behalf.

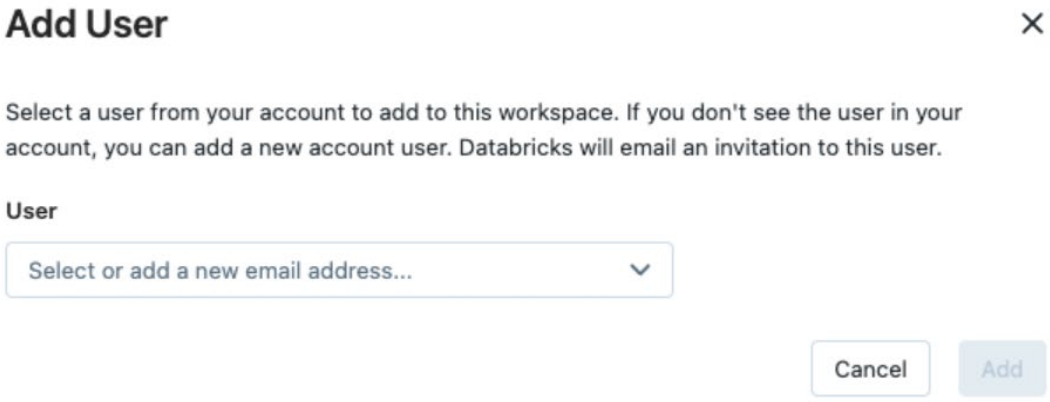
What is Databricks used for?

Our customers use Databricks to process, store, clean, share, analyze, model, and monetize their datasets with solutions from BI to machine learning. Use the Databricks platform to build and deploy data engineering workflows, machine learning models, analytics dashboards, and more.

See What is Databricks, DATABRICKS, https://docs.databricks.com/en/introduction/index.html (last visited October 17, 2023)

mlflow, DATABRICKS, https://web.archive.org/web/20190828125229/https://databricks.com/mlflow (accessible at least in August 28, 2019).

29. As shown below, a user may be added to a workspace and provided with credentials for registration and admission to the Databricks network system.

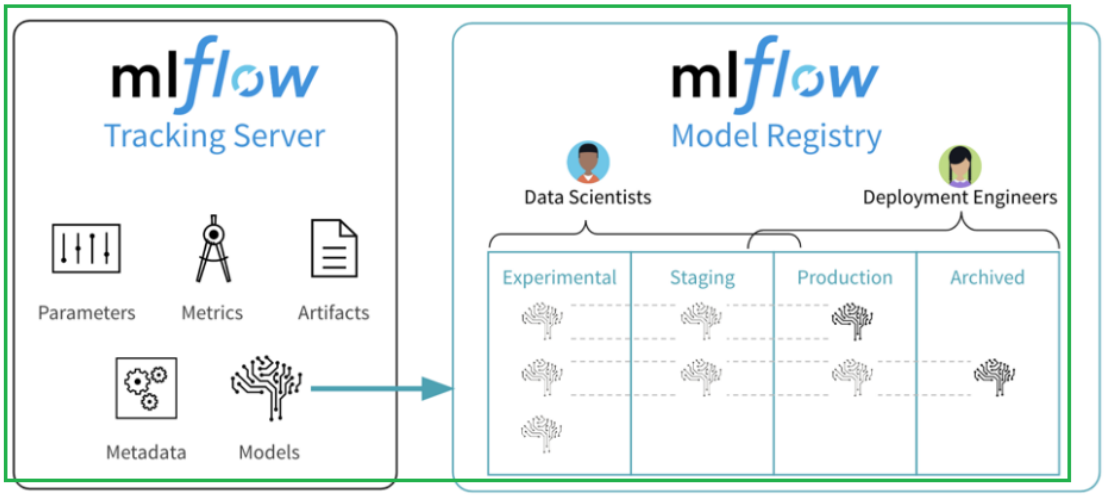


https://docs.databricks.com/en/administration-guide/users-groups/index.html

30. MLflow is a managed machine learning lifecycle system that includes a model management system.

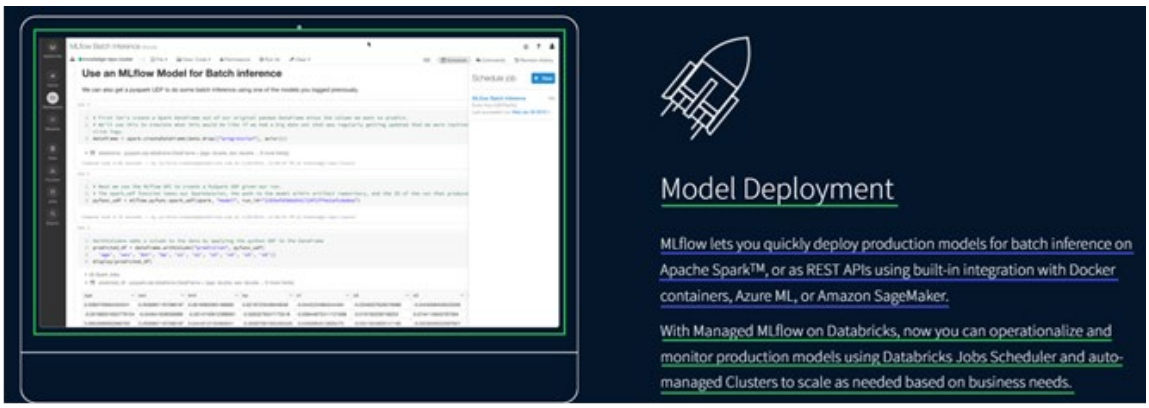
MLflow already has the ability to track metrics, parameters, and artifacts as part of experiments, package models and reproducible ML projects, and deploy models to batch or real-time serving platforms.

The MLflow Model Registry builds on MLflow's existing capabilities to provide organizations with one central place to share ML models, collaborate on moving them from experimentation to testing and production, and implement approval and governance workflows. Since we started MLflow, model management was the top requested feature among our open source users, so we are excited to launch a model management system that integrates directly with MLflow.

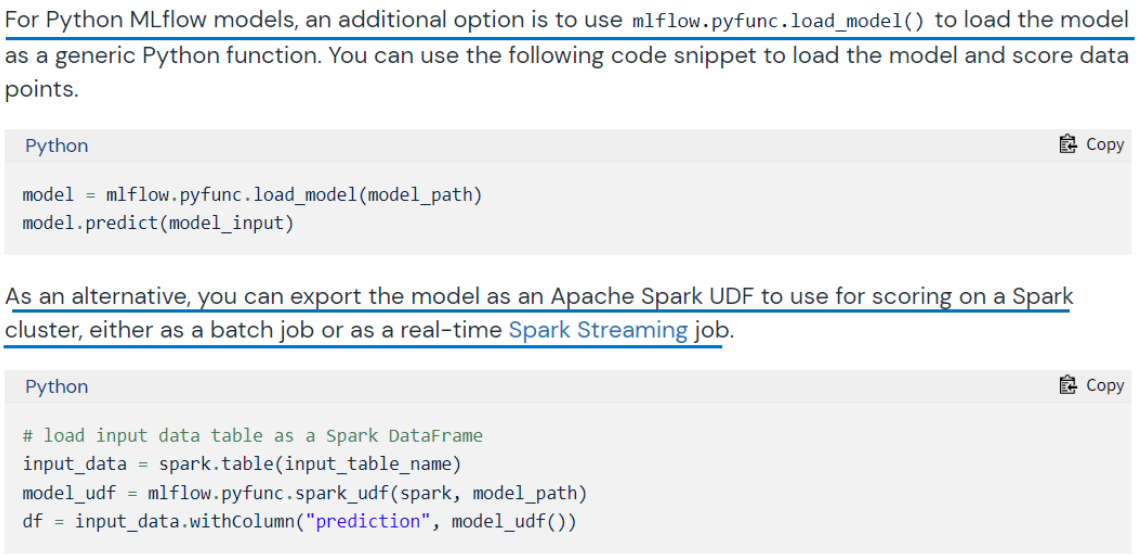


See Introducing the MLflow Model Registry, DATABRICKS, https://www.databricks.com/blog/2019/10/17/introducing-the-mlflow-model-registry.html (last visited Dec. 15, 2023).

31. Databricks provides access to the Databricks Jobs Scheduler and auto-managed clusters to use with MLflow, allowing users to schedule workflows and/or perform a variety of tasks.



See MLflow, DATABRICKS, *https://web.archive.org/web/20190828125229/https://databricks.com/mlflow*



For Python MLflow models, an additional option is to use `mlflow.pyfunc.load_model()` to load the model as a generic Python function. You can use the following code snippet to load the model and score data points.

Python Copy

```
model = mlflow.pyfunc.load_model(model_path)
model.predict(model_input)
```

As an alternative, you can export the model as an Apache Spark UDF to use for scoring on a Spark cluster, either as a batch job or as a real-time Spark Streaming job.

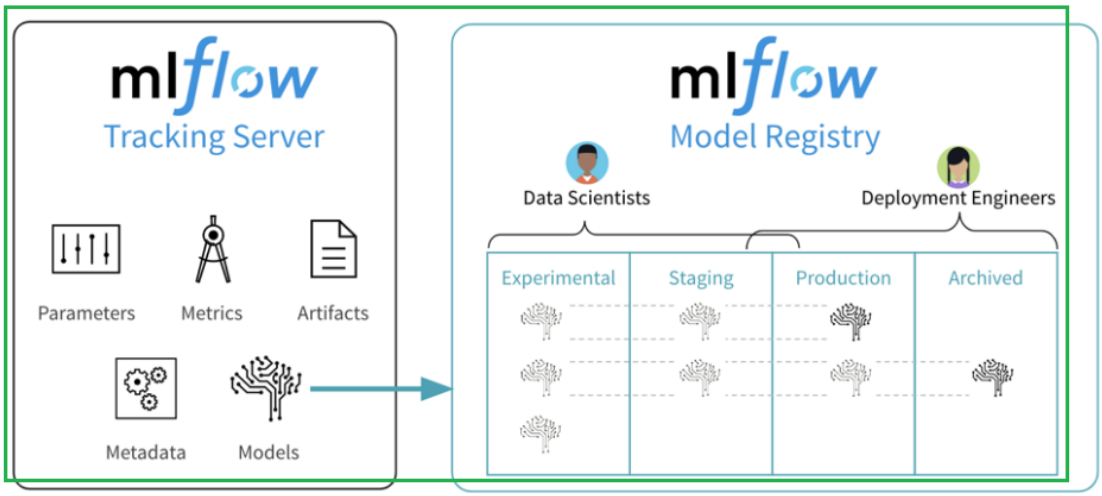
Python Copy

```
# load input data table as a Spark DataFrame
input_data = spark.table(input_table_name)
model_udf = mlflow.pyfunc.spark_udf(spark, model_path)
df = input_data.withColumn("prediction", model_udf())
```

See Log, load, register, and deploy MLflow models, DATABRICKS, https://docs.databricks.com/en/mlflow/models.html (last visited Dec. 15, 2023).

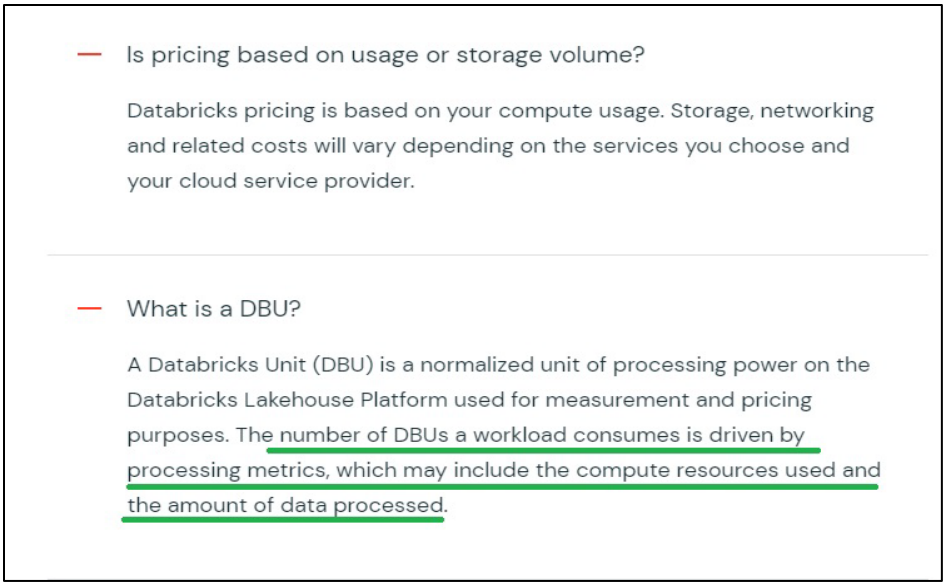
32. As shown below, the user creates an agent, via a MLflow model, which is operable to perform a task for the user, such as tracking metrics, parameters, and artifacts, packaging models and reproductible ML projects, and deploying models to batch or real-time serving platforms.

> MLflow already has the ability to track metrics, parameters, and artifacts as part of experiments, package models and reproducible ML projects, and deploy models to batch or real-time serving platforms.
>
> The MLflow Model Registry builds on MLflow's existing capabilities to provide organizations with one central place to share ML models, collaborate on moving them from experimentation to testing and production, and implement approval and governance workflows. Since we started MLflow, model management was the top requested feature among our open source users, so we are excited to launch a model management system that integrates directly with MLflow.



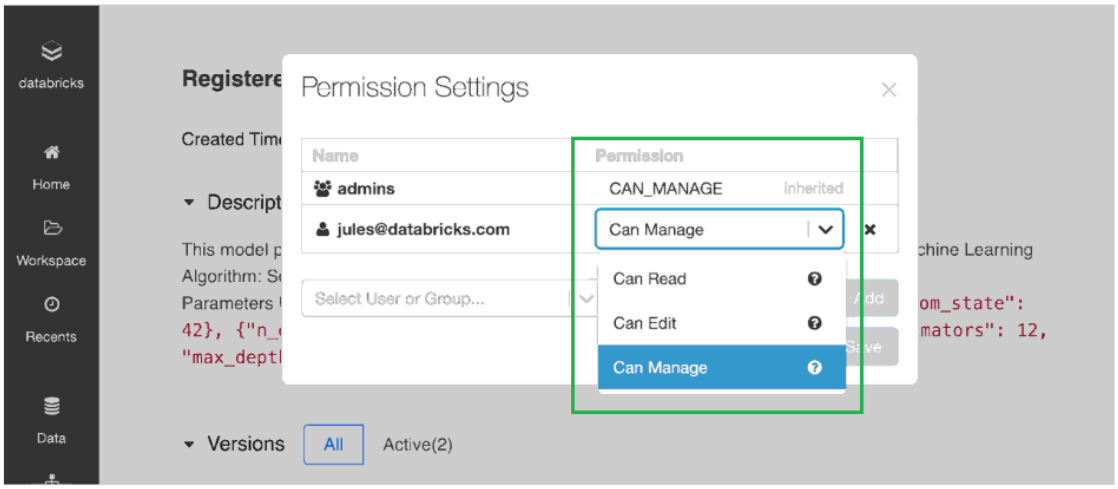
See Introducing the MLflow Model Registry, DATABRICKS, https://www.databricks.com/blog/2019/10/17/introducing-the-mlflow-model-registry.html (last visited Dec. 15, 2023).

33. Performance of tasks utilizing the Databricks Platforms, including Lakehouse, consume resources. For Lakehouse, Databricks quantifies the consuming of resources with DBUs. As explained below, a DBU quantifies the amount of data processed and the amount of compute resources used.



See Databricks Pricing FAQ, DATABRICKS, https://www.databricks.com/product/pricing (last visited October 17, 2023).

34. The user, can create, modify, or delete the agent (e.g., a model in MLflow) within the network system. For example, the MLflow editor allows a Databricks user create, modify, or delete the model.



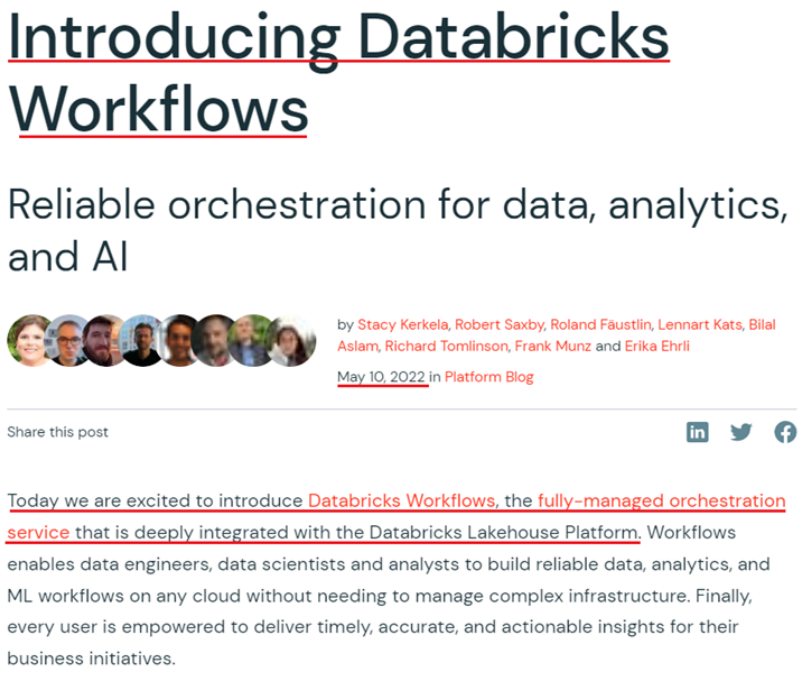
See Databricks Extends MLflow Model Registry with Enterprise Features, DATABRICKS, https://www.databricks.com/blog/2020/04/15/databricks-extends-mlflow-model-registry-with-enterprise-features.html (last visited December 15, 2023).

As shown in the table below, an administrator can assign four permission levels to models registered in the Model Registry: **No permissions**, **Read**, **Edit**, and **Manage**. Depending on team members' requirements to access models, you can grant permissions to individual users or groups for each of the abilities shown below.

Ability	No Permissions	Read	Edit	Manage
Create a model	X	X	X	X
View model and its model versions in a list		X	X	X
View model's details, its versions and their details, stage transition requests, activities, and artifact download URIs		X	X	X
Request stage transitions for a model version		X	X	X
Add a new version to model			X	X
Update model and version description			X	X
Rename model				X
Transition model version between stages				X
Approve, reject, or cancel a model version stage transition request				X
Modify permissions				X
Delete model and model versions				X

See Databricks Extends MLflow Model Registry with Enterprise Features, DATABRICKS, https://www.databricks.com/blog/2020/04/15/databricks-extends-mlflow-model-registry-with-enterprise-features.html (last visited December 15, 2023).

35. The Asserted Patents, including claim 24 of the '752 patent, cover Accused Instrumentalities of Defendant that practice a method comprising the steps of receiving, using a computing device (e.g., Databricks server), data (e.g., cluster definition, cluster name, etc.) for creating a network-based agent (e.g., a cluster). As shown below, Databricks Workflows allow a user to create and manage clusters.



Introducing Databricks Workflows

Reliable orchestration for data, analytics, and AI

by Stacy Kerkela, Robert Saxby, Roland Fäustlin, Lennart Kats, Bilal Aslam, Richard Tomlinson, Frank Munz and Erika Ehrli

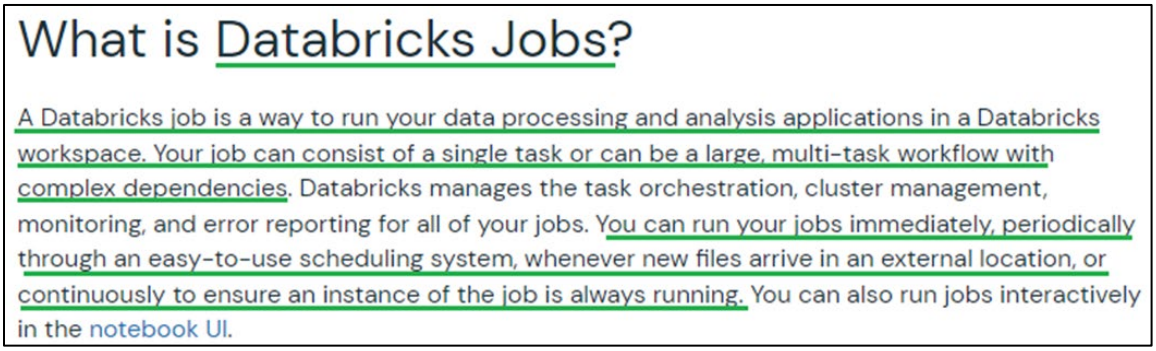
May 10, 2022 in Platform Blog

Share this post

Today we are excited to introduce Databricks Workflows, the fully-managed orchestration service that is deeply integrated with the Databricks Lakehouse Platform. Workflows enables data engineers, data scientists and analysts to build reliable data, analytics, and ML workflows on any cloud without needing to manage complex infrastructure. Finally, every user is empowered to deliver timely, accurate, and actionable insights for their business initiatives.

See Introducing Databricks Workflows, DATABRICKS, https://www.databricks.com/blog/2022/05/10/introducing-databricks-workflows.html (last visited December 15, 2023).

36. A cluster is a set of hosts running inter-dependent services. For creating a cluster, data such as cluster definition, number of nodes, types of service, cluster name, etc. are provided by the user.

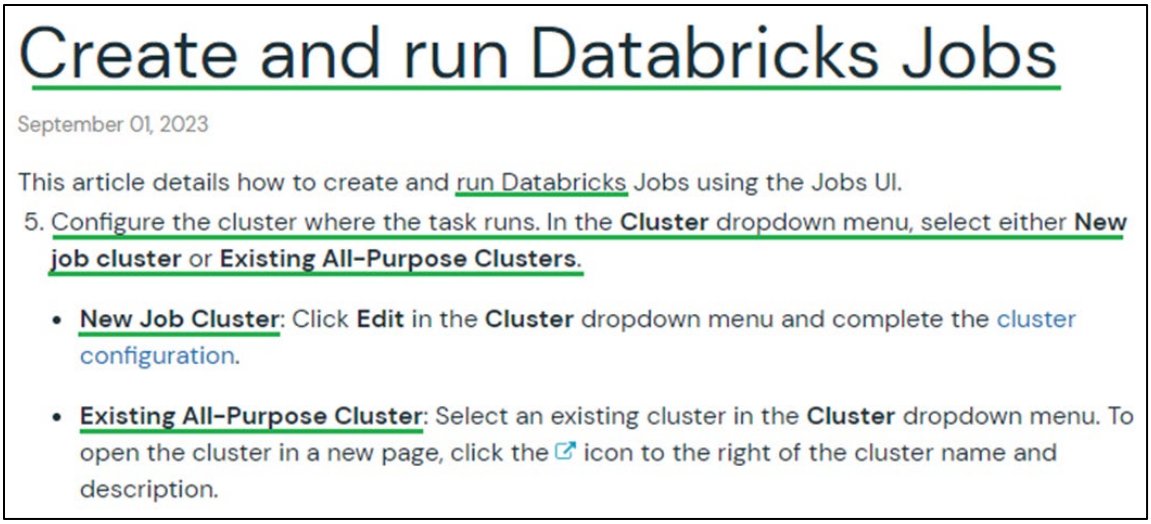


What is Databricks Jobs?

A Databricks job is a way to run your data processing and analysis applications in a Databricks workspace. Your job can consist of a single task or can be a large, multi-task workflow with complex dependencies. Databricks manages the task orchestration, cluster management, monitoring, and error reporting for all of your jobs. You can run your jobs immediately, periodically through an easy-to-use scheduling system, whenever new files arrive in an external location, or continuously to ensure an instance of the job is always running. You can also run jobs interactively in the notebook UI.

See Introduction to Databricks Workflows, DATABRICKS, https://docs.databricks.com/en/workflows/index.html (last visited December 15, 2023).

37. When the user clicks on 'New Job Cluster', the creation of a cluster is triggered.



Create and run Databricks Jobs

September 01, 2023

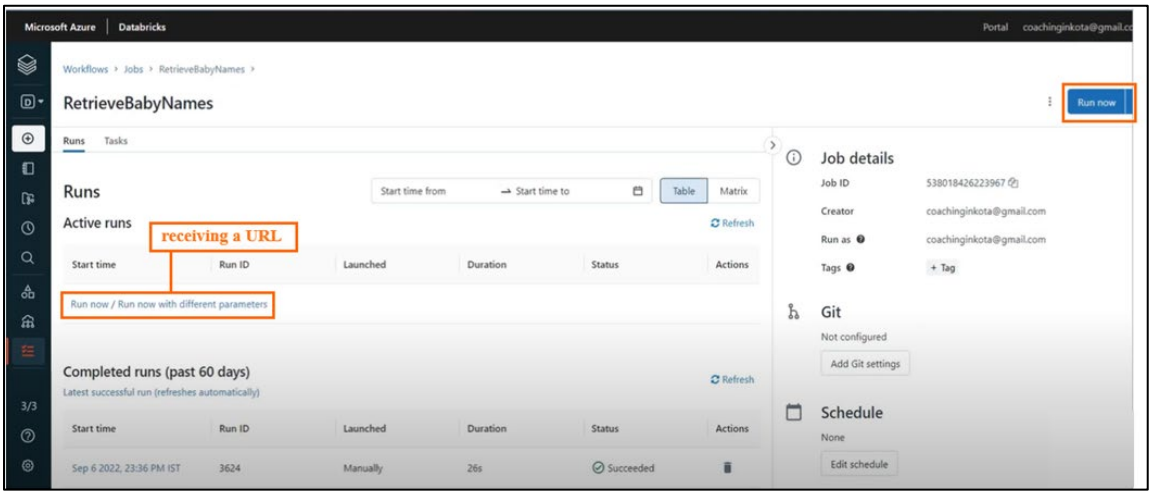
This article details how to create and run Databricks Jobs using the Jobs UI.

5. Configure the cluster where the task runs. In the **Cluster** dropdown menu, select either **New job cluster** or **Existing All-Purpose Clusters**.

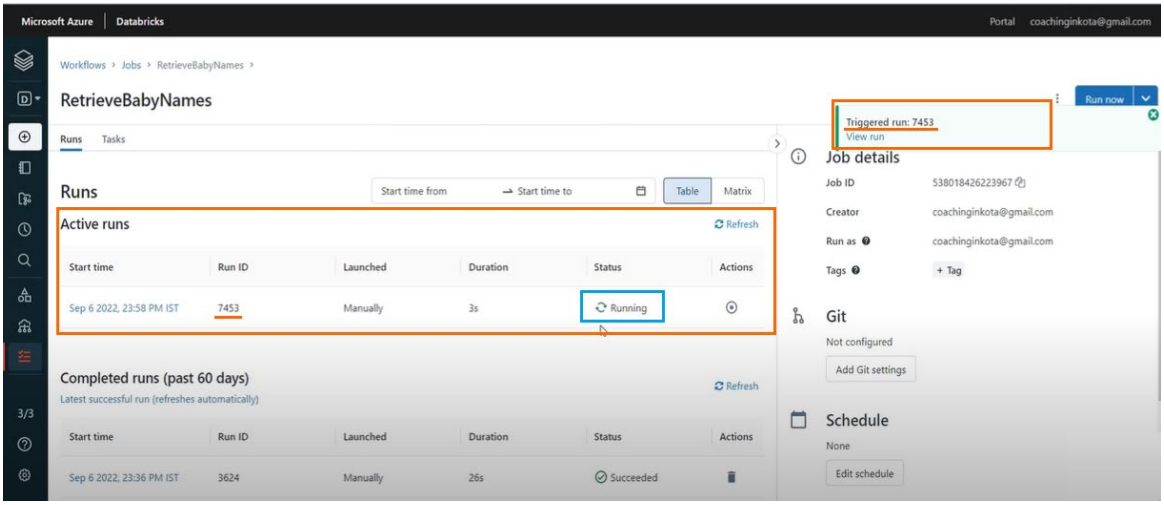
- **New Job Cluster**: Click **Edit** in the **Cluster** dropdown menu and complete the cluster configuration.
- **Existing All-Purpose Cluster**: Select an existing cluster in the **Cluster** dropdown menu. To open the cluster in a new page, click the icon to the right of the cluster name and description.

See Create and run Databricks Jobs, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/create-run-jobs.html (last visited December 15, 2023).

38. When the user clicks on 'Run now' URL, it displays a new ID corresponding to the job and starts the execution of the job by showing status as 'Running,' demonstrating that the Databricks server is invoking and beginning the execution of the cluster.

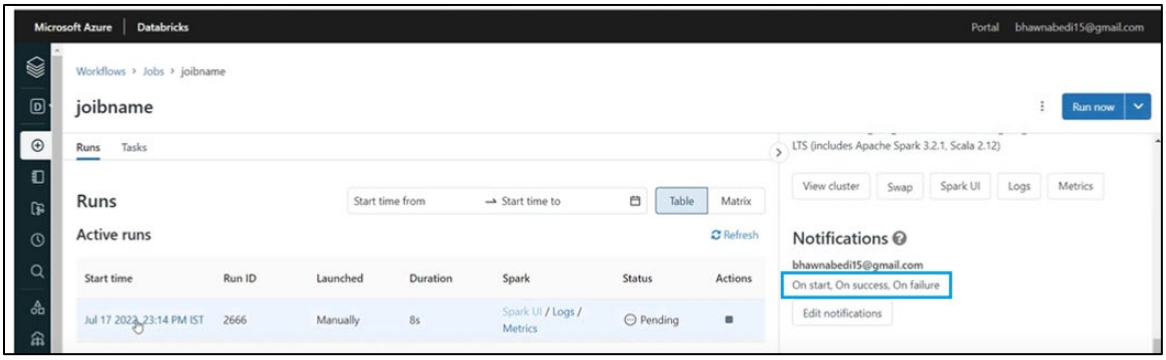


See Create Databricks Jobs, DATABRICKS, available on https://www.youtube.com/watch?v=z0I-vUTq8Z0 (last visited December 15, 2023).



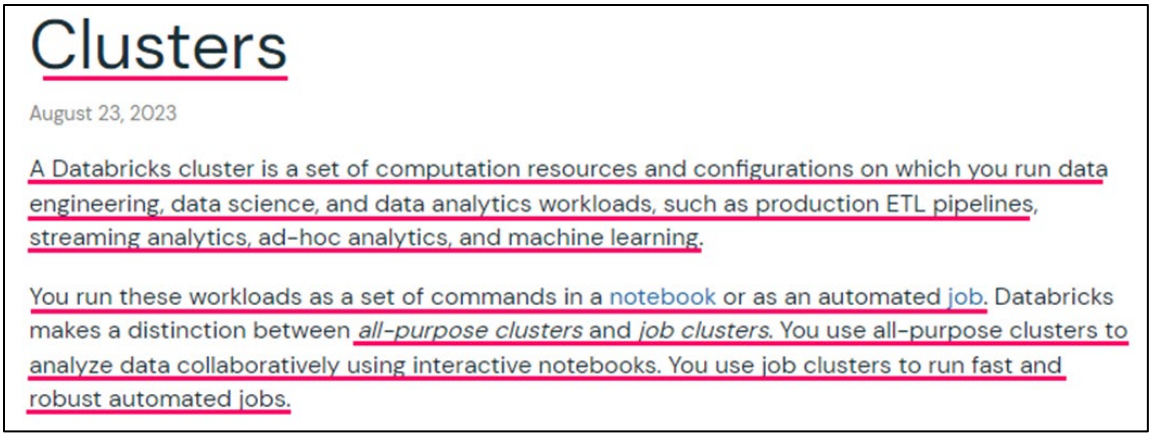
See Create Databricks Jobs, DATABRICKS, available on https://www.youtube.com/watch?v=z0I-vUTq8Z0 (last visited December 15, 2023).

39. In response to receiving a URL, the type of event is defined (e.g., notifications when a job run starts, completes successfully, or fails) and identifies the network-based agent to execute (e.g., a Job in Workflows).



See Configure Jobs in Databricks, CLOUDFITNESS, *https://www.youtube.com/watch?v=DwGi8AGE37M* (last visited December 15, 2023).

40. As shown below, the cluster executes multiple service types as defined for the cluster by the user.



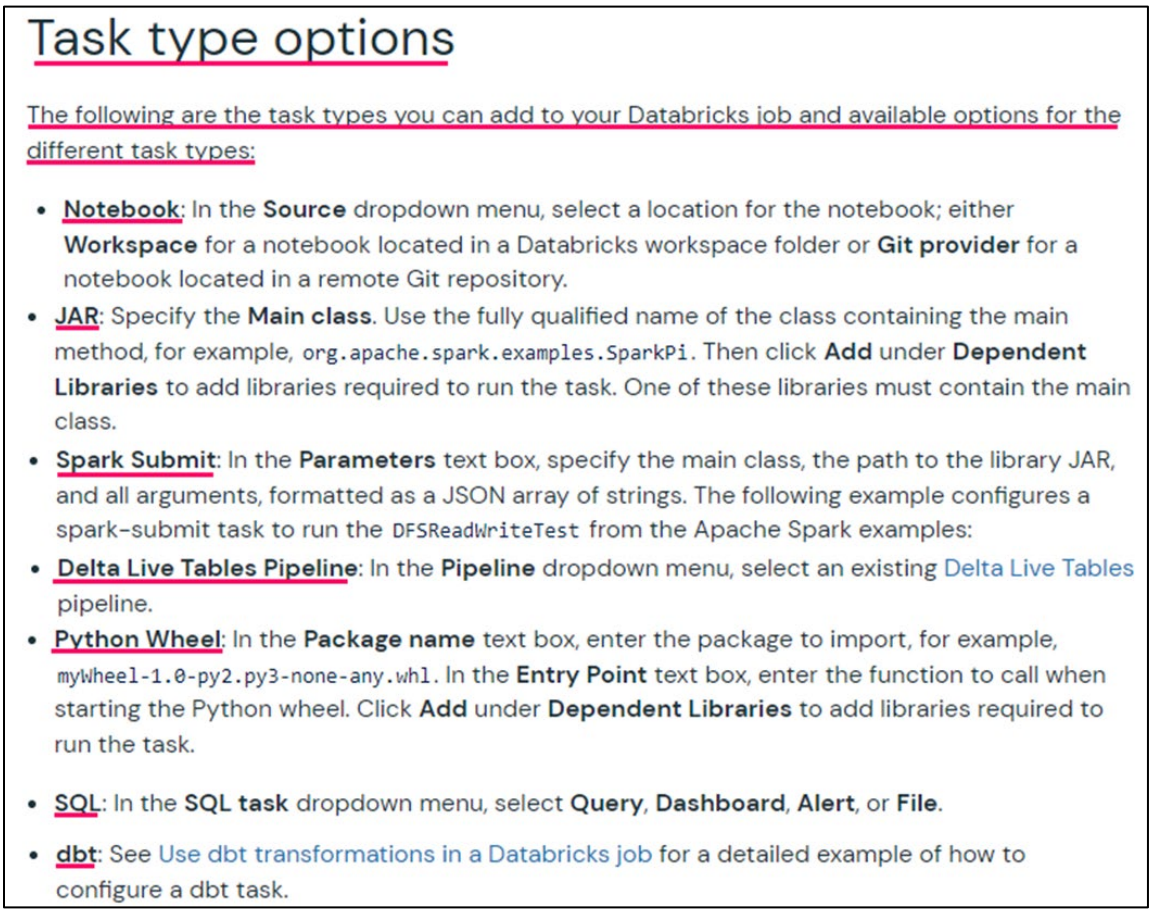
Clusters

August 23, 2023

A 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 or as an automated job. Databricks makes a distinction between *all-purpose clusters* and *job clusters*. You use all-purpose clusters to analyze data collaboratively using interactive notebooks. You use job clusters to run fast and robust automated jobs.

See Compute, DATABRICKS, *https://docs.databricks.com/en/clusters/index.html* (last visited December 15, 2023).



Task type options

The following are the task types you can add to your Databricks job and available options for the different task types:

- **Notebook**: In the **Source** dropdown menu, select a location for the notebook; either **Workspace** for a notebook located in a Databricks workspace folder or **Git provider** for a notebook located in a remote Git repository.
- **JAR**: Specify the **Main class**. Use the fully qualified name of the class containing the main method, for example, `org.apache.spark.examples.SparkPi`. Then click **Add** under **Dependent Libraries** to add libraries required to run the task. One of these libraries must contain the main class.
- **Spark Submit**: In the **Parameters** text box, specify the main class, the path to the library JAR, and all arguments, formatted as a JSON array of strings. The following example configures a spark-submit task to run the `DFSReadWriteTest` from the Apache Spark examples:
- **Delta Live Tables Pipeline**: In the **Pipeline** dropdown menu, select an existing Delta Live Tables pipeline.
- **Python Wheel**: In the **Package name** text box, enter the package to import, for example, `myWheel-1.0-py2.py3-none-any.whl`. In the **Entry Point** text box, enter the function to call when starting the Python wheel. Click **Add** under **Dependent Libraries** to add libraries required to run the task.
- **SQL**: In the **SQL task** dropdown menu, select **Query**, **Dashboard**, **Alert**, or **File**.
- **dbt**: See Use dbt transformations in a Databricks job for a detailed example of how to configure a dbt task.

See Create and run Databricks Jobs, DATABRICKS, *https://docs.databricks.com/en/workflows/jobs/create-run-jobs.html* (last visited December 15, 2023).

41. The resources for these services are consumed and subsequently measured by Databricks by using DBUs.

> Your configuration decisions will require a tradeoff between cost and performance. The primary cost of a cluster includes the Databricks Units (DBUs) consumed by the cluster and the cost of the underlying resources needed to run the cluster. What may not be obvious are the secondary costs such as the cost to your business of not meeting an SLA, decreased employee efficiency, or possible waste of resources because of poor controls.

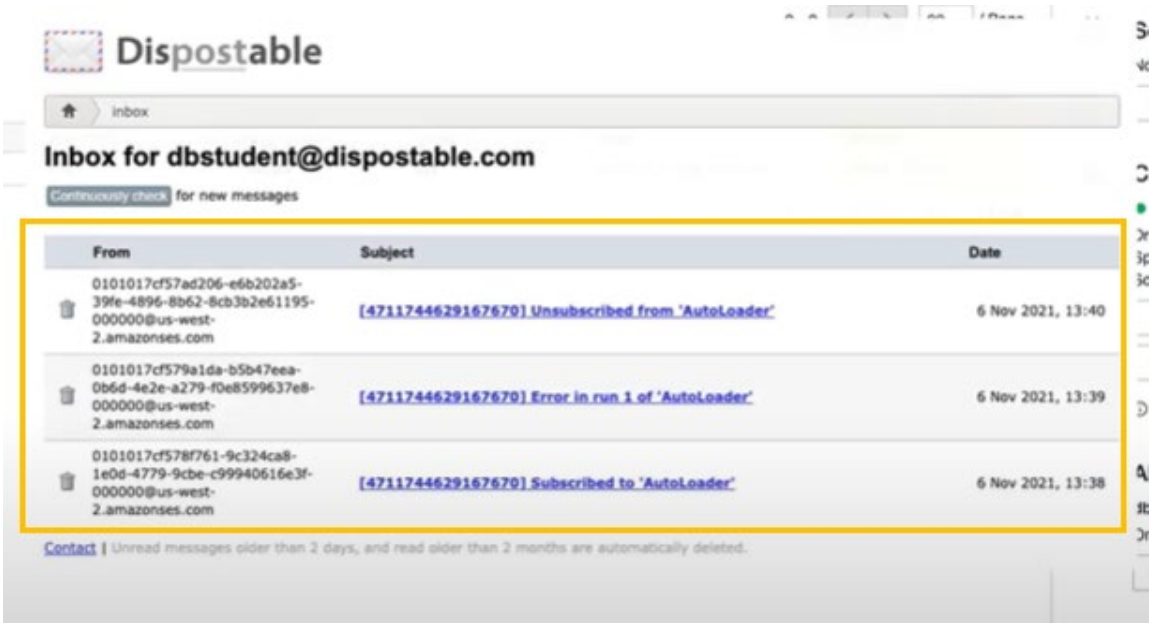
> **All-purpose clusters and job clusters**
>
> When you create a cluster you select a cluster type: an all-purpose cluster or a job cluster. All-purpose clusters can be shared by multiple users and are best for performing ad-hoc analysis, data exploration, or development. Once you've completed implementing your processing and are ready to operationalize your code, switch to running it on a job cluster. Job clusters terminate when your job ends, reducing resource usage and cost.

See Best practices: Cluster configuration, DATABRICKS, https://docs.databricks.com/en/clusters/cluster-config-best-practices.html (last visited December 15, 2023).

42. A result of the cluster creation and start operations (e.g., resource utilization status) is communicated by the Databricks server over a network communication link. As shown below, when the services execute their specific jobs, Databricks communicates the result to the user.

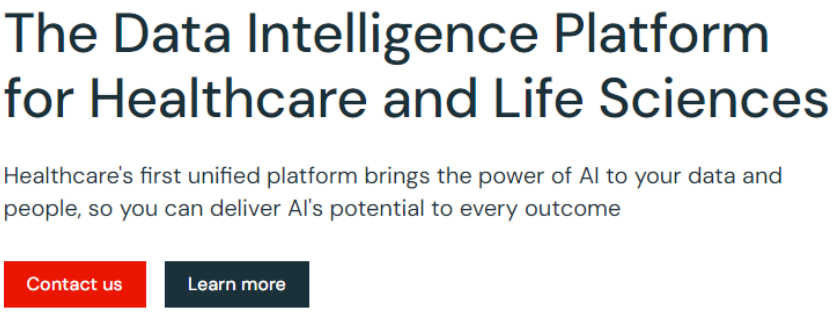
> **Add email and system notifications for job events**
>
> July 28, 2023
>
> You can monitor job runs by configuring notifications when a job run starts, completes successfully, or fails. Notifications can be sent to one or more email addresses or system destinations such as webhook destinations or Slack. You can also integrate email notifications with tools such as PagerDuty.

See Add email and system notifications for job events, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/job-notifications.html (last visited December 15, 2023).



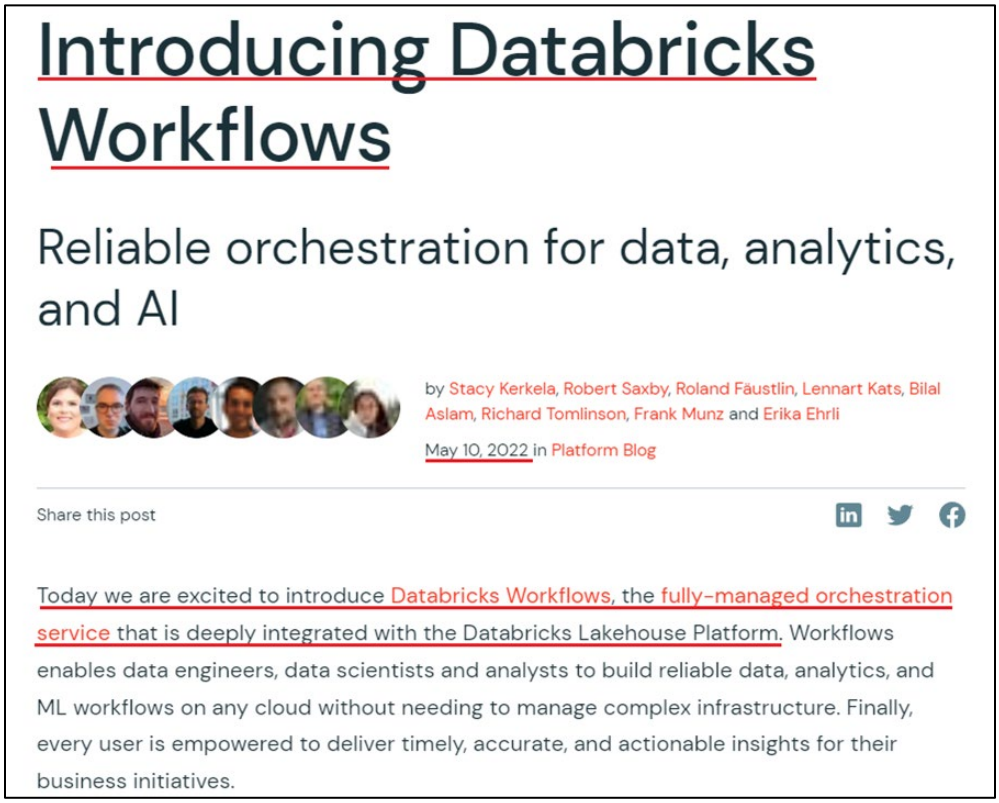
See Tutorial - Schedule a Job and Automate a Workload, DATABRICKS ACADEMY, https://www.databricks.com/resources/demos/videos/data-engineering/schedule-a-job-and-automate-a-workload (last visited December 15, 2023).

43. The Asserted Patents, including claim 11 of the '488 patent, cover Accused Instrumentalities of Defendant that practice in a computing environment (e.g., the Databricks Lakehouse Platform) a method to automate the validation of equipment and/or processes for use in a pharmaceutical and/or bio-technology manufacturing facility. As shown below, the Databricks Lakehouse Platform is being used for pharma and biotech application to automate analytics workflows.



See Healthcare and Life Sciences, DATABRICKS, https://www.databricks.com/solutions/industries/healthcare-and-life-sciences (last visited December 15, 2023).

44. As shown below, the Databricks Lakehouse Platform allows a user to automate the data flow validation process by using Databricks Workflow.



Introducing Databricks Workflows

Reliable orchestration for data, analytics, and AI

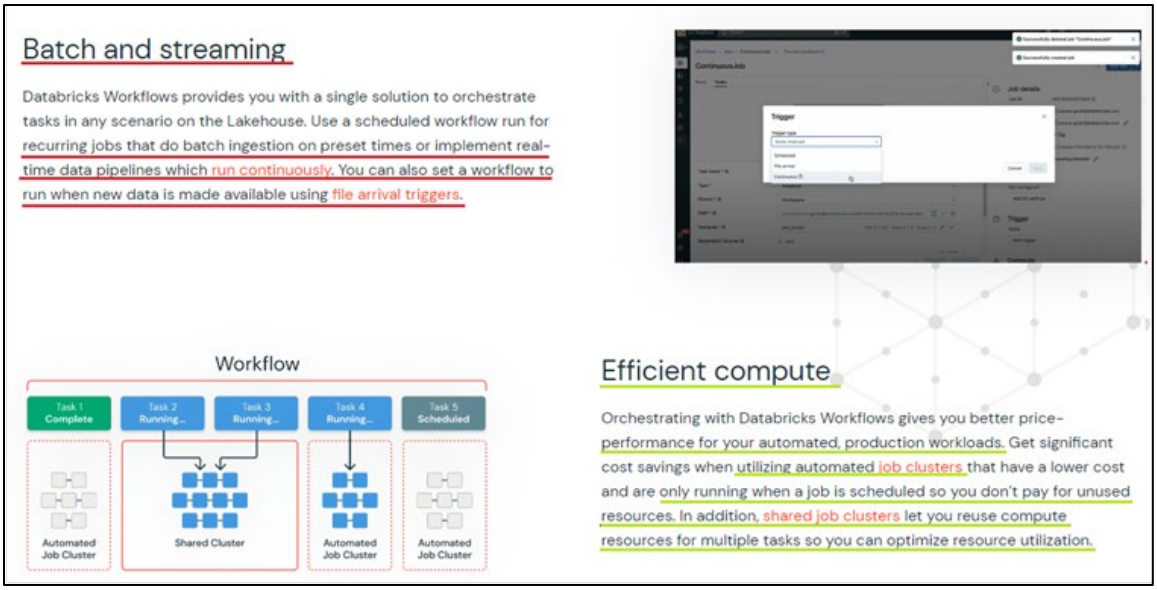
by Stacy Kerkela, Robert Saxby, Roland Fäustlin, Lennart Kats, Bilal Aslam, Richard Tomlinson, Frank Munz and Erika Ehrli

May 10, 2022 in Platform Blog

Share this post

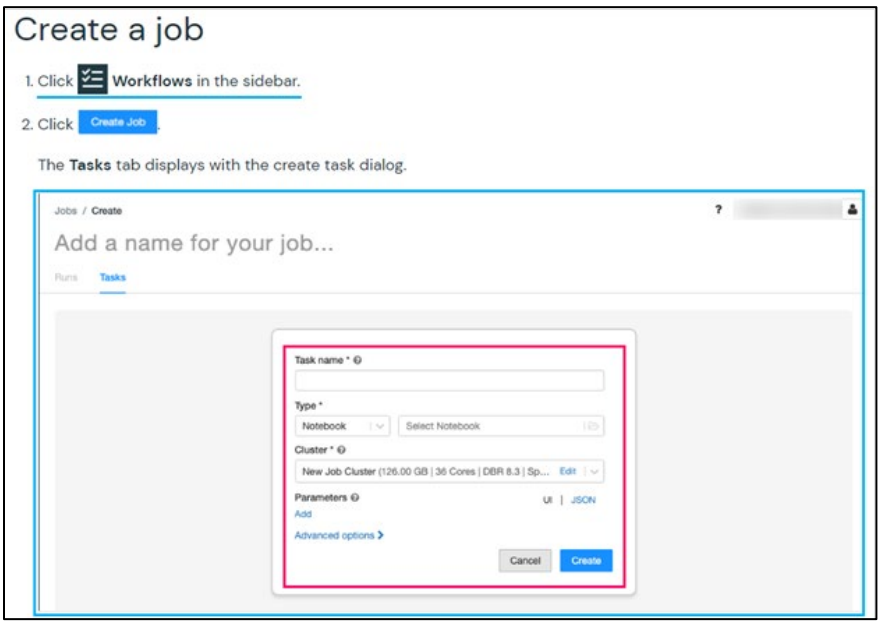
Today we are excited to introduce Databricks Workflows, the fully-managed orchestration service that is deeply integrated with the Databricks Lakehouse Platform. Workflows enables data engineers, data scientists and analysts to build reliable data, analytics, and ML workflows on any cloud without needing to manage complex infrastructure. Finally, every user is empowered to deliver timely, accurate, and actionable insights for their business initiatives.

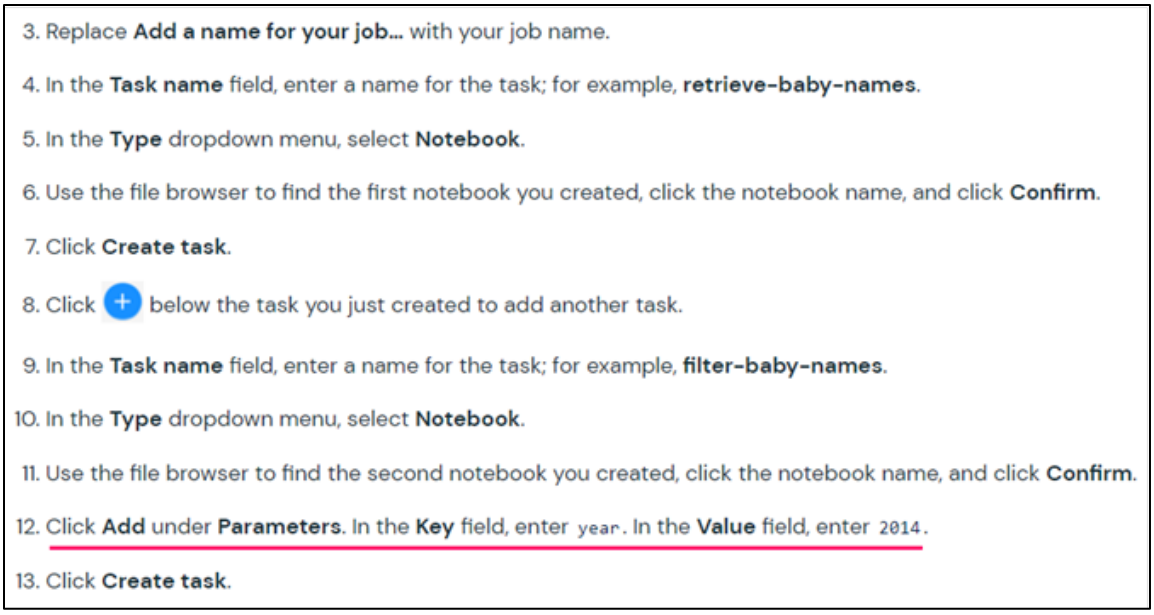
See Introducing Databricks Workflows, DATABRICKS, https://www.databricks.com/blog/2022/05/10/introducing-databricks-workflows.html (last visited December 15, 2023).



See Databricks Workflows, DATABRICKS, https://www.databricks.com/product/workflows (last visited December 15, 2023).

45. The Databricks Workflow provides a user interface (e.g., Databricks Workflow Job UI) with a dialog box capable of accepting and/or displaying data (e.g., details about creating jobs with various parameters).

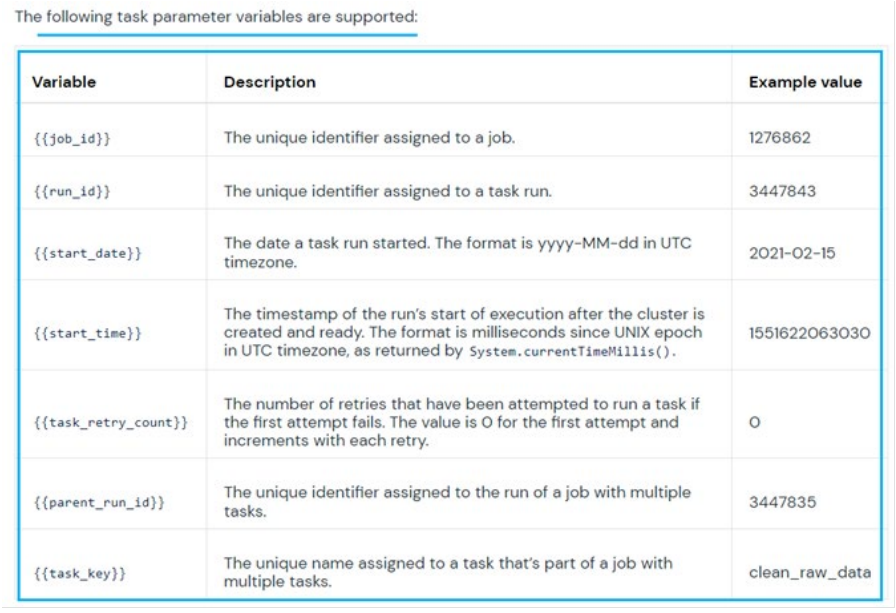




3. Replace **Add a name for your job...** with your job name.
4. In the **Task name** field, enter a name for the task; for example, **retrieve-baby-names**.
5. In the **Type** dropdown menu, select **Notebook**.
6. Use the file browser to find the first notebook you created, click the notebook name, and click **Confirm**.
7. Click **Create task**.
8. Click + below the task you just created to add another task.
9. In the **Task name** field, enter a name for the task; for example, **filter-baby-names**.
10. In the **Type** dropdown menu, select **Notebook**.
11. Use the file browser to find the second notebook you created, click the notebook name, and click **Confirm**.
12. Click **Add** under **Parameters**. In the **Key** field, enter `year`. In the **Value** field, enter `2014`.
13. Click **Create task**.

See Create your first workflow with a Databricks job, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/jobs-quickstart.html (last visited December 15, 2023).

46. These tasks, parameters, and/or data contained within the user interface and/or dialog box are representative of validation workflow management information.

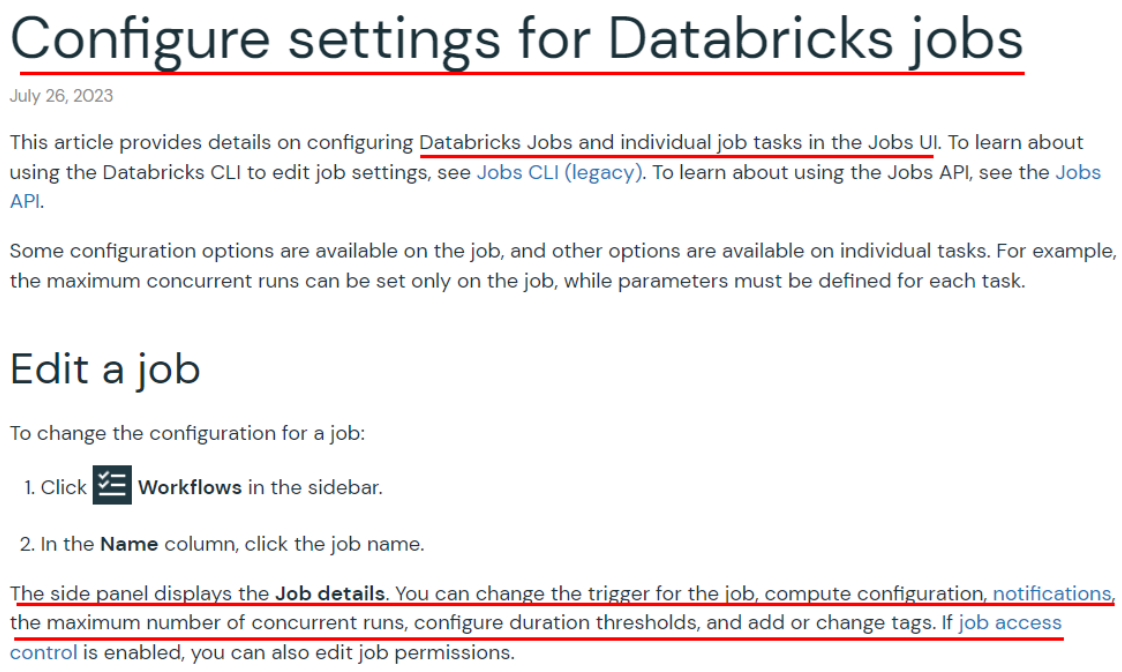


The following task parameter variables are supported:

Variable	Description	Example value
{{job_id}}	The unique identifier assigned to a job.	1276862
{{run_id}}	The unique identifier assigned to a task run.	3447843
{{start_date}}	The date a task run started. The format is yyyy-MM-dd in UTC timezone.	2021-02-15
{{start_time}}	The timestamp of the run's start of execution after the cluster is created and ready. The format is milliseconds since UNIX epoch in UTC timezone, as returned by `System.currentTimeMillis()`.	1551622063030
{{task_retry_count}}	The number of retries that have been attempted to run a task if the first attempt fails. The value is 0 for the first attempt and increments with each retry.	0
{{parent_run_id}}	The unique identifier assigned to the run of a job with multiple tasks.	3447835
{{task_key}}	The unique name assigned to a task that's part of a job with multiple tasks.	clean_raw_data

See Pass context about job runs into job tasks, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/task-parameter-variables.html (last visited December 15, 2023)

47. As shown below, the Databricks Lakehouse platform allows a user to automate the data flow validation process by using Databricks Workflow.



Configure settings for Databricks jobs

July 26, 2023

This article provides details on configuring Databricks Jobs and individual job tasks in the Jobs UI. To learn about using the Databricks CLI to edit job settings, see Jobs CLI (legacy). To learn about using the Jobs API, see the Jobs API.

Some configuration options are available on the job, and other options are available on individual tasks. For example, the maximum concurrent runs can be set only on the job, while parameters must be defined for each task.

Edit a job

To change the configuration for a job:

1. Click **Workflows** in the sidebar.
2. In the **Name** column, click the job name.

The side panel displays the **Job details**. You can change the trigger for the job, compute configuration, notifications, the maximum number of concurrent runs, configure duration thresholds, and add or change tags. If job access control is enabled, you can also edit job permissions.

See Configure settings for Databricks jobs, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/settings.html (last visited December 15, 2023).

48. The Databricks Workflow allows a user to automate jobs and validate configuration by using various parameters. The Databricks Workflow Job user interface provides the user with an option to enter values for configuration properties and/or notification parameters that are required for creating a validation workflow job. Databricks servers check whether the defined parameter information for the job, input during the execution of the job, is valid or invalid. The execution of the job is configured to run only if the properties and other parameters are valid.

Request structure

Field Name	Type	Description
run_id	INT64	The canonical identifier for the run. This field is required.
views_to_export	ViewsToExport	Which views to export (CODE, DASHBOARDS, or ALL). Defaults to CODE.

Response structure

Field Name	Type	Description
views	An array of ViewItem	The exported content in HTML format (one for every view item).

This endpoint validates that the `run_id` parameter is valid and for invalid parameters returns HTTP status code 400.

Example

Bash Copy

```
curl --netrc --request POST \
https://<databricks-instance>/api/2.0/jobs/runs/cancel \
--data '{ "run_id": <run-id> }'
```

See Jobs API 2.0, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/jobs-2.0-api.html (last visited December 15, 2023).

Run the job

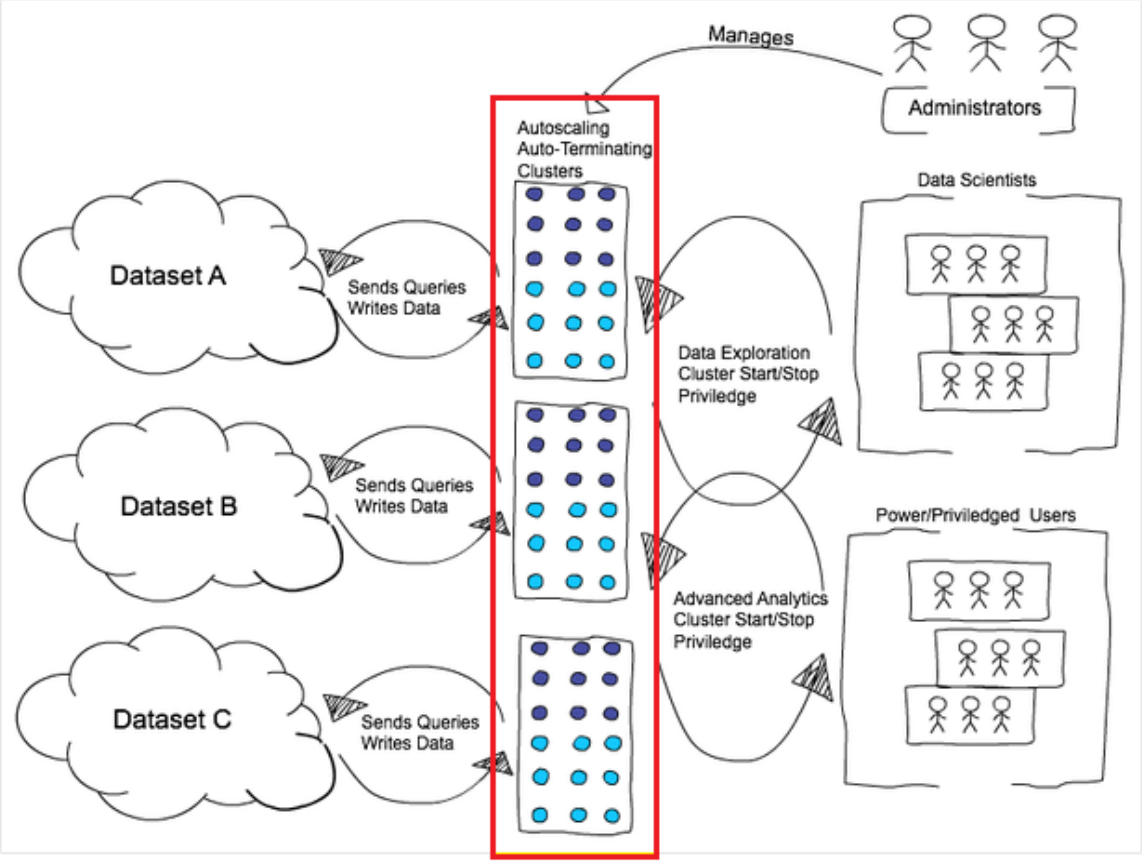
To run the job immediately, click Run Now in the upper right corner. You can also run the job by clicking the **Runs** tab and clicking **Run Now** in the **Active Runs** table.

View run details

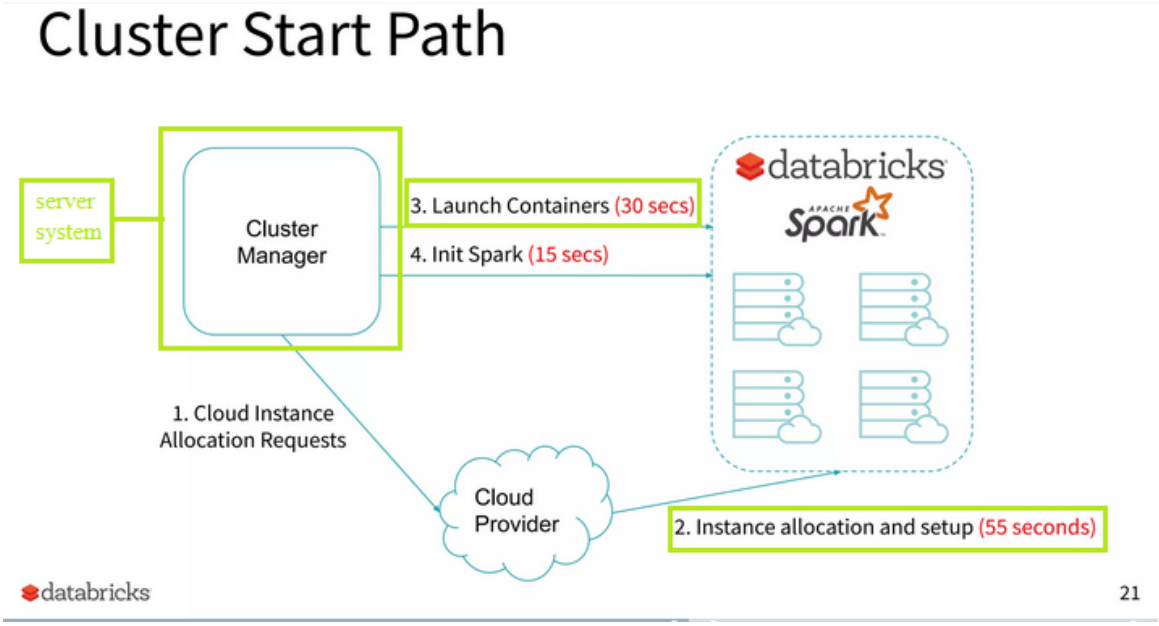
1. Click the **Runs** tab and click the link for the run in the **Active Runs** table or in the **Completed Runs (past 60 days)** table.
2. Click either task to see the output and details. For example, click the **filter-baby-names** task to view the output and run details for the filter task:

See Create your first workflow with a Databricks job, DATABRICKS, https://docs.databricks.com/en/workflows/jobs/jobs-quickstart.html (last visited December 15, 2023).

49. The Asserted Patents, including claim 1 of the '474 patent, cover Accused Instrumentalities of Defendant that practice a method comprising the steps of operating a distributed processing system having a network (e.g., virtual private network) coupling a multiplicity of Host distributed devices (e.g., multi-node clusters) for processing workloads (e.g., workloads) for the distributed processing system (e.g., distributed across multiple locations), a plurality of Client systems (e.g., user systems) requesting processing of the workloads, and a Server system (e.g., cluster manager) for selectively distributing the workloads from the plurality of Client systems for processing by the distributed processing system.



See Best practices: Cluster configuration, DATABRICKS, https://docs.databricks.com/en/clusters/cluster-config-best-practices.html (last visited December 15, 2023).

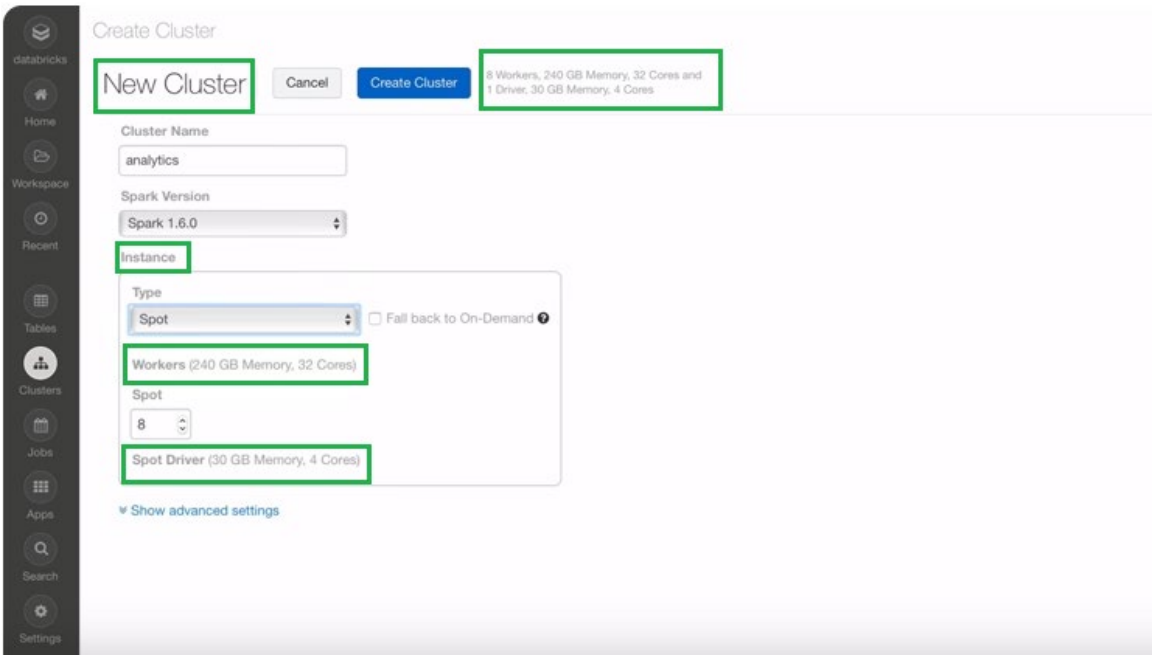


See Scaling Data Analytics Workloads on Databricks, DATABRICKS, https://www.slideshare.net/databricks/scaling-data-analytics-workloads-on-databricks (last visited December 15, 2023).

> The cluster manager controls physical machines and allocates resources to Spark Applications. This can be one of several core cluster managers: Spark's standalone cluster manager, YARN, or Mesos. This means that there can be multiple Spark Applications running on a cluster at the same time. We will talk more in depth about cluster managers in Part IV: Production Applications of this book. In the previous illustration we see on the left, our driver and on the right the four executors on the right. In this diagram, we removed the concept of cluster nodes. The user can specify how many executors should fall on each node through configurations. [glossary-cta]

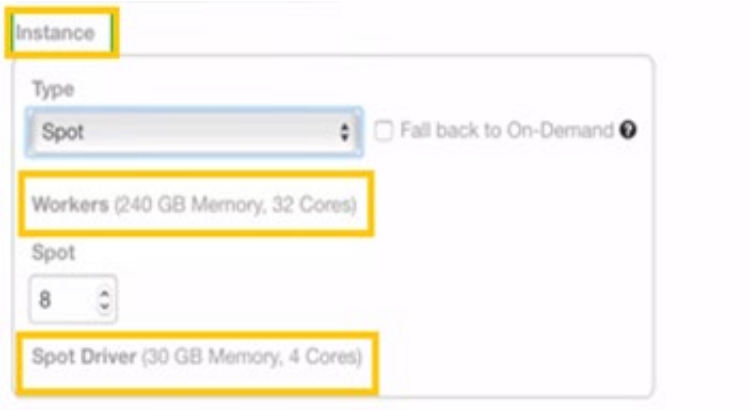
See Spark Applications, DATABRICKS, https://www.databricks.com/glossary/what-are-spark-applications (last visited December 15, 2023).

50. As shown below, a Databricks user can create a cluster for a workload through a server system (e.g. Cluster manager).



See Databricks Cluster Manager and Jobs, DATABRICKS, https://www.youtube.com/watch?v=R-lZxbgw81Y (last visited December 15, 2023).

51. After a user creates or manages a cluster to process a workload, a host distributed device (e.g. new or existing cluster performing the workload) is selected to run the workload in part by selecting or allocating worker nodes of the new cluster.



See id.

52. The Host distributed device (e.g. the new or existing cluster) is sent an index of data addresses (e.g. cloud object storage location) which defines a location for data that is required to process the workload.

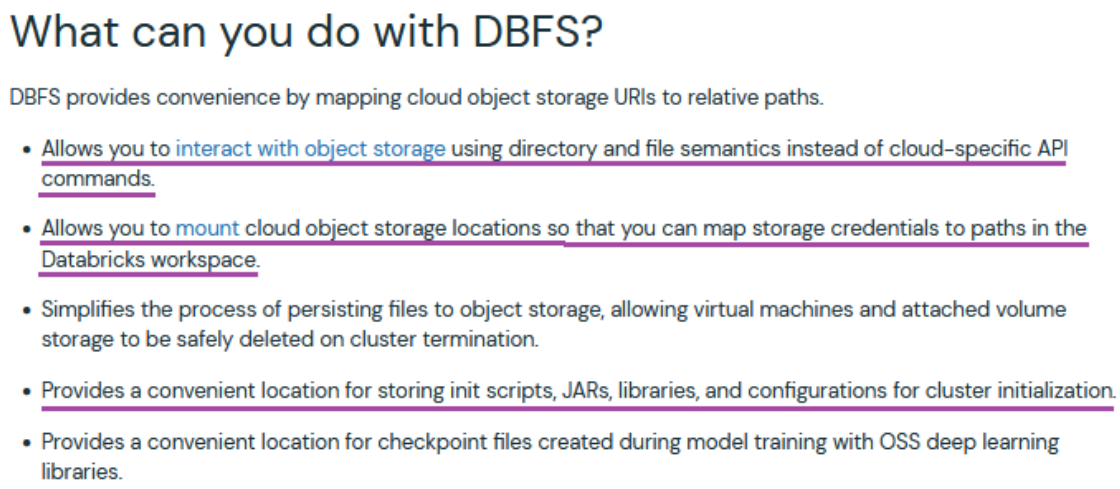
What is the Databricks File System (DBFS)?

October 10, 2023

The Databricks File System (DBFS) is a distributed file system mounted into a Databricks workspace and available on Databricks clusters. DBFS is an abstraction on top of scalable object storage that maps Unix-like filesystem calls to native cloud storage API calls.

Note

Databricks workspaces deploy with a DBFS root volume, accessible to all users by default. Databricks recommends against storing production data in this location.

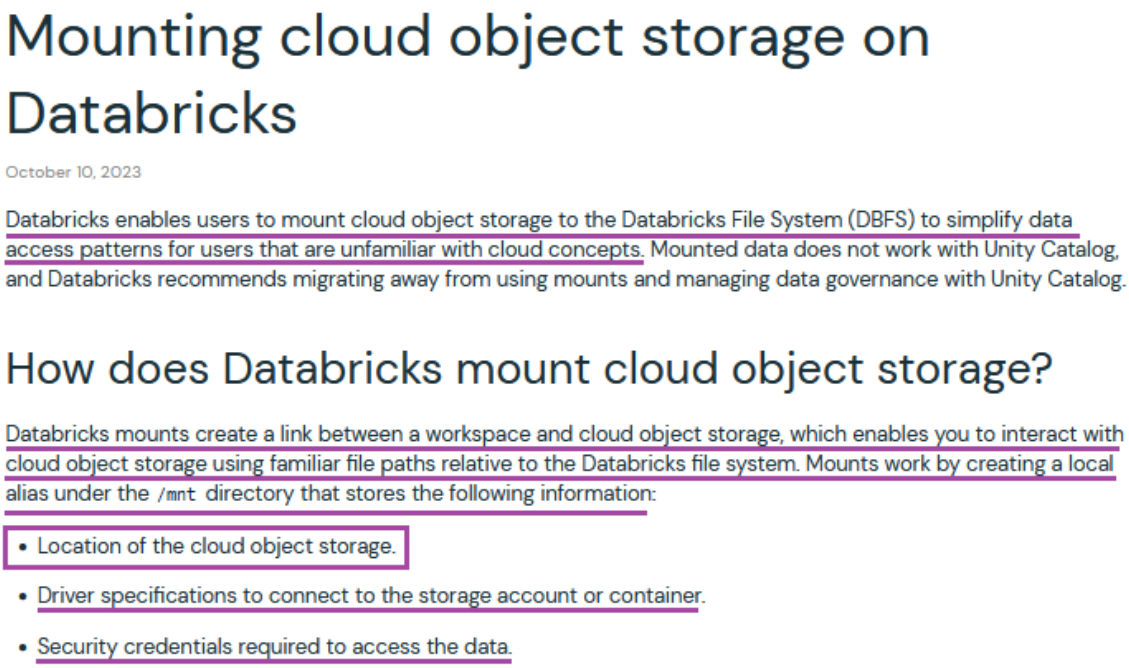


What can you do with DBFS?

DBFS provides convenience by mapping cloud object storage URIs to relative paths.

- Allows you to interact with object storage using directory and file semantics instead of cloud-specific API commands.
- Allows you to mount cloud object storage locations so that you can map storage credentials to paths in the Databricks workspace.
- Simplifies the process of persisting files to object storage, allowing virtual machines and attached volume storage to be safely deleted on cluster termination.
- Provides a convenient location for storing init scripts, JARs, libraries, and configurations for cluster initialization.
- Provides a convenient location for checkpoint files created during model training with OSS deep learning libraries.

See What is the Databricks File System (DBFS)?, DATABRICKS, https://docs.databricks.com/en/dbfs/index.html#interact-files (last visited December 15, 2023).



Mounting cloud object storage on Databricks

October 10, 2023

Databricks enables users to mount cloud object storage to the Databricks File System (DBFS) to simplify data access patterns for users that are unfamiliar with cloud concepts. Mounted data does not work with Unity Catalog, and Databricks recommends migrating away from using mounts and managing data governance with Unity Catalog.

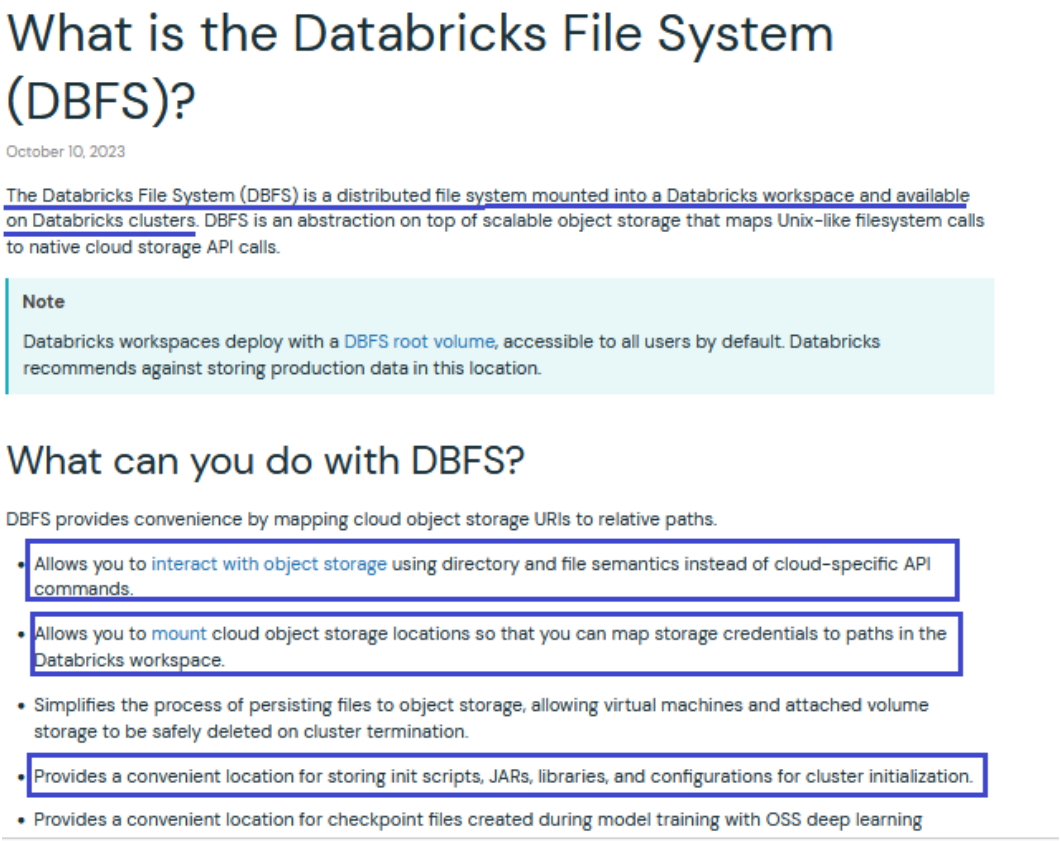
How does Databricks mount cloud object storage?

Databricks mounts create a link between a workspace and cloud object storage, which enables you to interact with cloud object storage using familiar file paths relative to the Databricks file system. Mounts work by creating a local alias under the `/mnt` directory that stores the following information:

- Location of the cloud object storage.
- Driver specifications to connect to the storage account or container.
- Security credentials required to access the data.

See Mounting cloud object storage on Databricks, DATABRICKS, https://docs.databricks.com/en/dbfs/mounts.html (last visited December 15, 2023).

53. The Databricks File System root storage is configured each time a new workspace is created to include the location of data to be accessed by a workspace for processing a cluster.



What is the Databricks File System (DBFS)?

October 10, 2023

The Databricks File System (DBFS) is a distributed file system mounted into a Databricks workspace and available on Databricks clusters. DBFS is an abstraction on top of scalable object storage that maps Unix-like filesystem calls to native cloud storage API calls.

Note

Databricks workspaces deploy with a DBFS root volume, accessible to all users by default. Databricks recommends against storing production data in this location.

What can you do with DBFS?

DBFS provides convenience by mapping cloud object storage URIs to relative paths.

- Allows you to interact with object storage using directory and file semantics instead of cloud-specific API commands.
- Allows you to mount cloud object storage locations so that you can map storage credentials to paths in the Databricks workspace.
- Simplifies the process of persisting files to object storage, allowing virtual machines and attached volume storage to be safely deleted on cluster termination.
- Provides a convenient location for storing init scripts, JARs, libraries, and configurations for cluster initialization.
- Provides a convenient location for checkpoint files created during model training with OSS deep learning

See What is the Databricks File System (DBFS)?, DATABRICKS, https://docs.databricks.com/en/dbfs/index.html#interact-files (last visited December 15, 2023).

54. The index is updated to include a storage address (e.g. data address of cached data in a node's local storage).

Mounting cloud object storage on Databricks

October 10, 2023

Databricks enables users to mount cloud object storage to the Databricks File System (DBFS) to simplify data access patterns for users that are unfamiliar with cloud concepts. Mounted data does not work with Unity Catalog, and Databricks recommends migrating away from using mounts and managing data governance with Unity Catalog.

How does Databricks mount cloud object storage?

Databricks mounts create a link between a workspace and cloud object storage, which enables you to interact with cloud object storage using familiar file paths relative to the Databricks file system. Mounts work by creating a local alias under the `/mnt` directory that stores the following information:

- Location of the cloud object storage.
- Driver specifications to connect to the storage account or container.
- Security credentials required to access the data.

See Mounting cloud object storage on Databricks, DATABRICKS, https://docs.databricks.com/en/dbfs/mounts.html (last visited December 15, 2023).

55. The storage address of storage is coupled to a Host distributed device (e.g. node in a cluster performing a workload) as a location of the first data.

Automatic and manual caching

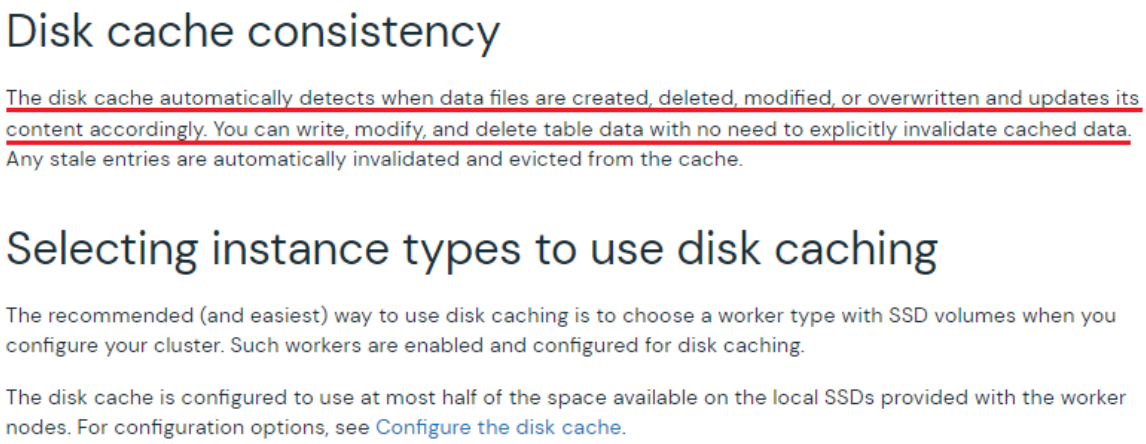
The Databricks disk cache differs from Apache Spark caching. Databricks recommends using automatic disk caching for most operations.

When the disk cache is enabled, data that has to be fetched from a remote source is automatically added to the cache. This process is fully transparent and does not require any action. However, to preload data into the cache beforehand, you can use the `CACHE SELECT` command (see Cache a subset of the data). When you use the Spark cache, you must manually specify the tables and queries to cache.

The disk cache contains local copies of remote data. It can improve the performance of a wide range of queries, but cannot be used to store results of arbitrary subqueries. The Spark cache can store the result of any subquery data and data stored in formats other than Parquet (such as CSV, JSON, and ORC).

The data stored in the disk cache can be read and operated on faster than the data in the Spark cache. This is because the disk cache uses efficient decompression algorithms and outputs data in the optimal format for further processing using whole-stage code generation.

Unlike the Spark cache, disk caching does not use system memory. Due to the high read speeds of modern SSDs, the disk cache can be fully disk-resident without a negative impact on its performance.



Disk cache consistency

The disk cache automatically detects when data files are created, deleted, modified, or overwritten and updates its content accordingly. You can write, modify, and delete table data with no need to explicitly invalidate cached data. Any stale entries are automatically invalidated and evicted from the cache.

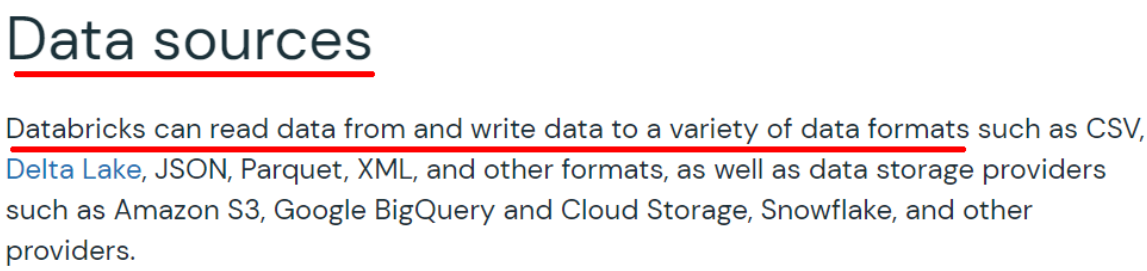
Selecting instance types to use disk caching

The recommended (and easiest) way to use disk caching is to choose a worker type with SSD volumes when you configure your cluster. Such workers are enabled and configured for disk caching.

The disk cache is configured to use at most half of the space available on the local SSDs provided with the worker nodes. For configuration options, see Configure the disk cache.

See Optimizing performance with caching on Databricks, DATABRICKS, https://docs.databricks.com/en/optimizations/disk-cache.html (last visited December 15, 2023).

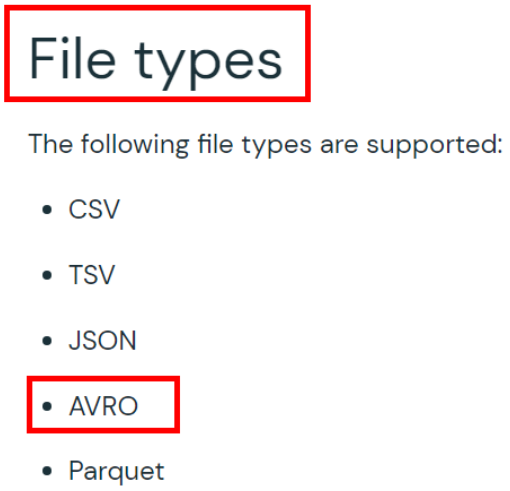
56. The Asserted Patents, including claim 1 of the '897 patent, cover Accused Instrumentalities of Defendant that practice improving compression (e.g., by utilizing a deflate algorithm) of data (e.g., SQL data), which comprises arranging (e.g. mapping) the data on a mixed format physical layout (e.g., Avro data format) having a plurality of fixed-sized fields (e.g., Int, BigInt, Double, etc.), a plurality of variable-sized fields (e.g., String, Date, etc.) and a plurality of offset slots (e.g., reference values to replace redundant data blocks), the fixed-sized fields being of a first size (e.g., 4 bytes, 8 bytes etc.) and the offset slots being of a second size (e.g., size of reference values).



Data sources

Databricks can read data from and write data to a variety of data formats such as CSV, Delta Lake, JSON, Parquet, XML, and other formats, as well as data storage providers such as Amazon S3, Google BigQuery and Cloud Storage, Snowflake, and other providers.

See Databricks integrations overview, DATABRICKS, https://docs.databricks.com/en/getting-started/connect/index.html (last visited December 15, 2023).



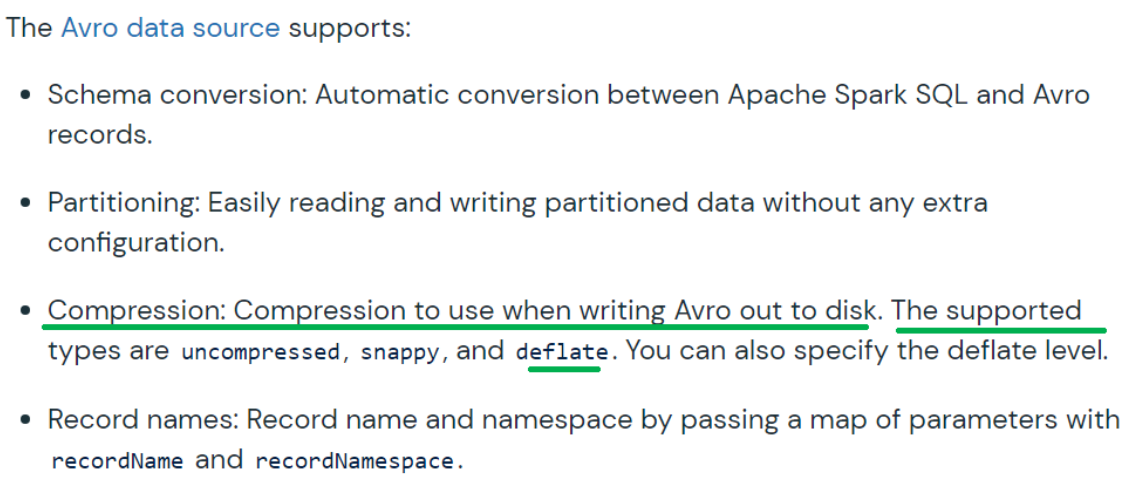
File types

The following file types are supported:

- CSV
- TSV
- JSON
- AVRO
- Parquet

See Load data using a Unity Catalog external location, DATABRICKS, https://docs.databricks.com/en/ingestion/add-data/add-data-external-locations.html (last visited December 15, 2023).

57. As shown below, Avro supports the compression type of "deflate."



The Avro data source supports:

- Schema conversion: Automatic conversion between Apache Spark SQL and Avro records.
- Partitioning: Easily reading and writing partitioned data without any extra configuration.
- Compression: Compression to use when writing Avro out to disk. The supported types are `uncompressed`, `snappy`, and `deflate`. You can also specify the deflate level.
- Record names: Record name and namespace by passing a map of parameters with `recordName` and `recordNamespace`.

See Avro file, DATABRICKS, https://docs.databricks.com/en/external-data/avro.html (last visited December 15, 2023).

58. Data is either a fixed size field or a variable sized field. This data is mapped from SQL to Avro.

The set of primitive type names is:

- *null*: no value
- *boolean*: a binary value
- *int*: 32-bit signed integer
- *long*: 64-bit signed integer
- *float*: single precision (32-bit) IEEE 754 floating-point number
- *double*: double precision (64-bit) IEEE 754 floating-point number
- *bytes*: sequence of 8-bit unsigned bytes
- *string*: unicode character sequence

Fixed-Sized Fields

Variable-Sized Fields

See Apache Avro Specification, APACHE, https://avro.apache.org/docs/1.11.1/specification/ (last visited December 15, 2023).

Supported types for Spark SQL -> Avro conversion

This library supports writing of all Spark SQL types into Avro. For most types, the mapping from Spark types to Avro types is straightforward (for example `IntegerType` gets converted to `int`); the following is a list of the few special cases:

Spark SQL type	Avro type	Avro logical type
ByteType	int	
ShortType	int	
BinaryType	bytes	
DecimalType	fixed	decimal
TimestampType	long	timestamp-micros
DateType	int	date

See Avro file, DATABRICKS, https://docs.databricks.com/en/external-data/avro.html (last visited December 15, 2023).

59. The deflate compression type utilized by Avro in the Databricks Lakehouse Platform, utilizes offset slots to compress the data.

LZ77 compression

LZ77 compression works by finding sequences of data that are repeated. The term ``sliding window'' is used; all it really means is that at any given point in the data, there is a record of what characters went before. A 32K sliding window means that the compressor (and decompressor) have a record of what the last 32768 (32 * 1024) characters were. When the next sequence of characters to be compressed is identical to one that can be found within the sliding window, the sequence of characters is replaced by two numbers: a distance, representing how far back into the window the sequence starts, and a length, representing the number of characters for which the sequence is identical.

I realize this is a lot easier to see than to just be told. Let's look at some highly compressible data:

```
Blah blah blah blah blah!
```

Our datastream starts by receiving the following characters: `B,' `l,' `a,' `h,' `␣,' and `b.' However, look at the next five characters:

```
  vvvvv
Blah blah blah blah blah!
      ^^^^^
```

There is an exact match for those five characters in the characters that have already gone into the datastream, and it starts exactly five characters behind the point where we are now. This being the case, we can output special characters to the stream that represent a number for length, and a number for distance.

The data so far:

```
Blah blah b
```

The compressed form of the data so far:

```
Blah b[D=5,L=5]
```

second size

offset

See An Explanation of the Deflate Algorithm, Antaeus Feldspar, https://www.zlib.net/feldspar.html (August 23, 1997).

60. Data is written to Avro files in a defined data structure which separates data into either a fixed or variable sized field.

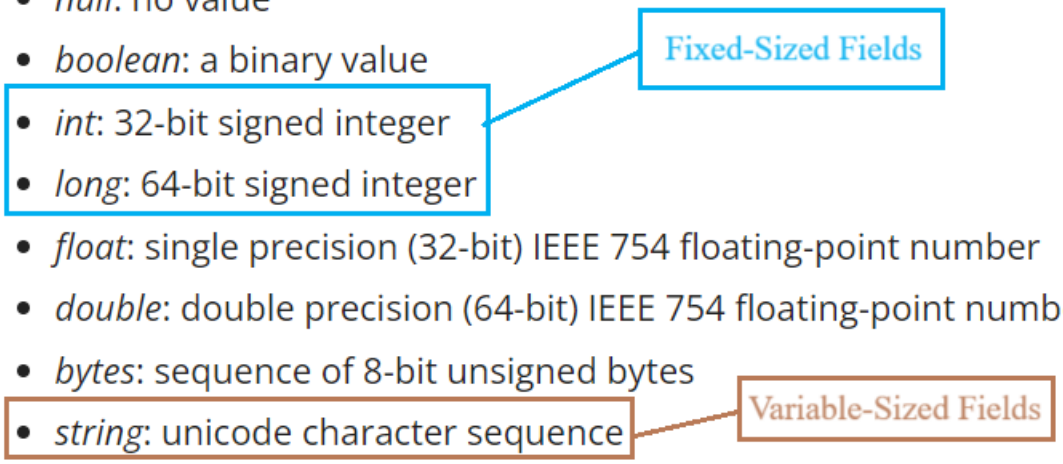
Supported types for Spark SQL -> Avro conversion

This library supports writing of all Spark SQL types into Avro. For most types, the mapping from Spark types to Avro types is straightforward (for example `IntegerType` gets converted to `int`); the following is a list of the few special cases:

See Avro file, DATABRICKS, https://docs.databricks.com/en/external-data/avro.html (last visited December 15, 2023).

The set of primitive type names is:

- *null*: no value
- *boolean*: a binary value
- *int*: 32-bit signed integer
- *long*: 64-bit signed integer
- *float*: single precision (32-bit) IEEE 754 floating-point number
- *double*: double precision (64-bit) IEEE 754 floating-point number
- *bytes*: sequence of 8-bit unsigned bytes
- *string*: unicode character sequence



See Apache Avro Specification, APACHE, https://avro.apache.org/docs/1.11.1/specification/ (last visited December 15, 2023).

61. Avro data can be compressed by the deflate compression type.

The Avro data source supports:

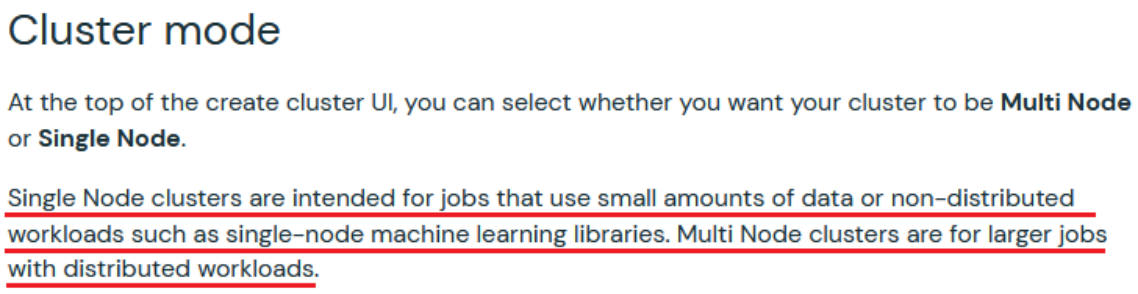
- Schema conversion: Automatic conversion between Apache Spark SQL and Avro records.
- Partitioning: Easily reading and writing partitioned data without any extra configuration.
- Compression: Compression to use when writing Avro out to disk. The supported types are `uncompressed`, `snappy`, and `deflate`. You can also specify the deflate level.
- Record names: Record name and namespace by passing a map of parameters with `recordName` and `recordNamespace`.

To configure compression when writing, set the following Spark properties:

- Compression codec: `spark.sql.avro.compression.codec`. Supported codecs are `snappy` and `deflate`. The default codec is `snappy`.
- If the compression codec is `deflate`, you can set the compression level with: `spark.sql.avro.deflate.level`. The default level is `-1`.

See Avro file, DATABRICKS, https://docs.databricks.com/en/external-data/avro.html (last visited December 15, 2023).

62. The Asserted Patents, including claims 2 and 14 of the '827 patent, cover Accused Instrumentalities of Defendant that practice configuring a distributed processing system of a plurality of distributed devices (e.g., cluster nodes) coupled to a network (e.g., virtual private network), wherein the plurality of distributed devices include respective client agents (e.g., executors) configured to process respective portions of a workload for the distributed processing system (e.g., distributed across multiple locations).

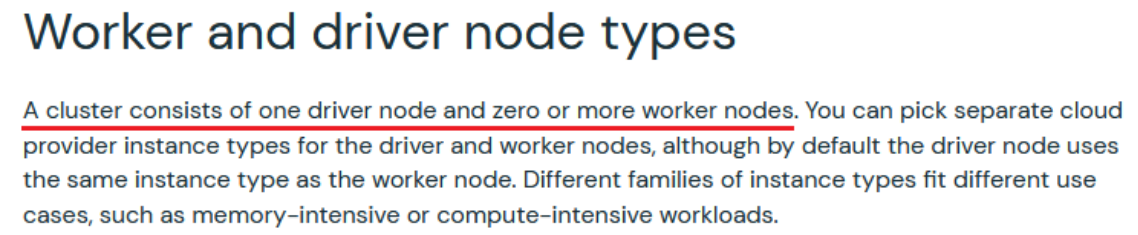


Cluster mode

At the top of the create cluster UI, you can select whether you want your cluster to be **Multi Node** or **Single Node**.

Single Node clusters are intended for jobs that use small amounts of data or non-distributed workloads such as single-node machine learning libraries. Multi Node clusters are for larger jobs with distributed workloads.

See Best practices: Cluster configuration, DATABRICKS, https://docs.databricks.com/en/clusters/cluster-config-best-practices.html (last visited December 18, 2023).

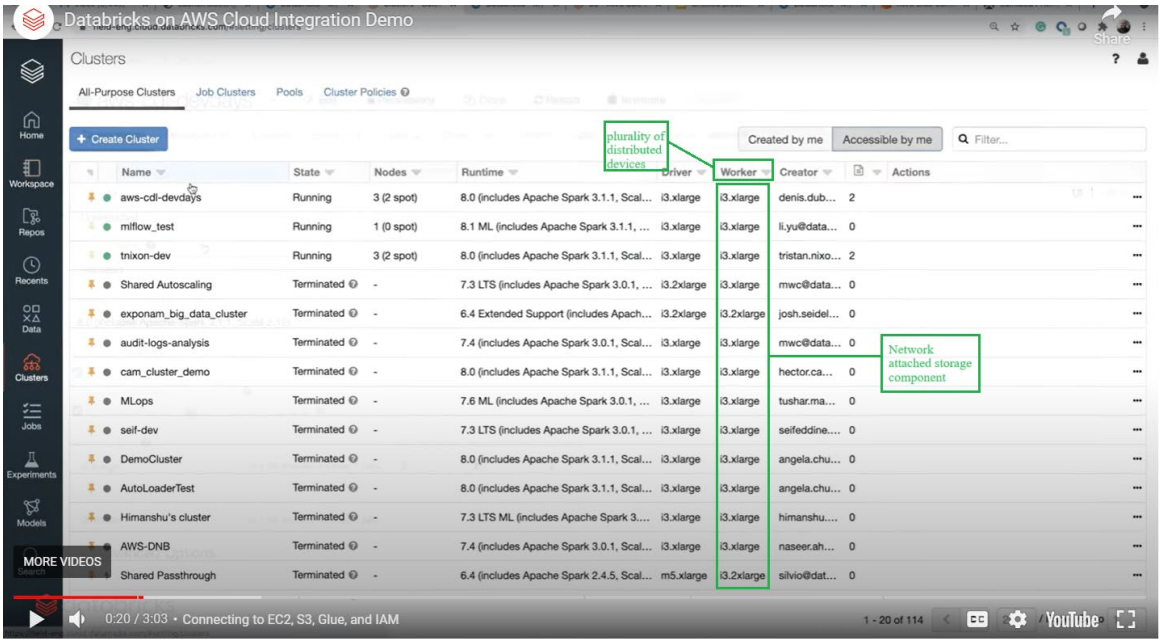


Worker and driver node types

A cluster consists of one driver node and zero or more 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.

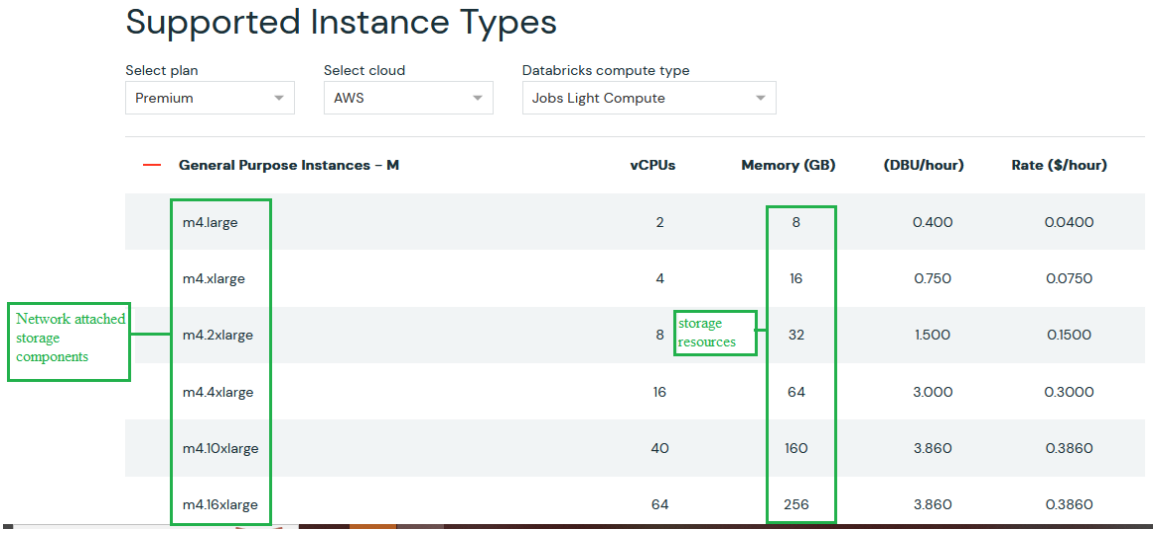
See Create a cluster, DATABRICKS, https://docs.databricks.com/en/clusters/configure.html (last visited December 18, 2023).

63. Particular distributed devices (e.g. a cluster and included worker nodes) within the plurality of distributed devices (e.g. clusters) have corresponding software-based network attached storage (NAS) components (e.g. instances).



See Databricks on AWS, DATABRICKS, https://www.databricks.com/resources/demos/videos/partner/databricks-on-aws (last visited December 18, 2023).

64. Instances define compute capacity such as CPU cores and storage resources available to the worker nodes for processing a workload.



See Pricing calculator, DATABRICKS, https://www.databricks.com/product/pricing/product-pricing/instance-types (last visited December 18, 2023).

65. When an instance is launched, dedicated host computer and storage space is allocated to the cluster, including the nodes within that cluster, to perform workload processing. For example, Databricks worker nodes "run Spark executors and other services for proper functioning clusters." Then "all the distributed processing happens on worker noders." The cluster or node (e.g., clusters A, B, C, and D shown below) therefore has an available amount of dedicated storage resources related to unused and under-utilized storage resources.

Cluster mode

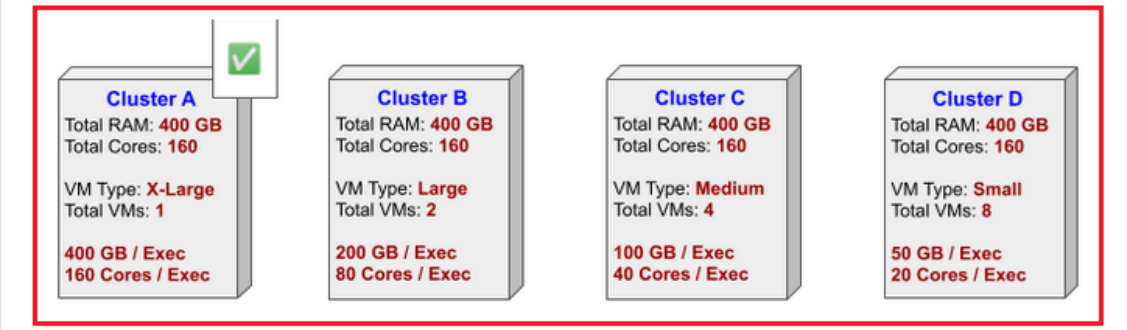
At the top of the create cluster UI, you can select whether you want your cluster to be **Multi Node** or **Single Node**.

See Best practices: Cluster configuration, DATABRICKS, https://docs.databricks.com/en/clusters/cluster-config-best-practices.html (last visited December 18, 2023).

Worker type

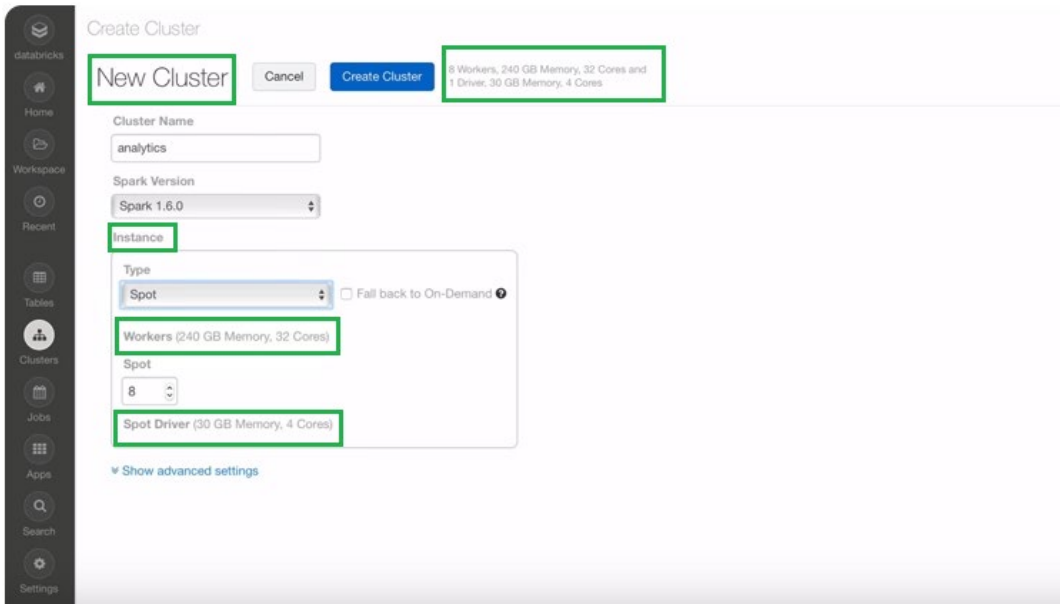
Databricks worker nodes run the Spark executors and other services required for proper functioning clusters. When you distribute your workload with Spark, all the distributed processing happens on worker nodes. Databricks runs one executor per worker node. Therefore, the terms executor and worker are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

https://docs.databricks.com/en/clusters/configure.html

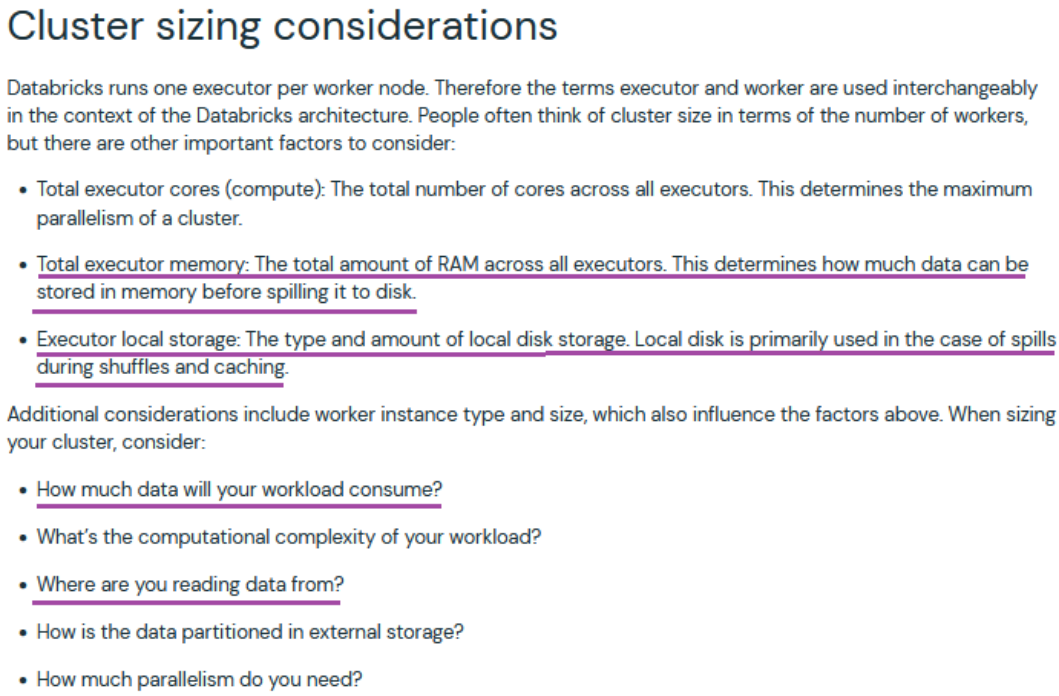


https://docs.databricks.com/en/clusters/cluster-config-best-practices.html#cluster-sizing-considerations

66. An instance allocated to a cluster or node (e.g., Databricks workers) has storage resources coupled to it. This storage is used by the node to read data from or write data to (e.g., intermediate results) during workload processing.



https://www.youtube.com/watch?v=R-lZxbgw81Y



Cluster sizing considerations

Databricks runs one executor per worker node. Therefore the terms executor and worker are used interchangeably in the context of the Databricks architecture. People often think of cluster size in terms of the number of workers, but there are other important factors to consider:

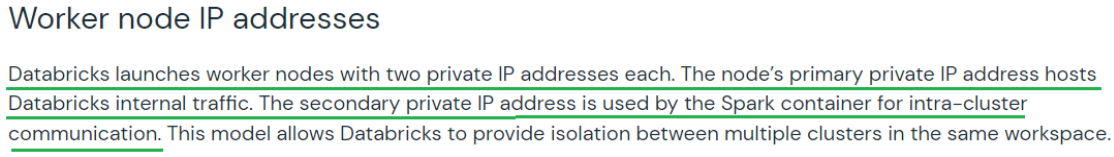
- Total executor cores (compute): The total number of cores across all executors. This determines the maximum parallelism of a cluster.
- Total executor memory: The total amount of RAM across all executors. This determines how much data can be stored in memory before spilling it to disk.
- Executor local storage: The type and amount of local disk storage. Local disk is primarily used in the case of spills during shuffles and caching.

Additional considerations include worker instance type and size, which also influence the factors above. When sizing your cluster, consider:

- How much data will your workload consume?
- What's the computational complexity of your workload?
- Where are you reading data from?
- How is the data partitioned in external storage?
- How much parallelism do you need?

See Best practices: Cluster configuration, DATABRICKS, https://docs.databricks.com/en/clusters/cluster-config-best-practices.html (last visited December 18, 2023).

67. Distributed devices (e.g., worker nodes performing a workload) function as a location distributed device to store location information (e.g., storing IP addresses of the node) associated with data stored by the selected distributed devices.



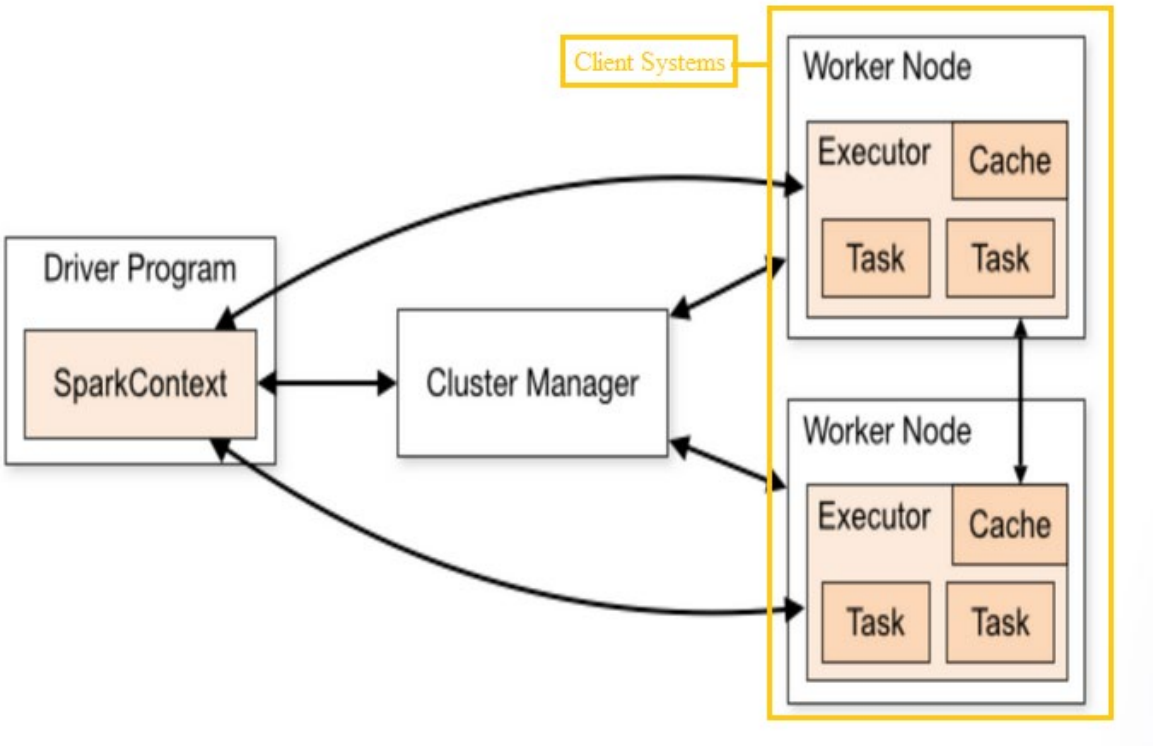
Worker node IP addresses

Databricks launches worker nodes with two private IP addresses each. The node's primary private IP address hosts Databricks internal traffic. The secondary private IP address is used by the Spark container for intra-cluster communication. This model allows Databricks to provide isolation between multiple clusters in the same workspace.

See Create a cluster, DATABRICKS, https://docs.databricks.com/en/clusters/configure.html (last visited December 18, 2023).

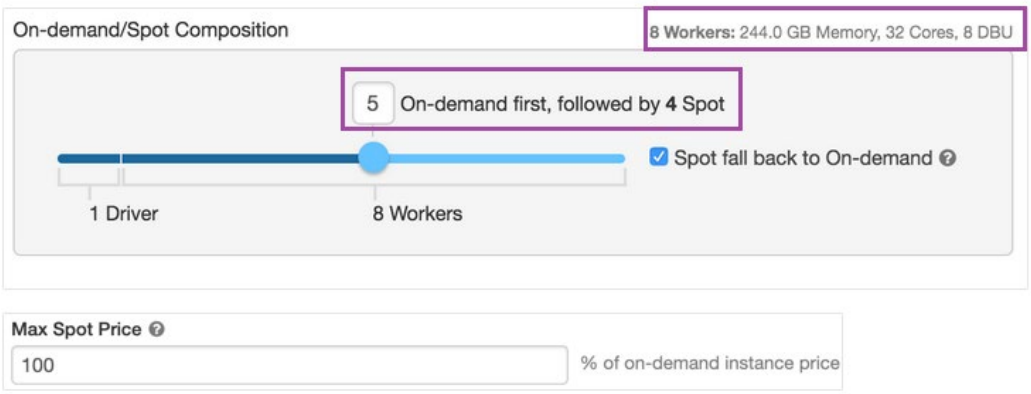
68. The Asserted Patents, including claim 1 of the '153 patent, cover Accused Instrumentalities of Defendant that practice a method of providing dynamic coordination of distributed client systems (e.g., Databricks worker nodes of a cluster) in a distributed computing

platform (e.g., in private or public subnets). The method provides a server system coupled to a network.



See Databricks driver sizing impact on cost and performance, JEFFREY CHOU, https://synccomputing.com/databricks-driver-sizing-impact-on-cost-and-performance/ (last visited December 18, 2023).

69. The client systems have under-utilized capabilities (e.g. on demand or spot instances) and runs a client agent program (e.g. worker nodes running a Spark executor) to process a workload for a job.



See Best practices: Cluster configuration, DATABRICKS, https://docs.databricks.com/en/clusters/cluster-config-best-practices.html (last visited December 18, 2023).

Worker node

Databricks worker nodes 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 worker nodes. Databricks runs one executor per worker node; therefore the terms *executor* and *worker* are used interchangeably in the context of the Databricks architecture.

See Configure Clusters, DATABRICKS, https://docs.databricks.com/en/archive/compute/configure.html (last visited December 18, 2023).

What is Spark Executor

rimmalapudi - Apache Spark - March 8, 2023

Spark Executor is a process that runs on a worker node in a Spark cluster and is responsible for executing tasks assigned to it by the Spark driver program. In this article, we shall discuss what is Spark Executor, the types of executors, configurations, uses, and the performance of executors.

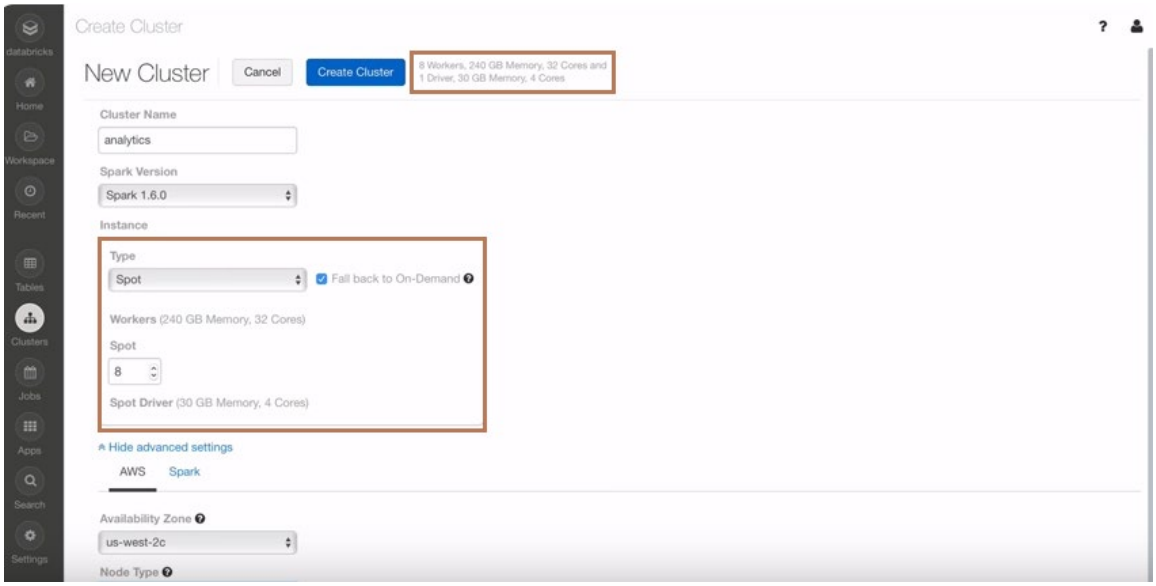
See What is Spark Executor, SPARK BY {EXAMPLES}, https://sparkbyexamples.com/spark/what-is-spark-executor/ (last visited December 18, 2023).

70. The initial project and poll parameters are distributed to the client systems.

Optimized autoscaling has the following characteristics:

- 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 job clusters, scales down if the cluster is underutilized over the last 40 seconds.
- On all-purpose clusters, scales down if the cluster is underutilized over the last 150 seconds.
- The `spark.databricks.aggressiveWindowDownS` Spark configuration property specifies in seconds how often a cluster makes down-scaling decisions. Increasing the value causes a cluster to scale down more slowly. The maximum value is 600.

See Monitor Delta Live Tables pipelines, DATABRICKS, https://docs.databricks.com/en/delta-live-tables/observability.html#autoscaling (last visited December 18, 2023).



See Databricks Cluster Manager and Jobs, DATABRICKS, https://www.youtube.com/watch?v=R-lZxbgw81Y (last visited December 15, 2023).

71. Worker nodes are monitored to identify under-utilized nodes.

> Databricks' optimized autoscaling solves this problem by periodically reporting detailed statistics on idle executors and the location of intermediate files within the cluster. The Databricks service uses this information to more precisely target workers to scale down when utilization is low. In particular the service can scale down and remove idle workers on an under-utilized cluster even when there are tasks running on other executors for the same Spark job. This behavior is different from traditional autoscaling, which requires the entire Spark job to be finished to begin scale-down. During scale-down, the Databricks service removes a worker only if it is idle and does not contain any shuffle data that is being used by running queries. Therefore running jobs and queries are not affected during down-scaling.

See Introducing Databricks Optimized Autoscaling on Apache Spark, DATABRICKS, https://www.databricks.com/blog/2018/05/02/introducing-databricks-optimized-auto-scaling.html (last visited December 18, 2023).

72. Databricks, based on the dynamic snapshot information, determines whether to add to or reduce the number of actively participating clients (e.g. nodes).

What is Enhanced Autoscaling?

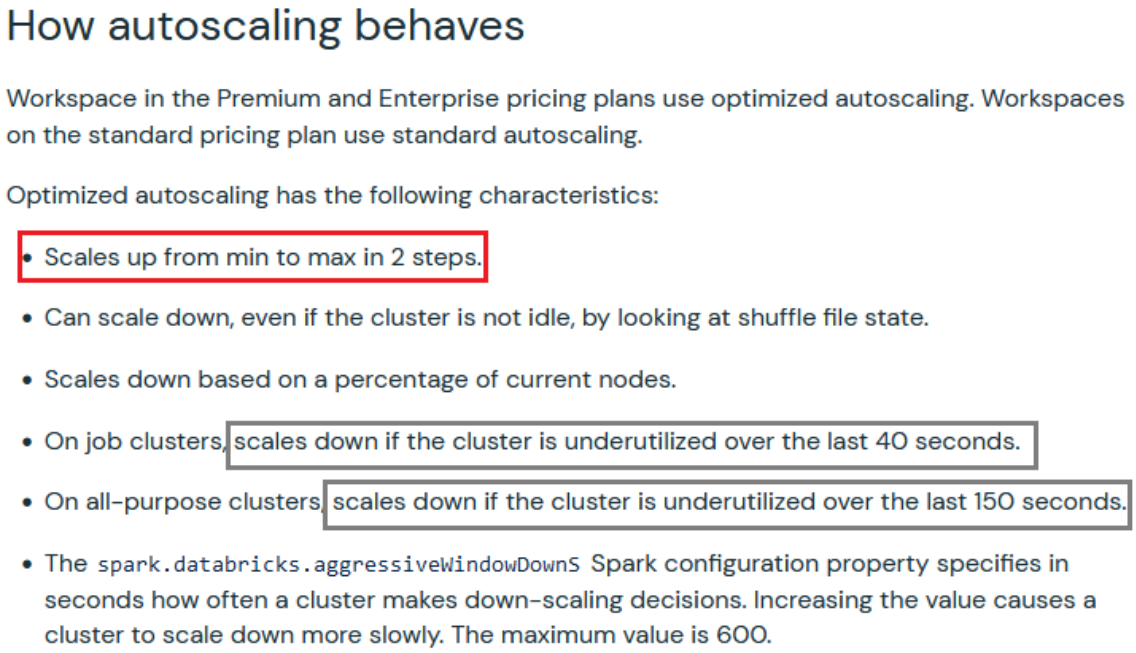
October 10, 2023

Databricks Enhanced Autoscaling optimizes cluster utilization by automatically allocating cluster resources based on workload volume, with minimal impact to the data processing latency of your pipelines.

Enhanced Autoscaling improves on the Databricks cluster autoscaling functionality with the following features:

- Enhanced Autoscaling implements optimization of streaming workloads, and adds enhancements to improve the performance of batch workloads. Enhanced Autoscaling optimizes costs by adding or removing machines as the workload changes.
- Enhanced Autoscaling proactively shuts down under-utilized nodes while guaranteeing there are no failed tasks during shutdown. The existing cluster autoscaling feature scales down nodes only if the node is idle.

See What is Enhanced Autoscaling?, DATABRICKS, https://docs.databricks.com/en/delta-live-tables/auto-scaling.html (last visited December 18, 2023).



How autoscaling behaves

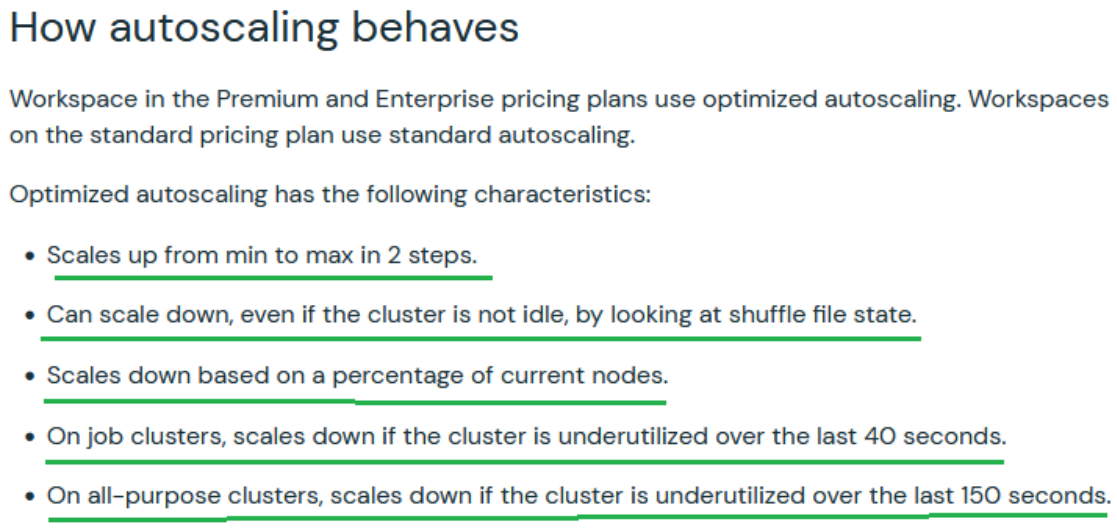
Workspace in the Premium and Enterprise pricing plans use optimized autoscaling. Workspaces on the standard pricing plan use standard autoscaling.

Optimized autoscaling has the following characteristics:

- 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 job clusters, scales down if the cluster is underutilized over the last 40 seconds.
- On all-purpose clusters, scales down if the cluster is underutilized over the last 150 seconds.
- The `spark.databricks.aggressiveWindowDownS` Spark configuration property specifies in seconds how often a cluster makes down-scaling decisions. Increasing the value causes a cluster to scale down more slowly. The maximum value is 600.

See Create a cluster, DATABRICKS, https://docs.databricks.com/en/clusters/configure.html#enable-autoscaling (last visited December 18, 2023).

73. The initial parameters of the project (e.g., number of worker nodes, spark.databricks.aggressiveWindowDownS parameter, automatic termination time, etc.) are modified depending on the decisions reached in the analyzing step (e.g. scaling up nodes, scaling down nodes, terminating nodes, etc.) and is/can be repeated.



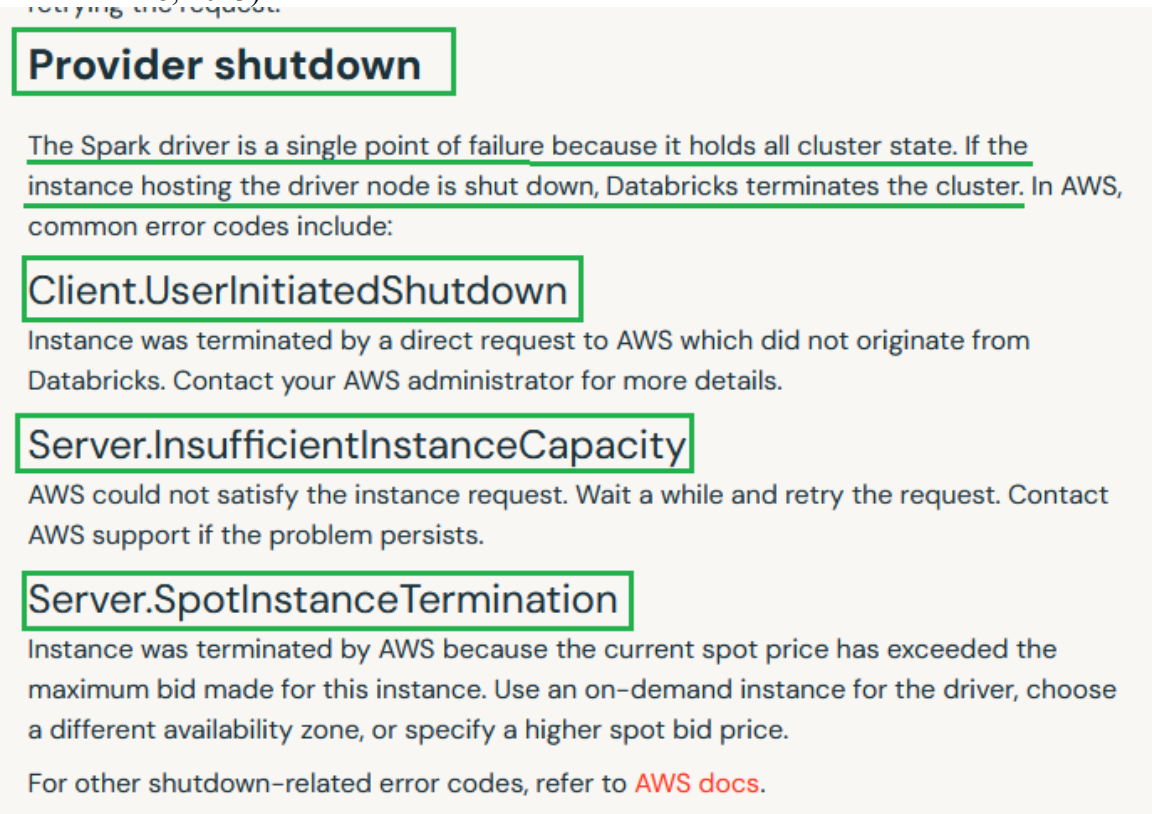
How autoscaling behaves

Workspace in the Premium and Enterprise pricing plans use optimized autoscaling. Workspaces on the standard pricing plan use standard autoscaling.

Optimized autoscaling has the following characteristics:

- 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 job clusters, scales down if the cluster is underutilized over the last 40 seconds.
- On all-purpose clusters, scales down if the cluster is underutilized over the last 150 seconds.

See Create a cluster, DATABRICKS, https://docs.databricks.com/en/clusters/configure.html#enable-autoscaling (last visited December 18, 2023).



Provider shutdown

The Spark driver is a single point of failure because it holds all cluster state. If the instance hosting the driver node is shut down, Databricks terminates the cluster. In AWS, common error codes include:

Client.UserInitiatedShutdown

Instance was terminated by a direct request to AWS which did not originate from Databricks. Contact your AWS administrator for more details.

Server.InsufficientInstanceCapacity

AWS could not satisfy the instance request. Wait a while and retry the request. Contact AWS support if the problem persists.

Server.SpotInstanceTermination

Instance was terminated by AWS because the current spot price has exceeded the maximum bid made for this instance. Use an on-demand instance for the driver, choose a different availability zone, or specify a higher spot bid price.

For other shutdown-related error codes, refer to AWS docs.

See Unexpected cluster termination, DATABRICKS, https://kb.databricks.com/clusters/termination-reasons.html (last visited December 18, 2023).

COUNT I

(INFRINGEMENT OF U.S. PATENT NO. 6,839,733)

74. Plaintiff incorporates paragraphs 1 through 74 herein by reference.

75. Plaintiff BYTEWEAVR is the assignee of the '733 patent, entitled "Network system extensible by users," with ownership of all substantial rights in the '733 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

76. The '733 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The '733 patent issued from U.S. Patent Application No.

09/712,712. The '733 patent was granted on January 1, 2004 and expired on or about October 23, 2018.

77. Defendant has directly infringed one or more claims of the '733 patent in this District and elsewhere in Texas and the United States.

78. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

79. Defendant has directly infringed the '733 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Products, their components, and/or products containing the same that incorporate the fundamental technologies covered by the '733 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief, Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Products outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the United States, thereby directly infringing the '733 patent. *See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc.*, 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad

but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

80. Furthermore, Defendant Databricks has directly infringed the '733 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments, companies, and/or brands of Defendant Databricks, including by designing the Accused Products for U.S. consumers and selling and offering for sale the Accused Products directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '733 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

81. For example, Defendant infringes claim 37 of the '733 patent via the Accused Instrumentalities, namely data management and analytics products and components, software, services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, and Databricks Spark Applications.

82. Those Accused Instrumentalities include a "method" comprising the limitations of claim 37. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of admitting a user to a network system wherein at least one agent is operable to consume a service resource while utilizing a service to perform a task for the user; and allowing the user to create, modify, or delete the agent within the network system.

83. At a minimum, Defendant has known of the '733 patent at least as early as the filing date of this Complaint.

84. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

COUNT II

(INFRINGEMENT OF U.S. PATENT NO. 7,949,752)

85. Plaintiff incorporates paragraphs 1 through 85 herein by reference.

86. Plaintiff BYTEWEAVR is the assignee of the '752 patent, entitled "Network system extensible by users," with ownership of all substantial rights in the '752 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

87. The '752 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The '752 patent issued from U.S. Patent Application No. 10/995,159. The '752 patent was granted on May 24, 2011 and expired on or about Aug. 13, 2022.

88. Defendant has directly infringed one or more claims of the '752 patent in this District and elsewhere in Texas and the United States.

89. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

90. Defendant has directly infringed the '752 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Products, their components, and/or products containing the same that incorporate the fundamental technologies covered by the '752 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief, Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Products outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the United States, thereby directly infringing the '752 patent. *See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc.*, 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

91. Furthermore, Defendant Databricks has directly infringed the '752 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments,

companies, and/or brands of Defendant Databricks, including by designing the Accused Products for U.S. consumers and selling and offering for sale the Accused Products directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '752 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

92. For example, Defendant infringes claim 24 of the '752 patent via the Accused Instrumentalities, namely data management and analytics products and components, software, services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, and Databricks Spark Applications.

93. Those Accused Instrumentalities include a "method" comprising the limitations of claim 24. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of receiving, using a computing device, data for creating a network-based agent; invoking, using the computing device, and in response to receiving a URL

defining a type of event and identifying the network-based agent, execution of the network-based agent, wherein the invoking comprises using a service and a service resource configured to be consumed by the network-based agent for performing the operation, and wherein a discrete unit of the service resource is exhausted upon being consumed by the network-based agent; and communicating, using the computing device, a result of the operation over a network communication link.

94. At a minimum, Defendant has known of the '752 patent at least as early as the filing date of this Complaint.

95. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

COUNT III

(INFRINGEMENT OF U.S. PATENT NO. 6,862,488)

96. Plaintiff incorporates paragraphs 1 through 96 herein by reference.

97. Plaintiff BYTEWEAVR is the assignee of the '488 patent, entitled "Automated validation processing and workflow management," with ownership of all substantial rights in the '488 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

98. The '488 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The '488 patent issued from U.S. Patent Application No. 10/190,368. The '488 patent was granted on March 1, 2005 and expired on or about April 9, 2023.

99. Defendant has directly infringed one or more claims of the '488 patent in this District and elsewhere in Texas and the United States.

100. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

101. Defendant has directly infringed the '488 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Products, their components, and/or products containing the same that incorporate the fundamental technologies covered by the '488 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief, Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Products outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the United States, thereby directly infringing the '488 patent. *See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc.*, 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

102. Furthermore, Defendant Databricks has directly infringed the '488 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments, companies, and/or brands of Defendant Databricks, including by designing the Accused Products for U.S. consumers and selling and offering for sale the Accused Products directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '488 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

103. For example, Defendant infringes claim 11 of the '488 patent via the Accused Instrumentalities, namely data management and analytics products and components, software, services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, and Databricks Spark Applications.

104. Those Accused Instrumentalities include, "[i]n a computing environment[,] a method to automate the validation of equipment and/or processes for use in a pharmaceutical and/or

bio-technology manufacturing facility" comprising the limitations of claim 11. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of providing a user interface capable of accepting and/or displaying data representative of validation processing and/or validation workflow management information, wherein said user interface has at least one dialog box populated with validation processing and/or validation workflow management information; providing a validation processing engine, said validation processing engine comprising at least one processing rule that operates on validation processing information selected through said user interface to produce validation protocol information..

105. At a minimum, Defendant has known of the '488 patent at least as early as the filing date of this Complaint.

106. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

COUNT IV

(INFRINGEMENT OF U.S. PATENT NO. 6,965,897)

107. Plaintiff incorporates paragraphs 1 through 107 herein by reference.

108. Plaintiff BYTEWEAVR is the assignee of the '897 patent, entitled "Data Compression Method and Apparatus," with ownership of all substantial rights in the '897 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

109. The ’897 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The ’897 patent issued from U.S. Patent Application No. 10/065,513. The ’897 patent was granted on November 15, 2005 and expired on or about August 10, 2023.

110. Defendant has directly infringed one or more claims of the ’897 patent in this District and elsewhere in Texas and the United States.

111. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

112. Defendant has directly infringed the ’897 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Instrumentalities, their components, and/or products containing the same that incorporate the fundamental technologies covered by the ’897 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief, Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Instrumentalities outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the United States, thereby directly infringing the ’897 patent. See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc., 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying

summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

113. Furthermore, Defendant Databricks has directly infringed the '897 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments, companies, and/or brands of Defendant Databricks, including by designing the Accused Instrumentalities for U.S. consumers and selling and offering for sale the Accused Instrumentalities directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '897 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

114. For example, Defendant infringes claim 1 of the '897 patent via the Accused Instrumentalities, namely data management and analytics products and components, software, services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data

Intelligence Platform, Databricks Spark Applications and related data storage and compression techniques.

115. Those Accused Instrumentalities include a "method for improving compression of data" comprising the limitations of claim 1. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of arranging the data on a mixed format physical layout having a plurality of fixed-sized fields, a plurality of variable-sized fields and a plurality of offset slots, the fixed-sized fields being of a first size and the offset slots being of a second size; dividing the data on the mixed format physical layout into the fixed-sized fields and the variable sized fields; and compressing the data of the variable sized fields and the fixed-sized fields.

116. At a minimum, Defendant has known of the '897 patent at least as early as the filing date of this Complaint.

117. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

COUNT V

(INFRINGEMENT OF U.S. PATENT NO. 7,082,474)

118. Plaintiff incorporates paragraphs 1 through 118 herein by reference.

119. Plaintiff BYTEWEAVR is the assignee of the '474 patent, entitled "Data sharing and file distribution method and associated distributed processing system," with ownership of all

substantial rights in the '474 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

120. The '474 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The '474 patent issued from U.S. Patent Application No. 09/602,803. The '474 patent was granted on July 25, 2006 and expired on or about December 3, 2022.

121. Defendant has directly infringed one or more claims of the '474 patent in this District and elsewhere in Texas and the United States.

122. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

123. Defendant has directly infringed the '474 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Instrumentalities, their components, and/or products containing the same that incorporate the fundamental technologies covered by the '474 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief, Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Instrumentalities outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the

United States, thereby directly infringing the '474 patent. *See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc.*, 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

124. Furthermore, Defendant Databricks has directly infringed the '474 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments, companies, and/or brands of Defendant Databricks, including by designing the Accused Instrumentalities for U.S. consumers and selling and offering for sale the Accused Instrumentalities directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '474 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

125. For example, Defendant infringes claim 1 of the '474 patent via the Accused Instrumentalities, namely data management and analytics products and components, software,

services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, Databricks Spark Applications, and related data storage and compression techniques.

126. Those Accused Instrumentalities include a "method operating a distributed processing system having a network coupling a multiplicity of Host distributed devices for processing workloads for the distributed processing system, a plurality of Client systems requesting processing of the workloads, and a Server system for selectively distributing the workloads from the plurality of Client systems for processing by the distributed processing system" comprising the limitations of claim 1. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of receiving a request by the Server system from one of the plurality of Client systems to use the distributed processing system to process a first workload; sending the first workload to a first Host distributed device selected from the multiplicity of Host distributed devices; sending to the first Host distributed device an index of one or more data addresses defining a location of first data required to process the first workload; accessing the first data from a first data address selected from the one or more data addresses in the index; and updating the index to include a storage address of storage coupled to the first Host distributed device as a location of the first data.

127. At a minimum, Defendant has known of the '474 patent at least as early as the filing date of this Complaint.

128. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that

adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

COUNT VI

(INFRINGEMENT OF U.S. PATENT NO. 8,275,827)

129. Plaintiff incorporates paragraphs 1 through 129 herein by reference.

130. Plaintiff BYTEWEAVR is the assignee of the '827 patent, entitled "Software-based network attached storage services hosted on massively distributed parallel computing networks," with ownership of all substantial rights in the '827 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

131. The '827 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The '827 patent issued from U.S. Patent Application No. 09/834,785.

132. Defendant has directly and/or indirectly infringed (by inducing infringement) one or more claims of the '827 patent in this District and elsewhere in Texas and the United States.

133. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

134. Defendant has directly infringed the '827 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Instrumentalities, their components, and/or products containing the same that incorporate the fundamental technologies covered by the '827 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief,

Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Instrumentalities outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the United States, thereby directly infringing the '827 patent. *See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc.*, 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

135. Furthermore, Defendant Databricks has directly infringed the '827 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments, companies, and/or brands of Defendant Databricks, including by designing the Accused Instrumentalities for U.S. consumers and selling and offering for sale the Accused Instrumentalities directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '827 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based

subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

136. For example, Defendant infringes at least claims 2 and 14 of the '827 patent via the Accused Instrumentalities, namely data management and analytics products and components, software, services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, Databricks Spark Applications, and related data storage and compression techniques.

137. Those Accused Instrumentalities include "[a] computer-implemented method" comprising the limitations of claim 1. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of configuring a distributed processing system of a plurality of distributed devices coupled to a network, wherein the plurality of distributed devices include respective client agents configured to process respective portions of a workload for the distributed processing system, wherein the respective client agents for particular distributed devices of the plurality of distributed devices have corresponding software-based network attached storage (NAS) components configured to assess unused or under-utilized storage resources in selected distributed devices of the plurality of distributed devices; representing with the corresponding software-based NAS component that the selected distributed devices respectively comprise NAS devices having an available amount of storage resources related to the unused and under-utilized storage resources for the selected distributed devices; processing one or more of data

storage or access workloads for the distributed processing system by accessing data from or storing data to at least a portion of the available amount of storage resources to provide NAS service to a client device coupled to the network; enabling at least one of the selected distributed devices to function as a location distributed device to store location information associated with data stored by the selected distributed devices through use of the respective client agents for the particular distributed device; and enabling at least one of the selected distributed devices to function as a stand-alone dedicated NAS device through use of the respective client agents for the particular distributed device.

138. At a minimum, Defendant has known of the ’827 patent at least as early as the filing date of this Complaint.

139. On information and belief, since at least the above-mentioned date when Defendant was on notice of its infringement, Defendant has actively induced, under 35 U.S.C. § 271(b), importers, distribution partners, vendors, reseller partners, dealers, customers, installers, consumers, users and other related service providers that import, distribute, purchase, offer for sale, sell, or use the Accused Instrumentalities that include or are made using all of the limitations of one or more claims of the ’827 patent to directly infringe one or more claims of the ’827 patent by using, offering for sale, selling, and/or importing the Accused Products. Since at least the date of notice provided above, Defendant conducts infringing activities with knowledge, or with willful blindness of the fact, that the induced acts constitute infringement of the ’827 patent. On information and belief, Defendant intends to cause, and has taken affirmative steps to induce, infringement by importers, distribution partners, reseller partners, vendors, dealers, customers, installers, consumers, users, and other related service providers by at least, *inter alia*, the following: 1) sales and marketing activities that promote the infringing use of the Accused Instrumentalities, 2)

utilizing partners to create and/or maintain established distribution channels for the Accused Instrumentalities into and within the United States, 3) designing, developing, manufacturing the Accused Instrumentalities in conformity with U.S. laws, regulations, and market standards, 4) distributing or making available training, certifications, demos, webinars, events, resource libraries, documentation, instructions and/or manuals for the Accused Instrumentalities to purchasers and prospective buyers, 5) testing and certifying the features in the Accused Instrumentalities, and/or 6) providing technical support, upgrades and migrations, professional or tutorial services for the Accused Instrumentalities to purchasers in the United States. *See, e.g.*, *Databricks Support Policy,* DATABRICKS, https://www.databricks.com/support (detailing the services provided to customers regarding the Databricks Platforms including data management, analytics products, components, software, services, and processes); *Partner Connect*, DATABRICKS, https://www.databricks.com/partnerconnect ("Partner Connect makes it easy for you to discover data, analytics and AI tools directly within the Databricks platform — and quickly integrate the tools you already use today") (last visited Dec. 22, 2023). Such support services and partnering provide convenience, added functionality and value that induces partners and consumers to license, use, and incorporate the Defendant's data management and analytics products and components, software, services, and processes into their own network systems and businesses. *See, e.g.*, *Databricks for Industry*, DATABRICKS, https://www.databricks.com/solutions (providing use cases for Databricks' products and services in different industries including "Communications, Media, & Entertainment," "Financial Services," "Healthcare and Life Sciences," and "Public Sector") (last visited Oct. 13, 2023). Thus, these activities further infringe or induce infringement of the '827 patent.

140. On information and belief, despite having knowledge of the '827 patent and knowledge that it is directly and/or indirectly infringing one or more claims of the '827 patent, Defendant has nevertheless continued its infringing conduct and disregarded an objectively high likelihood of infringement. Each of Defendant's infringing activities relative to the '827 patent have been, and continue to be, willful, wanton, malicious, in bad-faith, deliberate, consciously wrongful, flagrant, characteristic of a pirate, and an egregious case of misconduct beyond typical infringement such that Plaintiff is entitled under 35 U.S.C. § 284 to enhanced damages up to three times the amount found or assessed.

141. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

COUNT VII

(INFRINGEMENT OF U.S. REISSUED PATENT NO. RE42153)

142. Plaintiff incorporates paragraphs 1 through 142 herein by reference.

143. Plaintiff BYTEWEAVR is the assignee of the '153 patent, entitled "Dynamic coordination and control of network connected devices for large-scale network site testing and associated architectures," with ownership of all substantial rights in the '153 patent, including the right to exclude others and to enforce, sue, and recover damages for past and future infringements.

144. The '153 patent is valid, enforceable, and was duly issued in full compliance with Title 35 of the United States Code. The '153 patent issued from U.S. Patent Application No. 10/190,368. The '153 patent was granted on March 1, 2005 and expired on or about March 26, 2022.

145. Defendant has directly infringed one or more claims of the '153 patent in this District and elsewhere in Texas and the United States.

146. On information and belief, Defendant designs, develops, manufactures, imports, distributes, offers to sell, sells, and uses the Accused Instrumentalities, including via the activities of Databricks and its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers.

147. Defendant has directly infringed the '153 patent via 35 U.S.C. § 271(a) by making, offering for sale, selling, importing and/or using the Accused Instrumentalities, their components, and/or products containing the same that incorporate the fundamental technologies covered by the '153 patent to, for example, its alter egos, intermediaries, agents, distributors, importers, partners, customers, subsidiaries, affiliates, and/or consumers. Furthermore, on information and belief, Defendant develops and designs the Accused Instrumentalities for U.S. consumers, makes and sells the Accused Instrumentalities outside of the United States, delivers those products and services to related entities, subsidiaries, distribution partners, resellers, vendors, installers, customers and other related service providers in the United States, or in the case that it delivers the Accused Instrumentalities outside of the United States it does so intending and/or knowing that those products are destined for the United States and/or designing those products for sale and use in the United States, thereby directly infringing the '153 patent. *See, e.g., Lake Cherokee Hard Drive Techs., L.L.C. v. Marvell Semiconductor, Inc.*, 964 F. Supp. 2d 653, 658 (E.D. Tex. 2013) (denying summary judgment and allowing presentation to jury as to "whether accused products manufactured and delivered abroad but imported into the United States market by downstream customers … constitute an infringing sale under § 271(a)").

148. Furthermore, Defendant Databricks has directly infringed the '153 patent through its direct involvement in the activities of its subsidiaries, and related entities and other U.S.-based subsidiaries (e.g., Mosaic ML, 8080 Labs Gmbh, Redash, and Cortex Labs), members, segments, companies, and/or brands of Defendant Databricks, including by designing the Accused Instrumentalities for U.S. consumers and selling and offering for sale the Accused Instrumentalities directly to its related entities and importing the Accused Instrumentalities into the United States for its related entities. On information and belief, U.S.-based members, segments, companies, and/or brands conduct activities that constitute direct infringement of the '153 patent under 35 U.S.C. § 271(a) by importing, offering for sale, selling, and/or using those Accused Instrumentalities in the U.S. on behalf of and for the benefit of Defendant. Defendant is vicariously liable for the infringing conduct of members, segments, companies, and/or brands of Databricks (under both the alter ego and agency theories). On information and belief, Defendant Databricks and other U.S. based subsidiaries, members, segments, companies, and/or brands of Databricks are essentially the same company. Moreover, Databricks, as the parent company, has the right and ability to control the infringing activities of those entities such that Defendant receives a direct financial benefit from that infringement.

149. For example, Defendant infringes claim 1 of the '153 patent via the Accused Instrumentalities, namely data management and analytics products and components, software, services, and processes such as the Databricks Lakehouse Platforms and their components, including the Databricks SQL, Delta Lake, Unity Catalog, Databricks Marketplace, Data Intelligence Platform, Databricks Spark Applications, and related data storage and compression techniques.

150. Those Accused Instrumentalities include "[a] method of providing dynamic coordination of distributed client systems in a distributed computing platform" comprising the limitations of claim 1. The technology discussion above and the example Accused Instrumentalities provide context for Plaintiff's allegations that each of those limitations are met. For example, the Accused Instrumentalities include the steps of providing at least one server system coupled to a network; providing a plurality of network-connected distributed client systems, the client systems having under-utilized capabilities and running a client agent program to provide workload processing for at least one project of a distributed computing platform; utilizing the server system to distribute workloads for the at least one project to the client systems and to distribute initial project and poll parameters to the client systems; receiving poll communications from the client systems during processing of project workloads by the client systems, wherein a dynamic snapshot information of current project status is provided based at least in part upon the poll communications; analyzing the poll communications to determine whether or not to make one or more modification to the initial project and poll parameters, wherein the modifications to the initial project and poll parameters utilize the dynamic snapshot information to determine whether to change how many client systems are active in the at least one project, and if a fewer number is desired, including within a polling response communications a reduction in the number of actively participating clients, and if a greater number is desired, adding client systems to active participation in the at least one project; sending the poll response communications to the client systems to modify the initial project and poll parameters depending upon one or more decisions reached in the analyzing step; and repeating the receiving, analyzing and sending steps to dynamically coordinate project activities of the plurality of client systems during project operations.

151. At a minimum, Defendant has known of the '153 patent at least as early as the filing date of this Complaint.

152. Plaintiff BYTEWEAVR has been damaged as a result of Defendant's infringing conduct described in this Count. Defendant is thus liable to BYTEWEAVR in an amount that adequately compensates BYTEWEAVR for its infringements, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court under 35 U.S.C. § 284.

CONCLUSION

153. Plaintiff is entitled to recover from Defendant the damages sustained by Plaintiff as a result of Defendant's wrongful acts in an amount subject to proof at trial, which, by law, cannot be less than a reasonable royalty, together with interest and costs as fixed by this Court.

154. Plaintiff has incurred and will incur attorneys' fees, costs, and expenses in the prosecution of this action. The circumstances of this dispute may give rise to an exceptional case within the meaning of 35 U.S.C. § 285, and Plaintiff is entitled to recover its reasonable and necessary attorneys' fees, costs, and expenses.

JURY DEMAND

155. Plaintiff hereby requests a trial by jury pursuant to Rule 38 of the Federal Rules of Civil Procedure.

PRAYER FOR RELIEF

156. Plaintiff requests that the Court find in its favor and against Defendant, and that the Court grant Plaintiff the following relief:

A. A judgment that Defendant have infringed the Asserted Patents as alleged herein, directly and/or indirectly by way of inducing infringement of such patents;

B. A judgment for an accounting of damages sustained by Plaintiff as a result of the acts of infringement by Defendant;

C. A judgment and order requiring Defendant to pay Plaintiff damages under 35 U.S.C. § 284, including up to treble damages as provided by 35 U.S.C. § 284, and any royalties determined to be appropriate;

D. A judgment and order requiring Defendant to pay Plaintiff pre-judgment and post-judgment interest on the damages awarded;

E. A judgment and order finding this to be an exceptional case and requiring Defendant to pay the costs of this action (including all disbursements) and attorneys' fees as provided by 35 U.S.C. § 285; and

F. Such other and further relief as the Court deems just and equitable.

Dated: March 8, 2024

Respectfully submitted,

/s/ *Jeffrey R. Bragalone*
Jeffrey R. Bragalone (lead attorney)
Texas Bar No. 02855775
E-mail: jbragalone@bosfirm.com
Terry A. Saad
Texas Bar No. 24066015
E-mail: tsaad@bosfirm.com
Marcus Benavides
Texas Bar No. 24035574
E-mail: mbenavides@bosfirm.com
Brandon V. Zuniga
Texas Bar no. 24088720
E-mail: bzuniga@bosfirm.com
Mark M.R. Douglass
Texas Bar No. 24131184
Email: mdouglass@bosfirm.com
BRAGALONE OLEJKO SAAD PC
901 Main Street
Suite 3800
Dallas, Texas 75202
Telephone: (214) 785-6670
Facsimile: (214) 785-6680

ATTORNEYS FOR PLAINTIFF BYTEWEAVR, LLC