Cloudera Data Science Workbench

Cloudera Data Science Workbench Product Overview

Date published: 2020-02-28
Date modified:

https://docs.cloudera.com/

Legal Notice

Contents

Cloudera Data Science Workbench overview

Machine learning has become one of the most critical capabilities for modern businesses to grow and stay competitive today. From automating internal processes to optimizing the design, creation, and marketing processes behind virtually every product consumed, ML models have permeated almost every aspect of our work and personal lives.

ML development is iterative and complex, made even harder because most ML tools aren't built for the entire machine learning lifecycle. Cloudera Data Science Workbench on Cloudera Data Platform accelerates time-to-value by enabling data scientists to collaborate in a single unified platform that is all inclusive for powering any AI use case. Purpose-built for agile experimentation and production ML workflows, Cloudera Data Science Workbench manages everything from data preparation to MLOps, to predictive reporting. Solve mission critical ML challenges along the entire lifecycle with greater speed and agility to discover opportunities which can mean the difference for your business.

Each ML workspace enables teams of data scientists to develop, test, train, and ultimately deploy machine learning models for building predictive applications all on the data under management within the enterprise data cloud. ML workspaces support fully-containerized execution of Python, R, Scala, and Spark workloads through flexible and extensible engines.

Key differences between Cloudera Machine Learning (Public Cloud) and Cloudera Data Science Workbench

This topic highlights some key differences between Cloudera Data Science Workbench and its cloud-native counterpart, Cloudera Machine Learning.

How is Cloudera Machine Learning (CML) related to Cloudera Data Science Workbench (CDSW)?

CML expands the end-to-end workflow of Cloudera Data Science Workbench (CDSW) with cloud-native benefits like rapid provisioning, elastic autoscaling, distributed dependency isolation, and distributed GPU training.

It can run its own native distributed computing workloads without requiring a separate CDH cluster for scale-out compute. It is designed to run on CDP in existing Kubernetes environments, such as managed cloud Kubernetes services (EKS, AKS, GKE), Red Hat OpenShift, or ECS (Embedded Container Service), reducing operational costs for some customers while delivering multi-cloud portability.

Both products help data engineers and data science teams be more productive on shared data and compute, with strong security and governance. They share extensive code.

There is one primary difference:

- CDSW extends an existing CDH cluster, by running on gateway nodes and pushing distributed compute workloads to the cluster. CDSW requires and supports a single CDH cluster for its distributed compute, including Apache Spark.
- In contrast, CML is self-contained and manages its own distributed compute, natively running workloads - including but not limited to Apache Spark - in containers on Kubernetes.

  Note: It can still connect to an existing cluster to leverage its distributed compute, data, or metadata (SDX).

Table 1: Key Differences

	CDSW	CML
Architecture	CDSW can run on a CDP-DC, CDH (5 or 6), and HDP cluster and runs on one or more dedicated gateway nodes on the cluster.	CML is self-contained and does not require an attached CDH/HDP cluster.

	CDSW	CML
	Notion of 1 master and multiple worker hosts.	No designated master and worker hosts; all nodes are ephemeral.
Security	Kerberos authentication integrated via the CDH/ HDP cluster	Centralised identity management using FreeIPA via the Cloudera Data Platform (CDP).
	External authentication via LDAP/SAML.	
App Storage	Project files, internal postgresDB, and Livelog, are all stored persistently on the Master host.	All required persistent storage is on cloud-managed block store, NFS, and a relational data store. For example, for AWS, this is managed via EFS.
Compute	Python/R/Scala workloads run on the CDSW gateway nodes of the cluster.	Python/R/Scala workloads run on the CDP/cloud-provider-managed K8s cluster.
	CDSW pushes distributed compute workloads, such as Spark-on-YARN, to the CDH/HDP cluster.	Spark-on-YARN is not supported; Spark-on-K8s instead. Workloads will run on a dedicated K8s cluster provisioned within the customer environment.
	No autoscaling.	Autoscaling via your cloud service provider. Kubernetes/node-level autoscaling will be used to expand/contract the cluster size based on demand.
Packaging	Available as a downloadable RPM and CSD.	Available as a managed service on CDP.
	Spark is packaged with CDH.	Spark on K8s is packaged with CML - no dependency on an external cluster.
Data Access	Data usually resides on the attached CDH/HDP cluster in HDFS, Hive, HBase, and so on.	Data can reside on object storage such as S3 or any pre-existing workload clusters registered with CDP.