

Cloudera Data Engineering 1.23.0

## Cloudera Data Engineering Release Notes

Date published: 2020-07-30

Date modified: 2024-11-12

# CLOUDERA

<https://docs.cloudera.com/>

# Legal Notice

© Cloudera Inc. 2024. All rights reserved.

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

# Contents

<b>What's new in Cloudera Data Engineering Public Cloud?.....</b>	<b>5</b>
November 12, 2024.....	5
August 27, 2024.....	6
July 23, 2024.....	7
June 24, 2024.....	8
May 31, 2024.....	8
May 22, 2024.....	8
March 27, 2024.....	9
March 20, 2024.....	9
February 27, 2024.....	9
December 11, 2023.....	11
September 14, 2023.....	11
June 29, 2023.....	12
May 18, 2023.....	12
March 30, 2023.....	13
January 19, 2023.....	14
November 23, 2022.....	14
October 12, 2022.....	15
September 26, 2022.....	16
July 20, 2022.....	16
June 30, 2022.....	17
April 27, 2022.....	18
February 09, 2022.....	18
December 21, 2021.....	19
November 9, 2021.....	20
Older releases.....	20
October 18, 2021.....	20
August 30, 2021.....	21
August 2, 2021.....	22
June 23, 2021.....	22
May 20, 2021.....	23
April 7, 2021.....	24
March 9, 2021.....	24
February 4, 2021.....	24
December 21, 2020.....	25
November 9, 2020.....	26
September 21, 2020.....	26
July 30, 2020.....	27
 <b>Cloudera Data Engineering fix for CVE-2021-44228.....</b>	 <b>27</b>
 <b>Known issues and limitations in Cloudera Data Engineering Public</b>	
<b>Cloud.....</b>	<b>27</b>
General known issues with Cloudera Data Engineering.....	27
Technical service bulletins.....	39
Limitations.....	40

<b>Cloudera Data Engineering Runtime end of support.....</b>	<b>40</b>
<b>Compatibility for Cloudera Data Engineering and Runtime components.....</b>	<b>41</b>
<b>Using the Cloudera Runtime Maven repository for Cloudera Data Engineering.....</b>	<b>43</b>

# What's new in Cloudera Data Engineering Public Cloud?

This section lists major features and updates for the Cloudera Data Engineering (CDE) service in CDP Public Cloud.

## November 12, 2024

This release (1.23.0) of the Cloudera Data Engineering service on CDP Public Cloud introduces the following changes.

### External IDE connectivity through Spark Connect-based sessions (Technical Preview)

Cloudera Data Engineering supports Spark Connect sessions, which are Cloudera Data Engineering sessions that expose the Spark Connect interface. Spark Connect sessions allow you to run Spark commands from any remote Python environment. For more information, see [Spark Connect Sessions](#).

### In-place upgrade enhancements

In-place upgrade enhancements have been documented to improve handling Airflow version 2.9 and Python version 3.11 changes during the upgrade. For more information, see [In-place upgrade with Airflow Operators and Libraries](#).

### Backup-restore-based upgrade enhancements

Backup-restore-based upgrade enhancements have been implemented.

### Airflow version upgrade to 2.9

The Airflow version that Cloudera Data Engineering uses is upgraded to 2.9. For more information, see:

- [Apache Airflow 2.9 Release Notes](#)
- [Compatibility for Cloudera Data Engineering and Runtime components](#)

### Python 3.11 support for Apache Airflow

The Python version that Cloudera Data Engineering uses is upgraded to 3.11. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Iceberg version upgrade to 1.5

The Iceberg version that Cloudera Data Engineering uses is upgraded to 1.5.2. For more information, see: [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Kubernetes version upgrade to 1.30

The Kubernetes version that Cloudera Data Engineering uses is upgraded to 1.30. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Deprecation of CDW Operator

Starting from Cloudera Data Engineering service version 1.23.0, CDW Operator is deprecated. Cloudera recommends you to use SQL Operator in place of CDW Operator. For more information, see [Creating a connection to Cloudera Data Warehouse for CDW Operator](#).

### Active node count

The active node count has been removed from the following Cloudera Data Engineering UI pages:

- Administrator Overview
- Administrator Overview Service Details

For real-time node count information, check your Cloud Service Provider's website. For more information, see [Checking the node count on your Cloud Service provider's website](#).

### Fixed issues

#### **DEX-9617: CDE Jobs Page-Airflow UI: Audit Log tab does not allow further navigation**

The fix allows you to view the Airflow DAG run audit logs.

## August 27, 2024

This release (1.22.0-h1) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

This release does not contain new features, but includes the following fixes:

#### **DEX-13103: Fix dex-runtime-python-builder to use python 3.8 even for Spark 3.3+**

Apache Spark now comes with Python 3.8 installed for versions 3.3 and 3.5.

#### **DEX-14027: Spark 3.5.1 on RAZ; few jobs are failing with error 'org.apache.hadoop.fs.s3a.impl.InstantiationException'**

Jobs running on RAZ-enabled clusters on Apache Spark 3.5.1 failed with the org.apache.hadoop.fs.s3a.impl.InstantiationException error.

#### **DEX-14037: Restore tries to create deleted default VC**

When you performed the CDE jobs backup and restore operation, CDE tried to restore a default VC as well that had been deleted before the backup was created. This has been corrected, and now, only those VCs are restored that are included in the Backup archive.

#### **DEX-14231: Allow setting Spark configs to empty & multiple '=' separators values in the UI**

You can use the empty value and multiple '=' separator values in the UI for Spark configurations in the create-job and VC-level Spark configurations.

#### **DEX-14313: CDE Backup/Restore Upgrade Does Not Update K8 version**

When you performed the CDE jobs backup and restore operation, during the upgrade, the Kubernetes version was not upgraded.

#### **DEX-14412: Runtime API sometimes crashes due to panic: interface conversion: interface {} is cache.DeletedFinalStateUnknown, not \*v1.Event**

The Runtime API sometimes crashed with the panic: interface conversion: interface {} is cache.DeletedFinalStateUnknown, not \*v1.Event error.

#### **DEX-14424: TGT Secret name validation and sanitisation incorrect**

The TGT Kerberos tickets are saved as a secret with a name that is based on the CDP username. An issue related to the validation and sanitization of the CDP usernames has been fixed.

#### **DEX-14556: Unhide 1.22 CDP CLI changes related to custom private DNS zone**

The CDP CLI commands related to private DNS zones have been made available.

## July 23, 2024

This release (1.22.0) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### AWS Graviton support (GA)

AWS Graviton is a general purpose, ARM-based processor family. From the AWS Graviton family, CDE supports AWS Graviton 3. AWS Graviton delivers currently the best price performance for cloud workloads running in AWS EC2. With AWS Graviton, you can optimize costs and achieve better performance.

For more information, see [AWS Graviton instances in CDE](#).

### Support for in-place upgrade with private networking

If you use services that run on a private network, (for example, Private EKS), your CDE service is eligible for an in-place upgrade.

For more information, see [In-place upgrade with Apache Airflow Operators and Libraries](#)

### Kubernetes version upgrade to 1.28

The Kubernetes version that CDE uses is upgraded to 1.28. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Iceberg version upgrade to 1.4.3

The Iceberg version that CDE uses is upgraded to 1.4.3. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Support for Spark version 3.5.1

CDE supports Spark version 3.5.1. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Virtual Cluster (VC)-level Spark configurations (Technical Preview)

With the VC-level Spark configurations option, you can create and update Spark configurations that, by default, apply to all Spark jobs that are run in a VC.

For more information, see [Managing Virtual Cluster-level Spark configurations](#)

### Azure private DNS zones (Technical Preview)

With the Azure private DNS zones feature, you can optimize costs by leveraging existing Azure private DNS zone resources. When creating the CDE service, you can use an Azure private DNS zone for the Azure Kubernetes Service (AKS), the Storage Account File Share, and for the database.

For more information, see [Azure private DNS zones in a CDE service](#).

### Fixed issues

- DEX-12763 - Upgraded CDE Service to 1.20.3 and above with SSD Instances unable to run jobs
- DEX-13755 - Azure and AWS service B&R does not restore all the jobs
- DEX-13151 - UI keeps loading when jobs are unpaused if the user has readonly access
- DEX-12634 - VC Security/Access Control allows for duplicate user record
- DEX-14080 - Remove VC cluster maintenance mode configuration on restore
- DEX-13154 - View log for VC will not redirect to Logs tab

- DEX-13742 - UI should send all details for previous Service version
- DEX-13262 - UI: editing a schedule for the example job fails to send a request
- DEX-13385 - UI does not indicate user does not have access to CLI tool
- DEX-13639 - Without All Purpose details we are showing All Purpose OnDemand and Spot instances on Service Config page
- DEX-13692 - Include sub seconds into execution date for manually triggered Airflow runs
- DEX-13958 - In place upgrade fails from 1.19/1.20 -> 1.21/1.22 if custom runtime images are used with public docker registry

## June 24, 2024

This release (1.21.0-h2) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Azure and AWS service backup and restore

If there were Airflow jobs running on a Virtual Cluster (VC) and you performed a cluster backup and restore, the CDE service and the VC were restored, but the Airflow jobs on the VC were not restored.

This issue has been fixed and now if you perform a backup and restore, the Airflow jobs are restored as well.

### In-place upgrade using custom runtime images and public Docker registry

If you performed the in-place upgrade from CDE version 1.19 or version 1.20 to version 1.21 with a custom runtime image that uses a public Docker registry without a credential, the upgrade failed.

This issue has been fixed and now you can perform the in-place upgrade with a custom runtime image that uses a public Docker registry without a credential.

## May 31, 2024

This release (1.21.0-h1) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following change.

### Listing ACL-based users

Listing users governed by the Access Control List (ACL) for CDE Server versions lower than 1.20.3.

In CDE 1.21, users who had clusters lower than 1.20.3 with ACL-based access control were not able to interact with the ACL. Now, the cluster version is checked and the appropriate ACL-based, or RBAC-based UI is displayed, based on the cluster version.

## May 22, 2024

This release (1.21) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following change.

### Airflow version upgrade to 2.7

We have upgraded the Airflow version to 2.7. For more information, see:

- [Apache Airflow 2.7 Release Notes](#)
- [Compatibility for Cloudera Data Engineering and Runtime components](#)

.



## March 27, 2024

This release (1.20.3-h2) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### SSD Based Instance used on CDE Service

This release includes a fix for SSD based instance use in the CDE service. Jobs can now be run on an SSD based instances from 1.20.3-h2 and on.

### Bug fixes and improvements

This release includes bug fixes and performance improvements for in-place upgrades.

### Known issue added

When a customer creates a new CDE service with an SSD Instance enabled on CDE version greater than or equal to 1.19.4, Spark and Airflow jobs do not start at all. For more information on the workaround, see [General issues](#).

## March 20, 2024

This release (1.20.3-h1) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following change.

### Support for Azure Database for MySQL - Flexible Servers

CDE 1.20.3-H1 moves from Azure Single Server to Azure Flexible Server. This update is due to Azure no longer supporting Azure Single Server as of March 19, 2024.

## February 27, 2024

This release (1.20.3) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Sessions (GA) with enhancements

CDE Sessions is now GA as a default feature. Sessions is an interactive short-lived development environment for running Spark commands to help you iterate upon and build your Spark workloads. The Interaction tab was added so that you can run Java, Impala, and PySpark code in blocks to develop applications. Cloudera currently supports Sessions in the CDE CLI and UI. The Spark UI tab was also added to view active sessions. For more information, see [Creating and Managing CDE Sessions](#) and [Managing Sessions in CDE using the CLI](#).

### Updated CDE homepage 2.0

CDE now has a revamped landing page with a new design that focuses on a more simplified workflow: Develop, Deploy, and Monitor.

### In-place upgrade (GA)

CDE supports upgrades from two CDE versions 1.19.2 and above for AWS and 1.19.4 and above for Azure. Users will need to manually pause, backup, and restore each Virtual Cluster to account for upgrade failures. A way to handle upgrade failures has also been created. In-place upgrade also includes the following:

- Upgrades of CDE core components include: EKS, AKS Services, and Application Services
- Upgrades of dependencies include: Helm, K8s versions, YuniKorn

For more information, see [Upgrading CDE](#) and [Handling upgrade failures in CDE](#).

### Git repositories (Technical Preview)

You can now use Git repositories to collaborate, manage project artifacts, and promote applications from lower to higher environments. Cloudera currently supports Git providers such as GitHub, GitLab, and Bitbucket. Repository files can be accessed when you create a Spark or Airflow job. You can then deploy the job and use CDE's centralized monitoring and troubleshooting capabilities to tune and adjust your workloads. For more information, see [Creating a Git repository in CDE \(Technical Preview\)](#).

### Airflow custom operators and libraries for Python

CDE supports 3rd party python-based plugins and libraries to build custom Airflow pipelines using the CDE UI and API. For more information, see [Using custom operators and libraries for Apache Airflow](#) and [Using custom operators and libraries for Apache Airflow using API](#).

### New configuration parameters added for Airflow

New parameters were added for Airflow. For more information, see [CDE CLI Airflow flag reference](#) and [Submitting an Airflow job using the CLI](#).

### Support for Spark Streaming (Technical Preview)

CDE supports Spark Structured Streaming for both Spark 2 and Spark 3. For more information, see [Support for Spark Structured Streaming in Cloudera Data Engineering \(Technical Preview\)](#).

### Support for group-based access control for virtual clusters

You can now restrict or grant access to a virtual cluster for specific groups that you specify. For more information, see [Applying user and group access for virtual clusters](#).

### Edit all-purpose nodes for AWS and Azure

New sliders to edit all-purpose nodes for AWS and Azure have been added to allow users to control the size of your auto-scaling group. For more information, see [Enabling a Cloudera Data Engineering service](#).

### Kubernetes update

CDE now supports K8s 1.27. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### End of Service Notice

For more information, see [Support lifecycle policy](#).

### Support for Airflow 2.6

Support for Airflow 2.6 to version 2.6. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Update Automating data pipelines page with Impala VW connections

Impala VWs are supported and the CDWOperator is no longer needed for executing queries. For more information, see [Automating data pipelines using Apache Airflow in Cloudera Data Engineering](#).

### Support for Iceberg 1.3

When you upgrade to CDE 1.20.3, ensure that you also upgrade to Iceberg 1.3. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### Support for setting subnets for the Load Balancer

CDE now supports setting subnets for the Load Balancer during service creation. For more information, see [Enabling a Cloudera Data Engineering service](#).

### Enable Observability during service creation

You can select Enable Observability Analytics if you want diagnostic information about jobs and query execution sent to Cloudera Observability. This helps optimize troubleshooting. For more information, see [Enabling a Cloudera Data Engineering service](#)

## December 11, 2023

This release (1.19.4) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Kubernetes 1.26 support

- CDE now supports Kubernetes 1.26 for Azure and Amazon Web Services (AWS).
- You can upgrade to the Kubernetes 1.26 cluster through the CDE supported upgrade path.

### Amazon Relational Database Service (Amazon RDS) at rest encryption with Customer Managed Keys (CMK) (Technical Preview)

CDE Service deployed on AWS using this CMK enabled environment, will start using CMK based data at rest encryption for RDS.

### AWS Kubernetes secret encryption with Customer Managed Keys (CMK) (Technical Preview)

CDE Service deployed using this CMK enabled environment, will start using CMK based encryption for Kubernetes secrets.

### Amazon Elastic File System (AWS EFS) data at-rest encryption with Customer Managed Key (CMK) (Technical Preview)

Customer Managed Key is a feature supported by AWS that give customers ownership of their encryption keys.

### Amazon Elastic File System (AWS EFS) data in-transit encryption

Support for data in-transit encryption through EFS CSI Driver. The EFS data read/write over the wire are encrypted by TLS.

### Amazon Elastic File System (AWS EFS) Anonymous Access restriction

This feature includes security hardening by preventing anonymous user or machines from accessing EFS and its access points.

## September 14, 2023

This release (1.19.3) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Support for Spark 3.3 runtimes

CDE supports Spark 3.3. Note that Spark 3.3.0 is supported on Data Lake 7.2.15. For more information, see [Cloudera Data Engineering and Data Lake compatibility](#).

### Support for semi-private network for AWS and fully private network for AKS (Preview)

CDE now supports semi-private network for Amazon Web Services (AWS) and fully private network Azure Kubernetes Service (AKS) with Cluster Connectivity Manager v 2 (CCMv2). For more information, see [Enabling a Cloudera Data Engineering service](#), [Enabling a semi-private network on AWS](#), [Enabling a fully private network for a CDE service for Azure \(Preview\)](#), and [Cluster Connectivity Manager](#).

## June 29, 2023

This release (1.19.2) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Kubernetes update

CDE now supports K8s 1.25 for Azure and Amazon Web Services (AWS).

### Support for new AWS region

CDE now supports the EU Milan region for AWS.

### Support for user defined routing (UDR)

CDE now supports UDR when you enable a CDE service for Azure. For more information, see [Enabling a Cloudera Data Engineering service](#).

### Support for more AMD instances

CDE now includes more AMD instances for the Workload Type drop-down menu when you enable a CDE service.

### Workload Secrets

CDE now provides a secure way to create and store workload secrets for Cloudera Data Engineering (CDE) Spark Jobs. This is a more secure alternative to storing credentials in plain text embedded in your application or job configuration. For more information, see [Managing workload secrets with CDE Spark Jobs using the API](#).

## May 18, 2023

This release (1.19) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### [Technical Preview] Cloudera Data Engineering Sessions

CDE 1.19 introduces Sessions as an interactive short-lived development environment for running Spark commands to help you iterate upon and build your Spark workloads.

- The Sessions feature is available via CDE user interface and the CDE CLI.
- Python, Scala, and Java are supported session types.

For more information, see [Creating Sessions in Cloudera Data Engineering \[Technical Preview\]](#) and [Managing Sessions in Cloudera Data Engineering using the CLI](#).

### [Technical Preview] Lock and unlock

CDE supports locking of a CDE Service which freezes the configuration of a Service and its corresponding Virtual Clusters. When a Service is locked, you are unable to edit, add, or delete a Service and its Virtual Clusters. This will assist during planned upgrades to ensure changes are not made to the Service. For more information, see [Locking and unlocking a CDE Service](#).

### [Technical Preview] Airflow file based resource using the CDE CLI

CDE supports Airflow file based resources using the CDE CLI. By creating a pipeline in CDE using the CLI, you can add custom files that are available for tasks. For more information, see [Creating an Airflow pipeline with custom files using CDE CLI \[technical preview\]](#).

### Support for new AWS regions

CDE now supports the Hong Kong and Jakarta regions for AWS.

### Support for multiple Spark 3 runtimes

CDE supports Spark 3.3. Note that Spark 3.3 must use Data Lake 7.2.16. For more information, see [Cloudera Data Engineering and Data Lake compatibility](#).

### Creating and using multiple profiles using CDE CLI

You can now add a collection of CDE CLI configurations grouped together as profiles, to the config.yaml file. You can use these profiles while running commands. You can set the configurations either at a profile level or at a global level. For more information, see [Creating and using multiple profiles using CDE CLI](#).

### Using spark-submit drop-in migration tool for migrating Spark workloads to CDE

CDE provides a command line tool cde-env to help migrate your CDP Spark workloads running on CDP Private Cloud Base (“spark-on-YARN”) to CDE without having to completely rewrite your existing spark-submit command-lines. For more information, see [Using spark-submit drop-in migration tool for migrating Spark workloads to CDE](#).

### New Virtual Cluster types

CDE now provides a choice between two tiers during Virtual Cluster creation. Administrators will see the following two options:

- Core (Tier 1) - Batch-based transformation and engineering options.
- All-Purpose (Tier 2) - Develop using interactive sessions and deploy both batch and streaming workloads.

For more information, see [Creating virtual clusters](#).

### Default setting for external.table.purge property for migrating Hive tables

When migrating Iceberg tables in Spark 3, the external.table.purge property is now set to FALSE by default. For more information, see [Importing and migrating Iceberg table in Spark 3](#).

### New exit codes for the CDE CLI

CDE now provides exit codes for the CDE CLI. The exit codes help users better identify the error. For more information, see [Cloudera Data Engineering CLI exit codes](#).

### Support for Data Lake

CDE 1.19 supports Data Lake 7.2.14 through 7.2.16. For more information, see [Cloudera Data Engineering and Data Lake compatibility](#).

## March 30, 2023

This release (1.18.3) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces a new improvement that is described in this topic.

### Kubernetes update

CDE now supports K8s 1.24 for Azure and Amazon Web Services (AWS).

## January 19, 2023

This release (1.18.1) of the Cloudera Data Engineering Service on CDP Public Cloud introduces the following changes.

### Upgrade to Airflow 2.3.4

CDE 1.18.1 now runs with Airflow 2.3.4. This upgrade includes several fixes to improve performance and stability.

### Support for additional Grafana charts

Additional Grafana charts that specify metrics for Livy memory and API server have been added.

### Support for Data Lake

CDE 1.18.1 now supports Data Lake 7.2.16. For more information, see [Cloudera Data Engineering and Data Lake compatibility](#).

## November 23, 2022

This release (1.18) of the Cloudera Data Engineering Service on CDP Public Cloud introduces the following changes.

### Updated CDE user interface

The user interface for CDE 1.17 and above has been updated with easy access to commonly used pages, a new Home page, and a Virtual Cluster drop-down menu that allows you to view relevant content related to each Virtual Cluster that you select. Only users who have a CDE Service on 1.18 and create new Virtual Clusters on 1.18 will see the changes. Users on older versions will continue have access to the old UI. The following user interface changes were made:

- Left-hand menu displays the following:
  - Home- New landing page that displays Virtual Clusters and convenient quick-access links.
  - Jobs - Displays jobs for the Virtual Cluster that you select from the drop-down menu in the upper left-hand corner.
  - Job Runs - Displays the run history of all jobs within a selected Virtual Cluster.
  - Resources - Displays resources created within a selected Virtual Cluster.
  - Administration - Displays services and Virtual Clusters that can be customized (previously known as the Overview page).



**Note:** If you're using a browser in incognito mode, you'll need to allow all cookies in your browser settings so that you can view the following CDE pages: Pipelines, Spark, and Airflow.

### Airflow performance

Airflow scaling improvements include support for 1500 DAGs on AWS and about 300 to 500 DAGs when deploying on Azure. For more information, see [Apache Airflow scaling and tuning considerations](#).

### Support for the eu-1 (Germany) and ap-1 (Australia) regional Control Plane

The eu-1 (Germany) and ap-1 (Australia) regional Control Plane now supports CDE. For the list of all supported services for all supported Control Plane regions, see [CDP Control Plane regions](#).

### Java Virtual Machine Debugger (Tech preview)

Attaching a remote debugger (Java virtual machine (JVM) debugger) to a CDE Spark job is now supported as a technical preview feature. For more information, see [Using Java virtual machine \(JVM\) debugger with Apache Spark jobs in Cloudera Data Engineering \(Preview\)](#).

### Hive Warehouse Connector tables

Hive Warehouse Connector (HWC) tables are now supported in Spark 3 of CDE.

### Backup & Restore in object storage

Remote backup storage (object store) is now supported. Previously, only backup to and restore from local storage was supported. This is supported through the CLI and API only. For more information, see [Backing up Cloudera Data Engineering jobs](#) and [Restoring Cloudera Data Engineering jobs from backup](#).

### Limitations for raw Scala code in CDE

Limitations have been added to the raw Scala code. For limitation details, see [Running raw Scala code in Cloudera Data Engineering](#).

### Support for Iceberg V2

Iceberg table format version 2 (v2) is generally available (GA) in CDE. The latest specifications include the following key updates:

- UPDATE and DELETE operations follow the Iceberg format v2 row-level position delete specification and enforces snapshot isolation.
- DELETES, UPDATES, and MERGE operations use the merge-on-read function by default. Merge-on-read is more efficient than the copy-on-write function because it does not rewrite file data.

For more information, see [Prerequisites](#)

## October 12, 2022

This release (1.17-h1) of the Cloudera Data Engineering Service on CDP Public Cloud introduces the following changes.

### Support for Iceberg 0.14

When you upgrade to CDE 1.17-h1, ensure that you also upgrade to Iceberg 0.14. For more information, see [Using Apache Iceberg in Cloudera Data Engineering](#). The following features are included with Iceberg 0.14:

- MERGE operations allow for bulk updates and DELETES.
- CDE Azure deployments are now able to leverage Iceberg for Lakehouse architecture.
- [Technical Preview] Iceberg table format version 2 (v2) is the latest specification available and includes the following key updates:
  - UPDATE and DELETE operations follow the Iceberg format v2 row-level position delete specification and enforce snapshot isolation.
  - DELETES, UPDATES, and MERGE operations use the merge-on-read function by default. Merge-on-read is more efficient than the copy-on-write function because it does not rewrite file data.

### Set default values for the variables in CDE job specification

Using [--default-variable] flags you can now replace strings in job values. For more information, see [Creating and updating Apache Spark jobs using the CLI](#).

## September 26, 2022

This release (1.17) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Service reference architecture

Reference architecture has been published to outline best practices on scaling CDE service, including when running Airflow based pipelines. For more information, see [Recommendations for scaling CDE deployments](#) and [Apache Airflow scaling and tuning considerations](#).

### Kubernetes dashboard

CDE provides the option to view the Kubernetes dashboard to provide an easy user experience for monitoring your diagnostics. The dashboard is to be used when troubleshooting in coordination with Cloudera Support. For more information, see [Accessing the Kubernetes dashboard](#).

### Azure private storage

As of CDE 1.16, Azure private storage is supported. Details around deploying and configuring CDE with Azure private storage are now available. For more information, see [Supporting Azure private storage](#).

### SSL Support for Azure DB

For increased security, CDE on Azure will now deploy SSL enabled with TLS 1.2.

### Technical Service Bulletin (2022-587)

[Technical Service Bulletin \(2022-587\)](#) has been resolved.

## July 20, 2022

This release (1.16) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Airflow pipeline UI editor (GA)

- Airflow Pipeline UI editor is now GA as a default feature in new Virtual Clusters with support for all major browsers (Firefox, Chrome, and Safari).

### Upgrade to Airflow 2.2.5

CDE 1.16 now runs with Airflow 2.2.5.

- Several fixes to improve performance and stability have been bundled with the upgrade.
- New Virtual Clusters will automatically use the new Airflow version.
- This version deprecated the timezone package usage. The DAGs need to be updated to use the pendulum package instead. If your airflow DAGs need to be timezone aware then they should rely on the pendulum timezone library for start and end dates as described [here](#). Otherwise, the backup and restore process will not be able to restore these DAGs. For more information, see [CDE known issues](#).

### Spark 3 support for raw scala code

Spark 3 support for raw scala code.

Previously this feature was limited to Spark 2, it is now extended to Spark 3 based Virtual Clusters. This allows you to directly run raw scala via API & CLI in batch-mode without having to compile, similar to what spark-shell supports.



### Support for Azure private storage

CDE now supports Azure private storage. Both private ABFS and ADLS gen2 containers are now supported.

### Editing VC configurations post creation

You can now modify the virtual settings such as cluster quotas (CPU/memory) dynamically.

### Loading example jobs and sample data using new VCs

CDE provides an option to add in-product examples of data & jobs in new virtual clusters to facilitate smoother onboarding and learning for new customers.

### Kubernetes update

CDE now supports K8s 1.22.

- The CSP EOS for K8s 1.21 is as follows:

For Azure: July 2022

For AWS: February 2023

- Check for removals as per this upgrade:

[Kubernetes API and Feature Removals In 1.22](#) and [Removed APIs by release](#)

### Support for creation of a Default Virtual Cluster

CDE now provides support for default virtual clusters. This will help you get a jump start to create your jobs easily, without having to wait to create a CDE virtual cluster, making the onboarding smoother. You have the option to turn this selection off if you do not wish to use a default virtual cluster.

For more information, see [Enabling Cloudera Data Engineering service](#).

### [Technical Preview] In-place upgrades

CDE supports upgrades from two CDE versions prior for both AWS and Azure. For example, if the current CDE version is 1.18, then upgrades are supported from CDE 1.16. The upgrades can be triggered by an Admin from CDE UI.

- Users will need to manually pause/backup/restore each Virtual Cluster to account for upgrade failures.
- Upgrades of CDE core components include: EKS, AKS Services, and Application Services
- Upgrades of dependencies include: Helm, K8s versions, YuniKorn

## June 30, 2022

This release (1.15-h1) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud has certified support for Apache Iceberg v0.13.

### GA of Apache Iceberg

- You can use Cloudera Data Engineering virtual clusters running Spark 3 to interact with the latest version (0.13) of Apache Iceberg tables.
- CDE supports row level updates via copy-on-write MERGE / UPDATES/ DELETES operations. Copy-on-write is helpful in bulk updates in read heavy use-cases.
- Compaction is also supported using Spark Iceberg APIs.
- As support for Atlas lineage is still in progress, users should set the following Spark property in their jobs: `spark.lineage.enabled=false`.
- For more information, see [Using Apache Iceberg in Cloudera Data Engineering](#).

## April 27, 2022

This release (1.15) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### [Technical Preview] Support for Iceberg 0.13

- You can use Cloudera Data Engineering virtual clusters running Spark 3 to interact with the latest version (0.13) of Apache Iceberg tables.
- CDE now supports row level updates via copy-on-write MERGE / UPDATES/ DELETES operations. Copy-on-write is helpful in bulk updates in read heavy use-cases.
- For more information, see [Using Apache Iceberg in Cloudera Data Engineering](#).

### [Technical Preview] In-place upgrades

- CDE supports upgrades from two CDE versions prior for both AWS and Azure. For example, if the current CDE version is 1.18, then upgrades are supported from CDE 1.16. The upgrades can be triggered by an Admin from CDE UI.
- Users will need to manually pause/backup/restore each Virtual Cluster to account for upgrade failures.
- Upgrades of CDE core components include: EKS, AKS Services, and Application Services
- Upgrades of dependencies include: Helm, K8s versions, YuniKorn
- For more information, see [CDE In-place Upgrades \(Preview\)](#)

### Job email alerts

SLA miss and job failure conditions can be configured for email notifications.

### Job runtime notices

Active jobs will now provide notification to the user when certain conditions are met and jobs are not behaving as expected making it easier to understand why jobs might be stuck or not making progress.

For more information, see [Running jobs in Cloudera Data Engineering](#)

### Spark 3.2

CDE now supports Apache Spark 3.2.

### Data Lake upgrades

CDE has now been certified when Data Lake is resized..

## February 09, 2022

This release (1.14) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Improved handling of job resources to reduce EFS utilization

- Recursive copying of frequently used and large file resources can result in very high I/O throughput and can exhaust cloud storage burst credits, leading to poor performance. To avoid excessive file copying, CDE now uses hard linking in AWS by default.

### [Technical Preview] Apache Iceberg support

- Apache Iceberg tables are now supported with Spark 3 virtual clusters on AWS. Use tables at petabyte scale without impacting query planning, while benefiting from efficient metadata management, snapshotting, and time-travel.
- Run multi-analytic workloads by accessing those same tables in Cloudera Data Warehouse (CDW) with Hive and Impala for BI and SQL analytics (Expected in an upcoming CDW release).

### [Technical Preview] Remote Shuffle Service

- You can now store Spark shuffle data on remote servers. This improves resilience in case of executor loss.
- This feature is available as a Technical Preview. Contact your Cloudera account representative to enable access to this feature.

### Unified diagnostic bundle

- A single click now generates one unified bundle containing both service logs and summary status.
- The bundles are stored securely in the object storage of the environment.
- A historical list of previously generated bundles are available for access.

### Guardrails to prevent submitting jobs that do not fit resource capacity

- CDE now automatically prevents execution of jobs that do not fit on the available resources.
- CDE takes into account Kubernetes and system reserved resources, daemonset utilized resources, and Spark overhead factors.
- The API returns an error with run failed to start: requested *[\*\*\*TYPE AND AMOUNT OF RESOURCE\*\*\*]* is more than *[\*\*\*THE MAXIMUM AMOUNT OF AVAILABLE RESOURCES OF THAT TYPE\*\*\*]* allocatable per cluster node
- You can either reduce the Spark executor and driver CPU and/or memory requirements, or deploy on a larger cluster.

### Notification email configuration can now be verified

When configuring the optional email alerts feature [Technical Preview] during virtual cluster creation, you can now verify the SMTP settings before creating the virtual cluster.

### Streamlined resource creation and re-use during job creation

You can now create a resource on the fly when creating a job. Alternatively, you can select from a list of existing resources, if any, to upload your application or DAG file. This promotes re-usability of project artifacts across jobs.

### Kubernetes update

CDE now supports K8s 1.21.

## December 21, 2021

This release (1.13.0-h1-b1) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud addresses the Log4j2 security vulnerability CVE-2021-44228.

### Fixed Log4j2 security vulnerability CVE-2021-44228

- Removed JndiLookup.class from affected JAR files.

For instructions on upgrading your existing CDE services and virtual clusters, see [Cloudera Data Engineering fix for CVE-2021-44228](#).

## November 9, 2021

This release (1.13) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the following changes.

### Non-transparent proxy support

- CDE supports deploying into CDP environments using a non-transparent proxy.
- The proxy is registered and enabled during CDE environment creation.
- The proxy configuration is automatically added to the deployed CDE service and virtual clusters (VCs).

### UI support for Python virtual environments

- You can now create custom Python resources on the CDE UI, including virtual environments (venvs)
- These custom venvs are selectable in the job creation wizard when creating PySpark jobs.

### Support for Airflow core operators

- With Airflow 2, Cloudera now supports all core operators.

### Support for Ranger Authorization Service

- CDE now supports Ranger Authorization Service (RAZ) in AWS and Azure environments.
- For more information, see [RAZ support requirements](#)

## Older releases

Overview of new features, enhancements, and changed behavior introduced in earlier releases of Cloudera Data Engineering.

## October 18, 2021

This release (1.12) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### Apache Airflow 2

With this release, Apache Airflow 2.1 is the new default managed scheduler in CDE. It comes with governance, security and compute autoscaling enabled out-of-the-box, along with integration with CDE's job management APIs giving users the flexibility to deploy custom DAGs that tap into Cloudera Data Platform (CDP) data services like Spark in CDE and Hive in CDW.

For more information on what's new in Airflow 2, see the [upstream documentation](#).

### [Technical Preview] Airflow pipeline authoring UI

With the CDE Pipeline Authoring UI, any CDE user irrespective of their level of Airflow expertise can create multi-step pipelines with a combination of out-of-the-box operators (CDEOperator, CDWOperator, BashOperator, PythonOperator). Nevertheless, you can still deploy your own customer Airflow DAGs (Directed Acyclic Graphs) as before, or use the Pipeline Authoring UI to bootstrap your projects for further customization.

This feature is in Technical Preview and available on new CDE services only. When creating a Virtual Cluster, a new option allows you to enable the Airflow Authoring UI.

For best user experience, Cloudera suggests using Google Chrome for this feature

### [Technical Preview] Email alerts

You can now configure email alerts during Virtual Cluster setup and schedule them in custom Airflow DAGs.

### Kubernetes update

CDE now supports K8s 1.20.

## August 30, 2021

This release (1.11) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### GA support for virtual clusters powered by Apache Spark 3

- Support for virtual clusters powered by Apache Spark 3 is no longer a Technical Preview feature, and is now generally available (GA).
- The following functionalities are not currently supported:
  - Deep analysis (visual profiler)
  - HWC - that is, Hive managed ACID tables (Direct Reader & JDBC mode)
  - Phoenix Connector
  - SparkR
  - Kudu

### [Technical Preview] Fully private AKS cluster set up

- Fully private AKS clusters are now supported, for customers who want to restrict resources from being exposed via public IP addresses. This allows securing the Kubernetes cluster even more, an AKS API server can be created with a private IP address which is only accessible to the resources which are running inside of the Azure virtual network (VNet).
- A private AKS is deployed within customers' network and leverages CCMv2/Proxy for accessing the K8s APIs.
- Cloudera recommends using one single resource group per environment. You can accomplish this by selecting a (pre-created) resource group during CDP environment creation.

### Gang scheduling enabled by default

- YuniKorn Gang scheduling policy is now enabled by default within CDE.
- For more information on Gang scheduling, see the [Spark on Kubernetes – Gang Scheduling with YuniKorn](#) Cloudera Blog post.

### [Technical Preview] User-specified IAM roles

- CDE job pods can now run with a user-specified IAM role with the role credentials automatically supplied as instance credentials. This allows transparent usage of cloud SDKs or any code making use of the instance credentials provider. User roles are secured and allocated through the CDP environment IDBroker mappings.
- This feature is available as a Technical Preview. Contact your Cloudera account representative to enable access to this feature.

### Spark Analysis disabled by default

- Metric collection from Spark jobs is now disabled by default to provide the most optimal performance.

- During development and testing, you can turn on additional Spark profiling:
  - On the CDE UI:  
After creating the job, go to its Configuration tab and toggle the Spark Analysis option.  
For more information, see [Managing jobs in Cloudera Data Engineering](#).
  - From CLI/API:  
Set the following configuration parameter during job creation: `dex.safariEnabled=true`  
For more information, see [Managing Cloudera Data Engineering jobs using the CLI](#) and [Creating a Cloudera Data Engineering job using the API](#) respectively.

## August 2, 2021

This release (1.9) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### Kubernetes version updated to 1.19

The Kubernetes version has been updated to 1.19.

### Dynamic allocation enabled by default

Dynamic allocation is now enabled by default. You can configure the initial number of executors as well as a range of executors per job.

Dynamic allocation scales job executors up and down as needed for running jobs. This can provide large performance benefits by allocating as many resources as needed by the running job, and by returning resources when they are not needed so that concurrent jobs can potentially run faster.

Resources are limited by the job configuration (executor range) as well as the virtual cluster auto-scaling parameters. By default, the executor range is set to match the range of CPU cores configured for the virtual cluster. This improves resource utilization and efficiency by allowing jobs to scale up to the maximum virtual cluster resources available, without manually tuning and optimizing the number of executors per job.

This is a change from the default behavior (static allocation) in older releases. If you restore job configuration from an older release, the restored jobs will use dynamic allocation.

### Support for Amazon AWS S3 URLs in jobs

A previous issue with S3 URLs in job configurations has been fixed. You can now specify S3 URLs for your application code and Jar files. For jobs using this functionality, you must also add the following Apache Spark configuration option:

```
spark.hadoop.fs.s3a.delegation.token.binding=org.apache.knox.gateway.cloud.idbroker.s3a.IDBDelegationTokenBinding
```

### On-demand Python virtual environments

You can now submit a job with a Python requirements.txt file, as follows:

```
cde spark submit my_job.py --python-requirements /path/to/requirements.txt
```

This builds the Python virtual environment resource for the user, attaches it to the job, and sets it to be cleaned up when the job run terminates.

## June 23, 2021

This release (1.8) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### Python virtual environment improvements

Python virtual environments are now built in dedicated pods, and support C-based Python libraries.

### Open source CDE/CDW operators for Apache Airflow

You can now use CDE and CDW operators with your existing Apache Airflow deployment. For more information and instructions, see [Using CDE with an external Apache Airflow deployment](#).

### CDE jobs in different virtual clusters within the same DAG file

Airflow DAG files can now trigger CDE jobs in different virtual clusters. For more information, see [Automating data pipelines using Apache Airflow in Cloudera Data Engineering](#).

### CIDR notation support for IP whitelist

You can now add IP ranges to the whitelist using CIDR notation.

### Subnet selection option

You can now select a subnet to use for CDE when enabling a CDE service.

## May 20, 2021

This release (1.7) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### GA support for Microsoft Azure

- Support for Microsoft Azure is no longer a *Technical Preview* feature, and is now generally available (GA).

### GA support for gang scheduling

- Gang scheduling is no longer a *Technical Preview* feature, and is now generally available (GA).
- For a detailed explanation on gang scheduling, see the [blog post](#).

### Diagnostic bundles

- The Diagnostic page has been enhanced to support granular log selection as well as a snapshot of the service and cluster status.

### [Technical Preview] Support for virtual clusters powered by Apache Spark 3

- You can now create virtual clusters powered by Apache Spark version 3. You cannot use Spark 2 and Spark 3 within the same virtual cluster, but you can have multiple Spark 2 and Spark 3 virtual clusters within the same CDE service.
- This feature is available as a *Technical Preview*. Contact your Cloudera account representative to enable access to this feature.

### [Technical Preview] Bin packing resource scheduling

- Bin packing is a new Apache YuniKorn resource management policy. Bin packing makes more efficient use of available nodes when assigning executors to hosts.
- This feature is available as a *Technical Preview*. Contact your Cloudera account representative to enable access to this feature.

### Automatic TLS certificate renewal

- Cluster TLS certificates are now renewed automatically.

## April 7, 2021

This release (1.6) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### API keys

- CDE users can now use API keys, managed using the CDP user management service (UMS), to interact with the CDE Jobs API using the command line.

### Raw Scala code

- Users can now submit jobs with raw Scala code, without compiling. These jobs run spark-shell to process the application file.

### Diagnostic bundles

- Admins can now access summary diagnostic logs directly on their local machine.
- A new Diagnostic page has been added to the CDE Service details to generate and download the bundle.

### Force TLS certificate renewal

- CDE services older than 90 days will have expired TLS certificates. A new action has been added to the CDE service hamburger menu to renew the certificates and avoid access issues for DE users.

### [Technical Preview] GANG scheduling

- GANG is a new resource scheduling policy that overcomes scale-up challenges in situations where high rates of job submission lead to queuing. The new scheduling policy moves jobs off the queue in batches. This clears up the queue, forces scale-up of nodes to process the burst of incoming jobs, and reduces wait and startup time of jobs.
- By default, GANG scheduling is disabled. It can be turned on for specific jobs by adding a new job-level configuration option.

## March 9, 2021

This release (1.5) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces technical preview support for deploying CDE on Microsoft Azure.

### [Technical Preview] Support for Microsoft Azure

- Cloudera Data Engineering can now be deployed on Microsoft Azure. This functionality is provided as a technical preview, and is not supported for production environments.

## February 4, 2021

This release (1.4) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### Support for Airflow DAGs

- DE and ML practitioners can now define their own pipelines packaged as an Apache Airflow Python DAG. Currently supported CDP operators include running Spark jobs on CDE and Hive jobs on CDW.
- An embedded Airflow UI within the job & job run details pages gives users a “deep link” to the specific Airflow DAG making it easier to access within the context of the job runs.
- The Schedule page has been removed from the left panel of the virtual cluster jobs UI, and the full Airflow UI is now exposed through the Virtual Cluster details page.



### Improved service observability for service troubleshooting

Diagnostic bundles can now be collected through a new API end-point, which includes:

- Cloud resource status: (EKS, RDS, EFS, ELB)
- Helm status( helm version, helm ls -A)
- Kubernetes status (deployments, pods, services, ingresses, config maps)

### Virtual Cluster user-based ACL

- By default a Virtual Cluster is accessible to all DEUsers and DEAdmins, which includes the Jobs API, Airflow UI, along with any connections and credentials defined within Airflow.
- Enabling access control will now limit access to the API and UIs of the Virtual Cluster to a subset of users - normal or machine users. Groups are not yet supported.

### Kubernetes support

- CDE now supports EKS 1.18

## December 21, 2020

This release (1.3) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### Better scaling

- Larger compute instance types to accommodate heavier ETL workloads and tweaks to infra services for higher job submission API throughput.
- Better Spark defaults tuned for Kubernetes, improving scale up and stability.

### Run profiling analysis on demand for additional tuning metrics

Users can trigger additional profiling analysis for any job, providing memory and CPU utilization, and stage-level CPU flamegraphs.

### Workload Manager integration

CDE services can now share Spark application metadata and metrics with Workload Manager for better visibility into aggregate workloads across the entire CDP environment, manage SLAs, and identify additional tuning opportunities.

### Easy log configuration & access

- Full logs can now be downloaded as a new download option along with a quick bookmark to the S3 location of the Spark application logs.
- The UI now supports setting the log level from OFF to DEBUG and TRACE.

### Lightweight backup and restore

You can use the CDE CLI & API to backup and restore jobs from one virtual cluster into another virtual cluster within the same CDE service or completely new one. Helps support upgrades as it requires enabling a new service.

### Create/clone job from existing run for easier debugging

CLI now supports cloning job runs, carrying over configs and parameters, making it easier to troubleshoot runs with failures.

### [Tech Preview] Self-authored Airflow DAGs

DE and ML practitioners can now define their own pipelines packaged as an Apache Airflow Python DAG. Currently supported CDP operators include running Spark jobs on CDE and Hive jobs on CDW.

## November 9, 2020

This release (1.2) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### CDP CLI Integration

Administrators can now automate the enabling of CDE services and creation of Virtual Clusters through CDP CLI. Jobs will continue to be managed through the CDE CLI shipped with the service.

### Multiple CDE Services

It's now easier to enable CDE service multiple times within the same environment (datalake/SDX). This allows admins to set up multiple CDE services with differing instance profiles and allows for easier consumption tracking through AWS tags at the service level.

### Python virtual environments

Users can now specify a list of python libraries as dependencies for Pyspark jobs. This can be specified through a requirements.txt file that is uploaded and managed through CLI/API.

### CDP Trial Tours

The first trial tour for Data Engineering admins is now available.

## September 21, 2020

This release (1.1) of the Cloudera Data Engineering (CDE) service on CDP Public Cloud introduces the new features and improvements that are described in this topic.

### Spot instance support

Users now have an additional knob to control cost, by being able to choose between running on Spot or On-Demand instances, providing up to 90% discount in AWS resources.

### Local (native) SSD

For memory and shuffle heavy workloads, CDE now allows using instances with local (native) SSD for intermediate results boosting performance.

### Resource tags

Allows administrators to define tags during CDE service creation to track and audit cloud provider resources.

### IP Whitelisting

Administrators can lock down access to the EKS control plane components via a CIDR range.

### Stability & Security

New CDE service deployment will now use Kubernetes 1.15 and Helm 3; this improves the stability and security of the service moving forward

**July 30, 2020**

This release (1.0) marks the General Availability (GA) release of the Cloudera Data Engineering (CDE) service on CDP Public Cloud.

## Cloudera Data Engineering fix for CVE-2021-44228

On December 21, 2021, Cloudera [released](#) Cloudera Data Engineering (CDE) for Public Cloud version 1.13.0-h1-b1. It addresses the Log4j2 security vulnerabilities listed below. Cloudera urges all customers to upgrade their Data Engineering services to the latest version.

- [CVE-2021-44228](#) which affects Apache Log4j2 versions 2.0 through 2.14.1.
- [CVE-2021-45046](#) which affects Apache Log4j2 version 2.15.0

### Upgrade to the latest Data Engineering version

To upgrade your Cloudera Data Engineering service to the latest version, which addresses the log4j2 security vulnerability, follow these steps. These steps provide a comprehensive upgrade and are the recommended approach. To ensure compatibility with the CDP environment, you must also [upgrade the environment DataLake](#) to Runtime 7.2.11 or higher.

1. For each existing virtual cluster, [create a backup](#) of the jobs and resources.
2. [Enable a new Cloudera Data Engineering service](#) for each existing CDE service you have. Make sure to use the same settings as the existing service for each new corresponding service.
3. Within each new Data Engineering service, [create a new virtual cluster](#) for each existing virtual cluster in the pre-existing service. Make sure to use the same settings as the existing virtual cluster for each new corresponding virtual cluster.
4. After making sure that you have added CDE services and virtual clusters to match your existing deployment, [restore the backup file](#) for each pre-existing virtual cluster to the corresponding new virtual cluster.

Result

Your Cloudera Data Engineering service and all associated virtual clusters are upgraded to the latest version.

## Known issues and limitations in Cloudera Data Engineering Public Cloud

This section lists known issues and limitations that you might run into while using the Cloudera Data Engineering (CDE) service in CDP Public Cloud.

### General known issues with Cloudera Data Engineering

Learn about the general known issues with the Cloudera Data Engineering (CDE) service on public clouds, the impact or changes to the functionality, and the workaround.

#### **DEX-12616: Node Count shows zero in /metric request**

Cloudera Data Engineering (CDE) 1.20.3 introduced compatibility with Kubernetes version 1.27. With this update, the kube\_state\_metrics no longer provides label and annotation metrics by default.

Earlier, CDE used label information to calculate the Node Count for both Core and All-Purpose nodes, which was automatically exposed. However, due to the changes in kube\_state\_metrics, this functionality is no longer available by default. As a result, the Node count shows zero in /metrics, charts, and the user interface.

1. Make sure that you set a kube-config for the CDE service.

- Run the following command, which exposes the label manually in the prometheus-kube-state-metrics container:

```
`kubectl patch deployment monitoring-prometheus-kube-state-metrics --namespace monitoring --type='json' -p='{
  "[{"op": "add",
    "path": "/spec/template/spec/containers/0/args", "value": ["--metric-labels-allowlist=nodes=[role]"]}]'"`
```

For more information about how to see the node count, see [Checking the node count on your Cloud Service Provider's website](#).

#### **DEX-11340: Kill all the alive sessions in prepare-for-upgrade phase of stop-gap solution for upgrade**

If Spark sessions are running during the CDE upgrade, they are not automatically killed, leaving them in an unknown state during and after the upgrade.

Ensure that you do not have any Spark sessions running during the CDE upgrade. If they are running during the CDE upgrade, kill them before proceeding.

#### **DEX-14094: New pipeline editor can create a cde\_job step of its own pipeline job which causes recursion and looping**

If you add a CDE Job step in the pipeline editor and select the same job as the pipeline job from the Select Job drop-down list while configuring the pipeline job using the editor, then running the pipeline job results in a recursive loop.

For example, You have created a pipeline job named test-dag and selected the same job test-dag from the Select Job drop-down list while adding the CDE Job step, then running the pipeline job results in a recursive loop.

#### **DEX-14084: No error response for Airflow Python virtual environment at Virtual Cluster level for view only access user**

If a user with a view only role on a Virtual Cluster (VC) tries to create an Airflow Python virtual environment on a VC, the access is blocked with a 403 error. However, the no-access 403 error is not displayed on the UI.

#### **DEX-13465: CDE Airflow DAG Code Fetch is not working**

The embedded Airflow UI within the CDE Job pages does not correctly show the "Code" view.

Open the Jobs page through Virtual Cluster link and use the old CDE UI or manually open Airflow UI through Virtual Cluster details page and navigate to the appropriate DAG details.

#### **DEX-11639: "CPU" and "Memory" Should Match Tier 1 and Tier 2 Virtual Clusters AutoScale**

CPU and Memory options in the service or cluster edit page display the values for Core (tier 1) and All-Purpose (tier 2) together. However, they must be separate values for Core and All-Purpose.

#### **DEX-12482: [Intermittent] Diagnostic Bundle generation taking several hours to generate**

Diagnostics bundles can intermittently take very long to get generated due to low EBS throughput and IOPS of the base node.

Increase the EBS throughput and IOPS values in the CloudFormation template, then trigger new diagnostic bundles.



**Note:** Any existing job runs will fail when the CloudFormation template is changed as a new node comes up and the existing one is terminated.

#### **DEX-14253: CDE Spark Jobs are getting stuck due to the unavailability of the spot instances**

The unavailability of AWS spot instances may cause CDE Spark jobs to get stuck.

Re-create the Virtual Cluster with on-demand instances.



**Note:** Cloudera does not recommend using spot instances for creating VCs.

#### DEX-14192: Some Spark 3.5.1 jobs have slightly higher memory requirements

Some jobs running on Spark 3.5.1 have slightly higher memory requirements, resulting in the driver pods getting killed with a k8s OOMKilled.

Increase the driver pod memory from the default 1GB to 1.2GB in the job's configuration.

#### DEX-14173: VC Creation is failing with "Helm error: 'timed out waiting for the condition', no events found for chart"

In case of busy k8s clusters, installing VC/CDE may fail with an error message showing Helm error: 'timed out waiting for the condition', no events found for chart.

Try installing again. The failure is due to image pulls timing out. The installation will go through as more resources become available.

#### DEX-14067: Email alerting is not sending email to the recipient even after the job fails

When a Virtual Cluster is created through the UI, SMTP configuration and authentication is empty. Enabling Email Alerting after VC creation allows users to add the SMTP parameters, but the updated password is not fetched when sending email alerts.

Restart the runtime-api pod after adding the SMTP configuration. On restart the secrets are re-fetched and all services are reinitialized with the updated secrets.

1. List all pods in the namespace (for example, dex-app-dtznsgc5):

```
kubectl get pods -n [***NAMESPACE***]
```

2. Copy the runtime-api pod name. (For example, dex-app-dtznsgc5-api-b5f8d7cb9-gdbxb.)
3. Delete the runtime-api pod:

```
kubectl delete pod [***POD NAME***] -n [***NAMESPACE***]
```

4. A new instance of the runtime-api pod will get initialized.
5. Test email alerting. (For example, via a job run failure.)

#### DEX-14027: Spark 3.5.1 jobs are failing with error 'org.apache.hadoop.fs.s3a.impl.InstallationIOException'

Spark 3.5.1 RAZ-enabled CDE clusters fail to initialize RAZ S3 plugin library due to recent backward incompatible changes in the library, and jobs fail with error org.apache.hadoop.fs.s3a.impl.InstallationIOException.

Depending on the configuration level, multiple following workarounds are available through the API or CLI, since setting empty parameters in the UI is not possible.

- (preferred) as a VC-level config setting (not applicable to CDE sessions), with the API, use:

```
curl '[***CLUSTER APP INSTANCE API ENDPOINT***]'; \
-X 'PATCH' \
-H 'Connection: keep-alive' \
-H 'Content-Type: application/json' \
-H 'Cookie: [***COOKIE CONTAINING CDP-TOKEN***]' \
--data-raw '{"config":{"sparkConfigs":{"spark.hadoop.fs.s3a.http.signer.class":"org.apache.ranger.raz.hook.s3.RazS3SignerPlugin","spark.hadoop.fs.s3a.http.signer.enabled":"true","spark.hadoop.fs.s3a.custom.signers":"","spark.hadoop.fs.s3a.s3.signing.algorithm":""}}}'
```

- as a per session configuration (VC-level configuration will not be applied), in the CLI, run:

```
./cde --vcluster-endpoint [***CLUSTER APP INSTANCE API ENDPOINT***] session create
```

```
--name [***CDE SESSION NAME***] \
--conf "spark.hadoop.fs.s3a.custom.signers=" \
--conf "spark.hadoop.fs.s3a.http.signer.class=org.apache.range
r.raz.hook.s3.RazS3SignerPlugin" \
--conf "spark.hadoop.fs.s3a.http.signer.enabled=true" \
--conf "spark.hadoop.fs.s3a.s3.signing-algorithm="
```

- in the CLI, at job creation, run:

```
./cde --vcluster-endpoint [***VIRTUAL CLUSTER JOB API URL***]
job run --name [***CDE JOB NAME***]> \
--conf "spark.hadoop.fs.s3a.custom.signers=" \
--conf "spark.hadoop.fs.s3a.http.signer.class=org.apache.rang
er.raz.hook.s3.RazS3SignerPlugin" \
--conf "spark.hadoop.fs.s3a.http.signer.enabled=true" \
--conf "spark.hadoop.fs.s3a.s3.signing-algorithm="
```



**Note:** spark.hadoop.fs.s3a.custom.signers and spark.hadoop.fs.s3a.s3.signing-algorithm needs to be set to empty to unset them.

#### DEX-13975: 'spark.catalog.listTables()' command in job is failing in Python Spark for Spark 3.5.1

Using catalog.listTables() with Iceberg tables results in an exception org.apache.spark.sql.catalyst.parser.ParseException: [PARSE\_SYNTAX\_ERROR] Syntax error at or near end of input.

Set spark.sql.legacy.useV1Command=true for running catalog.listTables().

#### DEX-13957: CDE metrics and graphs show no data

CDE versions 1.20.3 and 1.21 use Kubernetes version 1.27. In Kubernetes version 1.27, by default, the kube\_state\_metrics does not provide label and annotation metrics. For this reason, the node count shows zero for core and all-purpose nodes in the CDE UI and in charts.

As a prerequisite, set a kube-config for the CDE service and run:

```
kubectrl patch deployment monitoring-prometheus-kube-state-metrics
--namespace monitoring --type='json' -p=\
'[{ "op": "add",
"path": "/spec/template/spec/containers/0/args", "value": [ "--metr
ic-labels-allowlist=nodes=[role]" ] } ]'
```

#### DEX-12630: CDE Service failed to run jobs on SSD Based clusters

When a customer creates a new CDE service with an SSD Instance enabled on CDE version greater than or equal to 1.19.4, Spark and Airflow jobs do not start at all. The same problem happens if an existing CDE service is upgraded to 1.19.3 or greater and has SSD Instance enabled.

Create a new CDE service without SSD Instance enabled until 1.20.3-h1. From 1.20.3-h2, you can create a new CDE service with SSD Instance enabled. However, you cannot upgrade existing SSD based service, therefore you must create a new one.

#### DEX 12451: Service creation fails when "Enable Public Loadbalancer" is selected in an environment with only private subnets

When creating a service with the Enable Public Load Balancer option selected, the service creation fails with the following error:

```
"CDE Service: 1.20.3-b15 ENV: dsp-storage-mow-priv (in mow-priv)
and dsp-storage-aws-dev-newvpc (mw-dev) - Environment is config
ured only with private subnets, there are no public subnets. dex
-base installation failed, events: [{ "ResourceKind": "Service", "
ResourceName": "dex-base-nginx-56g288bq-controller", "ResourceName
space": "dex-base-56g288bq", "Message": "Error syncing load balance
r: failed to ensure load balancer: could not find any suitable s
```

```
ubnets for creating the ELB", "Timestamp": "2024-02-08T09:55:28Z" }
]
```

When creating a service and enabling a public load balancer, configure at least one public subnet in the environment. For more information, see [Enabling a Cloudera Data Engineering service](#).

**DEX 11498: Spark job failing with error: "Exception in thread "main" org.apache.hadoop.fs.s3a.AWSBadRequestException:"**

When users in Milan and Jakarta region use Hadoop s3a client to access AWS s3 storage, that is using s3a://bucket-name/key to access the file, an error may occur. This is a known issue in Hadoop.

Set the region manually as: spark.hadoop.fs.s3a.endpoint.region=<region code> . For region codes, see [https://docs.aws.amazon.com/general/latest/gr/rande.html#s3\\_region](https://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region).

**ENGESC-22921: Livy pod failure (CrashLoopBackoff)**

There is not enough Livy-overhead-memory causing the Livy service to crash and trigger the pod to restart.

Increase the readiness/liveness timeout to 60 seconds and Livy pod will start.

**DEX-11086: Cancelled statement(s) not canceled by Livy**

Currently, Livy statements can't be cancelled immediately after using /sessions/{name}/statements/{id}/cancel. The status is returned as Cancelled but the background job continues to run.

There is a limitation on what can be cancelled. For example, if something is running on the driver exclusively, such as Thread.sleep(), it can not be cancelled.

**DEX-9939: Tier 2 Node groups are created with the same size as Tier 1 node groups during service creation. They cannot be edited during service edit**

If a service is created with 1 as the minimum on-demand scale limit, two nodes will run for Tier 1 and Tier 2. Even if a service is edited with the minimum reduced to 0, the Tier 2 node will still run. This will be fixed in the CDE 1.20 release.

You must manually edit the node group parameters from the AWS or Azure console. First, locate the log titled "Started provisioning a managed cluster, provisionerID: liftie-xxxxxxx" and locate the Liftie ID for the cluster so that you can continue with the steps below.

In AWS:

1. Log in to the AWS Management Console.
2. Navigate to EC2 > Auto Scaling groups.
3. Find <liftie-id>-spt2-<hash>-NodeGroup and click on the name to open the Instance Group Details page.
4. Under Group Details, click Edit.
5. Update Maximum capacity from 5 to 10, and click Update.
6. Repeat the steps above to update the Maximum capacity of the <liftie-id>-cmp2-5-<hash>-NodeGroup from 2 to 5. The Cluster Autoscaler implements the changes and the issue will resolve. The number of simultaneously running jobs will increase.

In Azure:

1. Navigate to the Kubernetes Service Details > Log page.
2. Navigate to the Node Pools page, and locate the Node Pool starting with cmp2.
3. Click on Scale Node pools and edit the capacity.

**DEX-9852: FreeIPA certificate mismatch issues for new Spark 3.3 Virtual Clusters**

In CDE 1.19, when creating a new Virtual Cluster based on Spark 3.3, and submitting any job in the pods, the following error occurs: "start TGT gen failed for user : rpc error: code = Unavailable desc = Operation not allowed while app is in recovery state."

Manually copy over the certificate from the FreeIPA server.

**DEX-10147: Grafana issue for virtual clusters with the same name**



In CDE 1.19, when you have two different CDE services with the same name under the same environment, and you click the Grafana charts for the second CDE service, metrics for the Virtual Cluster in the first CDE service will display.

After you have upgraded CDE, you must verify other things in the upgraded CDE cluster except the data shown in Grafana. Once you have verified everything in the new upgraded CDE service, the old CDE service should be deleted and the Grafana issue will be fixed.

**DEX-9932: Name length causes Pod creation error for CDE Sessions**

In CDE 1.19, the K8s pod name has a limitation of 63 Characters, and CDE Sessions has a name length of 56 maximum characters.

Create a CDE Session with a name of less than 56 characters to avoid the pod creation error.

**DEX-9895: CDE Virtual Cluster API response displays default Spark version as 2.4.7**

In CDE 1.19, the Spark version 3.2.3 is the expected default in a CDE Spark Virtual Cluster, but Spark 2.4.7 displays instead. This issue will be fixed in CDE 1.20.

**DEX-9112: VC deployment frequently fails when deployed through the CDP CLI**

In CDE 1.19, when a Virtual Cluster is deployed using the CDP CLI, it fails frequently as the pods fail to start. However, creating a Virtual cluster using the UI is successful.

Ensure that you are using proper units to --memory-requests in "cdp de" CLI, for example "--memory-requests 10Gi".

**DEX-9790: Single tab Session support for Virtual Cluster selection**

In CDE 1.19, the Virtual Cluster selection in the Jobs, Resources, Runs, and Sessions page is not preserved if the user attempts to open CDE in another browser tab/window.

When you open CDE in another tab, you must re-select the Virtual Cluster that you want to use in the new tab.

**DEX-10044: Handle adding tier 2 auto scaling groups during in-place upgrades**

Since auto scaling groups (ASGs) are not added or updated during the upgrade, the tier 2 ASGs are not created. This resulted in pods that were unable to be scheduled. This error applies to services created in CDE 1.18 and then upgraded to 1.19.

Create a new CDE service as this issue won't be seen on a new CDE 1.19 service because this error applies only to upgraded clusters.

**DEX-10107: Spark 3.3 in CDE 1.19 has a limitation of characters for job names**

Jobs with longer names over 23 characters can fail in Spark 3.3 with the following exception: 23-05-14 10:14:16 ERROR ExecutorPodsSnapshotsStoreImpl: Going to stop due to IllegalArgumentException Exception java.lang.IllegalArgumentException: '\$JOB\_NAME' in spark.kubernetes.executor.podNamePrefix is invalid. must conform <https://kubernetes.io/docs/concepts/overview/working-with-objects/names/#dns-label-names> and the value length <= 47

Change the name of the job:

1. Clone the job with a new name using the CDE UI, CLI, or API.
2. Set the app name in the job itself: `conf.setAppName("Custom Job Name")`.

**DEX-10055: Interacting with a killed session**

When you interact with a long-running killed Spark session, the session might become unresponsive. Refrain from interacting with the long-running killed session. This will be fixed in a future release of CDE.

**DEX-9879: Infinite while loops not working in CDE Sessions**

If an infinite while loop is submitted as a statement, the session will be stuck infinitely. This means that new sessions can't be sent and the Session stays in a busy state. Sample input:

```
while(True) {  
  print("hello")  
}
```



```
}

```

1. Copy and use the DEX\_API that can be found on the Virtual Cluster details page to cancel the statement: `POST $DEX_API/sessions/{session-name}/statements/{statement-id}/cancel`. The Statement ID can be found by running the `cde sessions statements` command from the CDE CLI.
2. Kill the Session and create a new one.

#### DEX-9898: CDE CLI input reads break after interacting with a Session

After interacting with a Session through the `sessions interact` command, input to the CDE CLI on the terminal breaks. In this example below, `^M` displays instead of proceeding:

```
> cde session interact --name sparkid-test-6
WARN: Plaintext or insecure TLS connection requested, take care
before continuing. Continue? yes/no [no]: yes^M
```

Open a new terminal and type your CDE commands.

#### DEX-9881: Multi-line command error for Spark-Scala Session types in the CDE CLI

In CDE 1.19, Multi-line input into a Scala session on the CDE CLI will not work as expected, in some cases. The CLI interaction will throw an error before reading the complete input. Sample input:

```
scala> type
|
```

Use the UI to interact with Scala sessions. A newline is expected in the above situation. In CDE 1.19, only unbalanced brackets will generate a new line. In CDE 1.20, all valid Scala newline conditions will be handled:

```
scala> customFunc(
|  (
|  )
|  )
|
```

#### DEX-9756: Unable to run large raw Scala jobs

Scala code with more than 2000 lines could result in an error.

To avoid the error, increase the stack size. For example, `"spark.driver.extraJavaOptions=-Xss4M"`, `"spark.driver.extraJavaOptions=-Xss8M"`, and so forth.

#### DEX-8679: Job fails with permission denied on a RAZ environment

When running a job that has access to files is longer than the delegation token renewal time on a RAZ-enabled CDP environment, the job will fail with the following error:

```
Failed to acquire a SAS token for get-status on ../../words.txt d
ue to org.apache.hadoop.security.AccessControlException: Permiss
ion denied.
```

#### DEX-8769: The table entity type on Atlas is `spar_tables` instead of `hive_tables` on Spark3 Virtual Clusters

Tables that are created using a Spark3 Virtual Cluster on an AWS setup will have `spark_tables` type instead of `hive_tables` on Atlas Entities.

On a Spark3 Virtual Cluster, `enableHiveSupport()` must be called in the following way: `spark = SparkSession.builder.enableHiveSupport().getOrCreate()` You may also use Spark2 in lieu of Spark3 as this issue does not occur in Spark2.

#### DEX-8774: Job and run cloning is not fully supported in CDE 1.17 through 1.18.1

Currently, cloning job and runs are not supported in CDE 1.17 through 1.18.1.

Clone jobs and run operations by navigating to the Administration page, clicking View Jobs on the respective Virtual Cluster.

**DEX-3706: The CDE home page not displaying for some users**

The CDE home page will not display Virtual Clusters or a Quick Action bar if the user is part of hundreds of user groups or subgroups.

The user must access the Administration page and open the Virtual Cluster of choice to perform all Job-related actions. This issue will be fixed in CDE 1.18.1

**DEX-8515: The Spark History Server user interface is not visible in CDE**

During job execution in CDE 1.18, the Spark History Server user interface is not visible. This error will be fixed in CDE 1.18.1.

**DEX-6163: Error message with Spark 3.2 and CDE**

For CDE 1.16 through 1.18, if you experience an error message, "Service account may have been revoked" with Spark 3.2 and CDE, note that this is not the core issue despite what the error message states. Look for other exceptions as it is a harmless error and only displays after a job fails due to another reason. The error message displays as part of the shutdown process. This issue will be fixed in CDE 1.18.1.

**DEX-7653: Updating Airflow Job/Dag file throws a 404 error**

A 404 error occurs when you update an Airflow Job/Dag file with a modified DAG ID or name when you initiate the following steps:

1. Create an Airflow job using a Simple Dag file. Use the Create Only option.
2. Edit the Airflow Job and delete the existing DAG file.
3. Upload the same DAG file with a modified DAG ID and Name from it's content.
4. Choose a different Resource Folder.
5. Use the Update and Run option.

The 404 error occurs.

To avoid this issue, ensure that you do not modify the DAG ID in step 3. If you must change your DAG ID in the dag file, then create a new file.

This issue will be fixed in CDE 1.18.1.

**DEX-8283: False Positive Status is appearing for the Raw Scala Syntax issue**

Raw Scala jobs that fail due to syntax errors are reported as succeeded by CDE as displayed in this example:

```
spark.range(3)..show()
```

The job will fail with the following error and will be logged in the driver stdout log:

```
/opt/spark/optional-lib/exec_invalid.scala:3: error: identifier
expected but '.' found.
  spark.range(3)..show()
                ^
```

This issue will be fixed in CDE 1.18.1.

**DEX-8281: Raw Scala Scripts fail due to the use of the case class**

Implicit conversions which involve implicit Encoders for case classes, that are usually supported by importing `spark.implicits._`, don't work in Raw Scala jobs in CDE. These include converting Scala objects, including RDD Dataset DataFrame, and Columns. For example, the following operations will fail on CDE:

```
import org.apache.spark.sql.Encoders
import spark.implicits._
```

```
case class Case(foo:String, bar:String)

// 1: an attempt to obtain schema via the implicit encoder for case class fails
val encoderSchema = Encoders.product[Case].schema
encoderSchema.printTreeString()

// 2: an attempt to convert RDD[Case] to DataFrame fails
val caseDF = sc
  .parallelize(1 to 3)
  .map(i => Case(f"$i", "bar"))
  .toDF

// 3: an attempt to convert DataFrame to Dataset[Case] fails
val caseDS = spark
  .read
  .json(List(List("{\"foo\":\"1\",\"bar\":\"2\"}")).toDS)
  .as[Case]
```

Whereas conversions that involve implicit encoders for primitive types are supported:

```
val ds = Seq("I am a Dataset").toDS
val df = Seq("I am a DataFrame").toDF
```

Notice that List, Row, StructField, and createDataFrame are used below instead of case class and .toDF():

```
val bankRowRDD = bankText.map(s => s.split(";")).filter(s => s(0) != "\"age\"").map(
  s => Row(
    s(0).toInt,
    s(1).replaceAll("\"", ""),
    s(2).replaceAll("\"", ""),
    s(3).replaceAll("\"", ""),
    s(5).replaceAll("\"", "").toInt
  )
)
val bankSchema = List(
  StructField("age", IntegerType, true),
  StructField("job", StringType, true),
  StructField("marital", StringType, true),
  StructField("education", StringType, true),
  StructField("balance", IntegerType, true)
)

val bank = spark.createDataFrame(
  bankRowRDD,
  StructType(bankSchema)
)

bank.registerTempTable("bank")
```

#### **DEX-7001: When Airflow jobs are run, the privileges of the user who created the job is applied and not the user who submitted the job**

If you have an Airflow job (created by User A) that contains Spark jobs, and the Airflow job is run by another user (User B), the Spark jobs are run as User A instead of the user who ran it. Regardless of who submits the Airflow job, the Airflow job is run with the user privileges of the user who created the job. This causes issues when the job submitter has lesser privileges than the job owner who has higher privileges. We recommend that the Spark and Airflow jobs must be created and run by the same user.

**CDPD-40396 Iceberg migration fails on partitioned Hive table created by Spark without location**

Iceberg provides a migrate procedure to migrate a Parquet/ORC/Avro Hive table to Iceberg. If the table was created using Spark and the location is not specified, and is partitioned, the migration fails.

If you are using Data Lake 7.2.15.2 or higher, the above known-issue will not occur. Otherwise, you'll need to unset the default table property of 'TRANSLATED\_TO\_EXTERNAL' from 'true' by completing the following:

1. Run 'ALTER TABLE ... UNSET TBLPROPERTIES ('TRANSLATED\_TO\_EXTERNAL') to unset the property.
2. Run the migrate procedure.

**DEX-5857 Persist job owner across CDE backup restores**

Currently, the user who runs the cde backup restore command has permissions, by default, to run the Jobs. This may cause CDE jobs to fail if the workload user differs from the user who runs the Jobs on Source CDE Service where the backup was performed. This failure is due to the Workload User having different privileges as the user who is expected to run the job.

Ensure that the cde job restore command is performed by the same user who is running the CDE Jobs in the Source CDE Cluster where the backup was performed. Additionally, you can ensure the User running the Restore has the same set of Permission as the User running the Job in Source CDE Cluster where the Backup was performed.

**DEX-7483 User interface bug for in-place upgrade (Tech Preview)**

The user interface incorrectly states that the Data Lake version 7.2.15 and above is required. The correct minimum version is 7.2.14.

**DEX-6873 Kubernetes 1.21 will fail service account token renewal after 90 days**

Cloudera Data Engineering (CDE) on AWS running version CDE 1.14 through 1.16 using Kubernetes 1.21 will observe failed jobs after 90 days of service uptime.

Restart specific components to force regenerate the token using one of the following options:

Option 1) Using kubectl:

1. [Setup kubectl for CDE](#).
2. Delete *calico-node* pods.

```
kubectl delete pods --selector k8s-app=calico-node --namespace kube-system
```

3. Delete Livy pods for all Virtual Clusters.

```
kubectl delete pods --selector app.kubernetes.io/name=livy --all-namespaces
```

If for some reason only one Livy pod needs to be fixed:

- a. Find the virtual cluster ID through the UI under Cluster Details.
- b. Delete Livy pod:

```
export VC_ID=<VC ID>
kubectl delete pods --selector app.kubernetes.io/name=livy --namespace ${VC_ID}
```

Option 2) Using K8s dashboard

1. On the Service Details page copy the RESOURCE SCHEDULER link.
2. Replace yunikorn part with the dashboard and open the resulting link in the browser.
3. In the top left corner find the namespaces dropdown and choose All namespaces.

4. Search for *calico-node*.
5. For each pod in the Pods table click the Delete option from the hamburger menu.
6. Search for *livy*.
7. For each pod in the Pods table click the Delete option from the hamburger menu.
8. If for some reason only one Livy pod needs to be fixed, find the Virtual Cluster ID through the UI under Cluster Details and only delete the pod with the name starting with Virtual Cluster ID.

#### DEX-7286 In place upgrade (Technical Preview) issue: Certificate expired showing error in browser

Certificates failure after an in-place upgrade from 1.14.

Start the certificate upgrade:

Get cluster ID

1. Navigate to the Cloudera Data Engineering Overview page by clicking the Data Engineering tile in the Cloudera Data Platform (CDP) management console.
2. Edit device details.
3. Copy cluster ID filed into click board.
4. In a terminal set the CID environment variable to this value.

```
export CID=cluster-1234abcd
```

Get session token

1. Navigate to the Cloudera Data Engineering Overview page by clicking the Data Engineering tile in the Cloudera Data Platform (CDP) management console.
2. Right click and select Inspect
3. Click the Application tab.
  4. Click Cookies and select the URL of the console.
  5. Select cdp-session-token.
  6. Double click the displayed cookie value and right click and select Copy.
  7. Open a terminal screen.

```
export CST=<Paste value of cookie here>
```

Force TLS certificate update

```
curl -b cdp-session-token=${CST} -X 'PATCH' -H 'Content-Type: application/json' -d '{"status_update": "renewTLSCerts"}' 'https://<URL OF CONSOLE>/dex/api/v1/cluster/$CID'
```

#### DEX-7051 EnvironmentPrivilegedUser role cannot be used with CDE

The role EnvironmentPrivilegedUser cannot currently be used by a user if a user wants to access CDE. If a user has this role, then this user will not be able to interact with CDE as an "access denied" would occur.

Cloudera recommends to not use or assign the EnvironmentPrivilegedUser role for accessing CDE.

#### CDPD-40396 Iceberg migration fails on partitioned Hive table created by Spark without location

Iceberg provides a migrate procedure for migrating a Parquet/ORC/Avro Hive table to Iceberg. If the table was created using Spark without specifying location and is partitioned, the migration fails.

By default, the table has a TRANSLATED\_TO\_EXTERNAL property and that is set to true. Unset this property by running ALTER TABLE ... UNSET TBLPROPERTIES ('TRANSLATED\_TO\_EXTERNAL') and then run the migrate procedure.

#### Strict DAG declaration in Airflow 2.2.5

CDE 1.16 introduces Airflow 2.2.5 which is now stricter about DAG declaration than the previously supported Airflow version in CDE. In Airflow 2.2.5, DAG timezone should be a `pendulum.tz.Timezone`, not `datetime.timezone.utc`.

If you upgrade to CDE 1.16, make sure that you have updated your DAGs according to the [Airflow documentation](#), otherwise your DAGs will not be able to be created in CDE and the restore process will not be able to restore these DAGs.

Example of valid DAG:

```
import pendulum
dag = DAG("my_tz_dag", start_date=pendulum.datetime(2016, 1, 1,
    tz="Europe/Amsterdam"))
op = DummyOperator(task_id="dummy", dag=dag)
```

Example of invalid DAG:

```
from datetime import timezone
from dateutil import parser
dag = DAG("my_tz_dag", start_date=parser.isoparse('2020-11-11T
    20:20:04.268Z').replace(tzinfo=timezone.utc))
op = DummyOperator(task_id="dummy", dag=dag)
```

#### COMPX-5494: Yunikorn recovery intermittently deletes existing placeholders

On recovery, Yunikorn may intermittently delete placeholder pods. After recovery, there may be remaining placeholder pods. This may cause unexpected behavior during rescheduling.

There is no workaround for this issue. To avoid any unexpected behavior, Cloudera suggests removing all the placeholders manually before restarting the scheduler.

#### DWX-8257: CDW Airflow Operator does not support SSO

Although Virtual Warehouse (VW) in Cloudera Data Warehouse (CDW) supports SSO, this is incompatible with the CDE Airflow service as, for the time being, the Airflow CDW Operator only supports workload username/password authentication.

Disable SSO in the VW.

#### COMPX-7085: Scheduler crashes due to Out Of Memory (OOM) error in case of clusters with more than 200 nodes

Resource requirement of the YuniKorn scheduler pod depends on cluster size, that is, the number of nodes and the number of pods. Currently, the scheduler is configured with a memory limit of 2Gi. When running on a cluster that has more than 200 nodes, the memory limit of 2Gi may not be enough. This can cause the scheduler to crash because of OOM.



**Important:** This change requires K8s cluster access.

Increase resource requests and limits for the scheduler. Edit the YuniKorn scheduler deployment to increase the memory limit to 16Gi.

For example:

```
resources:
  limits:
    cpu: "4"
    memory: 16Gi
  requests:
    cpu: "2"
    memory: 8Gi
```

#### COMPX-6949: Stuck jobs prevent cluster scale down

Because of hanging jobs, the cluster is unable to scale down even when there are no ongoing activities. This may happen when some unexpected node removal occurs, causing some pods to be stuck in Pending state. These pending pods prevent the cluster from downscaling.

Terminate the jobs manually.

#### **DEX-3997: Python jobs using virtual environment fail with import error**

Running a Python job that uses a virtual environment resource fails with an import error, such as:

```
Traceback (most recent call last):
  File "/tmp/spark-826a7833-e995-43d2-bedf-6c9dbd215b76/app.py",
line 3, in <module>
    from insurance.beneficiary import BeneficiaryData
ModuleNotFoundError: No module named 'insurance'
```

Do not set the `spark.pyspark.driver.python` configuration parameter when using a Python virtual environment resource in a job.

## Technical service bulletins

Learn about the technical service bulletins (TSBs) with the Cloudera Data Engineering (CDE) service on public clouds, the impact or changes to the functionality, and the workaround.

#### **TSB 2024-768: CDE graphs appear empty due to internal Prometheus OOM errors**

When using a CDE service that underwent an in-place upgrade from CDE 1.19.x to CDE 1.20.3/1.21.0, charts on the CDE administration page and Grafana dashboard might not show data. This issue occurs when the CDE internal Prometheus server encounters out of memory (OOM) errors, which is caused by the pre-upgrade Prometheus configurations not being carried over during upgrade. The upgrade path from CDE 1.20.3 to CDE 1.21.0 is not affected.

##### **Knowledge article**

For the latest update on this issue see the corresponding Knowledge Base article: [TSB 2024-768: CDE Graphs appear empty due to internal Prometheus OOM errors](#)

#### **TSB 2024-727: Upgrade YuniKorn in CDE deployments**

Apache YuniKorn (YuniKorn) 1.3.0, deployed as part of Cloudera Data Engineering (CDE) versions 1.19.3 and 1.19.4, can stop scheduling pods that cause workloads to get stuck even when cluster resources are available. Issues were found in the YuniKorn 1.3.0 release that get triggered by Spark jobs using gang scheduling in combination with preemption. These issues lead to accounting leaks and can result in YuniKorn stopping the scheduling of all pods in the cluster.

##### **Knowledge article**

For the latest update on this issue see the corresponding Knowledge Base article: [TSB 2024-727: Upgrade YuniKorn in CDE deployments](#)

#### **TSB 2023-658: Within CDP Public Cloud environments, CDE CLI access key authentication does not work for a[0-9].\*, b[1-9].\*, c[1-9].\* workload domains**

Using the Cloudera Data Engineering (CDE) command line interface (CLI) with access key-based authentication may cause authentication errors within CDP Public Cloud environments. The cause of the issue is the discrepancy in the way JSON Web Token (JWT) audience claim value is obtained between different components within the Cloudera Data Platform (CDP) Public Cloud.

##### **Knowledge article**

For the latest update on this issue see the corresponding Knowledge Base article: [TSB 2023-658: Within CDP Public Cloud environments, CDE CLI access key authentication does not work for a\[0-9\].\\*, b\[1-9\].\\*, c\[1-9\].\\* workload domains](#)

#### **TSB 2022-588: Kubeconfig and new version of aws-iam-authenticator**

Regenerate Kubeconfig and in conjunction use a newer version of aws-iam-authenticator on AWS. Kubeconfig in Cloudera Data Platform (CDP) Public Cloud Data Services needs to be regenerated because the Kubeconfig generated before June 15, 2022 uses an old APIVersion (client.authentication.k8s.io/v1alpha1) which is no longer supported. This causes compatibility issues with aws-iam-authenticator starting from **v0.5.7**. To be able to use the new aws-iam-authenticator, the Kubeconfig needs to be regenerated.

#### Knowledge article

For the latest update on this issue see the corresponding Knowledge Base article: [TSB 2022-588: Kubeconfig and new version of aws-iam-authenticator](#)

#### TSB 2022-587: CDE 1.14 and above using Kubernetes 1.21 will fail service account token renewal after 90 days

Cloudera Data Engineering (CDE) on Amazon Web Services (AWS) running version CDE 1.14 and above using Kubernetes 1.21 will observe failed jobs after 90 days of service uptime [1].

[1] “For Amazon Elastic Kubernetes Service (EKS) clusters, the extended expiry period is 90 days. Your Amazon EKS cluster's Kubernetes API server rejects requests with tokens older than 90 days.”

#### Knowledge article

For the latest update on this issue see the corresponding Knowledge Base article: [TSB 2022-587: CDE 1.14 and above using Kubernetes 1.21 will fail service account token renewal after 90 days](#)

## Limitations

Cloudera Data Engineering (CDE) has the following limitations.

#### Tier 2 node groups cannot be edited through the UI or API

Once a Tier 2 node group is created as part of service creation, the Tier 2 node group cannot be edited.

#### No support for Data Lake resizing

CDE does not support Data Lake resizing.

#### Running raw Scala code in Cloudera Data Engineering

- When setting the Log Level from the user interface, the setting is not applied to the raw Scala jobs.
- Do not use package <something> in the raw Scala job file as Raw Scala File is used for Scripting and not for Jar development and packaging.

## Cloudera Data Engineering Runtime end of support

Learn about Cloudera Data Engineering (CDE) Runtime end of support for Spark.

The following table specifies the planned End of Support (EoS) policy schedule for Spark. All future dates are provided for planning purposes only and are subject to change. In each case, the projected EoS date can be considered to be the last day of the month specified in the CDE Runtime end of support information table.

**Table 1: CDE Runtime end of support information**

Public Cloud Runtime version	Data Lake support	End of support	Long-term support	Notes
Spark 2.4.8	Up to 7.2.18	September 2025	Yes	Deprecated (will only receive bug fixes and security patches)
Spark 3.2.3	Up to 7.2.18	September 2025	-	-



Public Cloud Runtime version	Data Lake support	End of support	Long-term support	Notes
Spark 3.3.x	Up to 7.2.18	September 2025	-	-
Spark 3.5.x	Starting from 7.2.18	Will follow Data Lake support policy	Yes	-

## Frequently Asked Questions (FAQs)

### What is the relationship between CDE Spark runtimes and the Data Lake version?

Each CDE release is certified against one or more Data Lake versions. This ensures full compatibility between the Spark engine and the underlying platform capabilities, such as the Hive Catalog service and Ranger for security policies. Each Spark version is compatible with a Data Lake version. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).



**Important:** It is critical to update both CDE and Data Lake on a regular basis to ensure continued support and compatibility.

### Does CDE offer Long-term support (LTS) releases?

CDE offers LTS through underlying Spark runtimes. When running Spark jobs within CDE, you have the option to choose an older Spark version. Specific versions of Spark are designated as LTS. This allows you to continue running Spark jobs without any code changes. Since CDE job management APIs remain backwards compatible, existing automations are not impacted.

### What is the EoS timeline for Spark runtimes designated LTS?

Spark runtimes designated as LTS in the Public Cloud have certification that spans through multiple Data Lake versions and are limited to minor enhancements and critical bug and security enhancements. To use a Spark runtime that is tagged as LTS, you need to run a supported CDE version with a corresponding certified and currently supported Data Lake version.

### What is the EOS timeline for non-LTS Spark runtimes?

Spark runtimes that are not designated as LTS will be limited to the support lifetime of the corresponding certified Data Lake version. For more information, see [Compatibility for Cloudera Data Engineering and Runtime components](#).

### What is the support timeline for a given Data Lake version?

See the Cloudera Data Platform - Public Cloud section in [Support lifecycle policy](#).

## Compatibility for Cloudera Data Engineering and Runtime components

Learn about Cloudera Data Engineering (CDE) and compatibility for Runtime components across different versions. This document also includes component version compatibility information for AWS Graviton.

**Table 2: CDE compatibility with Runtime component details**

CDE Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Kubernetes
1.23.0	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Python 2.7 and 3.6</li> <li>Iceberg 1.3.1</li> <li>Data Lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Python 3.6</li> <li>Iceberg 1.3.1</li> <li>Data Lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Python 3.8</li> <li>Iceberg 1.3.1</li> <li>Data Lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.5.1</li> <li>Python 3.8</li> <li>Iceberg 1.5.2</li> <li>Data Lake 7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Airflow 2.9.3</li> <li>Python 3.11</li> </ul>	1.30
1.22.0	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Python 2.7 and 3.6</li> <li>Iceberg 1.3.1</li> <li>Data Lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Python 3.6</li> <li>Iceberg 1.3.1</li> <li>Data Lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Python 3.8</li> <li>Iceberg 1.3.1</li> <li>Data Lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.5.1</li> <li>Python 3.8</li> <li>Iceberg 1.4.3</li> <li>Data Lake 7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Airflow 2.7</li> <li>Python 3.8</li> </ul>	1.28
1.21	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Python 2.7 and 3.6</li> <li>Iceberg 1.3.1</li> <li>Data lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Python 3.6</li> <li>Iceberg 1.3.1</li> <li>Data lake 7.2.16-7.2.18</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Python 3.8</li> <li>Iceberg 1.3.1</li> <li>Data lake 7.2.16-7.2.18</li> </ul>	N/A	<ul style="list-style-type: none"> <li>Airflow 2.7</li> <li>Python 3.8</li> </ul>	1.27
1.20.3	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Python 2.7</li> <li>Iceberg 1.3.1</li> <li>Data lake 7.2.15-7.2.17</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Python 3.6</li> <li>Iceberg 1.3.1</li> <li>Data lake 7.2.15-7.2.17</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Python 3.8</li> <li>Iceberg 1.3.1</li> <li>Data lake 7.2.15-7.2.17</li> </ul>	N/A	<ul style="list-style-type: none"> <li>Airflow 2.6.3</li> <li>Python 3.8</li> </ul>	1.27
1.19.4	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.15-7.2.17</li> <li>Iceberg 1.1.0</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Java 11.0.20</li> <li>Python 3.6</li> <li>Data lake 7.2.15-7.2.17</li> <li>Iceberg 1.1.0</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Scala 2.12.15</li> <li>Java 11.0.20</li> <li>Python 3.8</li> <li>Data lake 7.2.15-7.2.17</li> <li>Iceberg 1.1.0</li> </ul>	N/A	<ul style="list-style-type: none"> <li>Airflow 2.3.4</li> <li>Python 3.8</li> </ul>	1.26
1.19.3	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.15-7.2.17</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Java 11.0.20</li> <li>Python 3.6</li> <li>Data lake 7.2.15-7.2.17</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Scala 2.12.15</li> <li>Java 11.0.20</li> <li>Python 3.8</li> <li>Data lake 7.2.15-7.2.17</li> </ul>	N/A	<ul style="list-style-type: none"> <li>Airflow 2.3.4</li> <li>Python 3.8</li> </ul>	1.25
1.19.2	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 1.1.0</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Java 11.0.20</li> <li>Python 3.6</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 1.1.0</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Scala 2.12.15</li> <li>Java 11.0.20</li> <li>Python 3.8</li> <li>Data lake: 7.2.16</li> <li>Iceberg 1.1.0</li> </ul>	N/A	<ul style="list-style-type: none"> <li>Airflow 2.3.4</li> <li>Python 3.8</li> </ul>	1.25

CDE Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Kubernetes
1.19	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 1.1.0</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Java 11.0.20</li> <li>Python 3.6</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 1.1.0</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.0</li> <li>Scala 2.12.15</li> <li>Java 11.0.20</li> <li>Python 3.8</li> <li>Data lake: 7.2.16</li> <li>Iceberg 1.1.0</li> </ul>	N/A	<ul style="list-style-type: none"> <li>Airflow 2.3.4</li> <li>Python 3.8</li> </ul>	1.25
1.18.3	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala: 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 0.14.1</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Java 11.0.20</li> <li>Python 3.6</li> <li>DL: 7.2.14-7.2.16</li> <li>Iceberg 0.14.1</li> </ul>	N/A	N/A	<ul style="list-style-type: none"> <li>Airflow 2.3.4</li> <li>Python 3.8</li> </ul>	1.24
1.18.1	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 0.14.1</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.0</li> <li>Scala 2.12.10</li> <li>Java 11.0.20</li> <li>Python 3.6</li> <li>Data lake 7.2.14-7.2.16</li> <li>Iceberg 0.14.1</li> </ul>	N/A	N/A	<ul style="list-style-type: none"> <li>Airflow 2.3.4</li> <li>Python 3.8</li> </ul>	1.23
1.18	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Data lake 7.2.14-7.2.15</li> <li>Iceberg 0.14.1</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.0</li> <li>Data lake 7.2.14-7.2.15</li> <li>Iceberg 0.14.1</li> </ul>	N/A	N/A	<ul style="list-style-type: none"> <li>Airflow 2.2.5</li> <li>Python 3.8</li> </ul>	1.23
1.17-h1	<ul style="list-style-type: none"> <li>Spark 2.4.8</li> <li>Scala 2.11.12</li> <li>Java 1.8.0_352</li> <li>Python 2.7</li> <li>Iceberg 0.14.1</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.1</li> <li>Iceberg 0.14.1</li> </ul>	N/A	N/A	<ul style="list-style-type: none"> <li>Airflow 2.2.5</li> <li>Python 3.8</li> </ul>	1.23

### AWS Graviton component version compatibility


To use AWS Graviton in CDE, you need the following runtime component versions:

- Spark 3.3 and 3.5
- Data Lake 7.2.18

## Using the Cloudera Runtime Maven repository for Cloudera Data Engineering

When building Spark applications to run on CDE, you can use the following table to determine which artifact versions to use from the Cloudera Runtime Maven repository.



You can determine the CDE version for your virtual cluster by clicking the  Cluster Details icon for the virtual cluster you want to use.

The following is a sample POM (pom.xml) file:

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <repositories>
    <repository>
      <id>cloudera</id>
      <url>https://repository.cloudera.com/artifactory/cloudera-repos/</url>
    </repository>
  </repositories>
</project>
```

CDE version	Cloudera Runtime Maven repository
<ul style="list-style-type: none"> <li>1.20.3</li> <li>1.19.4</li> <li>1.19.3</li> <li>1.19.2</li> </ul>	<ul style="list-style-type: none"> <li>Cloudera Runtime 7.2.15</li> <li>Cloudera Runtime 7.2.16</li> </ul>
<ul style="list-style-type: none"> <li>1.19</li> </ul>	<ul style="list-style-type: none"> <li>Cloudera Runtime 7.2.16</li> <li>Cloudera Runtime 7.2.15</li> </ul>
<ul style="list-style-type: none"> <li>1.18</li> <li>1.17</li> <li>1.16</li> </ul>	Cloudera Runtime 7.2.15
<ul style="list-style-type: none"> <li>1.15</li> </ul>	Cloudera Runtime 7.2.14
<ul style="list-style-type: none"> <li>1.14</li> </ul>	Cloudera Runtime 7.2.11.0
<ul style="list-style-type: none"> <li>1.13</li> </ul>	Cloudera Runtime 7.2.11.0
<ul style="list-style-type: none"> <li>1.12</li> <li>1.11</li> <li>1.9</li> </ul>	Cloudera Runtime 7.2.10.0
<ul style="list-style-type: none"> <li>1.8</li> <li>1.7</li> </ul>	Cloudera Runtime 7.2.8.0
<ul style="list-style-type: none"> <li>1.6</li> <li>1.5</li> <li>1.4</li> <li>1.3</li> <li>1.2</li> </ul>	Cloudera Runtime 7.2.2.0