Cloudera Data Engineering 1.25.0

Cloudera Data Engineering Release Notes

Date published: 2020-07-30
Date modified: 2025-11-20

CLOUDƎRA

Legal Notice

Contents

What's new in Cloudera Data Engineering on cloud?

This section lists major features and updates for the Cloudera Data Engineering service in Cloudera on cloud.

November 20, 2025

This release (1.25.0) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

Virtual Cluster-level suspend and resume [Technical Preview]

From Cloudera Data Engineering version 1.25.0, the Virtual Cluster-level suspend and resume feature is available, enabling you to temporarily pause one or more Cloudera Data Engineering Virtual Clusters during idle periods. This feature is supported on both AWS and Azure. Suspending a larger number of VCs can offer significant cost savings, primarily on the compute resources.

To start using the feature, you need to request the DE_ENABLE_VC_SUSPEND_RESUME entitlement.

For more information, see Overview of suspending and resuming Cloudera Data Engineering virtual clusters.

Cloudera Data Engineering Spark support for Hive Warehouse Connector

Cloudera Data Engineering offers the integration with Hive Warehouse Connector (HWC) to access and manage Hive-managed tables from Cloudera Data Engineering Spark. The HWC dependencies are built into Cloudera Data Engineering.

Through Cloudera Data Engineering Spark, you can use HWC to read data from Hive and to write data to Hive. HWC enables secure access and querying of Hive-managed tables directly from Apache Spark. To ensure secure data handling, HWC enables you to manage data access with fine-grained access control.

For more information, see Cloudera Data Engineering Spark support for Hive Warehouse Connector.

Amazon Linux distribution migration to AL2023

The OS image has been changed from AL2 to AL2023 to help you increase security and improve application performance. AL2023 optimizes boot times to decrease the interval from instance initialization to processing the customer workload. For more information, see Comparing AL2 and AL2023.

Cloudera Data Engineering performance improvement

NodeLocalDNS is a Kubernetes-supported DaemonSet that runs a local DNS cache on every node in the cluster. The deployment of NodeLocalDNS in Cloudera Data Engineering improves application reliability, as service discovery depends on DNS. Currently, NodeLocalDNS is supported for Cloudera Data Engineering services on AWS. Key benefits of deploying NodeLocalDNS:

- Reduced latency: queries are answered from a local cache, minimizing network hops
- Lower CoreDNS load: repeated queries are served locally, reducing load on CoreDNS
- Improved stability: cached entries remain available even if the CoreDNS pods restart

Cloudera Data Engineering system resilience improvement

The system resilience of Cloudera Data Engineering has been increased by enhancing the high availability capabilities of the Admission Controller to help make the system more robust in unreliable networking environments.

Upgrade precheck updates

As part of the upgrade preparation workflow, a check has been introduced to ensure that the Virtual Clusters (VCs) use a Spark version that is compatible with the Data Lake version. If the Spark version used for the VCs is incompatible with the Data Lake version, the upgrade preparation fails. For example, if the Data Lake version is upgraded from 7.2.18 to 7.3.1, and any of the VCs use a Spark version lower than 3.5.4, the upgrade preparation fails.

Kubernetes version upgrade to 1.32

The Kubernetes version that Cloudera Data Engineering uses is upgraded to Kubernetes 1.32. For more information, see Compatibility for Cloudera Data Engineering and Runtime components.

Apache Airflow version upgrade to 2.11.0

The Airflow version that Cloudera Data Engineering uses is upgraded to Airflow 2.11.0 for improved security and stability. For more information, see:

- Apache Airflow 2.11.0 Release Notes
- Compatibility for Cloudera Data Engineering and Runtime components

Fixed issues

- DEX-18145: Upgrade dbus-wxm-client version to 1.6.0-b7
- DEX-17969: Airflow failed to connect to Impala in Cloudera Data Engineering
- DEX-17565: Links to download cdeconnect and pyspark tars for Spark Connect are giving HTTP 404 error
- DEX-16414: Sessions GET endpoint not returning empty array
- DEX-15461: Writing Spark Dataframe to Hive using HWC Fails with java.util.NoSuchElementException: None.get
- DEX-17973: Implement NodeLocalDNS to improve DNS performance on deployed clusters
- DEX-17940: Scheduled Airflow DAG run syncing to Cloudera Data Engineering can miss runs
- DEX-17865: TZData package missing in chainguard image
- DEX-17806: Handle unavailability of the admission controller
- DEX-17760: Investigate the causes for intermittent RocksDBException in SHS pods
- DEX-17284: Gracefully handle quota reached situation
- DEX-17269: Upgrade tgtgenerator version to 1.0.3-b131 and testing basic jobs
- DEX-17260: Restoring manually packed archive failed with "could not find file   entry"
- DEX-17037: Jobs with python script referencing resources on airflow editor page failed to save
- DEX-16747: Cloudera Data Engineering 1.23.1-b114 - Driver container stderr, and stdout logs are missing for some Spark jobs
- DEX-16588: Cloudera Data Engineering in-place upgrade fails with "EKS API   server endpoint"
- DEX-16100: RunDAGMonitor may create duplicate Cloudera Data Engineering Jobs
- DEX-16048: Speed up Airflow Cloudera Data Engineering Job update
- DEX-15266: Get rid of Cluster#State check in the Cloudera Data Engineering code
- DEX-14276: Restrict CAP_NET_RAW permissions
- DEX-11181: Steps To Edit MountOptions Of Storage Class
- DEX-7390: Add support for backup/restore of workload credentials

July 30, 2025

This release (1.24.1-H1) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

⚠️ **Important:** If you are using Azure with Cloudera Data Engineering Services version 1.24.1 or lower versions, you will not be able to create new Virtual Clusters. To create new Virtual Clusters, upgrade your services to Cloudera Data Engineering version 1.24.1-H1.

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

DEX-17716: sparkuser home directory is not created after chainguard upgrade in livy-server image

The issue impacted customers using spark.jars.packages=mysql:mysql-connector-java:8.0.28. This issue has been fixed.

DEX-17567: Azure MySQL Root Certificate update

The root certificates for Azure MySQL Flexible database were updated due to security non-compliance of the SHA-1 certificate. The SHA-1 hashing algorithm is considerably insecure, due to discovered vulnerabilities. Cloudera resolved the issue by rotating the certificate to one signed by a compliant root certificate authority.

DEX-17519: Sessions are not killed as per the TTL configured in Azure and AWS

Sessions were not killed according to the Time To Live (TTL) configured in Azure and in AWS. The calculation of timeout was wrong in the isTimeout method in the Livy code. The isTimeout method of the Livy code incorrectly converted a timeout value that was already provided in nanoseconds, leading to miscalculation of the timeout. For this reason, the sessions were not killed after the timeout was reached.

DEX-17507: Restore of Scheduled Jobs are failing due to time format

Restoring the Spark Jobs with the Schedule Configuration failed if the start date or end date used a time format other than RFC3339Nano. This issue affected only jobs created through non-UI methods, such as the API or CLI. The fix standardizes schedule start and end times to RFC3339Micro format, ensuring alignment with Airflow API timestamps at microsecond precision.

Note: This fix resolves the restoration issue only for backup archives created using Cloudera Data Engineering version 1.24.1-H1 and higher versions. This fix does not resolve the issue for older Cloudera Data Engineering backup archives with different timestamp formats.

DEX-17500: [CDP Cli] Spark OsName "chainguard" not triggering error in Cloudera Data Engineering version 1.23.1 VC

Cloudera Data Engineering allowed the creation of a Virtual Cluster with the securityhardened spark.osname property in Cloudera Data Engineering versions lower than 1.24, without any error message. The underlying use of UBI [red hat] was technically correct; however, it caused confusion, as the Virtual Cluster property indicated securityhardened. With the fix, an error is now triggered if securityhardened is specified in Cloudera Data Engineering versions lower than 1.24.

DEX-17458: [Spark3.3.0] : Cloudera Data Engineering session creation is failing with javax.security.sasl.SaslException: Client closed before SASL negotiation finished

Cloudera Data Engineering sessions in a Spark 3.3.0 Virtual Cluster failed to be created. The following error was listed in the driver logs:

Exception in thread "main" java.util.concurrent.ExecutionException: javax.security.sasl.SaslException: Client closed before SASL negotiation finished

The root cause was determined to be a TTL issue rather than an SASL issue.

June 25, 2025

This release (1.24.1) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

Suspending and resuming Cloudera Data Engineering Services - Technical preview

The suspend and resume feature enables you to temporarily pause your entire Cloudera Data Engineering Service during idle periods. Suspending Services when they are not needed offers significant cost savings, primarily on compute resources, while ensuring that the Services remain available when required. During suspension, all Service and Virtual Cluster configurations and Jobs are preserved. By default, the entitlement for the suspend and resume feature is disabled. For more information, see Suspending and resuming Cloudera Data Engineering Services.

Security hardened image

Starting with Cloudera Data Engineering version 1.24.1, a new security-hardened image is introduced to enhance security and increase resilience against Common Vulnerabilities and Exposures (CVEs). In Cloudera Data Engineering version 1.24.1, you have the option to choose between adopting the new security hardened image or continuing to use the existing image from previous versions (Redhat UBI), based on your system needs and business requirements.

The runtime component versions are upgraded in the security hardened image; for compatibility, see Compatibility for Cloudera Data Engineering and Runtime components. For information on the upgrade behavior of the security hardened image, see Cloudera Data Engineering upgrade version compatibility.

> ⚠️ **Important:** If you switch from the Redhat UBI image to the security hardened image, note that the updated component versions and libraries can impact the Cloudera Data Engineering operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.
>
> For backward compatibility, you can continue using the Redhat UBI image. The security hardened image is the recommended image, however, note that it requires testing and validation.
>
> For more information, see Security hardened Spark image migration guide.

Git integration support available for GA

The Git integration support feature, previously available as a technical preview, has now reached General Availability (GA). For more information, see Creating a Git repository in Cloudera Data Engineering.

Autoscaling range setting method enhanced on the Cloudera Data Engineering UI

While creating a new Cloudera Data Engineering Service on the Cloudera Data Engineering UI (Administration Enable a Service), you have the option to set the autoscaling range for the Core and All-Purpose Tiers. In the Capacity & Costs section, you can configure the autoscaling range for On-demand instances and if you select an AWS environment, for Spot instances as well. For more information, see Cloudera Data Engineering tier types.

Runtime component version exposure on the Cloudera Data Engineering UI

On the Cloudera Data Engineering UI, you can view the Apache Airflow runtime components and Spark runtime components on the creation and details pages of a Service, Virtual Cluster, job, or session.

The following screenshot illustrates creating a new Virtual Cluster, where you can select the required runtime component versions from a drop-down menu on the UI.

Administration / Create a Virtual Cluster

Virtual Cluster Details

Setup your virtual cluster by completing the following fields. These settings cannot be changed once your cluster has been created.

* Name

Cluster Name

* Service

dashboard-test (cluster-4mkdwwz6)   ▼

* Spark Version

Spark 3.5.1 (Security Hardened)   Python 3.11 | Iceberg 1.5.2 | Java 17 | Scala 2.12.18   ▼

Need more information on compatibility of runtime components? Review our compatibility matrix

Airflow Runtime Version

Airflow 2.10.4   Python 3.11

Runtime component version behavior after upgrading

To maintain backwards compatibility, the in-place upgrade does not modify the image type, and after the upgrade to Cloudera Data Engineering version 1.24.1, the old RedHat (deprecated) images continue to be used. To use the new, security hardened image, after the upgrade, you must create new Virtual Clusters and select the new image. For more information on the supported runtime components, see Compatibility for Cloudera Data Engineering and Runtime components.

Cloudera Data Engineering private network for Elastic Kubernetes Service and Azure Kubernetes Service available for GA

Enabling a fully private network for a Cloudera Data Engineering service for Azure and Enabling a semi-private network for a Cloudera Data Engineering service with AWS are available for GA. For more information, see:

• Enabling a semi-private network for a Cloudera Data Engineering service with AWS
• Enabling a fully private network for a Cloudera Data Engineering service for Azure

Kubernetes version upgrade to 1.31

The Kubernetes version that Cloudera Data Engineering uses is upgraded to Kubernetes 1.31. For more information, see Compatibility for Cloudera Data Engineering and Runtime components.

Apache Airflow version upgrade to 2.10.4

The Airflow version that Cloudera Data Engineering uses is upgraded to Airflow 2.10.4. For more information, see:

• Apache Airflow 2.10.4 Release Notes
• Compatibility for Cloudera Data Engineering and Runtime components

Fixed issues

• DEX-15798: [Postgres][Azure] Private DNS zone is not created
• DEX-15989: Diagnostics Bundle lacks App Instances information
• DEX-16007: Spark 3.3+ on JDK 17 errors
• DEX-17281: Unmounted Airflow file mount still present at job runtime
• DEX-17264: [BnR] vc.vcApiUrl is not present in the VC Describe
• DEX-17495: In AKS 1.30+ Spark2 jobs running more than 1 hour are failing

June 9, 2025

This release (1.23.1-H3) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

This release does not contain new features, but includes a fix for Apache Parquet CVE-2025-30065.

⚠️ **Important:** The Cloudera Data Engineering 1.23.1-H3 hotfix release is a manual patch release and therefore it is not available through the in-place upgrade method.

Apache Parquet CVE-2025-30065 details

CVE-2025-30065 | Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code.

CVE: NVD - CVE-2025-30065

Severity (Critical): CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

For the latest update on this issue, see the corresponding Knowledge article: Cloudera Customer Advisory 2025-847: Cloudera's remediation actions for Apache Parquet CVE-2025-30065.

Spark images affected by Apache Parquet CVE-2025-30065 in 1.23.1-h2

The following Apache Spark images are affected by the Apache Parquet CVE:

Spark 3.2.x

Image Name:

- dex-livy-runtime-3.2.3-7.2.15.8:1.23.1-h2-b3
- dex-spark-runtime-3.2.3-7.2.15.8:1.23.1-h2-b3
- dex-livy-server-3.2.3-7.2.15.8:1.23.1-h2-b3

Spark 3.3.x

Image Name:

- dex-livy-server-3.3.0-7.2.16.200:1.23.1-h2-b3
- dex-spark-runtime-3.3.0-7.2.16.200:1.23.1-h2-b3
- dex-livy-runtime-3.3.0-7.2.16.200:1.23.1-h2-b3

Spark 3.5.x

Image Name:

- dex-livy-runtime-3.5.1-7.2.18.0:1.23.1-h2-b3
- dex-spark-runtime-3.5.1-7.2.18.0:1.23.1-h2-b3
- dex-livy-server-3.5.1-7.2.18.0:1.23.1-h2-b3

Spark images not affected by Apache Parquet CVE-2025-30065
Spark 2.x

Spark 2.x images are not affected by the Parquet CVE.

The Spark History server image for Cloudera Data Engineering 1.23.1 or lower versions does not include the shaded parquet jars.

Script incorporating the Parquet fixes

Go to the Cloudera Data Engineering section of Cloudera Customer Advisory 2025-847: Cloudera's remediation actions for Apache Parquet CVE-2025-30065 and follow the procedure described there. The script updates the Spark Runtime, Livy Runtime, and Livy Server with the updated images that incorporate the Parquet fixes.

April 4, 2025

This release (1.23.1-H2) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

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

DEX-16625: Nginx-ingress update to mitigate CVEs

- **CVE-2025-24513** (CVSS score: 4.8)

  An improper input validation vulnerability that could result in directory traversal within the container, leading to denial-of-service (DoS) or limited disclosure of secret objects from the cluster when combined with other vulnerabilities

- **CVE-2025-24514** (CVSS score: 8.8)

  The auth-url Ingress annotation can be used to inject configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller

- **CVE-2025-1097** (CVSS score: 8.8)

  The auth-tls-match-cn Ingress annotation can be used to inject configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller

- **CVE-2025-1098** (CVSS score: 8.8)

  The mirror-target and mirror-host Ingress annotations can be used to inject arbitrary configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller

- **CVE-2025-1974** (CVSS score: 9.8)

  An unauthenticated attacker with access to the pod network can achieve arbitrary code execution in the context of the ingress-nginx controller under certain conditions

March 27, 2025

This release (1.23.1-H1) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

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

DEX-15814: AirflowAPI list_runs_by_id fails with 'NoneType' object has no attribute 'strftime'

Cloudera Data Engineering Airflow Run Status updates did not work, which could lead to duration mismatch of Job duration versus the actual Airflow Dag Run duration, or it could lead to run status mismatch, or both. This happened when the same Airflow Job was triggered (AdHoc) multiple times and there was a limit on how many parallel DAGs could run at the same time. When the limit was reached, the DAG Runs were queued. As long as there were queued Airflow DAG Runs with None start_date, the Cloudera Data Engineering Job Run status updates did not work. Depending on the workload, this could take a long time. When this issue was present, Cloudera Data Engineering Jobs could not be killed. The state changed to Killed, but it changed back to the previous state. When this issue was present, the airflow-api logs listed: Could not search DAG runs: 'NoneType' object has no attribute 'strftime'.

DEX-15813: Cloudera Data Engineering killed job state can go back to previous state

Jobs could not be killed. When you tried to kill a Cloudera Data Engineering Airflow Job that was in running state, the Cloudera Data Engineering Job state changed to killed for a couple of seconds, but then it changed back to the previous state.

DEX-16304: Eliminate "Failed to parse DAG run start date" error logs when startDate is empty

If the start_date was zero for an Airflow DAG Run, an unnecessary error log was created.

DEX-16232: Fix already existing DAG check in airflow-api

A Cloudera Data Engineering Airflow Job could become corrupted during Cloudera Data Engineering Job creation, making it unusable for further management. The existing Airflow DAG check during Airflow Job creation did not handle OS-related issues properly (for example: NFS mount issue). If it happened, the metadata of an existing DAG could become corrupted. Corrupted Cloudera Data Engineering Airflow Jobs could not be deleted.

DEX-14724: Deletion logic prevents Cloudera Data Engineering Jobs from being deleted

A corrupted Airflow Job could not be deleted through the CLI, the REST API, or the UI.

DEX-15842: Cache the service account for consecutive job launches

When a large number of Jobs were submitted, during the Job launch, the service account check failed due to the Kubernetes API limit and the default timeout. Job submission failed with the following error: could not create user se rvice account: client rate limiter.

DEX-15957: runtime-api pod restart causes RunDagMonitor to fetch all the DAG runs (airflowapi DoS)

Cloudera Data Engineering Airflow Job CRUD operations and Job Run Status updates did not work on Virtual Clusters with big Airflow DAG Run history. The airflow-api crashed after the jobs-api restarted if there were at least 400 000 Airflow DAG Runs in the Virtual Cluster.

February 10, 2025

This release (1.23.1) of the Cloudera Data Engineering service on Cloudera on cloud introduces the following changes.

In-place upgrade enhancements

Using AWS, you can upgrade from Cloudera Data Engineering version 1.20.3 to 1.23.1. Using Azure, the minimum version of Cloudera Data Engineering for the upgrade is version 1.22.0. For more information, see Cloudera Data Engineering upgrade version compatibility and In-place upgrade with Airflow Operators and Libraries.

AWS Graviton spot instances support

With AWS Graviton, you can use spot instances as well. For more information, see AWS Graviton instances in Cloudera Data Engineering.

Data Lake 7.3.1 support

Cloudera Data Engineering version 1.23.1, besides supporting Data Lake 7.2.18, also supports Data Lake 7.3.1 with Apache Spark 3.5. For more information, see:

- Upgrading to Cloudera Data Lake 7.3.1 with Cloudera Data Engineering
- Compatibility for Cloudera Data Engineering and Runtime components

.

> **Note:** Data Lake 7.3.1 supports virtual cluster creation from Spark version 3.5 onwards. Earlier Spark versions are not supported with Data Lake 7.3.1.

Java upgrade to version 17

The Java version that Airflow uses is upgraded to Java 17. For more information, see Compatibility for Cloudera Data Engineering and Runtime components.

MySQL upgrade to version 8.0.39

The MySQL version that Cloudera Data Engineering 1.23.1 uses is upgraded to version 8.0.39.

Fixed issues

DEX-15143

Service backup failing due to Cadence size limits on the metadata

DEX-15229

UI glitch while accessing the Spark UI for a Cloudera Data Engineering Session

DEX-15398

Fix Helm upgrade failures: retrigger on context deadline exceeded and ensure correct revision on retry

DEX-15477

Rapid creation of successive job runs causes failures due to Jobs table locking

DEX-15479

Cannot upgrade Cloudera Data Engineering to 1.23 due to broken Spark Job

DEX-15498

Early unlock on the MutexMap causes incorrect locking behaviour

DEX-15587

RefreshRuns skips polling runs for Cloudera Data Engineering Job Status update for a while

DEX-15589

Livy marks the Batch failed when the monitoring thread is interrupted

DEX-15713

Jobs failing with keytab access issue

November 12, 2024

This release (1.23.0) of the Cloudera Data Engineering service on Cloudera on 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 External IDE connectivity through Spark Connect-based 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 Cloudera Data Warehouse Operator

Starting from Cloudera Data Engineering service version 1.23.0, Cloudera Data Warehouse Operator is deprecated. Cloudera recommends you to use SQL Operator in place of Cloudera Data Warehouse Operator. For more information, see Creating a connection to Cloudera Data Warehouse for Cloudera Data Warehouse 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: Cloudera Data Engineering 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.InstantiationIOException'

Jobs running on RAZ-enabled clusters on Apache Spark 3.5.1 failed with the org.apache.hadoop.fs.s3a.impl.InstantiationIOException 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 allp 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 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 efficiant 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 bases pipelines. For more informarion, 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 In-place upgrade with Airflow Operators and Libraries

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.i
dbroker.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 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 features 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 features 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.

Apache Parquet CVE-2025-30065

On April 1, 2025, a critical vulnerability in the parquet-avro module of Apache Parquet (CVE-2025-30065, CVSS score 10.0) was announced.

> **Note:** The issue communicated in this TSB has been fixed in Cloudera Data Engineering version 1.23.1-H3. For more information, see Release Notes, What's new, June 9, 2025.

Cloudera has determined the list of affected products, and is issuing this TSB to provide details of remediation for affected versions.

Upgraded versions are being released for all currently affected supported releases of Cloudera products. Customers using older versions are advised to upgrade to a supported release that has the remediation, once it becomes available.

Vulnerability Details

Exploiting this vulnerability is only possible by modifying the accepted schema used for translating Parquet files and subsequently submitting a specifically crafted malicious file.

CVE-2025-30065 | **Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code.**

CVE: NVD - CVE-2025-30065

Severity (Critical): CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:H/VI:H/VA:H/SC:H/SI:H/SA:H

Impact

Schema parsing in the parquet-avro module of Apache Parquet 1.15.0 and previous versions allows bad actors to execute arbitrary code. Attackers may be able to modify unexpected objects or data that was assumed to be safe from modification. Deserialized data or code could be modified without using the provided accessor functions, or unexpected functions could be invoked.

Deserialization vulnerabilities most commonly lead to undefined behavior, such as memory modification or remote code execution.

Releases affected

Cloudera Data Engineering on cloud

• All versions

Mitigation

Until Cloudera has released product version with the Apache Parquet vulnerability fix, please continue to use the the mitigations listed below:

Customers with their own FIM Solution:

1. Utilize a File Integrity Monitoring (FIM) solution. This allows administrators to monitor files at the filesystem level and receive alerts on any unexpected or suspicious activity in the schema configuration.

General advisory:

1. Use network segmentation and traffic monitoring with a device capable of deep packet inspection, such as a network firewall or web application firewall, to inspect all traffic sent to the affected endpoints.
2. Configure alerts for any suspicious or unexpected activity. You may also configure sample analysis parameters to include:

   • Parquet file format "magic bytes" = PAR1
   • Connections from sending hosts that are not expected source IP ranges.

3. Be cautious with Parquet files from unknown or untrusted sources. If possible, do not process files with uncertain origins or that can be ingested from outside the organization.
4. Ensure that only authorized users have access to endpoints that ingest Parquet files.

For the latest update on this issue, see the corresponding Knowledge article: Cloudera Customer Advisory 2025-847: Cloudera's remediation actions for Apache Parquet CVE-2025-30065

Kubernetes Ingress NGINX Controller vulnerabilities

Five vulnerabilities affecting the Ingress Nginx Controller for Kubernetes were publicly disclosed on March 24, 2025, and were given the nickname "IngressNightmare."

The 'IngressNightmare' vulnerabilities may allow Remote Code Execution (RCE) and potentially expose Kubernetes clusters to malicious configuration modifications. Exploitation requires specially crafted HTTP requests that bypass security measures, such as a Web Application Firewall (WAF). Successful exploitation may lead to complete cluster compromise, data exfiltration, and denial of service.

Details of the CVEs:

- CVE-2025-1974 (CVSS score: 9.8) – An unauthenticated attacker with access to the pod network can achieve arbitrary code execution in the context of the ingress-nginx controller under certain conditions
- CVE-2025-24514 (CVSS score: 8.8) – The auth-url Ingress annotation can be used to inject configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller
- CVE-2025-1097 (CVSS score: 8.8) – The auth-tls-match-cn Ingress annotation can be used to inject configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller
- CVE-2025-1098 (CVSS score: 8.8) – The mirror-target and mirror-host Ingress annotations can be used to inject arbitrary configuration into NGINX, resulting in arbitrary code execution in the context of the ingress-nginx controller and disclosure of secrets accessible to the controller
- CVE-2025-24513 (CVSS score: 4.8) – An improper input validation vulnerability that could result in directory traversal within the container, leading to denial-of-service (DoS) or limited disclosure of secret objects from the cluster when combined with other vulnerabilities

Mitigation

For mitigating CVE-2025-1974 on Cloudera Data Engineering on cloud, refer to the information below.

Note: Cloudera recommends limiting direct access to cluster hosts to only authorized administrators and auditing all activity as a security best practice.

Mitigation of CVE-2025-24514, CVE-2025-1097, CVE-2025-1098, and CVE-2025-24513 is secondary to the previous CVE. They require no immediate action, as attackers can only exploit these with direct access to cluster hosts and privileges to create arbitrary ingress objects via the Kubernetes API.

Note: Cloudera has tested these mitigation steps only on the currently supported releases. Customers using older versions are advised to upgrade to a supported release before attempting the mitigation actions.

Instructions

Cloudera Data Engineering does not enable the Validating Admission Controller feature of ingress-nginx and so is not exposed to the Remote Code Execution (RCE) vulnerability.

However, you can update the version of the ingress-nginx controller nevertheless by following the below mitigation steps.

1. Check your current Cloudera Data Engineering version

   a. In Cloudera Management Console, open "Data Engineering"
   b. In the left pane, click on the "Administrator" link, to list all the deployed Services.
   c. On the service card, click on the "Service Details" for the specific Service you want to mitigate:
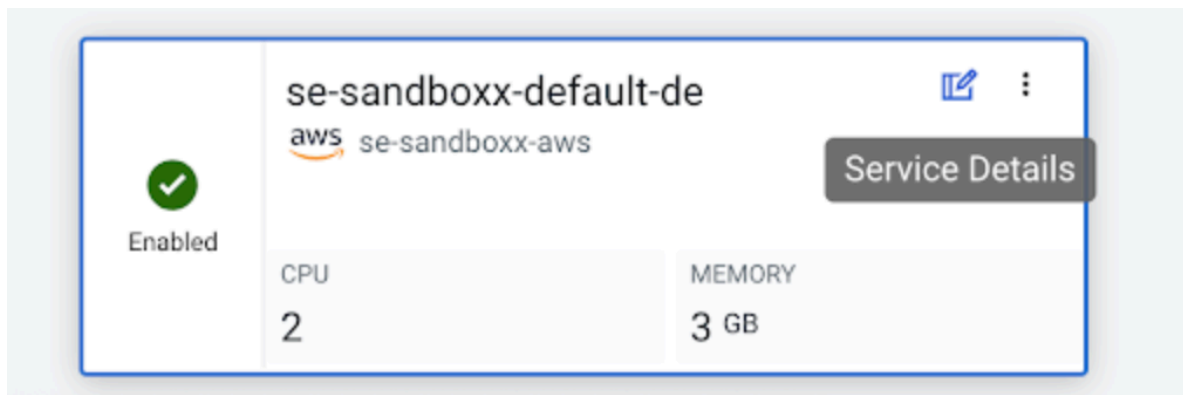

2. Retrieve your cluster id from the Cloudera Data Engineering on cloud UI

   Get the cluster-id of the service as shown in the below image. The id is the 8 character long alphanumerical string after "cluster-".
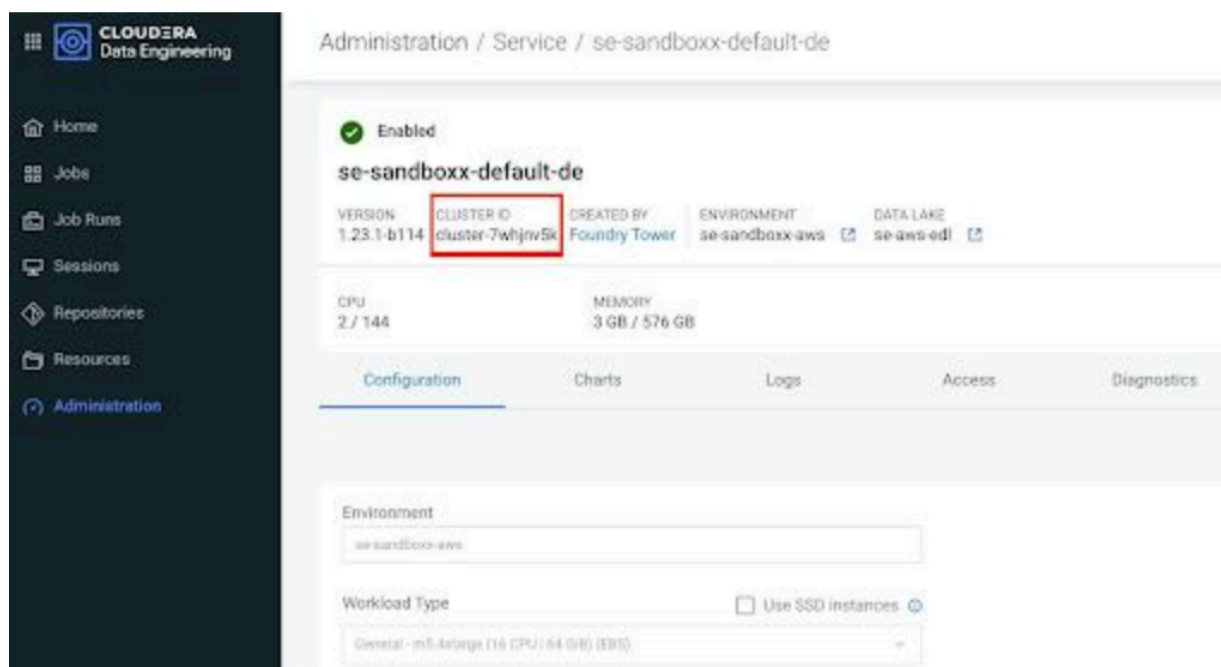
3. Download kubeconfig

    a. In the Cloudera Data Engineering UI, go to Administration.

    b. Select the service from the list, and in the Actions menu (3 vertical dots) select Download kubeconfig:
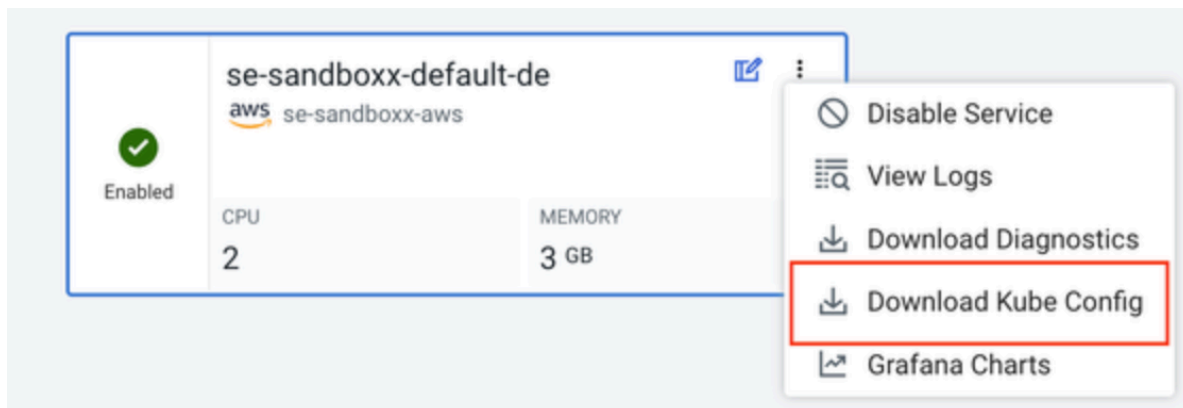
4. Patch the ingress-nginx controller image

    a. Substitute the below command with your 8 character long alpha numerical cluster id string from Step #2.

```
export CLUSTER_ID=<8 character alphanumeric cluster id>
```

    b. Set your kubeconfig to the kubeconfig downloaded in Step #3.

```
export KUBECONFIG=<kubeconfig_path>
```

5. Update nginx controller image to the verified patched version.

```
kubectl set image deployment dex-base-nginx-${CLUSTER_ID}-controller -n
dex-base-${CLUSTER_ID} controller=https://www.google.com/url?q=http://co
ntainer.repository.cloudera.com/cloudera_thirdparty/hardened/ingress-ngi
nx-controller:1.11.5-r0-202503251929
```

6. Verify the patch

    a. Verify the version of the nginx controller image is set to the correct version:

```
kubectl --kubeconfig kubeconfig.yaml -n dex-base-${CLUSTER_ID} get deplo
yment dex-base-nginx-${CLUSTER_ID}-controller -o jsonpath='{.spec.templa
te.spec.containers[?(@.name=="controller")].image}'
```

    b. Verify that regular operations (submitting jobs, creating a new virtual cluster, etc.) work after the patch has been applied.

> **Note:** Cloudera tested these mitigation steps only on Cloudera Data Engineering on cloud 1.22.0 or higher. Customers using older versions are advised to upgrade.

For the latest update on this issue, see the corresponding Knowledge articles:

- TSB 2025-839: Critical Kubernetes Ingress NGINX Controller Vulnerability Allows RCE Without Authentication
- TSB 2025-839 Mitigation steps for Cloudera Data Engineering on cloud

Cloudera Data Engineering fix for CVE-2021-44228

On December 21, 2021, Cloudera released Cloudera Data Engineering on 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 on cloud

This section lists known issues and limitations that you might run into while using the Cloudera Data Engineering service in Cloudera on cloud.

General known issues with Cloudera Data Engineering

Learn about the general known issues with the Cloudera Data Engineering service on cloud, the impact or changes to the functionality, and the workaround.

Note: Changes applied by some or all the workarounds can be lost once the Cloudera Data Engineering cluster is upgraded. After an upgrade, check whether the fix is permanently added or you need to apply the workaround again.

DEX-15929: Airflow pyenv certificates in PEM format are not converted to Base64 in the Cloudera Data Engineering UI

The Cloudera Data Engineering API for Airflow Operators and Libraries currently expects Base64-encoded certificates, but the Cloudera Data Engineering UI sends PEM certificates without conversion.

Go to Cloudera Data Engineering UI Administration Virtual Cluster Details of the selected virtual cluster Configuration tab Airflow Libraries and Operators Configure Repositories SSL Certificate and provide the certificate in Base64 format.

Cloudera Data Engineering 1.20.0 through 1.25.0.

N/A

DEX-18310: Azure Cloudera Data Engineering service stability or scaling issues in 1.25.0 (cde-fluentd OOMKilled)

Memory leak issue in cde-fluentd in Azure.

Contact Cloudera Support, as the workaround includes steps that require changes in Kubernetes objects.

Cloudera Data Engineering 1.25.0 and lower versions.

N/A

ENGESC-31314: Cloudera Data Engineering Spark UI is not available for finished jobs

By default, the priority class for the cde-fluentd daemonset is not set, so it uses the default priority class. As a result, if the cde-fluentd daemonset restarts for any reason, it can never preempt other pods and it cannot start running on every node. This issue is present on all cloud platforms.

This workaround primarily fixes the priority of the cde-fluentd daemonset manually.

Prerequisite

Ensure that no jobs are running in any virtual cluster. Although no explicit downtime occurs during the procedure, a few jobs might be terminated. Therefore, Cloudera recommends performing this procedure when the cluster is silent. You can expect the following procedure to take 15 to 20 minutes to perform.

Steps

1. Create a new priority class, for example cde-fluentd-node-critical, by creating a YAML file with the following content:

```
# cde-fluentd-node-critical.yaml
apiVersion: scheduling.k8s.io/v1
kind: PriorityClass
metadata:
  # This name will be referenced in the DaemonSet
  name: cde-fluentd-node-critical
value: 999999990
globalDefault: false
description: "High priority for dex agents like monitoring or
logging."
```

2. Apply the YAML file by using the following command:

```
kubectl apply -f cde-fluentd-node-critical.yaml
```

3. Patch the cde-fluentd daemonset with the new priorityClass by using the following command:

```
kubectl patch daemonset cde-fluentd -n [***DEX-BASE-***] -p
'{"spec": {"template": {"spec": {"priorityClassName": "cde-f
luentd-node-critical"}}}}'
```

Replace the namespace in the command with the namespace, where cde-fluentd is running.

Cloudera Data Engineering 1.25.0 and lower versions.

N/A

DEX-18727: Spark SQL functions do not work in Spark Connect sessions

Spark Connect functions require the SPARK_CONNECT_MODE_ENABLED environment variable to be set to true. Otherwise, Spark tries to use the normal Spark context functions, which do not work in Spark Connect.

To use Spark SQL functions in Spark Connect sessions, set the SPARK_CONNECT_MODE_
ENABLED environment variable to true in the session by running the following command:

```
import os
os.environ["SPARK_CONNECT_MODE_ENABLED"]="true"
```

Cloudera Data Engineering versions 1.23.0 through 1.25.0.

N/A

DEX-18832: Update CDP CLI with the latest DEX changes

With Data Lake 7.3.1 and higher, while using the CDP CLI to create a virtual cluster, the following
error is displayed:

```
An error occurred: can't find runtime catalog ID with CDE 1.25.0
, Datalake 7.3.2, Spark 3.5.1 and given osnames (Status Code: 50
0; Error Code: UNKNOWN; Service: de; Operation: createVc; Request
 ID: 1a8b575d-cfa9-4fb7-a65b-6be6ebf97a0c;)
```

Use the Cloudera Data Engineering UI, or the Cloudera Data Engineering API to create a virtual
cluster.

Cloudera Data Engineering 1.25.0

N/A

**DEX-18481: Retry of Cloudera Data Engineering service restore via script is failing with cluster with
same ID already exists error**

If a service is deleted and later restored with the same ID, and during this process the environment is
down or deleted, you will see a "cluster with same ID already exists" error. This error occurs even if
the service is not displayed on the Cloudera Data Engineering UI.

Raise a Support ticket to request the deletion of the Cloudera Data Engineering Base Entry from the
DEX database.

Control Plane for Cloudera Data Engineering versions 1.25.0 and lower.

N/A

DEX-18456: Airflow jobs fail during node upgrade

If you have Azure Node Image Auto Upgrade enabled, during the upgrade, Cloudera Data
Engineering Airflow jobs can fail, due to node failures. The root cause is that the node image is
upgraded automatically and causes all running Airflow schedule pods to terminate and Airflow jobs
to fail.

Disable Azure Node Image Auto Upgrade.

Cloudera Data Engineering 1.25.0 and lower versions.

N/A

DEX-18440: Cloudera Data Engineering Jobs are not reflecting in the spark-history Server

In Cloudera Data Engineering services running in AWS with Spark version higher than or equal to
3.5.1 with virtual clusters having the following versions: Cloudera Data Engineering 1.22.0, 1.23.0,
1.24.1 and 1.25.0, and running behind the Non Transparent Proxies (NTP), the Spark history server
is not able to discover the correct AWS S3 bucket region, due to a change in Hadoop AWS builds.
The affected runtimes are Cloudera Data Engineering running with 7.2.18 and 7.3.1 runtimes. In
such a case, the following error is displayed in SHS pod logs and the Spark UI does not display for
the finished jobs.

```
Exception in thread "main" org.apache.hadoop.fs.s3a.AWSApiCallTi
meoutException: getFileStatus on s3a://qe-s3-bucket-longrunning/
cluster-logs/dex/cluster-zff7f26d/6fwjjzgs/eventlog: software.am
azon.awssdk.core.exception.ApiCallTimeoutException: Client execu
```

```
tion did not complete before the specified timeout configuration:
 60000 millis
 at org.apache.hadoop.fs.s3a.S3AUtils.translateException(S3AUtils
.java:222)
 at org.apache.hadoop.fs.s3a.S3AUtils.translateException(S3AUtils
.java:154)
 at org.apache.hadoop.fs.s3a.S3AFileSystem.s3GetFileStatus(S3AFil
eSystem.java:402
```

The workaround is to set the correct bucket region in the Spark history configuration using the spar k.hadoop.fs.s3a.endpoint.region parameter.

Prerequisite: kubectl access to the Cloudera Data Engineering cluster.

Steps:

1. Determine the log bucket from the "summary" page of Data Lake. For example, us-west-2.
2. Determine the Virtual Cluster (VC) namespace of the affected VC, which is the same as its VC-ID. For example: dex-app-gk9kpt24. Export it in its current shell session.

```
export DEX_APP_NS=dex-app-gk9kpt24
```

3. Use the following command to edit the Spark default configmap.

```
kubectl edit cm -n $DEX_APP_NS $DEX_APP_NS-spark-defaults
```

4. Add the spark.hadoop.fs.s3a.endpoint.region: us-west-2 property as shown below in the screenshot and save the file.

5. Use the following command to verify the changes committed in the configmap.

```
kubectl get cm -n $DEX_APP_NS $DEX_APP_NS-spark-defaults -o
yaml
```

6. Restart the Spark History Server (SHS) pod for the changes to take effect.

   a. Get the pod name:

```
kubectl get pods  -n $DEX_APP_NS
```

   b. Delete the pod:

```
kubectl delete pod  -n $DEX_APP_NS dex-app-gk9kpt24-shs-7b7c
fd94dd-8mdqh
```

After this procedure, the Spark UI displays all previous and new job runs.

Cloudera Data Engineering 1.22.0, 1.23.0, 1.24.1 and 1.25.0

N/A

DEX-18305: Cannot create VC with latest 1.25.0 Control Plane on Cloudera Data Engineering Service 1.24.1-h1-b14 without specifying sparkVersion in the request

In Cloudera Data Engineering versions lower than 1.25.0, the createInstance API and the corresponding CDP CLI to create virtual clusters work when you do not specify a Spark version. In such a case, by default, a Spark version is selected.

In Cloudera Data Engineering 1.25.0, the createInstance API operates as follows:

- If you specify the Spark version, the createInstance API and the CDP CLI work.
- If you do not specify the Spark version, the createInstance API and the CDP CLI give an error.

Cloudera Data Engineering 1.25.0.

N/A

DEX-18203: Suspending multiple VCs simultaneously is leading to race condition

Suspending multiple Virtual Clusters (VCs) simultaneously can cause inconsistent state to the service.

Suspend VCs in a sequential manner.

Cloudera Data Engineering 1.25.0.

N/A

DEX-17369: Cloudera Data Engineering Service's Resource Edit is Failing with status 500: [error validating instance group update request]

You cannot edit the Auto-scaling Range for Spot and On-demand instances in both Core and All-Purpose tiers, after the Cloudera Data Engineering service is created. The Cloudera Data Engineering service edit operation fails if you try to edit these parameters after the service creation.

Raise a ticket with Cloudera Support to guide you through the changes needed for editing the Auto-scaling Range manually at the cloud provider.

Cloudera Data Engineering 1.23.0 and higher versions

N/A

DEX-16495: Write Spark DF/Stream to Hive using Hive Streaming fails with "org.apache.hive.streaming.InvalidTable"

In Spark 3.3.0, there is an issue with writing Spark Dataframe to Hive using Hive Streaming, that is DATAFRAME_TO_STREAM. The failure is related to compatibility issues between Spark 3.3.0 runtime components and Hive client libraries.

Cloudera Data Engineering Hive Warehouse Connector on Spark 3.3.0 with Cloudera Data Engineering 1.25.0

N/A

DEX-16492: Write Spark DF/Stream to Hive using Hive Streaming fails with "org.apache.hadoop.fs.s3a.impl.InstantiationIOException"

This issue affects both AWS and Azure. During the release of Spark 3.5.1, supported by the 7.2.18 runtime, the RAZ library has gone through backward incompatible changes and it requires in core-site.xml to remove two parameters and to add two new parameters. This has been achieved by unsetting and setting the required parameters through the runtime API server.

It has been observed that a few integration parameters, like Hive Warehouse Connector (HWC) parameters, do not accept these passed-in parameters and read only parameters defined in core-site.xml. As a result, a manual workaround is needed for the affected customers to fix the core-

site.xml by removing the required parameters and adding the new parameters for the affected Virtual Clusters (VCs).

Parameters to be removed from core-site.xml:

- fs.s3a.custom.signers
- fs.s3a.s3.signing-algorithm

Parameters to be added to core-site.xml:

- spark.hadoop.fs.s3a.http.signer.class: org.apache.ranger.raz.hook.s3.RazS3SignerPlugin
- spark.hadoop.fs.s3a.http.signer.enabled: true

Necessary user rights, privileges: Kubectl access is required to the Cloudera Data Engineering cluster.

Prerequisites

1. kubectl access to the Cloudera Data Engineering cluster
2. jq xmlstarlet base64 must be installed on the local machine

CLI procedure and example

Steps:

1. Download the raz-signer-params-workaround.sh script.
2. Make sure to install all the dependencies on the host machine.

    a. kubectl
    b. xmlstarlet
    c. base64
3. Make the script executable.

```
chmod +x ./raz-signer-params-workaround.sh
```

4. Find the namespace of the affected VC. It is the name of the VC. Example:

```
dex-app-c59x5d57
```

5. Use the run_test command to apply the changes and pass the VC namespace as an argument to the script, as shown in the following example.

```
./raz-signer-params-workaround.sh run_test dex-app-c59x5d57
```

After running the script successfully, the org.apache.hadoop.fs.s3a.impl.InstantiationIOException error is not displayed while running the jobs again.

Cloudera Data Engineering 1.23.0, 1.23.1, 1.24.1, and 1.25.0.

N/A

DEX-16426: Error fetching data from Hive Managed Table in JDBC_CLUSTER mode

If you use Spark Hive Warehouse Connector and Spark version 3.5.1, there is an issue related to fetching data from Hive managed tables in JDBC_CLUSTER read mode, which results in a syntax error.

Error message:

```
[PARSE_SYNTAX_ERROR] Syntax error at or near '.' (line 1, pos 21)
```

Pass the following job level configurations:

--conf   spark.sql.legacy.useV1Command=true

--conf   spark.sql.legacy.v1IdentifierNoCatalog=true

Cloudera Data Engineering 1.25.0.

N/A

DEX-17969: Airflow failed to connect to Impala in Cloudera Data Engineering

Due to a Python thrift-related error, which is a dependency of Impyla, Airflow fails to connect to Impala in Cloudera Data Engineering. This issue results in a system error related to the PY_SSIZE _T_CLEAN macro when using fast binary encoding.

While setting up the connection, include the following line after the connect command and before the cursor() elements:

```
con.service.client._iprot._fast_decode = None
```

Cloudera Data Engineering 1.24.1 and higher versions

Cloudera Data Engineering 1.25.0

DEX-17581: Cloudera Data Engineering-1.24.1 is not getting deployed in East US region

Only applicable to Azure. Cloudera Data Engineering service creation failed during the database server provisioning step. The issue occurred because the Azure API, which Cloudera Data Engineering uses to retrieve the supported database instance types for the specified region (for example, eastus), returned an empty response. As a result, the database server provisioning could not proceed. The following error message appeared in the Cloudera Data Engineering service logs:

unable to get MySQL flexible   server DB instance type for cluster, Error: no instance types availab le for MySQL   flexible server DB service tier: GeneralPurpose having vCores 2

Cloudera has raised a support ticket with Microsoft regarding this issue. According to their response, the empty API response for the specified location occurred because the quota for Azure MySQL Flexible Server was either unavailable or disabled in the given region for the subscription. If you encounter the same issue, contact Microsoft Support and request them to enable the quota for MySQL Flexible Servers in the affected region for your subscription.

DEX-17565: Links to download cdeconnect and pyspark tars for Spark Connect are giving HTTP 404 error

Links to download cdeconnect and pyspark   tars for Spark Connect give an HTTP 404 error.

Replace 7.2.18.800 with 7.2.18.0 in the URL.

Cloudera Data Engineering 1.25.0

DEX-17519: Sessions are not killed as per the ttl configured in Azure and AWS

Sessions are not killed as per the ttl configured in Azure and in AWS. The calculation of timeout has gone wrong in the isTimeout method in the Livy code. This method takes a calculated timeout in milliseconds and converts it into nano seconds. However, the caller is already passing the calculated timeout value in nano seconds. In the isTimeout method, the calculatedTimeout value is converted again, which provides a different value. Therefore, (toTime -   fromTime) will not be greater than the calculated timeout, as the calculated timeout value is higher. For this reason, the sessions are not killed after the timeout is reached.

Cloudera Data Engineering 1.24.1-H1

DEX-17507: Restore of Scheduled Jobs are failing due to time format

Restoring the Spark Jobs with the Schedule Configuration fails if the start date or end date uses a time format other than RFC3339Nano. This issue affects only jobs created using non-UI options, such as the API or CLI.

Before taking the backup, edit the schedule configuration of the affected Spark Job to use the RFC3339Nano time format.

Cloudera Data Engineering 1.24.1-H1

DEX-17500: [CDP Cli] Spark OsName "chainguard" Not Triggering Error in Cloudera Data Engineering Version 1.23.1 Virtual Cluster

Cloudera Data Engineering allows the creation of a Virtual Cluster with the securityhardened option in Cloudera Data Engineering version 1.23.1, without any error message. Technically, it is using UBI [redhat] underneath, which is correct, but it can lead to confusion, as the property in the Virtual Cluster states securityhardened.

Avoid using the CDP CLI to provide the spark.osname[="securityhardened"] to create a Virtual Cluster, as it is not supported in Cloudera Data Engineering versions lower than 1.24.1.

Cloudera Data Engineering 1.24.1-H1

DEX-17458: Cloudera Data Engineering session creation is failing with java.util.concurrent.ExecutionException: javax.security.sasl.SaslException

Cloudera Data Engineering sessions created in a Spark 3.3.0 Virtual Cluster fail to create. The following error is listed in the driver logs:

Exception in thread "main" java.util.concurrent.ExecutionException: javax.security.sasl.SaslException: Client closed before SASL negotiation   finished

Cloudera Data Engineering 1.24.1-H1

DEX-16747: Cloudera Data Engineering 1.23.1-b114 - Driver container stderr, and stdout logs are missing for some Spark jobs

For some job runs, intermittently, the driver stderr and stdout logs are missing.

DEX-16414: Sessions GET endpoint not returning empty array

When no sessions are present in a Virtual Cluster, the Sessions page on the Cloudera Data Engineering UI displays 'Loading' state, instead of empty state.

Upgrade to Cloudera Data Engineering version 1.24.1, or higher versions.

Alternatively, maintain at least one session in the Virtual Cluster. Using the CDE CLI, create at least one session in the Virtual Cluster, and to avoid resource consumption, kill the same session.

Cloudera Data Engineering 1.25.0

DEX-15884: Resource file upload did not pick the modified file intermittently

When you attempt to update a file by uploading a new version with the exact same filename, the operation appears to succeed, but the content of the file is not updated. The system continues to serve the previous version of the file. This issue has been observed to occur intermittently under the following conditions:

- Uploading a file to overwrite an existing file with the same name.
- Deleting the original file first and then uploading a new file with the same name.

Specify a different filename for the resource.

DEX-15714: Proxy settings are not propagating to Cloudera Data Engineering sessions

Proxy settings from a configured CDP proxy (configmap: cdp-proxy-config) are not propagated to Cloudera Data Engineering sessions. Proxy settings for Cloudera Data Engineering jobs are propagated through spark.driver.extraJavaOptions and spark.executor.extraJavaOptions, as standard JAVA_OPTS. For more information, see Cloudera public proxy documentation.

Add the proxy settings manually to spark.executor.extraJavaOptions and spark.driver.extraJavaOptions and after that create the session.

DEX-15461: Writing Spark Dataframe to Hive using HWC Fails with java.util.NoSuchElementException : None.get

This is a known issue while writing data in ORC format. The issue has been fixed internally, but more testing is needed. This issue will be part of the Hive Warehouse Connector and Cloudera Data Engineering certification in the future.

Cloudera Data Engineering 1.25.0

DEX-14725: virtualenv cannot access pypi mirror

When a Python virtual environment is created, virtual-env needs to access the internet to seed packages such as pip, setup-tools, and wheel. If you block the public internet access (for example, in case of a private network), certain packages fail to build. Example package: requests-kerberos

Use custom Docker images. Example: https://community.cloudera.com/t5/Community-Articles/ Creating-Custom-Runtimes-with-Spark3-Python-3-9-on-Cloudera/ta-p/368867.

DEX-14385: Backup fails if there is a Git repository resource

In the Cloudera Data Engineering 1.20.3 services, if there is a Git repository resource, the cluster backup fails.

Remove the Git repository.

Cloudera Data Engineering 1.21.0

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

Cloudera Data Engineering 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, Cloudera Data Engineering 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 Cloudera Data Engineering service.
2. Run the following command, which exposes the label manually in the prometheus-kube-state-metrics container:

```
`kubectl patch deployment monitoring-prometheus-kube-state-m
etrics --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 Cloudera Data Engineering 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 Cloudera Data Engineering upgrade. If they are running during the Cloudera Data Engineering upgrade, kill them before proceeding.

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-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: Cloudera Data Engineering Spark Jobs are getting stuck due to the unavailability of the spot instances

The unavailability of AWS spot instances may cause Cloudera Data Engineering 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/Cloudera Data Engineering 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-13957: Cloudera Data Engineering metrics and graphs show no data

Cloudera Data Engineering 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 Cloudera Data Engineering UI and in charts.

As a prerequisite, set a kube-config for the Cloudera Data Engineering service and run:

```
kubectl 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 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.

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

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

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

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

In Cloudera Data Engineering 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-9879: Infinite while loops not working in Cloudera Data Engineering 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 Cloudera Data Engineering commands.

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

In Cloudera Data Engineering 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 Cloudera Data Engineering 1.19, only unbalanced brackets will generate a new line. In Cloudera Data Engineering 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 Cloudera 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-3706: The Cloudera Data Engineering home page not displaying for some users

The Cloudera Data Engineering home page will not display Virtual Clusters or a Quick Action bar if the user is part of hundreds of user groups or subgrooups.

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 Cloudera Data Engineering 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 Cloudera Data Engineering 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 Cloudera Data Engineering 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 Cloudera Data Engineering. These include converting Scala objects, including RDD Dataset DataFrame, and Columns. For example, the following operations will fail on Cloudera Data Engineering:

```
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 ca
se 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("""{"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-7051 EnvironmentPrivilegedUser role cannot be used with Cloudera Data Engineering

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

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

Cloudera Data Engineering 1.19.2

Strict DAG declaration in Airflow 2.2.5

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

If you upgrade to Cloudera Data Engineering 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 Cloudera Data Engineering 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-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.

YK 0.10.2

Technical service bulletins

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

TSB 2025-837: Cloudera Data Engineering Airflow errors leading to hung jobs and SLA misses

Issues in Cloudera Data Engineering on cloud releases that have been upgraded to Airflow version 2.9 may lead to Directed Acyclic Graph (DAG) runs hanging and SLA misses. Cloudera has released a patch to address these issues, which needs to be manually applied in affected environments.

Knowledge article

For the latest update on this issue see the corresponding Knowledge Base article: TSB 2025-837: Cloudera Data Engineering Airflow errors leading to hung jobs and SLA misses

TSB 2025-797: Using Spark vectorized reads on Iceberg tables with Parquet files could result in malformed data

Due to an upstream Apache Iceberg issue, in specific situations, customers executing Spark vectorized reads on Iceberg tables with Parquet files may encounter malformed output and data incorrectness.

For the latest update on this issue see the corresponding Knowledge article: TSB 2025-797: Using Spark vectorized reads on Iceberg tables with Parquet files could result in malformed data

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

Learn about the Cloudera Data Engineering 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

Cloudera Data Engineering 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.

Spark job schedule or configuration changes modify the job owner

Any modifications to the schedule or the configuration of a Spark job in Cloudera Data Engineering changes the job owner to the user who last edited it. This behavior can lead to access issues, as it affects job run permissions.

You can assign the job ownership to a service account, but you cannot log in with a service account through the Cloudera Data Engineering UI. As a workaround, use the CDE CLI with the service account to edit the Spark job schedule or configuration and to reset the job ownership.

Usernames on Cloudera Data Engineering virtual clusters

Cloudera Data Engineering virtual clusters require lowercase usernames.

Cloudera Data Engineering API for Airflow Operators and Libraries

The Cloudera Data Engineering API for Airflow Operators and Libraries currently expects Base64-encoded certificates, but the Cloudera Data Engineering UI sends PEM certificates without conversion. As a workaround, go to Cloudera Data Engineering UI Administration Virtual Cluster Details of the selected virtual cluster Configuration tab Airflow Libraries and Operators Configure Repositories SSL Certificate and provide the certificate in Base64 format.

Cloudera Data Engineering Runtime end of support

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

In the Support lifecycle policy page, navigate to Cloudera Data Services on cloud (formerly, CDP Public Cloud, Data Services) Existing versions EOS dates Cloudera on Cloud - Spark Runtime .

The Cloudera on Cloud - Spark Runtime 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 table.

Frequently Asked Questions (FAQs)

What is the relationship between Cloudera Data Engineering Spark runtimes and the Data Lake version?

Each Cloudera Data Engineering 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 Cloudera Data Engineering and Data Lake on a regular basis to ensure continued support and compatibility.

Does Cloudera Data Engineering offer Long-term support (LTS) releases?

Cloudera Data Engineering offers LTS through underlying Spark runtimes. When running Spark jobs within Cloudera Data Engineering, 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 Cloudera Data Engineering 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 on 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 Cloudera Data Engineering 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 - on cloud section in Support lifecycle policy.

Compatibility for Cloudera Data Engineering and Runtime components

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

⚠️ **Important:** If you switch from the Redhat UBI image to the security hardened image, note that the updated component versions and libraries can impact the Cloudera Data Engineering operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

For backward compatibility, you can continue using the Redhat UBI image. The security hardened image is the recommended image; however, note that it requires testing and validation.

For more information, see Security hardened Spark image migration guide.

⚠️ **Important:** Cloudera Data Lake 7.3.1 supports virtual cluster creation from Apache Spark version 3.5 onwards. Earlier Spark versions are not supported with Cloudera Data Lake 7.3.1.

Table 1: Cloudera Data Engineering compatibility with Runtime component details

Cloudera Data Engineering Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Kubernetes
1.25.0 Security hardened	• N/A	• Spark 3.2.3 • Scala 2.12.10 • Java 11 • Python 3.9 • Iceberg 1.3.1 • Data Lake 7.2.17 and 7.2.18	• Spark 3.3.0 • Scala 2.12.15 • Java 17 • Python 3.10 • Iceberg 1.3.1 • Data Lake 7.2.17 and 7.2.18	• Spark 3.5.1 • Scala 2.12.18 • Java 17 • Python 3.11 • Iceberg 1.5.2 • Data Lake 7.2.18 and 7.3.1	• Airflow 2.11.0 • Python 3.11 • Java 17	1.32
1.25.0 Redhat UBI (deprecated)	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0 • Python 2.7 and 3.6 • Iceberg N/A • Data Lake 7.2.17 and 7.2.18	• Spark 3.2.3 • Scala 2.12.10 • Java 11 • Python 3.6 • Iceberg 1.3.1 • Data Lake 7.2.17 and 7.2.18	• Spark 3.3.0 • Scala 2.12.15 • Java 11 • Python 3.8 • Iceberg 1.3.1 • Data Lake 7.2.17 and 7.2.18	• Spark 3.5.1 • Scala 2.12.18 • Java 11 • Python 3.8 • Iceberg 1.5.2 • Data Lake 7.2.18 and 7.3.1	• Airflow 2.11.0 • Python 3.11 • Java 17	1.32
1.24.1 Security hardened	• N/A	• Spark 3.2.3 • Scala 2.12.10 • Java 11 • Python 3.9 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.3.0 • Scala 2.12.15 • Java 17 • Python 3.10 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.5.1 • Scala 2.12.18 • Java 17 • Python 3.11 • Iceberg 1.5.2 • Data Lake 7.2.18 and 7.3.1	• Airflow 2.10.4 • Python 3.11 • Java 17	1.31
1.24.1 Redhat UBI (deprecated)	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0 • Python 2.7 and 3.6 • Iceberg N/A • Data Lake 7.2.16-7.2.18	• Spark 3.2.3 • Scala 2.12.10 • Java 11 • Python 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.3.0 • Scala 2.12.15 • Java 11 • Python 3.8 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.5.1 • Scala 2.12.18 • Java 11 • Python 3.8 • Iceberg 1.5.2 • Data Lake 7.2.18 and 7.3.1	• Airflow 2.9.3 • Python 3.11 • Java 17	1.31

Cloudera Data Engineering Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Kubernetes
1.23.1	• Spark 2.4.8 • Python 2.7 and 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.2.3 • Python 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.3.0 • Python 3.8 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.5.1 • Python 3.8 • Iceberg 1.5.2 • Data Lake 7.2.18 and 7.3.1	• Airflow 2.9.3 • Python 3.11 • Java 17	1.30
1.23.0	• Spark 2.4.8 • Python 2.7 and 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.2.3 • Python 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.3.0 • Python 3.8 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.5.1 • Python 3.8 • Iceberg 1.5.2 • Data Lake 7.2.18	• Airflow 2.9.3 • Python 3.11	1.30
1.22.0	• Spark 2.4.8 • Python 2.7 and 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.2.3 • Python 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.3.0 • Python 3.8 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.5.1 • Python 3.8 • Iceberg 1.4.3 • Data Lake 7.2.18	• Airflow 2.7 • Python 3.8	1.28
1.21	• Spark 2.4.8 • Python 2.7 and 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.2.3 • Python 3.6 • Iceberg 1.3.1 • Data Lake 7.2.16-7.2.18	• Spark 3.3.0 • Python 3.8 • Iceberg 1.3.1 • Data lake 7.2.16-7.2.18	N/A	• Airflow 2.7 • Python 3.8	1.27
1.20.3	• Spark 2.4.8 • Python 2.7 • Iceberg 1.3.1 • Data lake 7.2.15-7.2.17	• Spark 3.2.3 • Python 3.6 • Iceberg 1.3.1 • Data lake 7.2.15-7.2.17	• Spark 3.3.0 • Python 3.8 • Iceberg 1.3.1 • Data lake 7.2.15-7.2.17	N/A	• Airflow 2.6.3 • Python 3.8	1.27
1.19.4	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.15-7.2.17 • Iceberg 1.1.0	• Spark 3.2.3 • Scala 2.12.10 • Java 11.0.20 • Python 3.6 • Data lake 7.2.15-7.2.17 • Iceberg 1.1.0	• Spark 3.3.0 • Scala 2.12.15 • Java 11.0.20 • Python 3.8 • Data lake 7.2.15-7.2.17 • Iceberg 1.1.0	N/A	• Airflow 2.3.4 • Python 3.8	1.26
1.19.3	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.15-7.2.17	• Spark 3.2.3 • Scala 2.12.10 • Java 11.0.20 • Python 3.6 • Data lake 7.2.15-7.2.17	• Spark 3.3.0 • Scala 2.12.15 • Java 11.0.20 • Python 3.8 • Data lake 7.2.15-7.2.17	N/A	• Airflow 2.3.4 • Python 3.8	1.25
1.19.2	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.14-7.2.16 • Iceberg 1.1.0	• Spark 3.2.3 • Scala 2.12.10 • Java 11.0.20 • Python 3.6 • Data lake 7.2.14-7.2.16 • Iceberg 1.1.0	• Spark 3.3.0 • Scala 2.12.15 • Java 11.0.20 • Python 3.8 • Data lake: 7.2.16 • Iceberg 1.1.0	N/A	• Airflow 2.3.4 • Python 3.8	1.25

Cloudera Data Engineering Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Kubernetes
1.19	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.14-7.2.16 • Iceberg 1.1.0	• Spark 3.2.3 • Scala 2.12.10 • Java 11.0.20 • Python 3.6 • Data lake 7.2.14-7.2.16 • Iceberg 1.1.0	• Spark 3.3.0 • Scala 2.12.15 • Java 11.0.20 • Python 3.8 • Data lake: 7.2.16 • Iceberg 1.1.0	N/A	• Airflow 2.3.4 • Python 3.8	1.25
1.18.3	• Spark 2.4.8 • Scala: 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.14-7.2.16 • Iceberg 0.14.1	• Spark 3.2.3 • Scala 2.12.10 • Java 11.0.20 • Python 3.6 • DL: 7.2.14-7.2.16 • Iceberg 0.14.1	N/A	N/A	• Airflow 2.3.4 • Python 3.8	1.24
1.18.1	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.14-7.2.16 • Iceberg 0.14.1	• Spark 3.2.0 • Scala 2.12.10 • Java 11.0.20 • Python 3.6 • Data lake 7.2.14-7.2.16 • Iceberg 0.14.1	N/A	N/A	• Airflow 2.3.4 • Python 3.8	1.23
1.18	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Data lake 7.2.14-7.2.15 • Iceberg 0.14.1	• Spark 3.2.0 • Data lake 7.2.14-7.2.15 • Iceberg 0.14.1	N/A	N/A	• Airflow 2.2.5 • Python 3.8	1.23
1.17-h1	• Spark 2.4.8 • Scala 2.11.12 • Java 1.8.0_352 • Python 2.7 • Iceberg 0.14.1	• Spark 3.2.1 • Iceberg 0.14.1	N/A	N/A	• Airflow 2.2.5 • Python 3.8	1.23

AWS Graviton component version compatibility

To use AWS Graviton in Cloudera Data Engineering, you need the following runtime component versions:

• Spark 3.3 and 3.5
• Data Lake 7.2.18 and 7.3.1

Using the Cloudera Runtime Maven repository for Cloudera Data Engineering

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

You can determine the Cloudera Data Engineering version for your virtual cluster by clicking the [icon] 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>
```

Cloudera Data Engineering version	Cloudera Runtime Maven repository
• 1.20.3 • 1.19.4 • 1.19.3 • 1.19.2	• Cloudera Runtime 7.2.15 • Cloudera Runtime 7.2.16
• 1.19	• Cloudera Runtime 7.2.16 • Cloudera Runtime 7.2.15
• 1.18 • 1.17 • 1.16	Cloudera Runtime 7.2.15
• 1.15	Cloudera Runtime 7.2.14
• 1.14	Cloudera Runtime 7.2.11.0
• 1.13	Cloudera Runtime 7.2.11.0
• 1.12 • 1.11 • 1.9	Cloudera Runtime 7.2.10.0
• 1.8 • 1.7	Cloudera Runtime 7.2.8.0
• 1.6 • 1.5 • 1.4 • 1.3 • 1.2	Cloudera Runtime 7.2.2.0