

Cloudera Data Engineering 1.15.1

Cloudera Data Engineering Release Notes

Date published: 2020-07-30

Date modified: 2023-06-13

CLOUDERA

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

Legal Notice

© Cloudera Inc. 2024. All rights reserved.

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

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

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

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

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

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

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

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

Contents

What's new in Cloudera Data Engineering Private Cloud.....	4
Known issues and limitations in Cloudera Data Engineering on CDP Private Cloud.....	5
Fixed issues in Cloudera Data Engineering on CDP Private Cloud.....	10
Creating Cloudera Data Engineering Virtual Cluster without installing Atlas in your CDP Base cluster.....	10
Compatibility for Cloudera Data Engineering and Runtime components.....	10

What's new in Cloudera Data Engineering Private Cloud

This release of Cloudera Data Engineering (CDE) on CDP Private Cloud 1.5.1 includes the following features:

Support for Apache Iceberg V2 (Technical Preview)

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

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

Apache Iceberg V1 and V2 is in technical preview in the Private Cloud 1.5.1 release and is not recommended for production deployments. Cloudera recommends that you try this feature in test or development environments.

For more information, see [Using Apache Iceberg in Cloudera Data Engineering](#).

Ozone integration using Data Connectors

Apache Ozone is an object store available on the CDP Private Cloud Base cluster which enables you to optimize storage for big data workloads. You can now create and manage CDE clusters with Ozone as a backend storage provider.

For more information, see [Using Ozone storage with Cloudera Data Engineering](#).

Sessions (Technical Preview)

A Cloudera Data Engineering (CDE) Session is an interactive short-lived development environment for running Spark commands to help you iterate upon and build your Spark workloads.

Sessions is in technical preview in this release and is not recommended for production deployments. Cloudera recommends that you try this feature in test or development environments.

For more information, see [Creating Sessions in Cloudera Data Engineering](#).

Updates to the Spark-submit migration tool

We now offer a container image where you can run the migration tool without installing it. It enables you to take the advantage of the migration tool on any operating system.

For more information, see [Using spark-submit drop-in migration tool](#).

Upgrading CDE Service with Endpoint Stability

You can seamlessly upgrade an old Cloudera Data Engineering (CDE) service to new version with endpoint stability. This enables you to access the CDE service of the new version with the previous endpoint. Thus, you can use the existing endpoints without changing configurations at the application level.

The CDE service endpoint migration process lets you migrate your resources, jobs, job run history, spark jobs' logs and event logs from your old cluster to the new cluster.

For more information, see [Upgrading CDE Service with Endpoint Stability](#).

Elastic Quota for Virtual Clusters (Technical Preview)

You can configure elastic quota to a virtual cluster (VC) to get a minimum guaranteed and maximum capacity of resources (CPU and memory) as guaranteed quota and maximum quota. The guaranteed quota dictates the minimum

amount of resources available for allocation for a VC at all times. The resources above the guaranteed quota and within the VC's maximum quota can be used by any VC on demand if the cluster capacity allows for it.

Elastic quotas allow the VC to acquire unused capacity in the cluster when their guaranteed quota limit gets exhausted. This ensures efficient use of resources in the cluster. At the same time, the maximum quota limits the threshold amount of resources a VC can claim in the cluster at any given time.

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

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](#).

Creating a custom Airflow Python environment (Technical Preview)

Cloudera Data Engineering (CDE) supports a custom Python environment in Airflow to manage job dependencies using the `airflow-python-env` resource type.

For more information, see [Creating a custom Airflow Python environment](#).

Pulse metrics

CDE provides Virtual Cluster-level metrics for tracking job runs, database requests, HTTP server requests, LDAP calls, Kubernetes nodes and other components related to the environment. These metrics are stored in and monitored by Prometheus.

For more information, see [Monitoring metrics for Data Engineering](#).

Gang scheduling support for Spark applications

Previously, Yunikorn scheduled the Spark applications only when the application's minimal resource request was satisfied. Otherwise, applications will be waiting in the queue. Applications are queued in hierarchy queues. Gang scheduling is now enabled by default and each resource queue is assigned the maximum number of applications running concurrently with minimum resources guaranteed.

Upgrade to Airflow 2.3.4

CDE PVC 1.5.1 now runs with Airflow 2.3.4. This upgrade includes several fixes to improve performance and stability.

Known issues and limitations in Cloudera Data Engineering on CDP Private Cloud

This page lists the current known issues and limitations that you might run into while using the Cloudera Data Engineering (CDE) service.

DEX-14676: Deep Analysis is not working in CDE PvC under analysis tab

If you are using Spark version 2.x for running your jobs, then the Run Deep Analysis feature present under the Analysis tab is not supported on Cloudera Data Engineering Private Cloud.

DEX-8540: Job Analysis tab is not working

When you access the Jobs Runs Analysis tab through the Cloudera Data Engineering UI, the Analysis tab fails to load data for Spark 2.

To view the data in the Job Analysis tab, open the JOBS API URL from the Virtual Cluster details page and access the Analysis tab.

DEX-10939: Running the `prepare-for-upgrade` command puts the workload side database into read-only mode.

Running the `prepare-for-upgrade` command puts the workload side database into read-only mode. If you try to edit any resources or jobs or run jobs in any virtual cluster under the CDE service for which the `prepare-for-upgrade` command was executed, The MySQL server is running with the `--read-only` option so it cannot execute this statement error is displayed.

This means that all the APIs that perform write operations will fail for all virtual clusters. This is done to ensure that no changes are done to the data in the cluster after the `prepare-for-upgrade` command is executed, so that the new restored cluster is consistent with the old version.

You must ensure that you have sufficient time to complete the entire upgrade process before running the `prepare-for-upgrade` command.

DOCS-17844: Logs are lost if the log lines are longer than 50000 characters in fluentd

This issue occurs when the `Buffer_Chunk_Size` parameter for the `fluent-bit` is set to a value that is lesser than the size of the log line.

The values that are currently set are:

```
Buffer_Chunk_Size=50000
Buffer_Max_Size=50000
```

When required, you can set higher values for these parameters in the `fluent-bit` configuration map which is present in the `dex-app-xxx` namespace.

DEX-10576: Builder job does not start automatically when the resource is restored from an archive.

For the airflow python environment resource, the restoration does not work as intended. Though the resource is restored, the build process is not triggered. Even if the resource was activated during backup, it is not reactivated automatically. This leads to job failure during restoration or creation, if there is a dependency on this resource.

You can use the CDE API or CLI to download the `requirements.txt` file and upload it to the resource. You can activate the environment if required.

```
# cde resource download --name <python-environment-name> --resource-path requirements.txt
# cde resource upload --name <python-environment-name> --local-path requirements.txt
```

DEX-10147: Grafana issue if the same VC name is used under different CDE services which share same environment

In CDE 1.5.1, when you have two different CDE services with the same name under the same environment, and you click the Grafana charts for the second CDE service, metrics for the Virtual Cluster in the first CDE service will display.

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

DEX-10116: Virtual Cluster installation fails when Ozone S3 Gateway proxy is enabled

Virtual Cluster installation fails when Ozone S3 gateway proxy is enabled. Ozone s3 gateway proxy gets enabled when more than one Ozone S3 Gateway is configured in the CDP Private Cloud Base cluster.

Add the `127.0.0.1 s3proxy-<environment-name>.<private-cloud-control-plane-name>-services.svc.cluster.local` entry in the `/etc/hosts` of all nodes in the CDP Private Cloud Base cluster where the Ozone S3 gateway is installed. For example, if the private cloud environment name is `cdp-env-1` and private cloud control plane name is `cdp`, then add

the `127.0.0.1 s3proxy-cdp-env-1.cdp-services.svc.cluster.local` entry in `/etc/hosts`.

DEX-10052: Logs are not available for python environment resource builder in CDP Private Cloud

When creating a python environment resource and uploading the `requirements.txt` file, the python environment is built using a k8s job that runs in the cluster. These logs cannot be viewed currently for debugging purposes using CDE CLI or UI. However, you can view the events of the job.

None

DEX-10051: Spark sessions is hung at the Preparing state if started without running the `cde-utils.sh` script

You might run into an issue when creating a spark session without initialising the CDE virtual cluster and the UI might hang in a Preparing state.

Run the `cde-utils.sh` to initialise the virtual cluster as well as the user in the virtual cluster before creating a Spark long-running session.

DEX-10055: Interacting with a killed session builder in CDP Private Cloud

The session might become unresponsive if you interact with a killed spark long-running session. Do not interact with a killed long running session.

None

DEX-9895: DEX VC API response shows spark version as 2.4.7

In CDE 1.5.1, the Spark version 3.2.3 is the expected default version in a CDE Spark Virtual Cluster, but Spark 2.4.7 is displayed instead in the CDE API. This issue will be fixed in a future release.

None

DEX-9783: While creating the new VC, it shows wrong CPU and Memory values

When clicking on the Virtual Cluster details for a Virtual Cluster that is in the Installing state, the configured CPU and Memory values that are displayed are inaccurate until the VC is created.

Refresh the Virtual Cluster details page to get the correct values, five minutes after the Virtual Cluster installation has started.

DEX-9692: CDE UI does not work when the port 80 is blocked on the k8s cluster

If your environment has blocked port 80 at the ingress level. Then the CDE UI does not work.

None

DEX-9961: CDE Service installation is failing when retrieving `aws_key_id`

CDE Service installation is failing when retrieving `aws_key_id` with the `Could not add shared cluster overrides, error: unable to retrieve aws_key_id from the env service error`.

1. Restart the Ozone service on the Cloudera Data Platform Base cluster and make sure all the components are healthy.
2. Create a new environment in Cloudera Data Platform Private Cloud using the Management Console.
3. Use the same environment for creating the CDE Service.

DEX-8996: CDE service stuck at the initialising state when a user who does not have correct permission tries to create it

When a CDE user tries to create a CDE service, it gets stuck at the initializing state and does not fail. Additionally, cleanup cannot be done from the UI and must be done on the backend.

Only the user who has the correct permission should create a CDE service. If you experience any issue, delete the stuck CDE service from the database.

DEX-8600: ECS 1.4.1 to 1.5.1 Upgrade: Virtual cluster creation and deletion is failing

Upgrading the ECS version while CDE service is enabled, causes the old CDE service and virtual cluster creation and deletion to fail. This is due to ECS upgrading to the kubernetes version 1.23 which removes the old ingress APIs used.

Back up the CDE jobs in the CDE virtual cluster, and then delete the CDE service and CDE virtual cluster. Restore it after the upgrade. For more information about backup and restore CDE jobs, see [Backing up and restoring CDE jobs](#).

DEX-7513: Patching an airflow DAG keeps the total run number even though the history is not retained

If you patches an airflow DAG with `catchup = true`, then the old run history will be deleted. You cannot see the old run logs.

Patch the DAG with `startDate` as current date to avoid rewrite the old run history that will ensure the old run history persist.

DEX-8682: CDE PvC 1.5.0 : CDP upgrade to 1.5.0 with OCP upgrade (4.8 to 4.10) Jobs UI is not opening

Upgrading the OCP version from 4.8 to 4.10 while CDE service is enabled, causes the Jobs UI to not open. This is due to OCP 4.10 upgrading to the Kubernetes version 1.23 which removes the old ingress APIs used.

Back up CDE jobs in the CDE virtual cluster, and then delete the CDE service and CDE virtual cluster. Restore it after the upgrade. For more information about backup and restore CDE jobs, see [Backing up and restoring CDE jobs](#).

DEX-8614: Sometimes Spark job is not getting killed even though its parent Airflow job gets killed

Sometimes if an issue is encountered while sending the request to kill a spark batch to the Livy API and the error is logged but not propagated properly to the Airflow job. In such cases, the underlying spark job might still be running, though the airflow job considers that the job is killed successfully.

Kill the spark job manually using CDE user interface, CLI, or API.

DEX-8601: ECS 1.4.x to 1.5.0 Upgrade: jobs fail after upgrade

Upgrading the ECS version while CDE service is enabled, causes the jobs launched in the old CDE virtual cluster fail. This is due to ECS upgrading to the kubernetes version 1.23 which removes the old ingress APIs used.

Back up CDE jobs in the CDE virtual cluster, and then delete the CDE service and CDE virtual cluster. Restore it after the upgrade. For more information about backup and restore CDE jobs, see [Backing up and restoring CDE jobs](#).

DEX-8226: Grafana Charts of new virtual clusters will not be accessible on upgraded clusters if virtual clusters are created on existing CDE service.

If you upgrade the cluster from 1.3.4 to 1.4.x and create a new virtual clusters on the existing CDE Service, Grafana Charts will not be displayed. This is due to broken APIs.

Create a new CDE Service and a new virtual cluster on that service. Grafana Charts of the virtual cluster will be displayed.

DEX-7000: Parallel Airflow tasks triggered at exactly same time by the user throws the 401:Unauthorized error.

Error 401:Unauthorized causes airflow jobs to fail intermittently, when parallel Airflow tasks using CDEJobRunOperator are triggered at the exact same time in an Airflow DAG.

Using the below steps, create a workaround bashoperator job which will prevent this error from occurring. This job will keep running indefinitely as part of the workaround and should not be killed.

1. Navigate to the Cloudera Data Engineering Overview page by clicking the Data Engineering tile in the Cloudera Data Platform (CDP) console.
2. In the CDE Services column, select the service containing the virtual cluster where you want to create the job.

3. In the Virtual Clusters column on the right, click the View Jobs icon on the virtual cluster where you want to create the job.
4. In the left hand menu, click Jobs.
5. Click Create Job.
6. Provide the job details:
 - a. Select Airflow for the job type.
 - b. Specify the job name as bashoperator-job.
 - c. Save the following python script to attach it as a DAG file.

```
from dateutil import parser
from airflow import DAG
from airflow.utils import timezone
from airflow.operators.bash_operator import BashOperator
default_args = {
    'depends_on_past': False,
}
with DAG(
    'bashoperator-job',
    default_args = default_args,
    start_date = parser.isoparse('2022-06-17T23:52:00.123Z')
    .replace(tzinfo=timezone.utc),
    schedule_interval = None,
    is_paused_upon_creation = False
) as dag:
    [ BashOperator(task_id = 'task1', bash_command = 'sleep
infinity'),
      BashOperator(task_id = 'task2', bash_command = 'sleep in
finity') ]
```

- d. Select File, click Select a file to upload the above python, and select a file from an existing resource.
7. Select the Python Version, and optionally select a Python Environment.
8. Click Create and Run.

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

Irrespective of who submits the Airflow job, the Airflow job is run with the user privileges who created the job. This causes issues when the job submitter has lesser privileges than the job owner who has higher privileges.

Spark and Airflow jobs must be created and run by the same user.

Changing LDAP configuration after installing CDE breaks authentication

If you change the LDAP configuration after installing CDE, as described in [Configuring LDAP authentication for CDP Private Cloud](#), authentication no longer works.

Re-install CDE after making any necessary changes to the LDAP configuration.

HDFS is the default filesystem for all resource mounts

For any jobs that use local filesystem paths as arguments to a Spark job, explicitly specify file:// as the scheme. For example, if your job uses a mounted resource called test-resource.txt, in the job definition, you would typically refer to it as /app/mount/test-resource.txt. In CDP Private Cloud, this should be specified as file:///app/mount/test-resource.txt.

Scheduling jobs with URL references does not work

Scheduling a job that specifies a URL reference does not work.

Use a file reference or create a resource and specify it

Limitations

Access key-based authentication will not be enabled in upgraded clusters prior to CDP PVC 1.3.4 release.

After you upgrade to PVC 1.3.4 version from earlier versions, you must create the CDE Base service and Virtual Cluster again to use the new Access Key feature. Otherwise, the Access Key feature will not be supported in the CDE Base service created prior to the 1.3.4 upgrade.

Fixed issues in Cloudera Data Engineering on CDP Private Cloud

Review the list of issues that are resolved in the Cloudera Data Engineering (CDE) service in the CDP Data Services 1.5.1 release.

DEX-9237: Job fails with the “Permission Denied” error after updating the virtual cluster resource quota.

With this fix, whenever the virtual cluster resource quota is updated, the newly launched jobs on the virtual cluster does not fail with the Permission Denied error.


Creating Cloudera Data Engineering Virtual Cluster without installing Atlas in your CDP Base cluster

If the Cloudera Data Engineering Virtual Cluster creation fails because Atlas is not installed, you must identify the CDE Namespace and set an environment variable prior to creating the Virtual Cluster.

Procedure

1. Identify the CDE Namespace

- a. In the Cloudera Data Platform (CDP) console, click the Data Engineering tile. The CDE Home page displays.
- b.

In the CDE Services column, click  for the CDE service you want to create a VC.

- c. Note the Cluster ID shown on the page and identify the CDE Namespace. For example, if the Cluster ID is cluster-sales8098, then the CDE Namespace is *dex-base-sales8098*.

2. Use this CDE Namespace (*dex-base-sales8098*) to run Kubernetes commands using kubectl or OpenShift's command line oc.

kubectl

```
kubectl set env deployment/dex-base-configs-manager -c dex-base-configs-manager ATLAS_CONFIGS_DISABLED=true --namespace <CDE Namespace>
```

oc

```
oc set env deployment/dex-base-configs-manager -c dex-base-configs-manager ATLAS_CONFIGS_DISABLED=true --namespace <CDE Namespace>
```

Compatibility for Cloudera Data Engineering and Runtime components

Learn about Cloudera Data Engineering (CDE) and compatibility for Runtime components across different versions.

Table 1: CDE compatibility with Runtime component details

Runtime Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Airflow	Iceberg	Kubernetes
7.1.7 SP 2	<ul style="list-style-type: none">• Spark 2.4.8• Scala 2.11• Python 2.7• Python 3.6	<ul style="list-style-type: none">• Spark 3.2.3• Scala 2.12.10• Python 3.6	NA	<ul style="list-style-type: none">• Airflow 2.3.4• Python 3.8	Iceberg 0.14.1	1.24
7.1.8	<ul style="list-style-type: none">• Spark 2.4.8• Scala 2.11• Python 2.7• Python 3.6	<ul style="list-style-type: none">• Spark 3.2.3• Scala 2.12.10• Python 3.6	NA	<ul style="list-style-type: none">• Airflow 2.6.3• Python 3.8	Iceberg 0.14.1	1.24