

Cloudera Data Engineering 1.5.4

# Cloudera Data Engineering Release Notes

Date published: 2020-07-30

Date modified: 2024-12-20

# CLOUDERA

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

# Legal Notice

© Cloudera Inc. 2025. All rights reserved.

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

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

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

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

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

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

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

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

# Contents

<b>What's new in Cloudera Data Engineering on premises.....</b>	<b>4</b>
<b>Known issues and limitations in Cloudera Data Engineering on Cloudera on premises.....</b>	<b>4</b>
<b>Fixed issues in Cloudera Data Engineering on Cloudera on premises.....</b>	<b>9</b>
<b>Creating Cloudera Data Engineering Virtual Cluster without installing Atlas in your Cloudera Base cluster.....</b>	<b>9</b>
<b>Compatibility for Cloudera Data Engineering and Cloudera Runtime components.....</b>	<b>10</b>
<b>Service Packs: Cloudera Data Engineering on premises.....</b>	<b>10</b>
Cloudera Data Engineering on premises 1.5.4 SP1.....	11
Cloudera Data Engineering on premises 1.5.4 SP2.....	12

## What's new in Cloudera Data Engineering on premises

Review the new features in the Cloudera Data Engineering service of the Cloudera Data Services 1.5.4 release.

### Support for password protected private key to initialize the virtual cluster

You can now use the password protected private keys to initialize the virtual cluster. Currently, the password protected private keys are supported with RSA and EC algorithms only. For more information, see [Initializing virtual clusters](#).

### 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](#).

## Known issues and limitations in Cloudera Data Engineering on Cloudera on premises

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

### DEX-8542: Newly created Iceberg tables are owned by "sparkuser"

The Iceberg tables created in Cloudera Data Engineering using Spark 3.2.3 are being displayed as owned by the "sparkuser" user. The Iceberg tables must be owned by the user who created them. For example,

```
hive=> SELECT "TBL_NAME", "OWNER" FROM "TBLS" WHERE "TBL_NAME" = '
iceberg_test';
  TBL_NAME      | OWNER
-----+-----
iceberg_test   | sparkuser
```

Spark 3.2.3 uses Iceberg version 0.14, which is causing this issue. Create and use a Cloudera Data Engineering Virtual Cluster with Spark version 3.3.2 which is not affected by this.

### DEX-14676: Deep Analysis is not working in Cloudera Data Engineering on premises 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 on premises.

### DEX-12150: Recursive search for a file in resources is not working

If you search for any file using the Search field in the Resources page, the result does not display any files present with that name inside the resources.

Navigate to the relevant resource and then locate the file in that resource.

### 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-12426: Data Connector UI does not load

Data Connector UI does not load for self-signed certificates if the browser certificates are not trusted.

When utilising the self-signed certificates, it is crucial to trust the certificates associated with the Cloudera Data Engineering URLs. Perform the following:

1. Using the Cloudera Data Engineering UI, install a virtual cluster (VC) on the cluster.
2. Initialize the installed VC using the `cde-utils.sh` script.
3. Open the **Jobs UI** in a new tab, which prompts the acknowledgement and trust of the certificates, accept it.
4. Refresh the **Data Connectors** page.

You can now access Grafana and the Data Connectors UI.

#### **DEX-11300: Editing the configuration of a job created using a Git repository shows Resources instead of Repository**

Jobs which use application file from *REPOSITORIES* when edited, shows Resources as a source under Select application file. This issue does not affect the functionality of the job but could confuse as it displays the source as a Resource for the application even if the selected file is from a repository. Though it would show Resource in this case, in the backend it is selected from the chosen repository.

#### **DEX-11340: Sessions go to unknown state if you start the Cloudera Data Engineering upgrade process before killing live Sessions**

If spark sessions are running during the Cloudera Data Engineering upgrade then they are not be automatically killed which can leave them in an unknown state during and after the upgrade.

You must kill the running Spark Sessions before you start the Cloudera Data Engineering upgrade.

#### **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 Cloudera Data Engineering 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-XXXX` namespace.

#### **DOCS-18585: Changes to the log retention configuration in the existing virtual cluster do not reflect the new configuration**

When you edit the log retention policy configuration for an existing virtual cluster, the configuration changes are not applied.

When you edit the log retention policy configuration, you must restart the runtime-api-server pod using the `kubectl rollout restart deployment/<deployment-name> -n <namespace>` command to apply the changes.

For example:

```
kubectl rollout restart deployment/dex-app-fw61rgm-api -n dex-app-fw61rgm
```

#### **DEX-11231: In OpenShift, the Spark 3.3 virtual cluster creation fails due to Airflow pods crashing**

This is an intermittent issue during virtual cluster installation in the OCP cluster where the airflow-scheduler and airflow-webserver pods are stuck in the `CRASHLOOPBACKOFF` state. This leads to virtual cluster installation failure.

Retry the virtual cluster installation because the issue is intermittent.

#### **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 Cloudera Data Engineering 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 Cloudera Data Engineering services which share same environment**

In Cloudera Data Engineering 1.5.1, 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. After you verified that everything in the new upgraded Cloudera Data Engineering service, the old Cloudera Data Engineering 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 Cloudera Base on premises 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 Cloudera Base on premises cluster where the Ozone S3 gateway is installed. For example, if the on premises environment name is `cdp-env-1` and on premises 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 Cloudera on premises**

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 Cloudera Data Engineering 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-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 for 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-9961: Cloudera Data Engineering Service installation is failing when retrieving aws\_key\_id**

Cloudera Data Engineering Service installation is failing when retrieving `aws_key_id` with the error: 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 Base cluster and make sure all the components are healthy.
2. Create a new environment in Cloudera on premises using the Management Console.
3. Use the same environment for creating the Cloudera Data Engineering Service.

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

When a Cloudera Data Engineering user tries to create a Cloudera Data Engineering 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 Cloudera Data Engineering service. If you experience any issue, delete the stuck Cloudera Data Engineering service from the database.

**DEX-8226: Grafana Charts of new virtual clusters will not be accessible on upgraded clusters if virtual clusters are created on existing Cloudera Data Engineering service**

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

Create a new Cloudera Data Engineering 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 console.
2. In the Cloudera Data Engineering 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 Cloudera Data Engineering breaks authentication**

If you change the LDAP configuration after installing Cloudera Data Engineering, as described in [Configuring LDAP authentication for Cloudera on premises](#), authentication no longer works.

Re-install Cloudera Data Engineering 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 Cloudera on premises, 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

**DEX-13775: The synchronization operation fails when using a non-default branch from the Git repository with Cloudera Data Engineering Git repositories**

When you use a non-default branch from a Git repository with the Cloudera Data Engineering Git repositories, the synchronization operation fails.

Clone the Git repository from the non-default branch again after the latest commit.



## Fixed issues in Cloudera Data Engineering on Cloudera on premises

Review the list of issues that are resolved in the Cloudera Data Engineering service in the Cloudera Data Services 1.5.4 release.

### **DEX-9692: Cloudera Data Engineering UI does not work when the port 80 is blocked on the Kubernetes cluster**

The Cloudera Data Engineering service UI loads correctly after authentication even if a firewall is configured to block all the requests going to the Kubernetes cluster at port 80.

### **DEX-11508: The file modification time in the Modified column is not updated on the Repositories page**

The GIT Repository modified time shows the correct modified time after syncing the files.

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

A Spark job gets successfully killed if a request to kill a Spark batch is sent to the Livy API.

### **DEX-13597: Cloudera Data Engineering jobs enabled with GPU acceleration gets stuck if gang scheduling is enabled on the virtual cluster**

The Cloudera Data Engineering jobs enabled with GPU acceleration run correctly even if they are running on a virtual cluster where gang scheduling is enabled.

### **DEX-13155: Restoring virtual cluster fails during upgrade if the SMTP username or password are missing**

A virtual cluster is restored successfully during the upgrade even if the SMTP username or password are missing or not available.

### **DEX-12096: Virtual clusters are not being restored with the same original app tier value and the log retention policy**

When you back up and restore a Cloudera Data Engineering service, virtual cluster is restored with the same original app tier value with which they are created and the same log retention policy.

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

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

### **Procedure**

#### **1. Identify the Cloudera Data Engineering Namespace**

**a.** In the Cloudera console, click the Data Engineering tile. The Cloudera Data Engineering Home page displays.

**b.**

In the Cloudera Data Engineering Services column, click  for the Cloudera Data Engineering service you want to create a VC.

**c.** Note the Cluster ID shown on the page and identify the Cloudera Data Engineering Namespace. For example, if the Cluster ID is cluster-sales8098, then the Cloudera Data Engineering Namespace is *DEX-BASE-SALES8098*.

- Use this Cloudera Data Engineering 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 Cloudera Runtime components

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

**Table 1: Cloudera Data Engineering compatibility with Cloudera Runtime component details**

Cloudera Runtime Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.4.x	Airflow	Iceberg	Kubernetes
7.1.7 SP 2	<ul style="list-style-type: none"> <li>Spark 2.4.7</li> <li>Scala 2.11</li> <li>Python 2.7</li> <li>Python 3.6</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Python 3.6</li> </ul>	NA	NA	<ul style="list-style-type: none"> <li>Airflow 2.6.3</li> <li>Python 3.8</li> </ul>	Iceberg 0.14.1	1.27
7.1.8	<ul style="list-style-type: none"> <li>Spark 2.4.7</li> <li>Scala 2.11</li> <li>Python 2.7</li> <li>Python 3.6</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.10</li> <li>Python 3.6</li> </ul>	NA	<ul style="list-style-type: none"> <li>Spark 3.4.1</li> <li>Scala 2.12.17</li> <li>Python 3.8</li> </ul>	<ul style="list-style-type: none"> <li>Airflow 2.6.3</li> <li>Python 3.8</li> </ul>	Iceberg 0.14.1 (Not applicable for Spark 3.4.x)	1.27
7.1.9	<ul style="list-style-type: none"> <li>Spark 2.4.7</li> <li>Spark 2.4.8</li> <li>Scala 2.11</li> <li>Python 2.7</li> <li>Python 3.6</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.2.3</li> <li>Scala 2.12.15</li> <li>Python 3.6</li> </ul>	<ul style="list-style-type: none"> <li>Spark 3.3.2</li> <li>Scala 2.12.15</li> <li>Python 3.8</li> </ul>	NA	<ul style="list-style-type: none"> <li>Airflow 2.6.3</li> <li>Python 3.8</li> </ul>	Iceberg 1.3.0	1.27



### Important:

- From the next major release after Cloudera Data Services on premises 1.5.4, Spark 3.5 will be supported and Spark 3.4 will no longer be available. Also, upgrading from Spark 3.4 to Spark 3.5 will not be supported. You must take a backup of the jobs and metadata that are created in the Spark 3.4 virtual cluster and then restore them in the Spark 3.5 virtual cluster.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported only on Spark 3.3.2.

## Service Packs: Cloudera Data Engineering on premises

Review the Service pack features and issues for Cloudera on premises 1.5.4.

## Cloudera Data Engineering on premises 1.5.4 SP1

Review the features, fixes, and known issues in the Cloudera Data Engineering 1.5.4 Service Pack 1 release.

### What's new in 1.5.4 SP1

This section lists the new features added in this release for Cloudera Data Engineering on premises.

#### Support for SAML authentication

Cloudera Data Engineering supports users to configure using SAML as Identity Provider along with LDAP. For more information about how to configure users using SAML authentication, see [Configuring users for Identity Providers](#).

#### Support for NTP proxy setup

Cloudera Data Engineering requires specific proxy configurations to manage virtual cluster connections efficiently in an air-gapped setup with restricted outbound connections. This setup ensures seamless access to external resources while adhering to network security and management policies. For more information about how to setup NTP proxy, see [NTP proxy setup on Cloudera Data Engineering](#).

### Known issues in 1.5.4 SP1

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

#### **DEX-15429: Data connectors jobs are failing with "UNKNOWN: Channel Pipeline: [ProtocolNegotiators \$ProxyProtocolNegotiationHandler#0, WriteBufferingAndExceptionHandler#0]" in the proxy setup**

Ozone data connector does not work when proxy is enabled.

Add the IP addresses of all the base cluster nodes to the **NO\_PROXY** list so that request to Ozone does not use proxy.

#### **DEX-15472: Cloudera Data Engineering service upgrade fails during backup and restore when SAML is enabled**


Script-based upgrade of Cloudera Data Engineering service doesn't not work when SAML is enabled.

Make sure that the Cloudera Data Engineering service is upgraded using the dex-upgrade-utils solution before enabling the SAML (both IdP and SP initiated) in the Control Plane.

#### **DEX-15735: The dbus-wxm-client pod appears in the dex-base namespace after editing the CPU and Memory quota the first time**

After editing the Resource Pool CPU or Memory for a Cloudera Data Engineering Service for the first time, dbus-wxm-client pod gets deployed in the Cloudera Data Engineering Service namespace. The dbus-wxm-client pod keeps failing and cause alerts in the Management Console. However, this will not cause any other issues in Cloudera Data Engineering.

The dbus-wxm-client pod deployment is not needed and can be safely deleted to fix the issue. To delete it, do these steps:

1. Identify the Cloudera Data Engineering Service namespace:
  - a. In the Cloudera console, click the Data Engineering tile. The Cloudera Data Engineering home page displays.
  - b. On the left navigation menu, click Administration.
  - c.
    - In the Services column, click  for the Cloudera Data Engineering service for which you want to identify the namespace.
  - d. Note the Cluster ID displayed on the page and identify the Cloudera Data Engineering service namespace. For example, if the Cluster ID is cluster-sales8098, then the Cloudera Data Engineering service namespace is dex-base-sales8098.
  - e. Note this namespace.
2. Using the Cloudera Data Engineering Service Namespace, run this Kubernetes command to delete the deployment:

```
kubectl delete deployment dex-base-dbus-wxm-client -n [***CDE-SERVICE-NAMESPACE***]
```

For example,

```
kubectl delete deployment dex-base-dbus-wxm-client -n dex-base-sales8098
```

#### **DEX-15825: Unable to update the resource quota for Cloudera Data Engineering virtual clusters**

Editing the virtual cluster resource quota on Administration Cluster Details Configuration Resource fails for Cloudera Data Engineering. The "Virtual Cluster could not be updated" error message is displayed on the UI if you try to update the virtual cluster CPU, Memory, or the GPU limits.

In the virtual cluster, update the CPU, Memory, and GPU Guaranteed and Maximum values together and click Update.

#### **DEX-15827: In Cloudera Data Engineering on premises, the GPU option is not displayed when creating the virtual cluster**

In Cloudera Data Engineering on premises, the GPU option is not displayed for Spark 3.3.2 and Spark 3.2.3 on the Administration Create a Virtual Cluster UI page.

Create a virtual cluster without GPU. Open the virtual cluster for editing and set the GPU limits.

#### **Fixed issues in 1.5.4 SP1**

Review the list of issues that are resolved in the Cloudera Data Engineering service in the Cloudera Data Services on premises 1.5.4 SP1 release.

#### **DEX-14484: Cloudera Data Services on premises- Fix dex-runtime-python-builder to use python 3.8 even for Spark 3.3+**

In Spark 3.3.2 virtual cluster, Python 3.8 is now used in Python virtual environment resource instead of old Python 3.6 which is not supported in Spark 3.3 or higher.

## **Cloudera Data Engineering on premises 1.5.4 SP2**

Review the features, fixes, and known issues in the Cloudera Data Engineering 1.5.4 Service Pack 2 release.

#### **Known issues in 1.5.4 SP2**

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

#### **DEX-15472: [stop-gap upgrade] Support for SAML enabled cluster during backup and restore**

Script-based upgrade does not work when SAML is enabled.

Make sure that Cloudera Data Engineering is upgraded using the dex-upgrade-utils solution before enabling SAML by initiating Identity Provider or Service Provider in the Cloudera Control Plane. For information on downloading the dex-upgrade-utils solution, see [Prerequisites for upgrading Cloudera Data Engineering service with endpoint stability](#).

### Fixed issues in 1.5.4 SP2

Review the list of issues that are resolved in the Cloudera Data Engineering service in the Cloudera Data Services on premises 1.5.4 SP2 release.

#### **DEX-10052: Logs are not available for Python environment resource builder in Cloudera Data Engineering on premises**

If you create a Python environment resource and upload the requirements.txt, it triggers a build job, whose logs were missing. This issue is fixed now, and you can view all the logs for the Python environment in Cloudera Data Engineering on premises.