Cloudera AI 1.5.4

Release Notes

Date published: 2020-07-16
Date modified: 2024-12-16

https://docs.cloudera.com/

Legal Notice

Contents

What's New

Cloudera on premises 1.5.4 includes the following features for Cloudera AI.

New features

Cloudera AI Service Accounts are available in Cloudera AI on premises

In Cloudera AI, the Kerberos principal for the Service Account may not be the same as your login information. Therefore, ensure you provide the Kerberos identity when you sign in to the Service Account. For more information, see Authenticating Hadoop for Cloudera AI Service Accounts.

Cloudera AI Registry is available in Cloudera on premises

Cloudera AI Registry is now generally available (GA) in on premises. Cloudera AI Registry in on premises uses Apache Ozone to store model artifacts. For creating a Cloudera AI Registry you need the Ozone S3 gateway endpoint, the Ozone access key, and the Ozone secret key.

If you deploy Cloudera AI Registry in an environment that contains one or more Cloudera AI Workbenches, you must synchronize the Cloudera AI Registry with the workbenches. For more information, see Prerequisites for creating Cloudera AI Registry and Synchronizing the Cloudera AI Registry with a workbench.

Heterogeneous GPU usage

When using heterogeneous GPU clusters to run sessions and jobs, the available GPU accelerator labels need to be selected during workload creation. For more information, see Heterogeneous GPU clusters.

Data connections without auto discovery

Cloudera AI is a flexible, open platform, supporting connections to many data sources. The provided code samples demonstrate how to access local data for Cloudera AI workloads. For more information, see Connecting to Cloudera Data Warehouse.

Spark Log4j Configuration

Cloudera AI allows you to update Spark's internal logging configuration on a per-project basis. Spark logging properties can be customized for every session, and job with a default file path found at the root of your project. You can also specify a custom location with a custom environment variable. For more information, see Spark Log4j Configuration.

ML Metrics Collector service

The Metrics Collector service gathers data about how users and groups use resource quota, like how much CPU, Memory and GPU capacity (if any) is allocated, and what the users or groups utilize from that. The Metrics Collector service is running by default, but to collect data about resource quota metrics, you need to enable the Quota Management feature. For more information, see ML Metrics Collector Service overview.

Quota Management for group level

Quota Management Technical Preview (TP) release enables you to control how resources are allocated within your Cloudera AI Workbench on user and on group level. Yunikorn Gang Scheduling is also available, which is the default scheduling mechanism in Cloudera AI. For more information, see Quota Management overview and Yunikorn Gang Scheduling.

Restarting a failed AMP setup

You can now retry failed AMP deployment steps and continue the AMP setup to handle intermittent and configuration issues. For more information, see Restarting a failed AMP setup.

New Hadoop CLI Runtime Addon versions are available

The HadoopCLI 7.1.8.3-601 Runtime Addon is released for the Cloudera on premises.

Related Information

Known Issues and Limitations

Fixed Issues

This section lists issues fixed in this release for Cloudera AI on premises.

DSE-24333: Embedded Spark UI fails to load in Spark pushdown mode for Spark 3

When Spark is run on yarn, the configuration attempts to disable proxying of the Spark UI through the Yarn Resource Manager. This setting was not respected by Spark 3 binaries and resulted in the Could not determine the proxy server for redirection error message when attempting to reach the embedded Spark UI on the driver.

This behavior works again in Spark 3.2.3 with CDH 7.1.7 and Spark 3.3.2 with CDH 7.1.9.

DSE-32887: Runtime updates cannot be enabled from the Cloudera Manager UI Site Administration dashboard

Runtime updates can now be enabled from the Cloudera AI UI Site Administration dashboard.

DSE-29976: Job Arguments fail in PBJ runtimes

PBJ (Powered by Jupyter) Runtimes were not able to access Job arguments as standard command line arguments. To mitigate this, a new environment variable was added to Jobs, called "JOB_ARGUMENTS". This environment variable can be used to access arguments specified in the Job Creation or Job Update UI.

DSE-27285: The calculation of User Resources was incorrect

The **User Resources** tab in the Cloudera AI Workbench UI calculated with the used and available resources for a cluster that also included the resources for the master node. As the workload cannot be scheduled in the master nodes, the master node resources shall not be considered in these calculations. This issue has been solved, the resources for the master are no longer included in the calculations for the User Resources.

Known Issues

You might run into some known issues while using Cloudera AI on premises.

DSE-44901: Possible occurrence for incorrect status for successful workloads

A potential race condition in the reconciler can result in the status of successful workloads being incorrectly updated to unknown or failed status.

The issue occurs due to high system load, which leads to incorrect status reporting after pod deletion.

DSE-6499: Using dollar character in environment variables in Cloudera AI

Environment variables with the dollar ($) character are not parsed correctly by Cloudera AI. For example, if you set PASSWORD="pass$123" in the project environment variables, and then try to read it using the echo command, the following output will be displayed: pass23

Workaround: Use one of the following commands to print the $ sign:

```
echo 24 | xxd -r -p
or
echo JAo= | base64 -d
```

Insert the value of the environment variable by wrapping it in the command substitution using $() or ``. For example, if you want to set the environment variable to ABC$123, specify:

```
ABC$(echo 24 | xxd -r -p)123
or
ABC`echo 24 | xxd -r -p`123
```

DSE-37827: Jupyter's RTC extension throws an error and notebooks become unusable

In certain cases, Jupyter's RTC (Real Time Collaboration) extension may cause errors claiming either that other sessions are active, or that other processes have accessed the notebook files. After these errors, the notebook becomes unusable due to the error messages and the Cloudera AI session needs to be restarted.

Workaround:

You must disable the Jupyter RTC extension by performing the following tasks:

1. Create a Session.
2. Open the terminal.
3. Enter `nano /home/cdsw/.jupyter/labconfig/page_config.json`.
4. Add the following lines to the file:

```
{
  "disabledExtensions": {
    "@jupyter/collaboration-extension": true
  },
  "lockedExtensions": {
    "@jupyter/collaboration-extension": true
  }
}
```

5. Save and close the file.

DSE-36718: Disable auto synchronization feature for users and teams

The automated team and user synchronization feature is disabled. Newly installed or upgraded workbenches do not have the automatic synchronization option in the Cloudera AI UI.

Workaround: none

DSE-36759: AMPs and Feature Announcement sections do not work in NTP setups

Cloudera AI on premises setups with Non Transparent Proxy do not function properly, that affects Cloudera Accelerators for Machine Learning Projects and Feature Announcements. The home page freezes, the feature announcement displays error message, and the AMPs do not load.

Workaround:

To avoid the home page freeze copy the following environment variables from the web deployment, and add them to the environment section of the API deployments:

- HTTP_PROXY
- HTTPS_PROXY
- NO_PROXY
- http_proxy
- https_proxy
- no_proxy

Note: The values are the same for both the lower and upper case variants.

DSE-32943: Enabling Service Accounts

Teams in the Cloudera AI Workbench can only run workloads within team projects with the Run as option for service accounts if they have previously manually added service accounts as a collaborator to the team.

DSE-35013: First Cloudera AI Workbench creation fails

On RHEL 8.8, during the first Cloudera AI Workbench installation on GPU with Cloudera Embedded Container Service external registry, pods might get stuck in the init or CrashLoop state.

First-time workbench installation is expected to fail. Consider this as a test workbench, and apply the following manual workaround for creating subsequent workbenches:

1. Restart or delete the pods which are in init or CrashLoop state in the test workbench.
2. Once all pods are in the running state, create new workbenches as needed.
3. Delete the test workbench from the Cloudera AI UI if no longer needed.

OPSX-4603: Buildkit in Cloudera Embedded Container Service in Cloudera AI on premises

Issue: BuildKit was introduced in Cloudera Embedded Container Service for building images of models and experiments. BuildKit is a replacement for Docker, which was previously used to build images of Cloudera AI's models and experiments in Cloudera Embedded Container Service. Buildkit is only for OS RHEL8.x and CentOS 8.x.

Buildkit in Cloudera AI on premises 1.5.2 is a Technical Preview feature. Hence, having Docker installed on the nodes/hosts is still mandatory for models and experiments to work smoothly. Upcoming release will be completely eliminating the dependency of Docker on the nodes.

Workaround: None.

DSE-32285: Migration: Migrated models are failing due to image pull errors

Issue: After CDSW to Cloudera AI migration (on-premises) via full-fledged migration tool, migrated models on Cloudera AI Workbench on premises fails on initial deployment. This is because the initial model deployment tries to pull images from on-premises's registry.

Workaround: Redeploy the migrated model. As this involves the build and deploy process, the image will be built, pushed to the on premises Cloudera AI Workbench's configured registry, and then the same image will be consumed for further usage.

DSE-28768: Spark Pushdown is not working with Scala 2.11 runtime

Issue: Scala and R are not supported for Spark Pushdown.

Workaround: None.

DSE-32304: On Cloudera AI on premises on Cloudera Embedded Container Service terminal and ssh connections can terminate

Issue: In Cloudera on premises on Cloudera Embedded Container Service, Cloudera AI Terminal and SSH connections can terminate after an uncertain amount of time, usually after 4-10 minutes. This issue affects the usage of local IDEs to work with Cloudera AI, as well as any customer application using a websocket connection.

Workaround: None.

DSE- 35251: Web pod crashes if a project forking takes more than 60 minutes

The web pod crashes if a project forking takes more than 60 minutes. This is because the timeout is set to 60 minutes using the `grpc_git_clone_timeout_minutes` property. The following error is displayed after the web pod crash:

```
2024-04-23 22:52:36.384   1737    ERROR      AppServer.VFS.grpc
                    crossCopy grpc error    data = [{"error":"1"},
{"code":4,"details":"2","metadata":"3"},"Deadline exceeded",{}]
          ["Error: 4 DEADLINE_EXCEEDED: Deadline exceeded\n
at callErrorFromStatus (/home/cdswint/services/web/node_modules/
```

```
@grpc/grpc-js/build/src/call.js:31:19)\n    at Object.onReceiveS
tatus (/home/cdswint/services/web/node_modules/@grpc/grpc-js/bui
ld/src/client.js:192:76)\n    at Object.onReceiveStatus (/home/c
dswint/services/web/node_modules/@grpc/grpc-js/build/src/client-
interceptors.js:360:141)\n    at Object.onReceiveStatus (/home/c
dswint/services/web/node_modules/@grpc/grpc-js/build/src/client-
interceptors.js:323:181)\n    at /home/cdswint/services/web/node
_modules/@grpc/grpc-js/build/src/resolving-call.js:94:78\n    at
 process.processTicksAndRejections (node:internal/process/task_q
ueues:77:11)\nfor call at\n    at ServiceClientImpl.makeUnaryReq
uest (/home/cdswint/services/web/node_modules/@grpc/grpc-js/buil
d/src/client.js:160:34)\n    at ServiceClientImpl.crossCopy (/ho
me/cdswint/services/web/node_modules/@grpc/grpc-js/build/src/mak
e-client.js:105:19)\n    at /home/cdswint/services/web/server-di
st/grpc/vfs-client.js:235:19\n    at new Promise (<anonymous>)\n
    at Object.crossCopy (/home/cdswint/services/web/server-dist/
grpc/vfs-client.js:234:12)\n    at Object.crossCopy (/home/cdswi
nt/services/web/server-dist/models/vfs.js:280:38)\n    at projec
tForkAsyncWrapper (/home/cdswint/services/web/server-dist/models/
projects/projects-create.js:229:19)"]
          node:internal/process/promises:288
          triggerUncaughtException(err, true /* fromPromise */);
          ^Error: 4 DEADLINE_EXCEEDED: Deadline exceeded
          at callErrorFromStatus (/home/cdswint/services/web/n
ode_modules/@grpc/grpc-js/build/src/call.js:31:19)
          at Object.onReceiveStatus (/home/cdswint/services/web/n
ode_modules/@grpc/grpc-js/build/src/client.js:192:76)
          at Object.onReceiveStatus (/home/cdswint/services/we
b/node_modules/@grpc/grpc-js/build/src/client-interceptors.js:36
0:141)
          at Object.onReceiveStatus (/home/cdswint/services/web/n
ode_modules/@grpc/grpc-js/build/src/client-interceptors.js:323:1
81)
          at /home/cdswint/services/web/node_modules/@grpc/grpc-
js/build/src/resolving-call.js:94:78
          at process.processTicksAndRejections (node:internal/p
rocess/task_queues:77:11)
          for call at
          at ServiceClientImpl.makeUnaryRequest (/home/cdswint/s
ervices/web/node_modules/@grpc/grpc-js/build/src/client.js:160:3
4)
          at ServiceClientImpl.crossCopy (/home/cdswint/services/
web/node_modules/@grpc/grpc-js/build/src/make-client.js:105:19)
          at /home/cdswint/services/web/server-dist/grpc/vfs-cli
ent.js:235:19
          at new Promise (<anonymous>)
          at Object.crossCopy (/home/cdswint/services/web/server-
dist/grpc/vfs-client.js:234:12)
          at Object.crossCopy (/home/cdswint/services/web/server-
dist/models/vfs.js:280:38)
          at projectForkAsyncWrapper (/home/cdswint/services/web/
server-dist/models/projects/projects-create.js:229:19) {
          code: 4,
          details: 'Deadline exceeded',
          metadata: Metadata { internalRepr: Map(0) {}, options:
 {} }
          }
```

Workaround: Increase the timeout limit, for example, to 120 minutes, using the `grpc_git_clone_timeout_minutes` property.

```
UPDATE site_config SET grpc_git_clone_timeout_minutes = <NEW
 VALUE>;
```

DSE-40198: Resolve painpoints with installations and updates of self-signed certificates

When rotating or updating the TLS certificate used by Cloudera AI, the Cloudera AI does not automatically pull the new certificate from the Cloudera Control Pane. To update Cloudera AI with a new TLS certificate, follow the steps below.

Workaround:

1. Backup the existing ConfigMap.

   Create a backup of the current private-cloud-ca-certs-pem-2 ConfigMap in your existing Cloudera AI Workbench using the following command:

   ```
   kubectl get configmap private-cloud-ca-certs-pem-2 -n [***ex
   isting CAI workbench namespace***] -o yaml > private-cloud-ca-
   certs-pem-2.backup
   ```

2. Create a temporary TLS-enabled workbench.

   Spin up a new, temporary TLS-enabled workbench in the same cluster and environment as the existing workbench. (It is not necessary for the workbench to start up correctly. You do not need to allocate a full set of resources for this cluster.)

3. Locate the ConfigMap in the new workbench.

   Once the Cloudera AI infrastructure pods in the new workbench are running, retrieve the priv ate-cloud-ca-certs-pem-2 ConfigMap using this command:

   ```
   kubectl get configmap private-cloud-ca-certs-pem-2 -n [***new
    CAI workbench namespace***] -o yaml
   ```

4. Update the existing workbench with the new certificate.

   Replace the binaryData: cacerts value in the existing ConfigMap of the Cloudera AI Workbench with the binaryData: cacerts value from the new workbench. The simplest way to perform this replacement is through the Cloudera Embedded Container Service UI. This data is a large base64-encoded string. To verify the new TLS certificate, decode the string and inspect its content using the OpenSSL tool:

   ```
     kubectl get configmap private-cloud-ca-certs-pem-2 -n [***
   new CAI workbench namespace***] -o yaml | grep cacerts | awk
    '{print $2}' | base64 -d > decoded-private-cloud-ca-certs-p
   em.pem
        while openssl x509 -noout -text; do :; done < decoded-pr
   ivate-cloud-ca-certs-pem.pem
   ```

5. Restart pods in the existing workbench. Restart the ds-cdh pod in the old namespace. Additionally, restart any other pods in the old namespace that fail to come up automatically.
6. Delete the temporary workbench. After confirming that the old Cloudera AI Workbench is functioning correctly with the updated certificate, delete the temporary workbench.

By following these steps, you can successfully update the TLS certificate for Cloudera AI while ensuring minimal disruption to your existing workbench.

DSE-39287: Job configuration update error: The job accelerator label ID is not of a valid type

You may encounter the error, The job accelerator label ID is not of a valid type when attempting to update a job. This issue prevents editing jobs through the UI. However, you can avoid recreating the job by performing the update through the API.

DSE-41898: Workload accelerators should have only one database entry for a given GPU type

Restarting Cloudera Embedded Container Service nodes creates duplicate entries in the node_labels database table for the same GPU type. This results in duplicate GPU registrations appearing under Site Administration Runtime Workload Accelerators and similarly, duplicate entries appear in the Resource Profile GPU Type dropdown menu when launching a session. Selecting a duplicate GPU entry while launching a session triggers the error: *This accelerator has been removed*.

Workaround:

1. Access the db-0 pod and specify the Workbench name:

```
kubectl exec db-0 -ti -n [***WORKBENCH NAME***] -c db - - psql
 -U sense
```

2. Check the existing entries in the node_labels table:

```
sense=# SELECT * FROM node_labels;
```

3. Delete inactive entries:

```
DELETE FROM node_labels WHERE availability = FALSE;
```

DSE-12064: Terminal remains functional after web session times out

The Cloudera AI terminal remains active even after the Cloudera AI web session has timed out.

Cumulative Hotfixes: Cloudera AI on premises

Review the cumulative hotfixes for Cloudera on premises 1.5.4.

Cloudera AI on premises 1.5.4 CHF1

Review the features, fixes, and known issues in the Cloudera AI 1.5.4 Cumulative hotfix 1 release.

Fixed issues in 1.5.4 CHF1

Review the fixed issues in the Cloudera AI 1.5.4 Cumulative hotfix 1 release.

DSE-37209: Upgrade of Cloudera AI Workbench with external NFS fails

For setups with an external network file system (NFS), the Cloudera AI Workbench upgrade flow was different from setups with internal NFSs. There was an issue with the persistent volume claim (PVC) size in external NFS flows which caused failures during the upgrade path to 1.5.4. Now, the issue has been fixed. For more details, see Cloudera Customer Advisory 2024-778: Cloudera AI Cloudera on premises Cloudera upgrade to 1.5.4 is failing with external NFS Technical Service Bulletin.

DSE-36078: Spark workload histories are not logged in Spark history server

The Spark history server did not log Spark workload history so far. This issue has been fixed and the Spark workload history is now logged in the base cluster Spark history server.

DSE-37192: Time zone discrepancy for usage API

Some data with respect to the workload run were stored in the local time zone, as a result of which the API returned empty results. By setting all the columns of the dashboard database to UTC time zone, there is no longer time zone discrepancy for the usage API. Note that the data stored prior to the upgrade remains in the previously set time zone.

DSE-36489: Opening existing JupyterLab session redirected to project page

After having logged out from the Cloudera AI UI and having logged back, the opening of an existing JupyterLab session led to the project page instead of opening the JupyterLab UI. This issue has been fixed and after a new login the JupyterLab UI can be opened.

DSE-32204: Improved error messaging during failed Kerberos authentication in session startup

If there was an issue with Kerberos credential authentication, Cloudera AI session startup hung until the regular session timed out and no warning message was displayed for the user. Now, a warning message with additional details on the status of the Kerberos credential authentication is displayed for the user along with a startup failure message after 1 minute.

DSE-36666: The migration readiness check is disrupted when engine readiness check is run

The shell command executor on CDSW could not handle the CDSW to Cloudera AI migration readiness timeout. This issue has been fixed, the timeout has been handled. Additionally, any SQL query-based readiness checks that remain unhandled will be displayed for manual validation.

DSE-36759: AMPs and Feature Announcement sections do not work in NTP setups

Cloudera AI Cloudera on premises setups with non-transparent proxy (NTP) did not function properly, affecting Cloudera Accelerators for Machine Learning Projects and feature announcements. The home page froze, the feature announcement displayed an error message, and the AMPs did not load. This issue has been fixed with the enablement of non-transparent proxy in a Cloudera on premises setup.

DSE-37454: Possible Cloudera AI terminal disconnections

The Cloudera AI terminal tended to disconnect on Cloudera Embedded Container Service because the proxy_read_timeout value on workload pods was set to a default value of 60 seconds. The proxy_read_timeout value has been increased to 121 seconds and the issue of the Cloudera AI terminal disconnecting connections has been resolved.

DSE-37610: Due to version mismatch of the jackson-module-scala_3 the mlops governance pod image failed

Earlier, the version of the jackson-module-scala_3 was updated in the mlopsgovernance/pom.xml file, but it was not updated in the parent/pom.xml file. This issue has been solved and there is no longer version mismatch for the jackson-module-scala_3.

DSE-36748: File upload issue fixed using APIv2

Earlier, when uploading a file to a sub-folder in a project, the file was added to the root folder. This issue has been fixed and now when uploading a file to a subfolder it is added to the project's sub folder.

Known issues in 1.5.4 CHF1

Review the known issues in the Cloudera AI 1.5.4 Cumulative hotfix 1 release.

DSE-36967: Namespace Termination issue when using Portworx storage

There is an issue with Portworx version lower than 3.1.1, as the namespace deletion gets stuck in terminating state. Portworx is not able to cleanly unmount and clean up the underlying resources.

Workaround: The issues is fixed with Portworx version 3.1.1. Upgrade to Portworx version 3.1.1 or to later versions.

DSE-37002: Download API using v2 python swagger client does not work

The download_project_file method does not function as expected. The v2 python swagger only allows content to be downloaded as a string and not in bytes, as described in Binary string response encoding with Python 3.

Workaround: Use download_project_file_v2 as an alternate method.

DSE-39287: Job configuration update error: The job accelerator label ID is not of a valid type

You may encounter the error, The job accelerator label ID is not of a valid type when attempting to update a job. This issue prevents editing jobs through the UI. However, you can avoid recreating the job by performing the update through the API.

Cloudera AI on premises 1.5.4 CHF2

There were no features, fixes, and known issues in the Cloudera AI 1.5.4 Cumulative hotfix 2 release.

Cloudera AI on premises 1.5.4 CHF3

Review the features, fixes, and known issues in the Cloudera AI 1.5.4 Cumulative hotfix 3 release.

Fixed issues in 1.5.4 CHF3

Review the fixed issues in the Cloudera AI 1.5.4 Cumulative hotfix 3 release.

DSE-37611: Cloudera AI v2 API deployed application did not inherit user level environment variables and site level environment variables

Applications created using APIv2 did not inherit user level environment variables and site level environment variables. This issue has been solved, and now an application created using APIv2 does not only inherit project level environment variables but also user level environment variables and site level environment variables. This ensures that the behaviour of APIv2 is in synchronisation with APIv1.

DSE-38499: Spark client configurations are not mounted to Spark executor pods

hadoop/conf/ files were mounted only on Spark engines but not to the Spark executors, resulting in a Kerberos issue in Spark applications using third-party file systems like PowerScale (Isilon). This issue has been fixed by including the hadoop_conf_dir in the Spark executor template YAML.

DSE-38627: Modifications for a job's resource profile could not be saved

The issue has been fixed and now the resource profile dropdown allows alternate selections and also the **Job Settings** page persists resource profile selection after saving.

Known issues in 1.5.4 CHF3

Review the known issues in the Cloudera AI 1.5.4 Cumulative hotfix 3 release.

DSE-39287: Job configuration update error: The job accelerator label ID is not of a valid type

You may encounter the error, The job accelerator label ID is not of a valid type when attempting to update a job. This issue prevents editing jobs through the UI. However, you can avoid recreating the job by performing the update through the API.

Service Packs: Cloudera AI on premises

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

Cloudera AI on premises 1.5.4 SP1

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

What's new in 1.5.4 SP1

Cloudera on premises 1.5.4 SP1 includes the following features for Cloudera AI.

Exploratory Data Science and Visualization

The Exploratory Data Science and Visualization feature automatically discovers the data sources available for you, from within the standard Cloudera AI user interface. Exploratory Data Science and Visualization enables the Data Scientist to connect to data sources within a project, to

explore data, its basic shape and characteristics, to create data sets to be used, to visualize data for understanding its nature, and to create dashboards for sharing purposes.

Fixed issues in 1.5.4 SP1

This section lists issues fixed in this release for Cloudera AI on premises.

DSE-39287: Job configuration update error on parameter type being invalid

The update on the job setting's workflow is fixed in on premises environments without GPUs configured.

DSE-38404: Model Monitoring AMP task failing due to a restart of the model metrics pod

The container in the Model Metrics pod restarted due to insufficient memory, causing a 502 Bad Gateway error on cdsw.read_metrics(**ipt). The memory limit for the container was set to 100 MB, which has now been increased to 1Gi.

DSE-39287: Job configuration update error: The job accelerator label ID is not of a valid type

Previously, you could encounter the error, The job accelerator label ID is not of a valid type when attempting to update a job through the UI. This issue has been resolved and now you can update job settings directly from the UI without any issues.

Known issues in 1.5.4 SP1

You might run into some known issues while using Cloudera AI on premises.

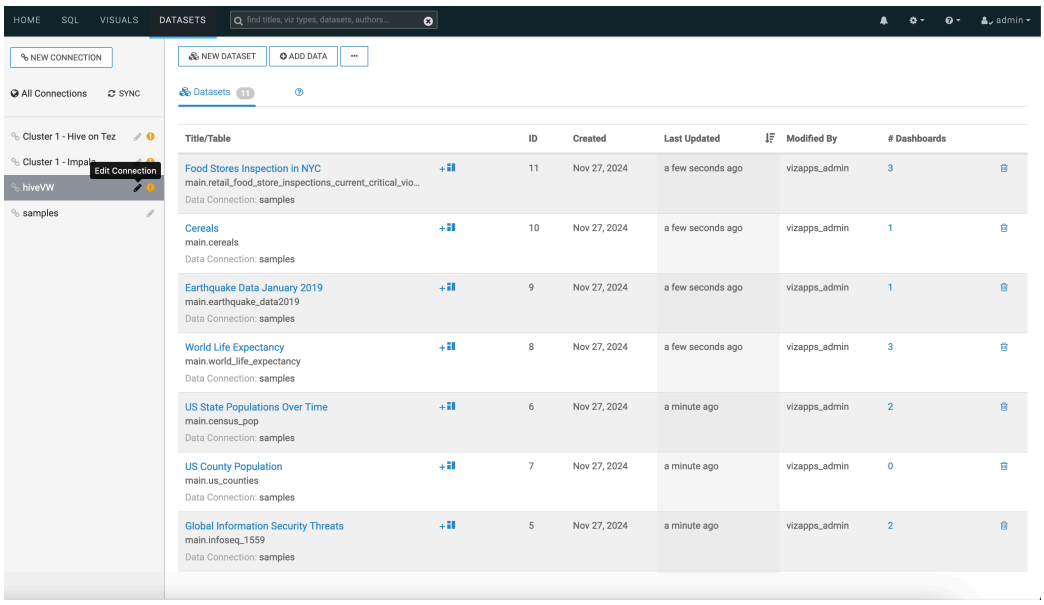
Required workarounds with Cloudera Data Visualization with Hive and Impala

Cloudera Data Visualization on premises for Hive and Impala Virtual Warehouses

For starting Cloudera Data Visualization on premises for Hive and Impala Virtual Warehouses, switch to LDAP **Authentication mode** and add your username and password in the **Edit Data Connection** page:

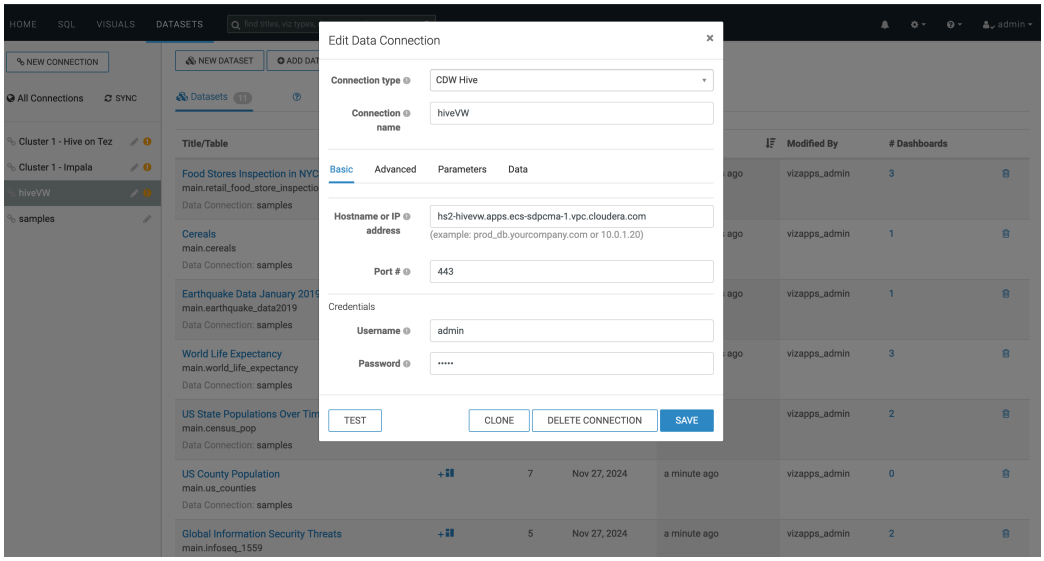
1. Select the required connection in the left navigation and click Edit Connection.

Figure 1: Selecting the required connection for editing



2. Provide your username and password.

Figure 2: Provide username and password

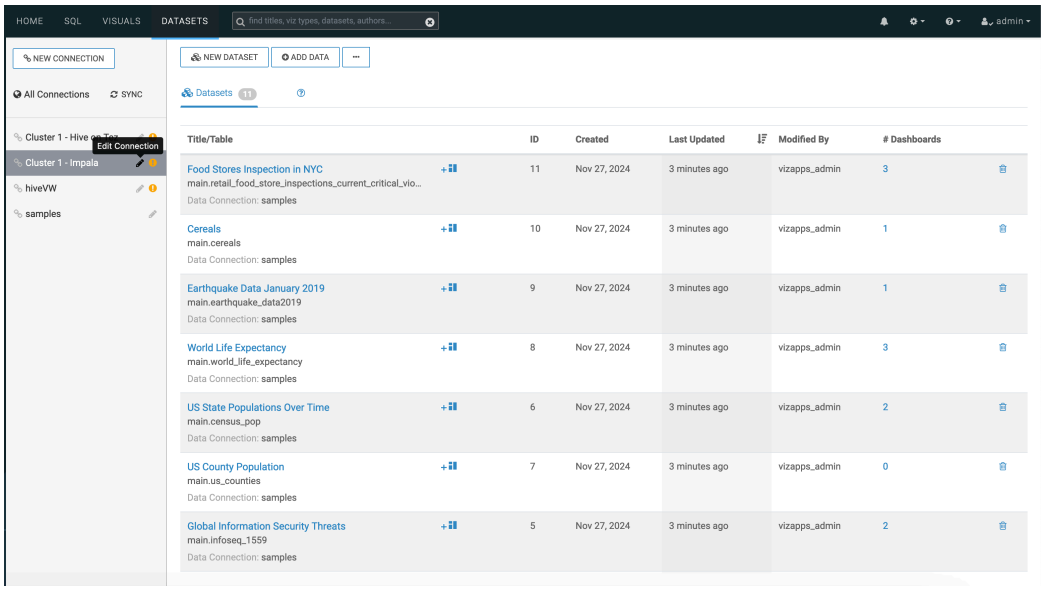


Cloudera Data Visualization on premises for Hive and Impala

For starting Cloudera Data Visualization on premises for Hive and Impala switch to Kerberos **Authentication mode** in the **Edit Data Connection** page:

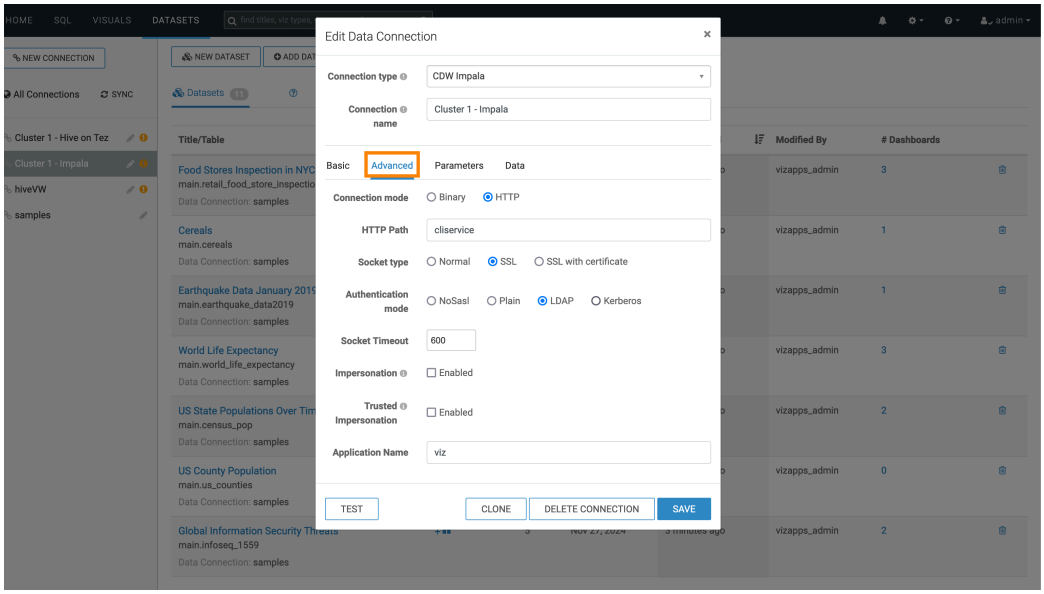
1. Select the required connection in the left navigation and click Edit Connection.

Figure 3: Selecting the required connection for editing



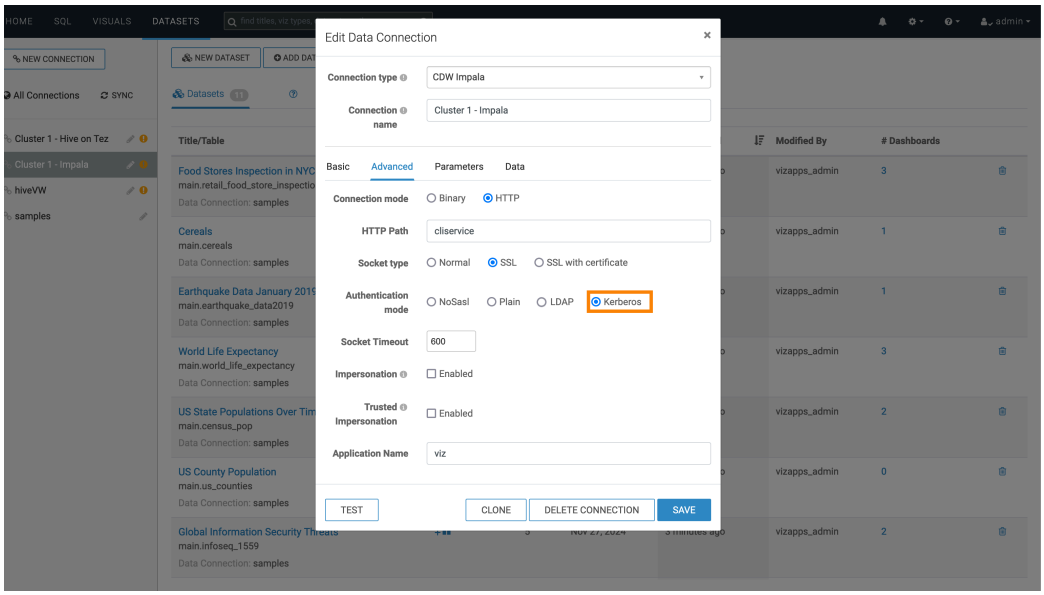
2. Select the **Advanced** tab in the **Edit Data Connection** window.

Figure 4: Selecting the Advanced tab



3. Select Kerberos as Authentication mode in the **Edit Data Connection** window.

Figure 5: Select Kerberos for authentication



Workaround:

For details see, Workarounds with Cloudera Data Visualization with Hive and Impala.

DSE-38164: AMP deployment is not possible in the proxy environment in Red Hat Openshift setup

When the model build is run, the proxy values are referenced from a file path and not from the environment variables. Consequently, the proxy values are not set properly for the models.

Workaround:

A new kubernetes secret needs to be created with the proxy values. Edit the web deployment and add this secret to the path specified.

1. Connect to the kubernetes cluster node and export KUBECONFIG.
2. Copy the script to the node and save it to file as enableProxy.sh

```
#!/bin/bash

# Function to display usage
usage() {
  echo "Usage: $0 -n namespace"
  exit 1
}

# Parse command-line arguments
while getopts ":n:" opt; do
  case ${opt} in
 n )
   NAMESPACE=$OPTARG
   ;;
 \? )
   echo "Invalid option: $OPTARG" 1>&2
   usage
   ;;
 : )
   echo "Invalid option: $OPTARG requires an argument" 1>&2
   usage
   ;;
  esac
done
shift $((OPTIND -1))

# Check if namespace is provided
if [ -z "$NAMESPACE" ]; then
  usage
fi

CONFIGMAP_NAME="cml-proxy-config"
SECRET_NAME="proxy-config"
NEW_SECRET_NAME="proxy-config-new"
DEPLOYMENT_NAME="web"
DATA_KEY="no_proxy"
SECRET_FIELD="no.proxy"
CONTAINER_NAME="web"
MOUNT_PATH="/config/proxy"

# Step 1: Retrieve the specified data from the ConfigMap
configmap_value=$(kubectl get configmap "$CONFIGMAP_NAME" -n
 "$NAMESPACE" -o json | jq -r --arg key "$DATA_KEY" '.data[$
key]')
if [ -z "$configmap_value" ]; then
  echo "Error: Key '$DATA_KEY' not found in ConfigMap '$CONFIG
MAP_NAME'."
  exit 1
fi
echo "Retrieved value from ConfigMap: $configmap_value"
# Step 3: Retrieve the existing Secret
existing_secret=$(kubectl get secret "$SECRET_NAME" -n "$NAM
ESPACE" -o json)
if [ -z "$existing_secret" ]; then
  echo "Error: Secret '$SECRET_NAME' not found."
  exit 1
fi

# Step 4: Modify the specified field in the Secret with the va
lue from the ConfigMap
```

```
modified_secret_data=$(echo "$existing_secret" | jq --arg fie
ld "$SECRET_FIELD" --arg value "$configmap_value" '
  .data[$field] = ($value | @base64)'
)
# Step 5: Create a new Secret with the modified data
temp_file=$(mktemp)
echo "$modified_secret_data" | jq '.metadata.name = "'$NEW_SEC
RET_NAME'" | del(.metadata.uid, .metadata.resourceVersion, .
metadata.creationTimestamp, .metadata.annotations)' > $temp_
file
kubectl apply -f $temp_file

# Clean up
rm $temp_file

echo "New secret '$NEW_SECRET_NAME' created successfully."

# Step 6: Mount the new Secret to the 'web' container in the
 specified Deployment
# Get the current deployment spec
deployment_spec=$(kubectl get deployment "$DEPLOYMENT_NAME" -n
 "$NAMESPACE" -o json)
if [ -z "$deployment_spec" ]; then
  echo "Error: Deployment '$DEPLOYMENT_NAME' not found."
  exit 1
fi

# Extract the existing volumes
existing_volumes=$(echo "$deployment_spec" | jq -r '.spec.t
emplate.spec.volumes // []')

# Find the index of the "web" container
container_index=$(echo "$deployment_spec" | jq -r --arg CONTAI
NER_NAME "$CONTAINER_NAME" '
  .spec.template.spec.containers | map(.name == $CONTAINER_
NAME) | index(true) // "null"'
)

# Check if the container named "web" exists
if [ "$container_index" == "null" ]; then
  echo "Error: Container with name '$CONTAINER_NAME' not found
 in deployment '$DEPLOYMENT_NAME'."
  exit 1
fi

# Extract the volumeMounts for the "web" container
existing_volume_mounts=$(echo "$deployment_spec" | jq -r --arg
json index "$container_index" '
  .spec.template.spec.containers[$index].volumeMounts // []'
)

# Define the new volume and volumeMount
new_volume=$(echo "[{\"name\": \"$NEW_SECRET_NAME\", \"secret
\": {\"secretName\": \"$NEW_SECRET_NAME\"}}]" | jq -r .)
new_volume_mount=$(echo "[{\"name\": \"$NEW_SECRET_NAME\", \"m
ountPath\": \"$MOUNT_PATH\"}]" | jq -r .)

# Combine existing and new volumes
updated_volumes=$(echo "$existing_volumes $new_volume" | jq -s
 'add')
updated_volume_mounts=$(echo "$existing_volume_mounts $new_v
olume_mount" | jq -s 'add')
# Patch the deployment with the new volumes and volume mounts
```

```
kubectl patch deployment "$DEPLOYMENT_NAME" -n "$NAMESPACE" --
type='json' -p="
[
  {
 \"op\": \"replace\",
 \"path\": \"/spec/template/spec/volumes\",
 \"value\": $updated_volumes
  },
  {
 \"op\": \"replace\",
 \"path\": \"/spec/template/spec/containers/$container_index/v
olumeMounts\",
 \"value\": $updated_volume_mounts
  }
]"

echo "New secret '$NEW_SECRET_NAME' mounted to the 'web' conta
iner in the deployment '$DEPLOYMENT_NAME' successfully."
```

3. Run the script with the following command and change the Cloudera AI namespace:

```
chmod +x enableProxy.sh && ./enableProxy.sh.sh -n <cml-names
pace>
```

4. Monitor the web pods under the Cloudera AI namespace.

OPSX-5766: The system is reaching the configured limit of inotify instances

Running too many sessions or nodes might lead to crashing fluentbit sidecar containers. Cloudera AI reports that the session is running but the users might experience limitations, that is, for example, Cloudera AI Workbench does not work. This is because the system reaches the configured limit of inotify instances.

Workaround:

The nodes must have the system variable fs.inotify.max_user_instances configured to a reasonable value.

This can be set by running the following in the node:

```
sudo sysctl fs.inotify.max_user_instances=8192
sudo sysctl -p
```

To persist the above configuration with node restarts, add the below line to /etc/sysctl.conf:

```
fs.inotify.max_user_instances=8192
```

DSE-42079: Cloudera AI Workbenches are not compatible with the *Unified Timezone* feature

When you enable the *Unified timezone* feature, the Cloudera Embedded Container Service cluster timezone is synchronized with the Cloudera Manager Base time zone, and the Cloudera AI sessions will fail to launch with Exit Code 34. You will also see timestamp discrepancies with workloads. For more information, see ENGESC-28507.

Workaround:

If you use Cloudera AI Workbenches, disable the *Unified Timezone* feature by following the instructions in Cloudera Embedded Container Service unified time zone.

DSE-34314: Templating Cloudera Base AutoTLS properties needed in Cloudera AI sessions

Spark pushdown does not work if Atlas and Spark Lineage checks are enabled for Spark.

Workaround:

To use Spark pushdown, you need to disable Atlas and Spark Lineage checks in the Spark configurations.

DSE-41424: Better handling for data connection validation errors

In Cloudera AI on premises, Cloudera AI Workbench fails to start under the following conditions:

- If the HIVE_ON_TEZ service is not present, or if the following configurations are missing:
  - hiveserver2_load_balancer
  - hive.server2.transport.mode
  - Kerberos_princ_name
- If the IMPALA service is not present on your Base cluster, or if the following configurations are missing:
  - Hs2_port or hs2_http_port
  - Kerberos_princ_name

Workaround:

1. Disable the data connection auto-discovery feature.

```
kubectl exec -it <db-pod> -n <workspace-namespace>
psql -U sense
# set the flag to false
sense=# update site_config set enable_discover_dataconn_res
tart = false;
UPDATE 1
```

2. Restart web pods to apply changes.

Note: The Data connection discovery button may not function properly until the invalid URLs in the database are corrected.

3. Check and ensure that there are no incorrect entries in the database.
4. Back up the workspace_data_connections and the project_data_connections tables.

```
[postgres@db-0 /]$ pg_dump -U sense  -t workspace_data_conne
ctions -f /data-versioned/workspace_data_connections_dump.sql
[postgres@db-0 /]$ pg_dump -U sense  -t project_data_connectio
ns -f /data-versioned/project_data_connections_dump.sql
```

5. Delete the tables.

```
DELETE FROM workspace_data_connections;
DELETE FROM project_data_connections;
```

Technical Service Bulletins

TSB 2025-822 Cloudera AI Workbench Web Service Crashes after Upgrading to Cloudera Data Services 1.5.4 SP1

Customers using Cloudera AI Workbench without Hive on TEZ or Impala, or using these services configured for High Availability on their Cloudera on premises cluster can experience crashes of the Cloudera AI Workbench web service at startup after upgrading to the latest Cloudera Data Services on premises 1.5.4 SP1 builds.

Knowledge article

For the latest update on this issue, see the corresponding Knowledge article: Cloudera Customer Advisory 2025-822: Cloudera AI Workbench Web Service Crashes after Upgrading to Cloudera Data Services 1.5.4 SP1.

Cloudera AI on premises 1.5.4 SP2

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

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.

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 Services on premises

- 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

What's new in 1.5.4 SP2

Cloudera on premises 1.5.4 SP2 includes the following features for Cloudera AI.

Introducing auto scaling performance-critical components

Some performance-critical components in Cloudera AI automatically scale based on demand, enhancing the scalability and the reliability of the product. To enable utilising this feature, we have increased the recommended resource requirements for Cloudera AI by 16 cores.

Fixed issues in 1.5.4 SP2

This section lists issues fixed in this release for Cloudera AI on premises.

DSE-40909: Disabled 'Run experiments' function does not work as expected

When the Administrator disabled 'running experiments' in Cloudera AI Workbench Site Administration Settings , the user was still able to see the **Experiments** menu item in the global- and project-level navigation.

With the help of implementing the sufficient conditions the UI can now hide the disabled function.

DSE-38645: spark.yarn.jars are not set properly in Cloudera AI on premises Spark Pushdown mode

In case of Spark pushdown, the spark.yarn.jars parameter can indicate which jars in the driver shall be transported to YARN executors. Without this parameter set, the YARN executors use the Spark jars from the base cluster provided by Cloudera Distribution of Spark. This can lead to version mismatch between the Spark driver (used Cloudera Data Engineering-provided jars) and the executors.

Now, the spark.yarn.jars parameter is set to use jars from the Cloudera AI driver (Cloudera AI session) to keep the versions matched between the Spark driver and the executors.

DSE-39798: API v1 stop session endpoint shall perform authentication check

Previously, users could stop sessions under projects that they were not authorized to access using the session’s Universally unique identifier (UUID). This issue is now resolved.

DSE-41431: Register engine stopped status

When the reconciler is overloaded with a large number of events, the deleted status is still propagated to ensure that engines do not remain in 'Stopping' status.

DSE-41424: Better handling for data connection validation errors

In Cloudera AI on premises 1.5.4 SP1 Cloudera AI Workbench failed to start under the following conditions:

- If the HIVE_ON_TEZ service was not present or the following configurations were missing:
  - hiveserver2_load_balancer
  - hive.server2.transport.mode
  - Kerberos_princ_name

- If the IMPALA service was not present on your Base cluster or the following configurations were missing:
  - Hs2_port or hs2_http_port
  - Kerberos_princ_name

This issue has been fixed.

DSE-41218: Restrict secrets and ingress access from User Service Account role

User access to Kubernetes secrets and ingresses in their own user namespace has been removed.

DSE-34314: Cloudera Base AutoTLS property set incorrectly

The correction to Cloudera Base AutoTLS configurations resulted in the atlas.kafka.ssl.truststore.location value to be set incorrectly.

This issue has been fixed.

Technical Service Bulletins

TSB 2025-822 Cloudera AI Workbench Web Service Crashes after Upgrading to Cloudera Data Services 1.5.4 SP1

For the latest update on this issue, see the corresponding Knowledge article: Cloudera Customer Advisory 2025-822: Cloudera AI Workbench Web Service Crashes after Upgrading to Cloudera Data Services 1.5.4 SP1.

Known issues in 1.5.4 SP2

You might run into some known issues while using Cloudera AI on premises.

DSE-42079: Cloudera AI Workbenches are not compatible with the *Unified timezone* feature

When you enable the *Unified timezone* feature, the Cloudera Embedded Container Service cluster timezone is synchronized with the Cloudera Manager Base time zone, and the Cloudera AI sessions fails to launch with Exit Code 34. Timestamp discrepancies with workloads are displayed.

Workaround:

If you use Cloudera AI Workbenches, disable the *Unified Timezone* feature by following the instructions in Cloudera Embedded Container Service unified time zone.

DSE-41757: Python workloads running multiple-line comments or strings might fail

Python workloads running multiple-line comments or strings might fail to run when using the Workbench Editor.

Workaround:

Run the code using the PBJ Workbench Editor.

DSE-42509: Creating a project does not work on NTP/Airgap proxy when using private repository with SSH key and SSH URL

When creating a project on NTP/Airgap proxy, using a private repository with an SSH key and an SSH URL, an error message is displayed that the project cannot be created.

No workaround is available.

DSE-42510: Fetching Cloudera or Huggingface AMP catalog is failing on NTP/Airgap proxy

When fetching Cloudera or Huggingface Cloudera Accelerators for Machine Learning Project (AMP) catalog on NTP/Airgap proxy, an error message is displayed Error fetching AMP catalog source URL.

No workaround is available.

DSE-44698: Restart Stabilty - Cloudera AI Workbench pod preemption by system-critical workloads

During control plane upgrades or cluster restarts, Cloudera AI Workbench pods may transition into the Init:Error or Init:ContainerStatusUnknown state.

This issue may arise during cluster startup or under resource pressure when the scheduler preempts lower-priority Cloudera AI Workbench pods to allocate resources for higher-priority system pods, such as critical system components. Additionally, Kubernetes does not automatically clean up preempted pods, leaving them in failed Init states.

This is the expected Kubernetes scheduler behavior. There is no permanent fix available, as pod preemption is controlled by the Kubernetes scheduler and is necessary for system stability during resource constraints.

Workaround:

Delete the affected pod, and Kubernetes automatically attempts to reschedule it if sufficient resources are available:

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

Cloudera AI on premises 1.5.4 SP2 HF1

Review the features, fixes, and known issues in the Cloudera AI 1.5.4 SP2 hotfix 1 release.

Fixed issues in 1.5.4 SP2 HF1

Review the fixed issues in the Cloudera AI 1.5.4 SP2 hotfix 1 release.

DSE-40325: Frequently received 504 timeout error for VFS stat calls

Concurrent project forks have caused 504 timeout errors when navigating the Projects page.

This issue has been resolved.

DSE-41027: Upgrade path from 1.5.0 -> 1.5.2 -> 1.5.4 -> 1.5.4 CHF3 fails with missing key

Due to a strict validation introduced, certain upgrade paths were missing the required key, resulting in upgrade failures. Users upgrading to version 1.5.4 might have encountered this issue if their original Cloudera Data Services on premises version at the time of initial installation was 1.5.0 or 1.5.1.

This issue has been resolved.

DSE-42379: Model registry page and its deletion result in errors on Cloudera AI Registry page

While navigating the Cloudera AI Registry pages, the following error was displayed: Could not interpret page token: provided token did not match parameters.

Additionally, when attempting to delete a model, the model disappeared from the page and an error message appeared: Failed to delete model from model registry &{} (*models.Error) is not supported by the TextConsumer, can be resolved by supporting TextUnmarshaler interface.

These issues occurred due to failures in page token validation on the Cloudera AI Registry pages.

These issues have been resolved.

DSE-43808: 2.0.44: Cloudera AI Registry workbench pagination issues

While navigating to the **next page** in the Cloudera AI Registry workbench UI, an error was displayed. Additionally, selecting the option to display 25-51 results per page often resulted in showing only 10 items.

These issues have been resolved.

DSE-42480: List of Cloudera AI Registry calls in workbench is invoked with page_size=10 10

When navigating on the Cloudera AI Registry pages to view the models from the workbench, it was always displayed with page_size=10, even if you had previously selected another value.

This issue has been resolved.

DSE-43104: Timezones in Cloudera on premises cause pods to be killed with exit code 34

In an on premises environment, the Kubernetes cluster could be configured to any timezone, leading to timezone discrepancies, particularly for engine pods. This issue affected engine timestamp fields, such as **scheduling_at**, **starting_at**, **running_at**, and **finished_at**, which were reported in varying timezones throughout the infrastructure. As a result, when the base cluster and the nodes were in different timezones, the engine pods were terminated.

This issue has been resolved.

Repository Locations for 1.5.4 SP2 HF1

The URLs for Cloudera AI on premises 1.5.4 SP2 HF1 are listed in the following table:

URL Type	Repository Location
Index	`https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.4-h19/`
Manifest	Repository: `https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.4-h19/manifest.json`
Parcels	Repository: `https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.4-h19/parcels/`

Cloudera AI on premises 1.5.4 SP2 CHF1

Review the features, fixes, and known issues in the Cloudera AI 1.5.4 SP2 Cumulative hotfix 1 release.

Fixed issues in 1.5.4 SP2 CHF1

Review the fixed issues in the Cloudera AI 1.5.4 SP2 Cumulative hotfix 1 release.

DSE-44586: Disabled Spark ML Runtime addons are reenabled

Previously, Spark ML Runtime addons were enabled automatically, which was unintended behavior. This issue is resolved, and Spark ML Runtime addons can only be enabled when explicitly configured by the administrator.

DSE-41027: Upgrade path from 1.5.0 -> 1.5.2 -> 1.5.4 -> 1.5.4 CHF3 fails with missing key

Due to a strict validation introduced, certain upgrade paths were missing the required key, resulting in upgrade failures. Users upgrading to version 1.5.4 might have encountered this issue if their original Cloudera Data Services on premises version at the time of initial installation was 1.5.0 or 1.5.1.

This issue has been resolved.

DSE-40225: Projects/cdn/XXXX-POD folders are not cleaned on NFS

This issue addressed a bug where projects/cdn/XXXX-POD folders were not being cleaned up on the Network File System (NFS) after workload termination, resulting in the accumulation of excessive stale directories. These folders, created per workload and mounted as writable volumes, were intended to persist until the session or project was deleted but were never automatically removed.

DSE-45050: Large file job attachment crashes web

Uploading excessively large files, for example, around 500MB, through the UI caused the web container to crash. This issue was triggered by an uncaught asynchronous exception in the `send email` method, which was not properly handled when processing attachments. The problem is resolved, ensuring that large file uploads no longer result in web container crashes.

DSE-43104: Timezones in Cloudera AI on premises cause pods to be killed with exit code 34

In Kubernetes clusters on premises, local timezones can be configured. This caused timezone discrepancies in engine pods, in which timestamp fields such as scheduling_at, starting_at, running_at, and finished_at were inconsistently stored and read across the Cloudera AI infrastructure.

The issue is now resolved by updating the dashboard timestamp fields to store all timestamps with explicit timezone information and enforcing UTC for all database writes. This fix ensures consistent timestamp handling and prevents premature pod termination due to timezone offsets.

DSE-44091: StartJobRun Kubernetes client failure error lost during processing

Kubernetes client failure errors during StartJobRun API were not properly reported, leading to misleading success logs. The operator pod failed to correctly translate certain asynchronous action errors, resulting in false Finish StartJobRun success messages even when pod start failures occurred.

The root cause was identified as insufficient error handling in the GetFailureResponse method, which allowed nil errors to bubble up and obscure the actual failures. This issue is now resolved, ensuring accurate error reporting and preventing misleading success logs.

DSE-44083: Web service requests to other services have malformed UUID in logs

This issue addressed a bug where web service requests logged malformed Universally Unique Identifiers (UUIDs), making debugging difficult. The problem occurred because the length of the contextId UUID was limited to 30 characters, causing 36-character requestIds UUID to be truncated in operator logs.

The issue is now resolved by increasing the contextId length to 36 characters, ensuring full UUIDs are preserved in logs and improving log accuracy for debugging purposes.

DSE-44088: Operator pod start failure log missing ID

This issue addressed the lack of detailed error logging in operator pod failures, where logs did not include the engineId UUID or request UUID, making it difficult to trace and debug issues.

The fix implemented now includes both the engineId UUID and request UUID in failure logs, significantly improving traceability and aiding in debugging operator pod failures.

DSE-41733 - Spark logs not cleaned up by Livelog cleaner

Spark logs were not cleaned up by the Livelog cleaner because the getCleanableEngines internal API did not include Spark executors. This issue is now resolved, and the getCleanableEngines API now includes Spark executors, ensuring that Spark logs are properly cleaned up by the Livelog cleaner.

DSE-42231: When a workbench experiences heavy usage the new Session page starts to load slowly

The new Session page and related workload creation pages, such as new Jobs, and Applications, experienced significant slowdowns, with load times exceeding 60 seconds and frequent timeouts.

The issue is now resolved by removing unnecessary /usage API calls from workload creation pages, including Sessions, Jobs, Applications and so on, ensuring faster load times and improved responsiveness.

DSE-40029: The job timeout is not of a valid type

Previously, the job timeout value could not be cleared once set and attempting to update a job with an empty timeout field resulted in a The job timeout is not of a valid type validation error . This issue is now resolved, and the timeout field can now be cleared as expected.

DSE-43950: Workbench installation is failing as buildkit pod is crashing due to port bind issues

Buildkitd pods in Cloudera AI Workbench could intermittently fail with a CrashLoopBackOff error because the BuildKit 1234 port was not properly released during pod restarts or was occupied by another process. This resulted in errors, such as:

```
buildkitd: listen tcp 0.0.0.0:1234: bind: address already in use.
```

This issue is now resolved, and Buildkitd pods no longer enter a crash loop state due to port binding issues.

DSE-44700: Fixing web pod crash while fetching data connections from Cloudera Base cluster

This issue addressed a bug that prevented the use of underscores in data connection names. The fix now allows underscores to be used, ensuring greater flexibility in naming data connections.

DSE-43774 Reconciler unresponsiveness issue

Previously, a logging-pipeline issue could cause the reconciler and other microservices utilizing Cloudera customized tee binary to freeze. The issue surfaced when the reconciler tried to log a very large Kubernetes object from a DeletedFinalStateUnknown pod event. The object itself was harmless, but the single log line it produced exceeded 64 KB, triggering a hard line-size limit in the customized tee, which blocked the log stream and stalled the entire service. The issue is now resolved and the tee can now stream log lines of any length, removing the 64 KB constraint and preventing similar hangs across all components.

Repository Locations for 1.5.4 SP2 CHF1

The URLs for Cloudera AI on premises 1.5.4 SP2 CHF1 are listed in the following table:

URL Type	Repository Location
Index	`https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.4-h30/`
Manifest	Repository: `https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.4-h30/manifest.json`
Parcels	Repository: `https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.4-h30/parcels/`