

Cloudera AI

Release Notes

Date published: 2020-07-16

Date modified: 2025-12-18

CLOUDERA

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

Legal Notice

© Cloudera Inc. 2026. 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 AI on premises 1.5.5.....	5
Compatibility for Cloudera AI 1.5.5, Cloudera Runtime components and Data Visualization.....	6
Fixed issues in Cloudera AI on premises 1.5.5.....	8
Known issues for Cloudera AI on premises 1.5.5.....	11
Service Packs: Cloudera AI on premises.....	18
Cloudera AI on premises 1.5.5 SP4.....	18
What's new in 1.5.5 SP4.....	18
Compatibility for Cloudera AI 1.5.5 SP4, Cloudera Runtime components and Data Visualization.....	19
Supported models and servers in Cloudera AI Inference service.....	20
Supported vLLM command line arguments.....	25
Fixed issues in Cloudera AI on premises 1.5.5 SP4.....	26
Known issues for Cloudera AI on premises 1.5.5 SP4.....	27
Cloudera AI on premises 1.5.5 SP3.....	31
What's new in 1.5.5 SP3.....	31
Compatibility for Cloudera AI 1.5.5 SP3, Cloudera Runtime components and Data Visualization.....	33
Supported models and servers in Cloudera AI Inference service.....	34
Supported vLLM command line arguments.....	38
Fixed issues in Cloudera AI on premises 1.5.5 SP3.....	38
Known issues for Cloudera AI on premises 1.5.5 SP3.....	41
Cloudera AI on premises 1.5.5 SP2.....	45
What's new in 1.5.5 SP2.....	45
Compatibility for Cloudera AI 1.5.5 SP2, Cloudera Runtime components and Data Visualization.....	46
Supported transformer model architectures for vLLM 0.8.5 in Cloudera AI Inference service 1.5.5 SP2.....	47
Supported vLLM command line arguments.....	50
Fixed issues in 1.5.5 SP2.....	51
Known issues for Cloudera AI on premises 1.5.5 SP2.....	52
Cloudera AI on premises 1.5.5 SP1.....	55
What's new in 1.5.5 SP1.....	55
Compatibility for Cloudera AI 1.5.5 SP1, Cloudera Runtime components and Data Visualization.....	57
Supported transformer model architectures for vLLM 0.8.4 in Cloudera AI Inference service 1.5.5 SP1.....	58
Supported vLLM command line arguments.....	61
Fixed issues in 1.5.5 SP1.....	62
Known issues for Cloudera AI on premises 1.5.5 SP1.....	65
Cumulative Hotfixes: Cloudera AI on premises.....	68
Cloudera AI on premises 1.5.5 SP3 CHF3.....	68

Fixed issues in Cloudera AI on premises 1.5.5 SP3 CHF3.....	68
Known issues for Cloudera AI on premises 1.5.5 SP3 CHF3.....	70
Cloudera AI on premises 1.5.5 SP3 CHF2.....	71
Cloudera AI on premises 1.5.5 SP2 CHF1 HF1.....	71
Repository Locations for 1.5.5 SP2 CHF1 HF1.....	71
Fixed issues in Cloudera AI on premises 1.5.5 SP2 CHF1 HF1.....	71
Cloudera AI on premises 1.5.5 SP2 CHF1.....	72
What's new in Cloudera AI on premises 1.5.5 SP2 CHF1.....	72
Compatibility for Cloudera AI 1.5.5 SP2 CHF1, Cloudera Runtime components and Data Visualization.....	73
Supported transformer model architectures for vLLM 0.8.5 in Cloudera AI Inference service 1.5.5 SP2.....	74
Supported vLLM command line arguments.....	77
Fixed issues in Cloudera AI on premises 1.5.5 SP2 CHF1.....	78
Cloudera AI on premises 1.5.5 CHF1.....	80
Fixed issues in 1.5.5 CHF1.....	81

What's new in Cloudera AI on premises 1.5.5

Understand the functionalities and improvements in Cloudera AI on premises 1.5.5

For information on the Repository locations, see [Repository Locations for Data Services on premises 1.5.5](#).

New features and improvements

AI Studios [Technical Preview]

Cloudera AI Studios is a comprehensive suite of low-code tools designed to simplify the development, customization, and deployment of generative AI solutions within enterprises. This suite empowers organizations to operationalize AI workflows quickly and efficiently by leveraging real-time enterprise data.

For more information, see [AI Studios Overview](#)

Cloudera AI Inference service [Technical Preview]

The Cloudera AI Inference service is a data service available in Technical Preview. Cloudera AI Inference service is a production-grade serving environment for traditional, generative AI, and Large Language Models. It is designed to handle the challenges of production deployments, such as high availability, fault tolerance, and scalability. The service is now available to carry out inference on the following categories of models:

- Optimized open-source Large Language Models.
- Traditional machine learning models like classification, regression, and so on. Models need to be imported to the Cloudera AI Registry to be served using the Cloudera AI Inference service.

For more information, see [Using Cloudera AI Inference service](#).

Cloudera Copilot [Technical Preview]

Cloudera Copilot is an AI-powered coding assistant designed for seamless integration within JupyterLab ML Runtimes. With its chat interface and comprehensive code completion features, Cloudera Copilot enhances the development experience for machine learning projects. It offers compatibility with model endpoints deployed in Cloudera AI Inference service model, providing developers with flexibility and efficiency in their workflows.

For details, see [Cloudera Copilot Overview](#)

Model Hub [Technical Preview]

Model Hub is a catalog of top-performing LLM and generative AI models. You can now easily import the models listed in the Model Hub into the Cloudera AI Registry and then deploy it using the Cloudera AI Inference service. This streamlines the workflow of developers working on AI use cases by simplifying the process of discovering, deploying, and testing models.

For more information, see [Using Model Hub](#)

Certification Manager

Cert-manager is an open-source tool for Kubernetes that automates the provisioning, management, and renewal of TLS certificates. Its documentation at <https://cert-manager.io/docs/> provides comprehensive guidance on installing, configuring, and using cert-manager to secure workloads with trusted X.509 certificates. Cloudera provides out-of-the-box support for Venafi TPP as part of the Cloudera Embedded Container Service installation. By integrating cert-manager, the Cloudera Data Services on premises achieve secure communication, reduced manual overhead, and compliance with security standards, leveraging its robust automation and flexibility. For more information on setting up Cert-manager using Venafi TPP, see [Certification Manager service for increased security](#).

Workbench-level Spark defaults

Custom Spark settings can now be configured at workbench level. When set, the custom Spark configuration provided by the administrator will be merged with the default Spark configuration used in Cloudera AI sessions. These settings will automatically apply to all newly launched Spark sessions within the workbench. The configuration option is available under **Site Administration Runtimes**.

Spark pushdown

The Administrator can set Spark pushdown to be enabled during project creation by default.

Auto-synchronization for user and team is enabled by default

The efficiency and usability of the auto-synchronization features have been enhanced for user and team management. Key updates include:

- Auto-synchronization is enabled by default: Auto synchronization for users and teams is now enabled by default, with a synchronization interval set to 12 hours.
- User management service: User management is now handled by a new service, reducing overhead on the web pod. It now prevents multiple synchronization operations from running in parallel.
- Logging: Detailed logging has been added for the failure cases.
- Synchronization trigger sequence: The team synchronization now internally triggers user synchronization to pull the most recent user details from the Cloudera control plane.
- You can switch on or off the User and Team auto synchronization feature.

These improvements are aimed at optimizing performance and streamlining the synchronization process for users and teams.

Multiple docker registry accounts

Cloudera AI now supports storing multiple Docker credentials for your custom runtimes and provides a dedicated UI and API for managing them. Additionally, Cloudera AI no longer retrieves Docker registry credentials from the regcred secret.

If you previously relied on credentials stored in the regcred secret, you must add these credentials to Cloudera AI under **Site Administration Runtime** to ensure your workloads continue functioning seamlessly.

Related Information

[Known Issues and Limitations](#)

[Certification Manager service for increased security](#)

[Managing Users](#)

Compatibility for Cloudera AI 1.5.5, Cloudera Runtime components and Data Visualization

Explore Cloudera AI 1.5.5 and its compatibility with Cloudera Runtime components and with Cloudera Data Visualization across various versions.

**Important:**

- If you switch from the Red Hat Universal Base Image (Redhat UBI image) to the security hardened image, note that the updated component versions and libraries can impact the Cloudera AI operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

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

For more information, see [Security hardened Spark image migration guide](#).

- From Cloudera Data Services on premises 1.5.5 onwards, Spark 3.4 is no longer available.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported on Spark 3.3.2 and higher.

Table 1: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for Security Hardened image

Cloudera Runtime Version	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.7 SP 3	- Spark 3.2.3 - Scala 2.12.15 - Python 3.9.22 - Java 11.0.27	NA	NA	- Airflow 2.10.4 - Python 3.11.12 - Java 17.0.15	Iceberg 0.14.1	1.30	8.0.2-b34
7.1.9 SP1	- Spark 3.2.3 - Scala 2.12.15 - Python 3.9.22 - Java 11.0.27	- Spark 3.3.2 - Scala 2.12.15 - Python 3.10.17 - Java 17.0.15	NA	- Airflow 2.10.4 - Python 3.11.12 - Java 17.0.15	Iceberg 1.3.0	1.30	8.0.2-b34
7.3.1	NA	NA	- Spark 3.5.1 - Scala 2.12.18 - Python 3.11.13 - Java 17.0.15	- Airflow 2.10.4 - Python 3.11.13 - Java 17.0.15	Iceberg 1.5.2	1.30	8.0.2-b34

Table 2: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for Redhat insecure image

Cloudera Runtime Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.7 SP3	- Spark 2.4.7 - Scala 2.11.12 - Python 2.7.18 - Python 3.6.8 - Java 1.8.0_452	- Spark 3.2.3 - Scala 2.12.15 - Python 2.7.18 - Python 3.6.8 - Java 11.0.25	NA	NA	- Airflow 2.10.4 - Python 3.11.12 - Java 17.0.15	Iceberg 0.14.1	1.30	8.0.2-b34
7.1.9 SP1	- Spark 2.4.7 - Spark 2.4.8 - Scala 2.11.12 - Python 2.7.18 - Python 3.6.8 - Java 1.8.0_452	- Spark 3.2.3 - Scala 2.12.15 - Python 2.7.18 - Python 3.6.8 - Java 11.0.25	- Spark 3.3.2 - Scala 2.12.15 - Python 2.7.18 - Python 3.8.17 - Java 11.0.25	NA	- Airflow 2.10.4 - Python 3.11.12 - Java 17.0.15	Iceberg 1.3.0	1.30	8.0.2-b34
7.3.1	NA	NA	NA	- Spark 3.5.1 - Scala 2.12.18 - Python 2.7.18 - Python 3.8.17 - Java 11.0.25	- Airflow 2.10.4 - Python 3.11.13 - Java 17.0.15	Iceberg 1.5.2	1.30	8.0.2-b34

Fixed issues in Cloudera AI on premises 1.5.5

This section lists the issues that have been fixed since the last release of Cloudera AI on premises.

Cloudera AI Workbench

DSE-38715: Improve the Last Modified column for Files in the UI

This update standardises the use of absolute timestamps across the entire application, replacing relative timestamps such as 1 week ago with precise date and time information. The **Files** section in the Project Overview page now displays exact Last Modified dates, while the **Jobs** section in the **Job Overview** page shows accurate Latest Run times for improved clarity and consistency.

DSE-41733 - Spark logs are not cleaned up by livelog cleaner

The Livelog cleaner previously cleaned up input and output log topics but did not delete the content of container log topics. It now includes engine containers and Spark executor log topics in the cleanup process.

DSE-42499: ZSTD compression codec support in Cloudera AI Spark while reading Hive tables

Previously, Spark workloads were unable to read the ZSTD-encoded Hive tables when running in the default Spark on Kubernetes mode.

This issue has now been resolved, enabling Spark to successfully read ZSTD-encoded data from Hive tables.

DSE-42183: Shared Memory Not Set on Applications

The Shared Memory configuration set in `Project Settings Advanced` is now applied to applications in addition to sessions.

This configuration is retrieved from the Projects table when a user creates a new application.

DSE-41429 The timezone for duration for experiment details is not correct

The **Experiment** page on the Details tab occasionally displayed a negative value for the duration of time, due to an incorrect timezone configuration in certain setups. This issue has been solved.

DSE-42858 - [Cloudera Embedded Container Service Restart Stability] db-0 Pod CrashLoops with error after Cloudera Embedded Container Service restart

The following error was intermittently observed in the db-0 pod during restarts or upgrades:

- CDSW_PG FATAL: lock file "postmaster.pid" already exists
- CDSW_PG HINT: Is another postmaster (PID 7) running in data directory

This issue occurred because the db-migrate-xxx job was still in the Running state and was attempting to access the embedded database.

The issue has now been resolved. The db-0 pod no longer enters a crash loop due to a stale postmaster.pid file.

ML Runtimes**DSE-41772: Opening session opens up a wrong file intermittently**

This update addressed an issue where the Workbench editor occasionally opened the wrong file. The problem stemmed from timing conflicts in DocumentManager.js, where a rapid refresh could retrieve an outdated file from local storage instead of the file that was just clicked.

The fix ensures that the DocumentManager.js consistently selects the most recently clicked file during the refresh process, even in scenarios involving quick refreshes.

DSE-40874: Workbench brackets editor search retains the same string until there is a click on the editor

There was an issue in the Workbench editor where the Find input box (Ctrl+F or Command+F) retained the previous search string upon reopening. The input box now clears automatically when clicking anywhere in the editor text area, enabling you to enter a new search string seamlessly.

DSE-42595: Sharing generated images does not work with PBJ Runtimes

Earlier, the HTML code generated to embed an image in PBJ Workbench did not work. Now, you can embed images generated in PBJ Workbench-based sessions similarly to how you can embed images from Workbench-based sessions using the share icon next to the generated images.

DSE-42962: PBJ log is not truncated (default value and MAX_TEXT_LENGTH value)

PBJ Workbench Runtime images now comply with the value set in the environmental variable MAX_TEXT_LENGTH. This limits the maximum number of characters that can be displayed by each command executed.

DSE-42344: The Interrupt button is not working with PBJ Workbench Runtime

Earlier, the Interrupt button did not work in Cloudera AI sessions that ran a PBJ Workbench Runtime. This issue has been fixed.

DSE-43297: Non-Python custom editor custom Runtimes fail to start

Sessions using PBJ-based custom Runtimes with a custom editor could not start previously. This issue has been fixed.

DSE-42077: PBJ Workbench not pretty-printing outputs

When PBJ Workbench R Runtime was used, the tables and help text were not properly displayed. This issue has been fixed.

DSE-42966: R PBJ Session commands' documentation is not displayed correctly

When PBJ Workbench R Runtime was used, the help text, required as a session command's output, was not properly displayed. This issue has been fixed.

DSE-42967: R PBJ does not have rich visualization for tables

When PBJ Workbench R Runtime was used, the content of the tables was not properly displayed. This issue has been fixed.

DSE-35299: Stop logging every livelog entry in PBJ Runtimes

When PBJ Workbench R Runtime was used, the start of each command execution was logged with INFO severity, and was consequently displayed. Now, the command execution is logged with DEBUG severity and is not displayed.

DSE-42345: Using PBJ variants code completion in Editor is not working

When PBJ Workbench R Runtime was used, the editor was not able to show code completion. This issue has been fixed.

DSE-42498: Workloads are not starting up: /api/kernels: context deadline exceeded

A previously unauthenticated endpoint is now securely authenticated through a reverse proxy, using a mechanism called Browser Accessible Service.

DSE-43192: Jupyter Kernelgateway in PBJ Workloads is exposed publicly without authentication

A previously unauthenticated endpoint is now securely authenticated via a reverse proxy, using a mechanism known as Browser Accessible Service.

DSE-41438: JOB_MAXIMUM_MINUTES does not affect Jobs using PBJ workbench

PBJ Workbench jobs now respect the JOB_MAXIMUM_MINUTES environment variable. PBJ Workbench jobs are automatically terminated once they exceed the time limit specified by JOB_MAXIMUM_MINUTES.

DSE-42473: Jupyter sessions do not follow SESSION_MAXIMUM_MINUTES

When SESSION_MAXIMUM_MINUTES is set, sessions are expected to expire after the specified duration, regardless of activity (default is 7 days). However, unlike PBJ Sessions and Remote Editor Sessions, Jupyter Sessions previously continued running beyond this time and did not time out. This issue has now been resolved.

DSE-42814: Enhanced Job Arguments UX via JOB_ARGUMENTS environment variable for PBJ Workbench ML Runtimes

This update improves the UI on both the Job Creation page and the Job Settings page to clarify that the PBJ-based ML Runtime transforms job arguments into the JOB_ARGUMENTS environment variable. It introduces a dedicated label and a help block, including a link to a code snippet for user guidance.

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

Previously, Spark failed when running in Pushdown mode with the Scala runtime, while it functioned correctly on Spark on Kubernetes. The issue was resolved by adding the Python interpreter to the Scala 2.11 runtime image.

Openshift container platform

DSE-42509: Project creation using private repository with SSH key and SSH URL is not working on NTP/Airgap Openshift Container Platform

Project creation using a private repository with SSH key and SSH URL did not work on NTP setups. This issue has been fixed.

DSE-42510: Fetching Cloudera or Huggingface AMP catalog is failing on NTP/Airgap Proxy Openshift Container Platform

Previously there has been an error fetching the AMP catalog on NTP/Airgap Proxy Openshift setup when selecting Site Administration AMP tab.

This issue has been solved.

Known issues for Cloudera AI on premises 1.5.5

This section lists known issues that you might run into while using Cloudera AI on premises.

Cloudera AI Workbench known issues

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-44699: Provisioning workbench is failing with error pool guaranteed resources larger than parent's available guaranteed resources

With the Quota Management feature enabled, creating Cloudera AI Workbench might fail with the error pool guaranteed resources larger than parent's available guaranteed resources.

DSE-44367: Buildkitd Pod CrashLoopBackOff due to port conflicts

During the creation or upgrade of a Cloudera AI Workbench, buildkitd pods may occasionally enter a CrashLoopBackOff state. This typically happens when the port used by BuildKit is not properly released during pod restarts or is occupied by another process. You may encounter errors such as:

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

Workaround:

If you experience this issue, follow this step to resolve it:

- Perform a rollout restart of the buildkitd pods to ensure they start correctly:

```
kubectl rollout restart daemonset buildkitd -n [***WORKBENCH  
NAMESPACE***]
```

DSE-43704: Rename custom tee binary to cml-tee

Certain vulnerability scanners may incorrectly flag Cloudera AI as using a vulnerable version of coreutils and tee.

Cloudera AI services include a custom tee binary, developed entirely in-house by Cloudera, which is not based on the open-source coreutils library. The current version of Cloudera's custom tee command is 0.9, which may be mistakenly identified as the tee command from coreutils that contains known vulnerabilities.

DSE-44827: The model is failing with unknown status

Model deployments may fail to start if the model build relies on an add-on that was hotfixed in the release. This issue occurs when the model deployment restarts and the add-on hotfixing process overlap, leading to conflicts.

Workaround:

To resolve this issue, create a new build and deploy it for the affected model deployments.

DSE-44913: Spark executor fluentd init container fails

A bug introduced in a recent change to handle dynamic volume association with Spark pods causes the Fluentbit executor, responsible for log collection, to crash. As a result, logs from affected pods are not included in debug bundles, and the Spark executor pod logs will not appear in the session's **Logs** tab in the UI.

Despite this, the Spark executors will continue to function as expected—the engine container will still start, and the script will execute.

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-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 [***existing 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 private-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-private-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-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.

DSE-44698: Restart Stability - 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***]
```

DSE-48606 Team creation and synchronization in Cloudera AI on premises adds default user permission

When creating and synchronizing a team, Cloudera AI on premises does not store the selected user permission, viewer, collaborator, or administrator, in the database. Instead, it assigns a default permission, granting write access to all team members.

There is no workaround.

DSE-40756: Resource Profile field displays empty on Job Settings page

After creating or upgrading a job and saving a **Resource Profile**, the **Resource Profile** field on the **Job Settings** pages might appear empty.

There is no workaround.

DSE-47904: Resource Profile is not visible for jobs and applications

After creating or upgrading a job and saving a **Resource Profile**, the **Resource Profile** field on the **Application Settings** pages might appear empty.

There is no workaround.

DSE-56330: Specific GPU quota support with the default Kubernetes scheduler

In Cloudera AI on premises 1.5.5, when the workbench is not configured to use YuniKorn, the UI allows Administrators to configure individual quotas for each GPU type, for example, 1 GPU each for A100, H100, H200, and V100, for a total of 4 GPUs. However, you can still launch sessions that consume up to the total configured GPU count from a single GPU type, for example, 4 V100 GPUs, instead of being limited to 1 V100 GPU.

Workaround:

In Site Administration, navigate to Runtimes and configure the **Maximum GPUs per Workload** setting. This setting limits the total number of GPUs that you can select when configuring and launching an individual session profile. The limit applies to the overall GPU quota and not to a specific GPU type.

DSE-57724: Sessions fail with argument list too long error message

In environments with high levels of concurrent workloads, or where stale Kubernetes Service (svc) objects accumulate within a user namespace, new sessions and jobs can fail during the init container phase and enter the Init:Error state. This issue is typically caused by exhaustion of available NodePort service allocations within the namespace.

Workaround:

1. Limit concurrency: Keep the number of concurrent sessions within a single user namespace below approximately 700. Use lower limits when Spark Pushdown is enabled, because it creates additional NodePort services.
2. Remove stale services.
 - a. Verify that no active workloads are running in the affected user namespace.
 - b. Use the OpenShift or Kubernetes CLI to manually delete stale `cdsw-runtime-*` service objects by running the following command:

```
oc delete svc -n [***USER-NAMESPACE***] -l [***SELECTOR***]
```

Cloudera AI Registry known issues

DSE-44319 The model import from Model Hub is failing with error to communicate with Cloudera AI Registry

If you have used self-signed certificates or your trust store lacks the certificate required to trust the Cloudera AI Registry domain, the Cloudera AI Registry calls from the UI might fail.

Figure 1: Importing a model fails

Import Model


Mistral Instruct

Details

Mistral Instruct is a language model that can follow instructions, complete requests, and generate creative text formats. The Mistral Instruct Large Language Model (LLM) is an instruct fine-tuned version of the Mistral.

Tags

Mistral
Instruct
Large Language Model
TensorRT-LLM
Language Generation
NeMo
NVIDIA Validated

Unexpected Error

Error occurred while communicating with AI registry 'https://modelregistry.apps.nlaille-sandbox-ecs-1.vpc.cloudera.com'.

Request ID: Unknown

* Select AI Registry

nlaille-sandbox-ecs-1 model-registry-ml-6b2935b2-c03

nlaille-sandbox-ecs-1 model-registry-ml-6b2935b2-c03

Mistral 7B Instruct

* Select Optimization

Select Optimization

* Enter Name or Select Model

Enter Name or Select Model

Warning

IMPORTANT: Please read the following before proceeding. This AI model is a third party software package that is not validated or maintained by Cloudera, Inc. ("Cloudera"). By configuring and launching this AI model, you will cause such third party software to be downloaded and installed into your environment direct from an external website. If you do not wish to

Import

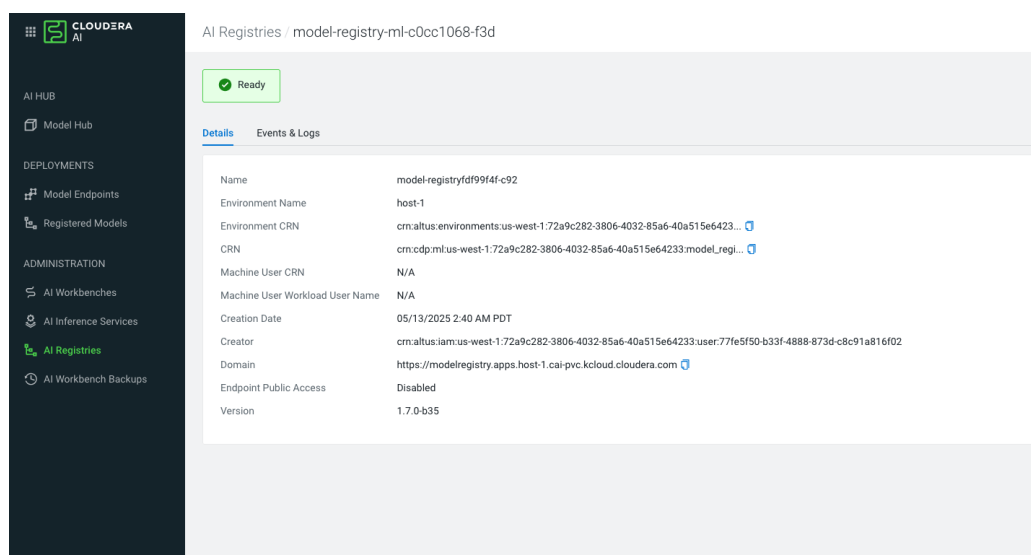
View On NVIDIA

Cancel

Workaround:

When facing issues with an untrusted certificate:

- Select Cloudera AI Registry from the left navigation pane. The Cloudera AI Registry page is displayed.
- Open up the Cloudera AI Registry **Details** page.
- Copy the domain and open it in a new browser.



- Add the certificate permanently to your device's trust store to avoid the risk with the current session.

For permanent trust, export the certificate from your browser and import it into your operating system's certificate manager.

DSE-45572: Cloudera AI Registry failure after upgrading from Cloudera 1.5.4 to 1.5.5

After upgrading Cloudera from version 1.5.4 to 1.5.5, you may find that the Cloudera AI Registry does not appear on the **List AI Registries** page. This issue occurs if you had an active Cloudera AI Registry prior to the upgrade.

Workaround:

To fix the issue and restore the visibility of Cloudera AI Registry, update the tenant field in the Cloudera database:

```
# exec into Cloudera database using kubectl
$kubectl exec -it cdp-embedded-db-0 -n <cdp namespace> -- bash
$ psql
psql (10.23)
Type "help" for help.
postgres=# \c db-mlx
update model_registry set tenant = SPLIT_PART(environment_crn, ':', 5) where tenant = '';
```

DSE-44682: Model deployment is failing at building stage due to TLS issues

TLS-related issues may occasionally occur during the Cloudera AI model build process in Cloudera Embedded Container Service clusters, specifically when pulling images from the container registry. These issues are typically caused by missing registry certificates on the worker nodes, which should be located at the following path: `/etc/docker/certs.d/`.

Workaround:

To address this issue, ensure that the required registry certificates are present on all worker nodes. Follow these steps to recover:

1. Identify a reference worker node.

Select a worker node where the model build process completes successfully without TLS errors. This node is expected to have the correct registry certificates in place. In general, any worker node running `s2i-builder` pods is likely to have the necessary certificates on the nodes of the cluster.

2. Locate the registry certificates.

On the reference worker node, navigate to `/etc/docker/certs.d/[***REGISTRY NAME***]/` and verify that the `registry.crt` file exists.

3. Distribute certificates to affected nodes.

Copy the certificate files (`registry.crt`) from the reference node to the same path (`/etc/docker/certs.d/[***REGISTRY NAME***]/`) on the affected worker nodes that lack the required certificates. Make sure that the certificates for both the image pull and push registries are present and are correctly placed on all worker nodes.

4. Perform rollout restart of the `buildkitd` daemonset so that the certificates are applied properly to the `buildkit` pods.

Cloudera AI Inference service known issues

DSE-44238: Cannot create Cloudera AI Inference service application deployment via CDP CLI when ozone credentials are passed

Cloudera AI Inference service cannot be created via CDP CLI. Create the Cloudera AI Inference service only via the UI.

DSE-44141: Failed to delete deployment in executing `DeleteML-servingApp`

Cloudera AI Inference service fails to remove all namespaces if the Cloudera AI Inference service is deleted post installation failure.

Workaround:

Manually delete the below namespaces from the cluster:

- `knative-serving`
- `kserve`
- `cml-serving`
- `knox-caii`
- `serving-default`



Note:

If you encounter errors related to missing namespaces, you can safely ignore them. These namespaces may have already been deleted as part of the Cloudera AI Inference service removal process.

DSE-46352: Kserve fails to pull images in air-gapped environments where the docker registry is not listed in the trusted list

In air-gapped environments, model endpoints fail with an error message due to the Docker registry not being included in the trusted list. The trusted list is defined in the `registries-skipping-tag-resolving` configuration, which is part of the `config-deployment` configuration map located in the `knative-serving` namespace.

Further known issues with Cloudera AI Inference service:

- Updating the description after a model has been added to a model endpoint will lead to a UI mismatch in the model builder for models listed by the model builder and the models deployed.
- Embedding models function in two modes: query or passage. This has to be specified when interacting with the models. There are two ways to do this:
 - suffix the model id in the payload by either `-query` or `-passage` or
 - specify the `input_type` parameter in the request payload.

For more information, see [NVIDIA documentation](#).

- Embedding models only accept strings as input. Token stream input is currently not supported.
- Llama 3.2 Vision models are not supported on AWS on A10G and L40S GPUs.

- Llama 3.1 70B Instruct model L40S profile needs 8 GPUs to deploy successfully, while NVIDIA documentation lists this model profile as needing only 4 L40S GPUs.
- Mistral 7B models for NIM version 1.1.2 require the max_tokens parameter in the request payload. This API regression is known to affect the Test Model UI functionality for this specific NIM version.
- NIM endpoints will reply with a 307 temporary redirect if the URL ends with a trailing /. Make sure not to have a trailing slash character at the end of NIM endpoint URLs.

Service Packs: Cloudera AI on premises

Review the Service pack features and issues for Cloudera AI on premises 1.5.5 Service Pack 1 release.

Cloudera AI on premises 1.5.5 SP4

What's new in 1.5.5 SP4

Cloudera AI on premises 1.5.5 SP4 delivers a set of new features for Cloudera AI.

Cloudera AI Workbench

NVIDIA GPUDirect Remote Direct Memory Access

NVIDIA GPUDirect Remote Direct Memory Access (RDMA) enables applications to directly access remote GPU memory without routing data through the operating system kernel. By minimizing CPU involvement in network transfers, GPUDirect RDMA reduces latency and overhead while freeing CPU resources for application processing. For more information see [NVIDIA GPUDirect Remote Direct Memory Access for Cloudera AI](#).

Multi-instance GPU support in Cloudera AI Workbench

Multi-Instance GPU (MIG) support in Cloudera AI Workbench enables you to partition a physical GPU into isolated virtual instances to run multiple workloads, such as sessions, models, jobs or applications, simultaneously and increase hardware utilization. This enhancement allows multiple resource-isolated model inference workloads to run concurrently on the same GPU, improving GPU utilization and reducing the need to dedicate an entire physical GPU to a single deployment. For more information see [Multi-instance GPU support in Cloudera AI](#).

Cloudera AI Workbench

The PostgreSQL database of Cloudera AI Registry is now upgraded to version 17 (PG17).

Cloudera AI Inference service

Persistent Volume usage for model storage in Cloudera AI Inference service

Persistent volume storage in Cloudera AI Inference service provides dedicated storage for model artifacts, eliminating dependency on the underlying node storage capacity.

You can use persistent volume storage to host large models when repeated downloads are impractical, such as when the model size exceeds the available root volume capacity. Cloudera AI Inference service downloads the model artifacts once and reuses them across workload restarts, avoiding the need to download the artifacts each time you recreate a workload. For more information see [Persistent Volume usage for model storage in Cloudera AI Inference service](#).

Multi-instance GPU support in Cloudera AI Inference service

Cloudera AI Inference service supports NVIDIA GPU Operator and dynamic fractional GPU allocation (MIG), enabling users to run efficient, resource-isolated inference workloads on MIG-supported GPUs. This enhancement allows users to deploy models by selecting granular GPU memory profiles, helping smaller workloads avoid consuming entire high-cost GPU resources unnecessarily. The Model deployment panel provides predefined and

standardized GPU resource profiles that can be selected based on the memory requirements of the model. For more information see [Creating a Model Endpoint using UI](#).

Enhanced Replica monitoring for Model Endpoints

The Model Endpoint Details page header now features a real-time **Replicas** status row to improve endpoint monitoring and operational visibility. This status provides immediate breakdown counts for Total, Healthy, Pull Image, and Pending replicas across model deployments. By tracking active Kubernetes pod events alongside revision metadata, you can quickly identify deployment progress and image-pull delays directly from the UI without checking underlying cluster logs.

Low-Rank Adaptation adapters with Cloudera AI Inference service

Cloudera AI Inference service supports deploying LoRA (Low-Rank Adaptation) adapters in Cloudera AI 1.5.5 SP4 and higher releases.

This capability allows you to serve multiple fine-tuned model variants from one endpoint, significantly reducing GPU resource consumption compared to deploying each variant as an independent inference service. For more information, see [Low-Rank Adaptation adapters with Cloudera AI Inference service](#).

Filter and discover certified models in Model Hub

Model Hub now provides advanced filtering capabilities using a new **Certification** drop-down list and workload certification badges to help you quickly identify optimized models. You can easily filter the catalog by certification type, such as **Agent Studio Certified**, as well as model family, machine learning task, and provider registry. Certified models include preverified hardware configurations, compatible runtimes, and validated schemas to ensure smooth deployment across Cloudera AI tools. Enterprise administrators can also define custom certification labels directly within air-gapped ModelHubCatalog YAML configurations.

Improvements

Agent Studio

Standardized prompts for Large Language Model (LLM) validation in Agent Studio

When testing a registered LLM in Cloudera AI Agent Studio, both the Test Model and Test Function Calling actions now use a standard, non-editable prompt. Previously, these actions accepted a user-supplied prompt; standardizing the prompt ensures consistent validation of model credentials, network connectivity, and function-calling support across environments. For more information, see [Testing registered models in Agent Studio](#).

Compatibility for Cloudera AI 1.5.5 SP4, Cloudera Runtime components and Data Visualization

Explore Cloudera AI 1.5.5 SP4 and its compatibility with Cloudera Runtime components and with Cloudera Data Visualization across various versions.



Important:

- If you switch from the Red Hat Universal Base Image (Redhat UBI image) to the security hardened image, note that the updated component versions and libraries can impact the Cloudera AI operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

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

For more information, see [Security hardened Spark image migration guide](#).

- From Cloudera Data Services on premises 1.5.5 onwards, Spark 3.4 is no longer available.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported on Spark 3.3.2 and higher.

Table 3: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for security hardened image and for Redhat insecure image

Cloudera Runtime Version	Spark 2.4.x ¹	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.9 SP2	- Spark 2.4.8¹ - Scala 2.11.12¹ - Python 2.7.18¹ - Python 3.6.8¹ - Java 1.8.0_492¹	- Spark 3.3.2 - Scala 2.12.15 - Python 2.7.18¹ - Python 3.8.17¹ - Python 3.10.20 - Java 11.0.25¹ - Java 17.0.19	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.16 - Java 17.0.20	- Airflow 2.11.2 - Python 3.11.16 - Java 17.0.20	Iceberg 1.3.0	1.34	8.1.4.1000-4
7.3.1		NA	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.16 - Java 17.0.20	- Airflow 2.11.2 - Python 3.11.16 - Java 17.0.20	Iceberg 1.5.2	1.34	8.1.4.1000-4
7.3.2		NA	- Spark 3.5.4 - Scala 2.12.19 - Python 3.11.16 - Java 17.0.20	- Airflow 2.11.2 - Python 3.11.16 - Java 17.0.20	Iceberg 1.5.2	1.34	8.1.4.1000-4

Supported models and servers in Cloudera AI Inference service

This topic provides a comprehensive inventory of model servers, runtimes, and supported model architectures available in Cloudera AI Inference service.

Hugging Face model server

Table 4: Hugging Face model server components

Component	Version
Hugging Face Model Server (KServe)	unknown
- vLLM	0.27.1

Triton Inference server

Table 5: Triton Inference server components

Component	Version
Triton Inference Server	26.03.07
- PyTorch	2.11.0
- TensorFlow	2.18.0
- scikit-learn	1.8.0
- XGBoost	3.1.2
- LightGBM	4.6.0
- CatBoost	1.2.7
- MLflow	3.12

NVIDIA NIM models



Note: The gemma2, nemotron#nano#12b, and paddleocr models are planned for deprecation. Cloudera recommends avoiding these models and selecting alternative options available in Modelhub.

Table 6: Supported NVIDIA NIM models

Vendor	Model	NVIDIA GPU Cloud Container	Versions
Baidu	NeMo Retriever PaddleOCR	unknown	1.4.0, 1.5.0
BigCode	StarCoder2 7B	nvcr.io/nim/bigcode/starcoder2-7b	1.8.1, 1.14.1, 1.15.3, 2.0.9
DeepSeek AI	DeepSeek R1	unknown	1.7.3
DeepSeek AI	DeepSeek R1 Distill Llama 70B	unknown	1.5.2
DeepSeek AI	DeepSeek R1 Distill Llama 8B	unknown	1.5.2
DeepSeek AI	DeepSeek V4 Flash	nvcr.io/nim/deepseek-ai/deepseek-v4-flash	1.10.0-variant
Google	Gemma 4 31B IT	nvcr.io/nim/google/gemma-4-31b-it	1.7.1-variant
Meta	Llama 3.1 405B Instruct	unknown	1.3.0
Meta	Llama 3.1 70B Instruct	nvcr.io/nim/meta/llama-3.1-70b-instruct	1.2.1, 1.3.3, 1.8.5, 1.14.0, 1.14.0-pb5.5-stig-fips-x86-64, 1.14.0-pb5.2-stig-fips-x86-64, 2.0.11
Meta	Llama 3.1 8B Instruct	nvcr.io/nim/meta/llama-3.1-8b-instruct	1.2.2, 1.3.3, 1.8.6, 1.13.1, 1.14.0-pb5.5-stig-fips-x86-64, 1.14.0-pb5.2-stig-fips-x86-64, 2.0.12
Meta	Llama 3.2 11B Vision	unknown	1.1.1
Meta	Llama 3.2 1B Instruct	unknown	1.6.0, 1.8.6, 1.12.0
Meta	Llama 3.2 3B Instruct	unknown	1.6.0, 1.8.6, 1.10.1
Meta	Llama 3.2 90B Vision	unknown	1.1.1
Meta	Llama 3.3 70B Instruct	nvcr.io/nim/meta/llama-3.3-70b-instruct	1.8.2, 1.8.5, 1.14.0, 1.15.1, 2.0.3, 2.0.9
MiniMax AI	MiniMax M2.5	nvcr.io/nim/minimax-ai/minimax-m25	1.7.1
MIT	MIT Boltz2	nvcr.io/nim/mit/boltz2	1.1.0, 1.3.0, 1.5.0, 1.8.0
Mistral AI	Mistral 7B Instruct v0.3	unknown	1.1.2, 1.3.0, 1.12.0

Vendor	Model	NVIDIA GPU Cloud Container	Versions
Mistral AI	Mistral Medium 3.5 128B	nvcr.io/nim/mistralai/mistral-medium-3.5-128b	2.0.6-variant
Mistral AI	Mixtral 8x22B Instruct v0.1	unknown	1.2.2
Mistral AI	Mixtral 8x7B Instruct v0.1	unknown	1.2.1, 1.3.0, 1.8.4, 1.12.0
NVIDIA	Cosmos 3 Nano Reasoner	nvcr.io/nim/nvidia/cosmos3-nano-reasoner	1.7.0
NVIDIA	Cosmos 3 Super Reasoner	nvcr.io/nim/nvidia/cosmos3-super-reasoner	1.7.0
NVIDIA	Cosmos Reason2 8B	nvcr.io/nim/nvidia/cosmos-reason2-8b	1.7.0, 1.8.0
NVIDIA	Llama 3.1 Nemotron Nano 4B V1.1	unknown	1.8.4, 1.8.5
NVIDIA	Llama 3.1 Nemotron Nano 8B	unknown	1.8.2, 1.8.4
NVIDIA	Llama 3.2 EmbedQA 1B	nvcr.io/nim/nvidia/llama-3.2-nv-embedqa-1b-v2	1.5.0, 1.10.0, 1.11.3-stig-fips-x86, 1.14.4-stig-fips-x86
NVIDIA	Llama 3.2 RerankQA 1B v2	nvcr.io/nim/nvidia/llama-3.2-nv-rerankqa-1b-v2	1.3.1, 1.8.0, 1.9.3-stig-fips-x86, 1.12.4-stig-fips-x86
NVIDIA	Llama 3.3 Nemotron Super 49B V1	unknown	1.8.5, 1.8.6, 1.10.1
NVIDIA	Llama 3.3 Nemotron Super 49B V1.5	unknown	1.14.0, 1.14.0-pb5.3, 2.0.3
NVIDIA	NVIDIA ASR Parakeet 1.1B CTC EN-US - Offline	nvcr.io/nim/nvidia/riva-asr/parakeet-1-1b-ctc-en-us-offline	1.4.0, 1.5.3
NVIDIA	NVIDIA Riva ASR Whisper Large v3	nvcr.io/nim/nvidia/riva-asr/whisper	1.3.0, 1.3.1, 1.4.0, 1.5.1
NVIDIA	NVIDIA Riva TTS Magpie Multilingual	nvcr.io/nim/nvidia/riva-tts/magpie-tts-multi-lingual	1.6.0, 1.10.0
NVIDIA	NeMo Retriever Graphic Elements	unknown	1.3.0, 1.6.0
NVIDIA	NeMo Retriever Page Elements	unknown	1.3.0, 1.6.0
NVIDIA	NeMo Retriever Page Elements v3	unknown	1.7.0
NVIDIA	NeMo Retriever Table Structure	unknown	1.3.0, 1.6.0
NVIDIA	NeMo Retriever-Parse	unknown	1.2.0
NVIDIA	Nemotron 3 Nano 30B	nvcr.io/nim/nvidia/nemotron-3-nano	1.7.0, 2.0.3, 2.0.11
NVIDIA	Nemotron 3 Nano Omni 30B A3B Reasoning	nvcr.io/nim/nvidia/nemotron-3-nano-omni-30b-a3b-reasoning	2.0.4-variant
NVIDIA	Nemotron 3 Super 120B A12B	nvcr.io/nim/nvidia/nemotron-3-super-120b-a12b	1.8.1, 2.0.3, 2.0.11
NVIDIA	Nemotron 3 Ultra 550B A55B	nvcr.io/nim/nvidia/nemotron-3-ultra-550b-a55b	2.0.12
NVIDIA	Nemotron Graphic Elements	nvcr.io/nim/nvidia/nemotron-graphic-elements-v1	1.8.0
NVIDIA	Nemotron Nano 12B V2 VL	unknown	1.5.0
NVIDIA	Nemotron Page Elements v3	nvcr.io/nim/nvidia/nemotron-page-elements-v3	1.8.0
NVIDIA	Nemotron Parse	nvcr.io/nim/nvidia/nemotron-parse	1.5.0
NVIDIA	Nemotron Table Structure	nvcr.io/nim/nvidia/nemotron-table-structure-v1	1.8.0
OpenAI	OpenAI GPT-OSS 120B	nvcr.io/nim/openai/gpt-oss-120b	1.12.1, 1.12.3, 1.12.4, 2.0.3, 2.0.10

Vendor	Model	NVIDIA GPU Cloud Container	Versions
OpenAI	OpenAI GPT-OSS 20B	nvc.io/nim/openai/gpt-oss-20b	1.12.1, 1.12.3, 1.12.4, 2.0.3, 2.0.11

Hugging Face supported model architectures

Text generation models (139)

afmoeforcausal, apertusforcausal, aquilaforcausal, aquilamodel, arceeforcausal, arcticforcausal, axk1forcausal, baichuanforcausal, bailingmoeforcausal, bailingmoev2_5forcausal, bailingmoev2forcausal, bambaforcausal, bloomforcausal, chatglmforconditionalgeneration, chatglmmodel, cohere2forcausal, cohereforcausal, cwmforcausal, dbrxforcausal, decilmforcausal, deepseekforcausal, deepseekv2forcausal, deepseekv32forcausal, deepseekv3forcausal, deepseekv4forcausal, dots1forcausal, ernie4_5_moeforcausal, ernie4_5forcausal, exaone4forcausal, exaoneforcausal, exaonemoeforcausal, fairseq2llamaforcausal, falconforcausal, falconh1forcausal, falconmambaforcausal, flexolmoforcausal, gemma2forcausal, gemma3forcausal, gemma3nforcausal, gemma4forcausal, gemmaforcausal, glm4forcausal, glm4moeforcausal, glm4moeliteforcausal, glmforcausal, glmmoedsaforcausal, gpt2lmheadmodel, gptbigcodeforcausal, gptjforcausal, gptneoxforcausal, gptossforcausal, graniteforcausal, granitemoeforcausal, granitemoehybridforcausal, granitemoesharedforcausal, gritlm, grok1forcausal, grok1modelforcausal, hcxvisionforcausal, hcxvisionv2forcausal, hunyuandensev1forcausal, hunyuanmoev1forcausal, hyperclovaxforcausal, hyv3forcausal, internlm2forcausal, internlm2veforcausal, internlm3forcausal, internlmforcausal, iquestloopcoderforcausal, iquestloopcoderforcausal, jais2forcausal, jaislmheadmodel, jambaforcausal, kimilinerforcausal, lfm2forcausal, lfm2moeforcausal, llama4forcausal, llamaforcausal, longcatflashforcausal, mamba2forcausal, mambaforcausal, mimoforcausal, mimov2flashforcausal, minicpm3forcausal, minicpmforcausal, minimaxforcausal, minimaxm1forcausal, minimaxm2forcausal, minimaxtext01forcausal, ministral3forcausal, mistralforcausal, mistrallarge3forcausal, mixtralforcausal, mptforcausal, nemotronforcausal, nemotronhforcausal, nemotronhpuzzleforcausal, olmo2forcausal, olmo3forcausal, olmoeforcausal, olmoforcausal, olmoahybridforcausal, optforcausal, orionforcausal, ouroforcausal, panguembaddedforcausal, pangupromoev2forcausal, panguultramoeforcausal, param2moeforcausal, persimmonforcausal, phi3forcausal, phiforcausal, phimoeforcausal, plamo2forcausal, plamo3forcausal, qwen2forcausal, qwen2moeforcausal, qwen3forcausal, qwen3moeforcausal, qwen3nextforcausal, qwenlmheadmodel, rnj1forcausal, rwwforcausal, sarvammlaforcausal, sarvammoeforcausal, seedossforcausal, solarforcausal, stablelmepochforcausal, stablelmforcausal, starcoder2forcausal, step1forcausal, step3p5forcausal, step3textforcausal, telechat2forcausal, telechat3forcausal, telechatforcausal, teleflmforcausal, xverseforcausal, zamba2forcausal

Embedding models (11)

bertmodel, bertspadesparseembeddingmodel, bgem3embeddingmodel, erniemodel, gemma2model, gemma3textmodel, gtemodel, gtenewmodel, jinaembeddingsv5model, llamabidirectionalmodel, llamamodel

Late interaction models (11)

colbertjinarobertamodel, colbertlfm2model, colbertmodernbertmodel, colmodernvbertforretrieval, colpaliforretrieval, colqwen3, colqwen3_5, hf_colbert, jinaforranking, opscolqwen3model, qwen3vlnemotronembedmodel

Reward models (3)

internlm2forrewardmodel, qwen2forprocessrewardmodel, qwen2forrewardmodel

Token classification models (4)

bertfortokenclassification, erniefortokenclassification, modernbertfortokenclassification, qwen3asrforcedalignerfortokenclassification

Sequence classification models (11)

bertforsequenceclassification, ernieforsequenceclassification, gpt2forsequenceclassification, gtenewforsequenceclassification, jambaforsequenceclassification, jinaforranking, llamabidirectionalforsequenceclassification, llamaonemotronv1forsequenceclassification, modernbertforsequenceclassification, robertaforsequenceclassification, xlmrobertaforsequenceclassification

Multimodal models (106)

ariaforconditionalgeneration, audioflamingo3forconditionalgeneration, ayavisionforconditionalgeneration, bagelforconditionalgeneration, beeforconditionalgeneration, blip2forconditionalgeneration, chameleonforconditionalgeneration, cheers, cheersforconditionalgeneration, cohere2visionforconditionalgeneration, cohereasrforconditionalgeneration, deepseekocr2forcausallm, deepseekocrforcausallm, deepseekvlv2forcausallm, dotsocrforcausallm, eagle2_5_vlforconditionalgeneration, ernie4_5_vlmoeforconditionalgeneration, exaone4_5_forconditionalgeneration, fireredasr2forconditionalgeneration, fireredlidforconditionalgeneration, funasrforconditionalgeneration, funaudiochatforconditionalgeneration, fuyuforcausallm, gemma3forconditionalgeneration, gemma3nforconditionalgeneration, gemma4forconditionalgeneration, glm4vforcausallm, glm4vforconditionalgeneration, glm4vmoeforconditionalgeneration, glmarsrforconditionalgeneration, glmocrforconditionalgeneration, granite4visionforconditionalgeneration, granitespeechforconditionalgeneration, h2ovlchatmodel, hunyuanvlforconditionalgeneration, idefics3forconditionalgeneration, interns1forconditionalgeneration, interns1proforconditionalgeneration, internvlchatmodel, internvlforconditionalgeneration, isaacforconditionalgeneration, kananavforconditionalgeneration, keyeforconditionalgeneration, keyevl1_5forconditionalgeneration, kimiaudioforconditionalgeneration, kimik25forconditionalgeneration, kimivlforconditionalgeneration, lfm2vlforconditionalgeneration, lightonocrforconditionalgeneration, llama4forconditionalgeneration, llama_nemotron_nano_vl, llavaforconditionalgeneration, llavanextforconditionalgeneration, llavanextvideoforconditionalgeneration, lavaonevisionforconditionalgeneration, mantisforconditionalgeneration, midashenglmmodel, minicpmo, minicpmv, minimaxvl01forconditionalgeneration, mistral3forconditionalgeneration, molmo2forconditionalgeneration, molmo2forcausallm, moonshotkimiaforcausallm, musicflamingoforconditionalgeneration, nemotronh_nano_omni_reasoning_v3, nemotronh_nano_vl_v2, nemotronh_super_omni_reasoning_v3, nemotronparseforconditionalgeneration, nvlm_d, opencuaforconditionalgeneration, openpanguvlforconditionalgeneration, ovis, ovis2_5, ovis2_6_moeforcausallm, ovis2_6forcausallm, paddleocrvlforconditionalgeneration, paligemmaforconditionalgeneration, phi3vforcausallm, phi4forcausallmv, phi4mmforcausallm, pixtralforconditionalgeneration, qwen2_5_vlforconditionalgeneration, qwen2_5omniforconditionalgeneration, qwen2_5omnimodel, qwen2audioforconditionalgeneration, qwen2vlforconditionalgeneration, qwen3_5forconditionalgeneration, qwen3_5moeforconditionalgeneration, qwen3asrforconditionalgeneration, qwen3asrrealtimegeneration, qwen3omnimoeoforconditionalgeneration, qwen3vlforconditionalgeneration, qwen3vlmoeforconditionalgeneration, qwenvlforconditionalgeneration, rforconditionalgeneration, skyworkr1vchatmodel, smolvlmforconditionalgeneration, step3vlforconditionalgeneration, stepvlforconditionalgeneration, tarsier2forconditionalgeneration, tarsierforconditionalgeneration, ultravoxmodel, voxtralforconditionalgeneration, voxtralrealtimegeneration, whisperforconditionalgeneration

Speculative Decoding (33)

deepseekmtpmodel, deepseekv4mtpmodel, dflashdraftmodel, eagle3deepseekv2forcausallm, eagle3deepseekv3forcausallm, eagle3llamaforcausallm, eagle3minimaxm2forcausallm, eagle3qwen2_5vlforcausallm, eagle3qwen3vlforcausallm, eagledeepseekmtpmodel, eaglellama4forcausallm, eaglellamaforcausallm, eagleminicpmforcausallm, eaglemistrallarge3forcausallm, erniemtpmodel, exaone4_5_mtp, exaonemoemtp, extracthiddenstatesmodel, glm4moelitemtpmodel, glm4moemtpmodel, glmocrmtpmodel, hyv3mtpmodel, llamaforcausallmeagle3, longcatflashmtpmodel, medusamodel, mimomtpmodel, mlpspeculatorpretrainedmodel, nemotronhmtpmodel, openpangumtpmodel, qwen3_5moemtp, qwen3_5mtp, qwen3nextmtp, step3p5mtp

Transformers supported models (2)

emu3forconditionalgeneration, smollm3forcausallm

Transformers Backend models (10)

transformerembeddingmodel, transformersforcausallm, transformersforsequenceclassification, transformersmoembeddingmodel, transformersmoeforcausallm, transformersmoeforsequenceclassification, transformersmultimodalembeddingmodel, transformersmultimodalforcausallm, transformersmultimodalforsequenceclassification, transformersmultimodalmoeoforcausallm

Using Hugging Face models

Deploying Hugging Face models might require additional configuration details.

Table 7: Using Hugging Face models

Model name	vLLM parameters	GPU	Test payload
google/gemma-4-31B-it	--tensor-parallel-size 2 --quantization fp8 --max-model-len 8192 --max-num-batched-tokens 4096	2 x NVIDIA L40S	<pre>{ "model": "google/gemma-4-31B-it", "messages": [{ "role": "user", "content": "What is Generative AI?" }], "stream": true }</pre>

Supported vLLM command line arguments

Cloudera AI on premises 1.5.5 SP4 uses vLLM 0.27.1 The command line arguments are detailed in [vLLM arguments](#).

The following command line arguments are supported with vLLM 0.27.1:

- --block-size
- --calculate-kv-scales
- --cpu-offload-gb
- --disable-cascade-attn
- --disable-chunked-mm-input
- --disable-sliding-window
- --dtype
- --enable-auto-tool-choice
- --enable-chunked-prefill
- --enable-prefix-caching
- --enforce-eager
- --gpu-memory-utilization
- --kv-cache-dtype
- --load-format
- --logprobs-mode
- --long-prefill-token-threshold
- --max-logprobs
- --max-long-partial-prefills
- --max-model-len
- --max-num-batched-tokens
- --max-num-partial-prefills
- --max-num-seqs
- --max-seq-len-to-capture
- --multi-step-stream-outputs
- --no-enable-prefix-caching
- --num-lookahead-slots
- --num-scheduler-steps
- --pipeline-parallel-size, -pp
- --preemption-mode
- --prefix-caching-hash-algo
- --quantization

- --rope-scaling
- --rope-theta
- --scheduling-policy
- --seed
- --tensor-parallel-size, -tp
- --tool-call-parser
- --trust-remote-code

You can find details on the above listed command line arguments here: [vLLM arguments](#).



Note: Only the above listed vLLM command line arguments are supported.

Fixed issues in Cloudera AI on premises 1.5.5 SP4

This section lists the issues that have been fixed since the last release of Cloudera AI on premises.

DSE-56443: Cloudera AI user interface returned a 431 Request Header Fields Too Large error for users with multiple group memberships

Previously, if you were a member of a large number of groups, you might have encountered a 431 Request Header Fields Too Large error when you accessed the Cloudera AI Registry by using the user interface (UI). This issue occurred because the authentication token grew proportionally with group memberships. When the token size exceeded the header size limit of 8 KB, Knox rejected the request.

This issue is now resolved. Cloudera AI Registry now successfully processes authentication tokens for users with multiple group memberships without exceeding the 8 KB header size limit.

DSE-59641: Missing automatic cleanup of orphaned Istio VirtualServices and Gateway routes

Previously, orphaned Istio VirtualServices, Gateway API routes, and uncollected Persistent Volumes (PVs) were no longer retained in user namespaces when updating application subdomains or deleting associated workloads. This issue is now resolved. The cleanup workflow now automatically identifies and removes these lingering resources from affected namespaces, resolving the resource leak issue.

DSE-55355: Unredacted sensitive data in Cloudera AI service logs

Previously, Cloudera AI service pods (Web, API, Reconciler, Operator, and DB) included unredacted sensitive information, such as environment variables, tokens, passwords, SSH configs, and Kerberos keytabs in service logs and diagnostic bundles. This issue is now resolved. Service logs are now automatically masked as [REDACTED] both in transit and at rest within diagnostic bundles.

DSE-60253: Default accelerator quota is not updated when requested value equals max_gpu_count

Previously, when updating the default accelerator quota for a heterogeneous GPU type, setting the quota value equal to max_gpu_count, the maximum number of that accelerator type available on any node in the cluster, could cause the API to return a success response without saving the update to the database. This issue has been resolved.

DSE-60254: No UI error is displayed when user or team quotas exceed the Workbench resource pool quota

Previously, when a Workbench resource pool quota was configured for CPU, memory, or GPU resources, the UI allowed administrators to enter user or team quota values that exceeded the resource pool quota without displaying a validation error. Although the backend API correctly rejected the request and returned an error message, the validation was not surfaced in the UI. This issue has been resolved for CPU, memory, and GPU resource types.

DSE-59835: Heterogeneous GPU custom quota is not reset correctly after reverting to default quota

Previously, when resetting a user's custom heterogeneous GPU quota to the default quota, the system could fail to persist the default quota configuration. As a result, workloads launched afterward could continue using the previously configured custom quota values instead of the default quota settings. This issue has been resolved.

Known issues for Cloudera AI on premises 1.5.5 SP4

You might run into some known issues while using Cloudera AI on premises 1.5.5 SP4.

Cloudera AI on premises 1.5.5, 1.5.5 SP1, 1.5.5 SP2 and 1.5.5 SP3 existing known issues are carried into Cloudera AI on premises 1.5.5 SP4.

For more details, see:

- [Known issues for Cloudera AI on premises 1.5.5](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP1](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP2](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP3](#)

For known issues fixed since Cloudera AI on premises 1.5.5, see:

- [Fixed issues in 1.5.5 SP1](#)
- [Fixed issues in 1.5.5 SP2](#)
- [Fixed issues in 1.5.5 SP3](#)

Cloudera AI Workbench

DSE-59571: MIG variant profiles are not supported in Cloudera AI

Cloudera AI supports standard Multi-instance GPU (MIG) profiles only, MIG variant profiles are not supported, including profile variants such as +me, +me.all, +gfx, and -me (for example, 1g.10gb +me).

DSE-58791: Not supported GPU usage tracking for MIG MIX GPU in Grafana Quota Management Dashboard

When using Multi-Instance GPU (MIG) slicing with mixed GPU profiles in Cloudera AI Workbench, GPU usage metrics are not tracked or displayed within the Grafana Quota Management Dashboard. GPU usage tracking for Multi-Instance GPU (MIG) slicing and mixed GPU configuration profiles is not supported in the Grafana Quota Management Dashboard.

Workaround:

View GPU utilization through the standard Usage Dashboard or the Resource Usage Dashboard, that actively supports and displays resource tracking for MIG GPUs.

DSE-61563: MIG slice usage is displayed as GPU count in the GPU Usage graph

When using Multi-Instance GPU (MIG) slicing, each MIG slice is currently counted as one GPU in the **GPU Usage graphs** on the **Usage** and **Resource Usage** dashboards.

DSE-43884: Default timeout insufficient for large databases to be migrated from Cloudera Data Science Workbench to Cloudera AI

When migrating a Cloudera Data Science Workbench workspace containing large database tables, such as a dashboards table with several hundred thousand unstopped rows, the database update process inside the cml-db-migrate pod can take several hours to complete. Because the default job completion timeout is set to 10 minutes, the migration fails prematurely due to a timeout waiting for the cml-db-migrate job to finish.

DSE-58875: Workbench applications can remain indefinitely in the Starting state when the web server is unreachable

Previously, if a deployed application web server was unreachable on localhost:\$CDSW_APP_PORT, for example, because of an incorrect port binding, a crashed framework, or HTTPS being

served on an HTTP port, the application could remain indefinitely in the Starting state. Since the underlying Kubernetes pod remained healthy and in the Running state, the failure was not detected or reported to users or administrators.

A startup timeout reconciler has been added to address this issue. If an application remains in the Starting state beyond the configured timeout after the workload begins running, the reconciler automatically transitions the application to the Failed state and displays the APP_WEBSERVER_UNREACHABLE error code on the application card and in the logs.

DSE-57724: Sessions fail with argument list too long error at high session concurrency

If you scale Cloudera AI Workbench to high concurrent session counts (approximately 850 through 1,000 sessions), new session pods fail to start and enter an Init:Error state. The engine-deps init container displays the following error:

```
/deps/engine-deps-install: argument list too long
```

This issue occurs because Kubernetes automatically injects environment variables for all namespace Services into newly created pods. At high session concurrency or when stale runtime Service records accumulate, the volume of injected environment variables exceeds environment size limits, preventing container initialization scripts from running.

Workaround:

- Limit concurrency: Keep the number of active concurrent sessions in any single user namespace below ~700.
- Resource cleanup: Manually clean up any unremoved or stale cdsw-runtime-* Service objects in affected namespaces to reduce injected environment variable overhead.

Cloudera AI Registry

DSE-61353, DSE-61273: Model privacy and Administrator roles

Private models are visible only to their owners. Similarly, only the model owner can delete a model, regardless of whether the model is private or public.

This implies the following:

- Users assigned elevated roles, MLAdmin and PowerUser, cannot override these ownership constraints.
- Only the user who owns the model can perform the deletion.

DSE-61665: Post-upgrade relinking required for remote Cloudera AI Registries to deploy models in Cloudera AI Inference service

After upgrading to Cloudera AI 1.5.5 SP4, deploying a model from a Cloudera AI Registry in a different environment than the Cloudera AI Inference service fails. Deploying endpoint API requests, like `/api/v1alpha1/deployEndpoint`, returns an HTTP 403 Forbidden error, similar to:

```
{
  "err": "rpc error: code = PermissionDenied desc = user does not have access to the remote registry environment"
}
```

After the upgrade from Cloudera AI 1.5.5 SP3 CHF3 to 1.5.5 SP4, if the Cloudera AI Inference service and the Cloudera AI Registry are in different environments, they need to be relinked to allow Cloudera AI Registry access across environments.

Workaround:

To restore model deployment functionality from the remote Cloudera AI Registry, relink the registry from the Cloudera AI Inference service application:

1. In the **Cloudera** console, click the Cloudera AI tile.

The **Cloudera AI** page is displayed.

- Click AI Inference Services under ADMINISTRATION in the left navigation menu.

The AI Inference Services page is displayed.

- For a selected Cloudera AI Inference service instance, click the  icon from the Actions menu and select the Update Storage Configuration option.

The Update Storage Configuration page is displayed.

- Select the Cloudera AI Registry from the Select AI Registry drop-down list that you want to relink to the Cloudera AI Inference service instance.



Note: Cloudera AI Registries linked to the Cloudera AI Inference service in

Cloudera AI 1.5.5 SP3 are displayed as Linked items after the upgrade. However, Cloudera AI Registries located in a different environment than the Cloudera AI Inference service must be relinked.

Once relinked, the secret is updated with the required roles and you can deploy models as expected.

Cloudera AI Inference service

Cloudera AI Inference service

In Cloudera AI 1.5.5 SP4, HGX support in the Cloudera AI Inference service is limited to single-node deployments.

DSE-61664: Cloudera AI Inference service instance appears disabled during endpoint creation in cross-registry environments

In cross-registry configurations where the Cloudera AI Inference service instance and Cloudera AI Registry are deployed in different environments, when you attempt to create a model endpoint through the Cloudera AI Inference service user interface (UI) by clicking Create Endpoint, the Cloudera AI Inference service instance appears disabled or unavailable because of a strict environment registry check. This issue occurs in both fresh installations and upgraded environments.

Workaround:

Deploy the model directly from the Cloudera AI Registry interface instead of the Cloudera AI Inference service user interface:

- Go to Registered Models under Deployments in the left navigation menu.
- Select the model that you want to deploy.
- Click Deploy Model and select your target Cloudera AI Inference service instance.

DSE-49451: Tensor parallelism fails when multiple MIG slices are scheduled on the same host

Tensor parallelism is not supported when you schedule multiple Multi-Instance GPU (MIG) slices on the same host. If you attempt to deploy a model using tensor parallelism across multiple MIG slices on a single host, deployments might fail with an NVIDIA Collective Communications Library (NCCL) error: unhandled system error, including after pod restarts.

Workaround:

To work around this issue, perform one of the following actions:

- Use a single MIG slice instead of multiple slices.
- Select a larger GPU resource profile.
- Deploy a smaller model, or reduce memory requirements by setting the `--max_model_len` parameter.

DSE-49451: Host displays as Invalid when GPUs use inconsistent Multi-Instance GPU profile types

When you configure a single Multi-Instance GPU (MIG) profile across GPUs on a host, each physical GPU must use one uniform profile slice type. If a GPU is configured with multiple slice types, the host status displays as Invalid, and available resource profiles do not appear in the user interface (UI).

Workaround:

Before deploying a model endpoint, reconfigure MIG on the host so that each GPU uses one consistent profile type:

1. Go to Cloudera Management Console UI.
2. Select **Administration**.
3. Select the **MIG Configuration** tab.

DSE-61631: Knox API token authentication fails on Cloudera AI Inference service endpoints after upgrade

Following the upgrade from Cloudera AI 1.5.5 SP2 to 1.5.5 SP4, in OpenShift Container Platform environment with upgrades from Openshift 4.19 to 4.21, API requests to Cloudera AI Inference service model serving endpoints that use Knox-generated API tokens fail with an HTTP 401 error. Requests authenticated with standard JSON Web Token (JWT) credentials continue to work as expected.

This authentication failure originates within the Knox external authorization chain (cml-serving-remoteauth-remoteauthprovider) at the ExtAuthz layer before reaching the model endpoint. During the upgrade from version Cloudera AI 1.5.5 SP2 to 1.5.5 SP4, the Knox serving topology descriptor or token validation trust chain might become mismatched.

Workaround:

1. Determine the Knox Authentication URL.
 - a. Identify your Knox Gateway host in Cloudera Management Console. On Cloudera on premises, your target URL must follow the following exact structure:

```
https://[***KNOX-GATEWAY-HOST***]:8443/gateway/cdp-proxy-token/auth/api/v1/pre
```

Example

```
https://ccycloud-10.cai-719.root.comops.site:8443/gateway/cdp-proxy-token/auth/api/v1/pre
```

2. Patch Knox via kubectl.
 - a. Set your specific URL as an environment variable, then apply the patch to update the Knox capabilities in the cml-serving namespace.

```
REMOTE_AUTH_URL="https://ccycloud-10.cai-719.root.comops.site:8443/gateway/cdp-proxy-token/auth/api/v1/pre"
kubectl patch kxapabilities -n cml-serving remoteauth --type=merge -p "{
  \"spec\": {
    \"capabilities\": [{
      \"type\": \"remoteauthprovider\",
      \"overrides\": {
        \"knox-auth-service.acl\": \"*:*:*\",
        \"remote.auth.url\": \"${REMOTE_AUTH_URL}\"
      }
    }]
  }
}"
```

3. Verify the update.

- a. Query the configuration to confirm the new URL was successfully written to the cluster.

```
kubectl get knoxcapabilities -n cml-serving remoteauth -o ya  
ml | grep remote.auth.url
```

Expected outcome:

```
kubectl get knoxcapabilities -n cml-serving remoteauth -o ya  
ml | grep remote.auth.url  
remote.auth.url: https://ccycloud-10.cai-719.root.co  
mops.site:8443/gateway/cdp-proxy-token/auth/api/v1/pre
```

Limitations

DSE-56558: Canary deployment

Canary deployment is supported only for MLflow models in Cloudera AI 1.5.5 SP4 and higher releases. It is not supported for LLM workloads.

Spark External Shuffle Service for Spark on Kubernetes

Cloudera AI does not support Spark External Shuffle Service for Spark on Kubernetes.

Cloudera AI on premises 1.5.5 SP3

What's new in 1.5.5 SP3

Cloudera AI on premises 1.5.5 SP3 delivers a set of new features for Cloudera AI.

Cloudera AI Registry

Cloudera AI Registry in-place upgrade

Cloudera AI Registry can be upgraded in place on the existing release within the same namespace in Cloudera AI 1.5.5 SP3 and higher releases.

The in-place upgrade preserves all Kubernetes resources, allowing the process to complete in a short time, typically within 15 minutes. For more details, see [Cloudera AI Registry upgrade](#).

Multiple Cloudera AI Registries working with a single Cloudera AI Inference service

Cloudera supports deploying Cloudera AI Inference service connecting to multiple Cloudera AI Registries in Cloudera AI. For more details, see [Multiple Cloudera AI Registries connected to a single Cloudera AI Inference service](#).

API endpoint for MLFlow model download

This new API endpoint allows you to download a model artifact for a specific model version from the Cloudera AI Registry. It supports models that use the MLFLOW repository type; other repository types are not compatible. For more details, see [Downloading MLFlow models using API endpoint](#).

Cloudera AI Inference service

Fine-grained access control

Fine-grained access control allows Administrators to define specific access levels for Model Endpoints for individual users or groups. For more details, see [Configuring fine-grained access control for Model Endpoints](#).

Knox API key support

The Cloudera AI Inference service now accepts the new Knox API keys, enabling long-lived connectivity to Cloudera AI Inference service Model Endpoints.



Note: The new Knox API key support is certified with Cloudera Runtime 7.1.9 SP2 and 7.3.2.

Cloudera AI Workbench

Quota management for General Availability

Quota Management enables you to control how resources are allocated within your Cloudera AI Workbench on user and on team level. In order to prevent a single session, job, or other workload from consuming all the available cluster resources, you can limit the number of CPUs, the amount of memory allocated by user, business units, or Data Service by defining resource pools that set resource limits. For more details, see [Quota Management overview](#).

New alert for insufficient quota

When creating a session, job, or application, an alert notifies the user if the configuration does not have sufficient quota.

User ID decoupled from user namespace

With the introduction of dynamic pool assignment, UserIDs no longer map predictably to user namespace sequence numbers, and the ds-username label is no longer included in new user namespaces.

Diagnostic bundles for a Cloudera AI Workbench

New types of logs, the User logs, can be generated and downloaded from the Diagnostic page in the Site Administration settings of the Cloudera AI. For more details, see [Downloading diagnostic bundles for a workbench](#).

Automated application restart functionality

Failed applications are now automatically restarted up to three times (with a five-minute delay between attempts) to minimize downtime. This feature is enabled by default, but Site Administrators can disable it globally under Site Administration settings. For more information, see [Disabling global Application restarts](#).

Refined Project list filters and scoping

The Projects list page now features a refined **My Projects** filter that displays only projects explicitly owned by the logged-in user, moving shared work to a brand-new **My Team Projects** filter for individual collaborators and team members. This behavioral change makes it easier to isolate personal projects without scrolling through shared repositories.

Self-service Run As service account assignment for contributors

Project Contributors can now independently assign service accounts (machine users) to workloads using the "Run as" feature, eliminating the need for Site Administrator intervention. To maintain platform security, Contributors are strictly limited to selecting service accounts that hold an Operator role within the project or team. Service accounts with Administrator privileges remain restricted to Site Administrators, Project Administrators, and Project Owners. This self-service capability applies to creating and updating jobs, applications, and models across both v1 and v2 APIs. For more information, see, [Creating a Workload as a Contributor](#).

ML Runtimes

Hardened Chainguard ML Runtimes

This release introduces new Hardened Edition ML Runtimes based on Chainguard images. These runtimes are designed to meet strict security standards and provide enhanced protection for your workloads and are released behind a paywall.

Hardened Runtime workloads do not use the Java version provided by the Hadoop Runtime add-on. Instead, they rely on the Java 17 installation included in the Runtime image.

Hadoop Runtime add-on images

Hadoop Runtime add-on images are upgraded to Ubuntu 24.04 LTS.

Compatibility for Cloudera AI 1.5.5 SP3, Cloudera Runtime components and Data Visualization

Explore Cloudera AI 1.5.5 SP3 and its compatibility with Cloudera Runtime components and with Cloudera Data Visualization across various versions.



Important:

- If you switch from the Red Hat Universal Base Image (Redhat UBI image) to the security hardened image, note that the updated component versions and libraries can impact the Cloudera AI operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

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

For more information, see [Security hardened Spark image migration guide](#).

- From Cloudera Data Services on premises 1.5.5 onwards, Spark 3.4 is no longer available.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported on Spark 3.3.2 and higher.

Table 8: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for security hardened image and for Redhat insecure image

Cloudera Runtime Version	Spark 2.4.x ²	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.9 SP2	- Spark 2.4.8² - Scala 2.11.12² - Python 2.7.18² - Python 3.6.8² - Java 1.8.0_492²	- Spark 3.3.2 - Scala 2.12.15 - Python 2.7.18² - Python 3.8.17² - Python 3.10.20 - Java 11.0.25² - Java 17.0.19	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.15 - Java 17.0.19	- Airflow 2.11.2 - Python 3.11.15 - Java 17.0.19	Iceberg 1.3.0	1.32	8.1.2-b41
7.3.1		NA	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.15 - Java 17.0.19	- Airflow 2.11.2 - Python 3.11.14 - Java 17.0.19	Iceberg 1.5.2	1.32	8.1.2-b41
7.3.2		NA	- Spark 3.5.4 - Scala 2.12.19 - Python 3.11.15 - Java 17.0.19	- Airflow 2.11.2 - Python 3.11.15 - Java 17.0.19	Iceberg 1.5.2	1.32	8.1.2-b41

Supported models and servers in Cloudera AI Inference service

This topic provides a comprehensive inventory of model servers, runtimes, and supported model architectures available in Cloudera AI Inference service.

Hugging Face model server

Table 9: Hugging Face model server components

Component	Version
Hugging Face Model Server (KServe)	0.18.0
- vLLM	0.20.0

Triton Inference server

Table 10: Triton Inference server components

Component	Version
Triton Inference Server	25.08.06
- PyTorch	2.11
- TensorFlow	2.18
- scikit-learn	1.8
- XGBoost	3.1.2
- LightGBM	4.6.0
- CatBoost	1.2.7
- MLflow	3.12

NVIDIA NIM models



Note: The gemma2, nemotron#nano#12b, and paddleocr models are planned for deprecation. Cloudera recommends avoiding these models and selecting alternative options available in Modelhub.

Table 11: Supported NVIDIA NIM models

Vendor	Model	NVIDIA GPU Cloud Container	Versions
Baidu	NeMo Retriever PaddleOCR	nvcr.io/nim/baidu/paddleocr	1.5.0
BigCode	StarCoder2 7B	nvcr.io/nim/bigcode/starcoder2-7b	1.14.1, 1.15.3
DeepSeek AI	DeepSeek R1	nvcr.io/nim/deepseek-ai/deepseek-r1	1.7.3
DeepSeek AI	DeepSeek R1 Distill Llama 70B	unknown	1.5.2
DeepSeek AI	DeepSeek R1 Distill Llama 8B	unknown	1.5.2
Meta	Llama 3.1 405B Instruct	nvcr.io/nim/meta/llama-3.1-405b-instruct	1.3.0
Meta	Llama 3.1 70B Instruct	nvcr.io/nim/meta/llama-3.1-70b-instruct	1.14.0, 1.14.0-pb5.2-stig-fips-x86-64, 1.14.0-pb5.5-stig-fips-x86-64
Meta	Llama 3.1 8B Instruct	nvcr.io/nim/meta/llama-3.1-8b-instruct	1.13.1, 1.14.0-pb5.2-stig-fips-x86-64, 1.14.0-pb5.5-stig-fips-x86-64
Meta	Llama 3.2 1B Instruct	nvcr.io/nim/meta/llama-3.2-1b-instruct	1.12.0
Meta	Llama 3.2 3B Instruct	nvcr.io/nim/meta/llama-3.2-3b-instruct	1.10.1

Vendor	Model	NVIDIA GPU Cloud Container	Versions
Meta	Llama 3.3 70B Instruct	nvcr.io/nim/meta/llama-3.3-70b-instruct	1.14.0, 1.15.1, 2.0.3
MiniMax AI	MiniMax M2.5	nvcr.io/nim/minimax-ai/minimax-m25	1.7.1
MIT	MIT Boltz2	nvcr.io/nim/mit/boltz2	1.1.0, 1.3.0, 1.5.0
Mistral AI	Mistral 7B Instruct v0.3	nvcr.io/nim/mistralai/mistral-7b-instruct-v0.3	1.12.0
Mistral AI	Mixtral 8x7B Instruct v0.1	nvcr.io/nim/mistralai/mixtral-8x7b-instruct-v0.1	1.8.4, 1.12.0
NVIDIA	Cosmos Reason2 8B	nvcr.io/nim/nvidia/cosmos-reason2-8b	1.7.0
NVIDIA	Llama 3.1 Nemotron Nano 4B V1.1	nvcr.io/nim/nvidia/Llama-3.1-Nemotron-Nano-4B-v1.1	1.8.5
NVIDIA	Llama 3.1 Nemotron Nano 8B V1	nvcr.io/nim/nvidia/llama-3.1-nemotron-nano-8b-v1	1.8.4
NVIDIA	Llama 3.2 EmbedQA 1B	nvcr.io/nim/nvidia/llama-3.2-nv-embedqa-1b-v2	1.10.0, 1.11.3-stig-fips-x86
NVIDIA	Llama 3.2 RerankQA 1B v2	nvcr.io/nim/nvidia/llama-3.2-nv-rerankqa-1b-v2	1.8.0, 1.9.3-stig-fips-x86
NVIDIA	Llama 3.3 Nemotron Super 49B V1	nvcr.io/nim/nvidia/llama-3.3-nemotron-super-49b-v1	1.10.1
NVIDIA	Llama 3.3 Nemotron Super 49B V1.5	nvcr.io/nim/nvidia/llama-3.3-nemotron-super-49b-v1.5	1.14.0, 1.14.0-pb5.3, 2.0.3
NVIDIA	NVIDIA ASR Parakeet 1.1B CTC EN-US - Offline	nvcr.io/nim/nvidia/riva-asr/parakeet-1-1b-ctc-en-us-offline	1.4.0
NVIDIA	NVIDIA Riva ASR Whisper Large v3	nvcr.io/nim/nvidia/riva-asr/whisper	1.3.1, 1.4.0
NVIDIA	NVIDIA Riva TTS Magpie Multilingual	nvcr.io/nim/nvidia/riva-tts/magpie-tts-multi-lingual	1.6.0
NVIDIA	NeMo Retriever Graphic Elements	nvcr.io/nim/nvidia/nemoretriever-graphic-elements-v1	1.6.0
NVIDIA	NeMo Retriever Page Elements v3	nvcr.io/nim/nvidia/nemoretriever-page-elements-v3	1.7.0
NVIDIA	NeMo Retriever Table Structure	nvcr.io/nim/nvidia/nemoretriever-table-structure-v1	1.6.0
NVIDIA	Nemotron 3 Nano 30B	nvcr.io/nim/nvidia/nemotron-3-nano	1.7.0, 2.0.3
NVIDIA	Nemotron 3 Super 120B A12B	nvcr.io/nim/nvidia/nemotron-3-super-120b-a12b	1.8.1, 2.0.3
NVIDIA	Nemotron Nano 12B V2 VL	nvcr.io/nim/nvidia/nemotron-nano-12b-v2-vl	1.5.0
NVIDIA	Nemotron Parse	nvcr.io/nim/nvidia/nemotron-parse	1.5.0
OpenAI	OpenAI GPT-OSS 120B	nvcr.io/nim/openai/gpt-oss-120b	1.12.4, 2.0.3
OpenAI	OpenAI GPT-OSS 20B	nvcr.io/nim/openai/gpt-oss-20b	1.12.4, 2.0.3

Hugging Face supported model architectures

Text generation models (139)

afmoeforcausallm, apertusforcausallm, aquilaforcausallm, aquilamodel, arceeforcausallm, arcticforcausallm, axk1forcausallm, baichuanforcausallm, bailingmoeforcausallm, bailingmoev2_5forcausallm, bailingmoev2forcausallm, bambaforcausallm, bloomforcausallm, chatglmforconditionalgeneration, chatglmmodel, cohere2forcausallm, cohereforcausallm, cwmforcausallm, dbrxforcausallm, decilmforcausallm, deepseekforcausallm, deepseekv2forcausallm, deepseekv32forcausallm, deepseekv3forcausallm, deepseekv4forcausallm, dots1forcausallm, ernie4_5_moeforcausallm, ernie4_5forcausallm, exaone4forcausallm, exaoneforcausallm, exaonemoeforcausallm, fairseq2llamaforcausallm,

falconforcausallm, falconh1forcausallm, falconmambaforcausallm, flexolmoforcausallm, gemma2forcausallm, gemma3forcausallm, gemma3nforcausallm, gemma4forcausallm, gemmaforcausallm, glm4forcausallm, glm4moeforcausal
lm, glm4moeliteforcausallm, glmforcausallm, glmmoedsaforcausallm, gpt2lmheadmodel, gptbigcodeforcausallm, gptjforcausallm, gptneoxforcausallm, gptossforcausallm, graniteforcausallm, granitemoeforcausallm, granitemoehy
bridforcausallm, granitemoessharedforcausallm, gritlm, grok1forcausallm, grok1modelforcausallm, hcxvisionforcaus
allm, hcxvisionv2forcausallm, hunyuandensev1forcausallm, hunyuanmoev1forcausallm, hyperclovaxforcausallm, hyv3forcausallm, internlm2forcausallm, internlm2veforcausallm, internlm3forcausallm, internlmforcausallm, ique
stcoderforcausallm, iquestloopcoderforcausallm, jais2forcausallm, jaislmheadmodel, jambaforcausallm, kimilinearfo
rcausallm, lfm2forcausallm, lfm2moeforcausallm, llama4forcausallm, llamaforcausallm, longcatflashforcausallm, mamba2forcausallm, mambaforcausallm, mimoforcausallm, mimov2flashforcausallm, minicpm3forcausallm, mini
cpmforcausallm, minimaxforcausallm, minimaxm1forcausallm, minimaxm2forcausallm, minimaxtext01forcausallm, ministral3forcausallm, mistralforcausallm, mistrallarge3forcausallm, mixtralforcausallm, mptforcausallm, nemotron
forcausallm, nemotronhforcausallm, nemotronhpuzzleforcausallm, olmo2forcausallm, olmo3forcausallm, olmoefor
causallm, olmoforcausallm, olmoahybridforcausallm, optforcausallm, orionforcausallm, ouroforcausallm, panguemb
eddedforcausallm, pangupromoev2forcausallm, panguultramoeforcausallm, param2moeforcausallm, persimmonfor
causallm, phi3forcausallm, phiforcausallm, phimoeforcausallm, plamo2forcausallm, plamo3forcausallm, qwen2for
causallm, qwen2moeforcausallm, qwen3forcausallm, qwen3moeforcausallm, qwen3nextforcausallm, qwenlmheadmo
del, rnj1forcausallm, rwforcausallm, sarvammlaforcausallm, sarvammoeforcausallm, seedossforcausallm, solarfor
causallm, stablelmepochforcausallm, stablelmforcausallm, starcoder2forcausallm, step1forcausallm, step3p5forcausal
lm, step3textforcausallm, telechat2forcausallm, telechat3forcausallm, telechatforcausallm, teleflmforcausallm, xver
seforcausallm, zamba2forcausallm

Embedding models (11)

bertmodel, bertspladesparseembeddingmodel, bgem3embeddingmodel, erniemodel, gemma2model, gemma3textmo
del, gtemodel, gtenewmodel, jinaembeddingsv5model, llamabidirectionalmodel, llamamodel

Late interaction models (11)

colbertjinarobertamodel, colbertlfm2model, colbertmodernbertmodel, colmodernvbertforretrieval, colpaliforretrieval,
colqwen3, colqwen3_5, hf_colbert, jinaforranking, opscolqwen3model, qwen3vlnemotronembedmodel

Reward models (3)

internlm2forrewardmodel, qwen2forprocessrewardmodel, qwen2forrewardmodel

Token classification models (4)

bertfortokenclassification, erniefortokenclassification, modernbertfortokenclassification, qwen3asrforcedalignerfor
tokenclassification

Sequence classification models (11)

bertforsequenceclassification, ernieforsequenceclassification, gpt2forsequenceclassification, gtenewforsequencecla
ssification, jambaforsequenceclassification, jinavlforkranking, llamabidirectionalforsequenceclassification, llamanem
otronvlforksequenceclassification, modernbertforsequenceclassification, robertaforsequenceclassification, xlmrober
taforsequenceclassification

Multimodal models (106)

ariaforconditionalgeneration, audioflamingo3forconditionalgeneration, ayavisionforconditionalgeneration, bagelfor
conditionalgeneration, beeforconditionalgeneration, blip2forconditionalgeneration, chameleonforconditionalgener
ation, cheers, cheersforconditionalgeneration, cohere2visionforconditionalgeneration, cohereasrforconditionalgener
ation, deepseekocr2forcausallm, deepseekocrforcausallm, deepseekv2forcausallm, dotsocrforcausallm, eagle2_5
_vlforkconditionalgeneration, ernie4_5_vlmoeforkconditionalgeneration, exaone4_5_forconditionalgeneration, firereda
sr2forconditionalgeneration, fireredlidforconditionalgeneration, funasrforconditionalgeneration, funaudiochatforcondi
tionalgeneration, fuyuforcausallm, gemma3forconditionalgeneration, gemma3nforconditionalgeneration, gemma4fo
rconditionalgeneration, glm4vforcausallm, glm4vforconditionalgeneration, glm4vmoeforkconditionalgeneration, glma
srforconditionalgeneration, glmocrforconditionalgeneration, granite4visionforconditionalgeneration, granitespeechfor
conditionalgeneration, h2ovlchatmodel, hunyuanvlforkconditionalgeneration, idefics3forconditionalgeneration, inte
rns1forconditionalgeneration, interns1proforconditionalgeneration, internvlforkchatmodel, internvlforkconditionalgenera
tion, isaacforconditionalgeneration, kananavforconditionalgeneration, keyeforkconditionalgeneration, keyevl1_5for

conditionalgeneration, kimiaudioforconditionalgeneration, kimik25forconditionalgeneration, kimivlforconditionalgeneration, lfm2vlforconditionalgeneration, lightonocrforconditionalgeneration, llama4forconditionalgeneration, llama_nemotron_nano_vl, llavaforconditionalgeneration, llavanextforconditionalgeneration, llavanextvideoforconditionalgeneration, lavaonevisionforconditionalgeneration, mantisforconditionalgeneration, midashengllmmodel, minicpm, minicpmv, minimaxvl01forconditionalgeneration, mistral3forconditionalgeneration, molmo2forconditionalgeneration, molmo2forcausal, moonshotkimiaforcausal, musicflamingoforconditionalgeneration, nemotronh_nano_omni_reasoning_v3, nemotronh_nano_vl_v2, nemotronh_super_omni_reasoning_v3, nemotronparseforconditionalgeneration, nvlm_d, opencuaforconditionalgeneration, openpanguvlforconditionalgeneration, ovis, ovis2_5, ovis2_6_moeforcausal, ovis2_6forcausal, paddleocrvlforconditionalgeneration, paligemmaforconditionalgeneration, phi3vforcausal, phi4forcausal, phi4mmforcausal, pixtralforconditionalgeneration, qwen2_5_vlforconditionalgeneration, qwen2_5omniforconditionalgeneration, qwen2_5omnimodel, qwen2audioforconditionalgeneration, qwen2vlforconditionalgeneration, qwen3_5forconditionalgeneration, qwen3_5moeforconditionalgeneration, qwen3asrforconditionalgeneration, qwen3asrrealtimegeneration, qwen3omnimoeoforconditionalgeneration, qwen3vlforconditionalgeneration, qwen3vlmoeforconditionalgeneration, qwenvlforconditionalgeneration, rforconditionalgeneration, skyworkr1vchatmodel, smolvlmforconditionalgeneration, step3vlforconditionalgeneration, stepvlforconditionalgeneration, tarsier2forconditionalgeneration, tarsierforconditionalgeneration, ultravoxmodel, voxtralforconditionalgeneration, voxtralrealtimegeneration, whisperforconditionalgeneration

Speculative Decoding (33)

deepseekmtpmodel, deepseekv4mtpmodel, dflashdraftmodel, eagle3deepseekv2forcausal, eagle3deepseekv3forcausal, eagle3llamaforcausal, eagle3minimaxm2forcausal, eagle3qwen2_5vlforcausal, eagle3qwen3vlforcausal, eagledeepseekmtpmodel, eaglellama4forcausal, eaglellamaforcausal, eagleminicpmforcausal, eaglemistrallarge3forcausal, erniemtpmodel, exaone4_5_mtp, exaonemoemtp, extrathiddenstatesmodel, glm4moelitemtpmodel, glm4moemtpmodel, glmocrmtpmodel, hyv3mtpmodel, llamaforcausalmeagle3, longcatflashmtpmodel, medusamodel, mimomtpmodel, mlpspeculatorpretrainedmodel, nemotronhmtpmodel, openpangumtpmodel, qwen3_5moemtp, qwen3_5mtp, qwen3nextmtp, step3p5mtp

Transformers supported models (2)

emu3forconditionalgeneration, smollm3forcausal

Transformers Backend models (10)

transformerembeddingmodel, transformersforcausal, transformersforsequenceclassification, transformersmoembeddingmodel, transformersmoeforcausal, transformersmoeforsequenceclassification, transformersmultimodalembeddingmodel, transformersmultimodalforcausal, transformersmultimodalforsequenceclassification, transformersmultimodalmoeoforcausal

Using Hugging Face models

Deploying Hugging Face models might require additional configuration details.

Table 12: Using Hugging Face models

Model name	vLLM parameters	GPU	Test payload
google/gemma-4-31B-it	<pre>--tensor-parallel-size 2 --quantization fp8 --max-model-len 8192 --max-num-batched-tokens 4096</pre>	2 x NVIDIA L40S	<pre>{ "model": "google/gemma-4-31B-it", "messages": [{ "role": "user", "content": "What is Generative AI?" }], "stream": true }</pre>

Supported vLLM command line arguments

Cloudera AI on premises 1.5.5 SP3 uses vLLM 0.20. The command line arguments available in vLLM 0.20.0 are detailed in [vLLM arguments](#).

The following command line arguments are supported with vLLM 0.20.0:

- `--block-size`
- `--calculate-kv-scales`
- `--cpu-offload-gb`
- `--disable-cascade-attn`
- `--disable-chunked-mm-input`
- `--disable-sliding-window`
- `--dtype`
- `--enable-auto-tool-choice`
- `--enable-chunked-prefill`
- `--enable-prefix-caching`
- `--enforce-eager`
- `--gpu-memory-utilization`
- `--kv-cache-dtype`
- `--load-format`
- `--logprobs-mode`
- `--long-prefill-token-threshold`
- `--max-logprobs`
- `--max-long-partial-prefills`
- `--max-model-len`
- `--max-num-batched-tokens`
- `--max-num-partial-prefills`
- `--max-num-seqs`
- `--max-seq-len-to-capture`
- `--multi-step-stream-outputs`
- `--no-enable-prefix-caching`
- `--num-lookahead-slots`
- `--num-scheduler-steps`
- `--pipeline-parallel-size, -pp`
- `--preemption-mode`
- `--prefix-caching-hash-algo`
- `--quantization`
- `--rope-scaling`
- `--rope-theta`
- `--scheduling-policy`
- `--seed`
- `--tensor-parallel-size, -tp`
- `--tool-call-parser`
- `--trust-remote-code`

You can find details on the above listed command line arguments here: [vLLM arguments](#).



Note: Only the above listed vLLM command line arguments are supported.

Fixed issues in Cloudera AI on premises 1.5.5 SP3

This section lists the issues that have been fixed since the last release of Cloudera AI on premises.

Workloads

DSE-54380: Missing Spark Executor start times caused dropped workloads in Observability

Previously, some Spark Executor records contained null startTime variables, leading to a NullPointerException exception and causing those workloads to be dropped. This stemmed from a logic that rewrote all failed executors to stopped status, even when they had never started due to issues such as OOM errors or image#pull failures. This issue is now fixed. The logic checks the running_at variables so executors that never started remain failed, while only those that actually ran become stopped. Additional fixes ensure that Kubernetes failure reasons surface correctly, interactive#session executors inherit the proper session CRN, and sessions failing before pod creation are no longer excluded. These changes improve failure reporting and prevent workload records from being lost.

DSE#55381: Internal services routed externally, causing model deployment failures

Previously, when the Enable Public IP Address for Load Balancer option was enabled, model deployment workloads attempted to access internal services, such as the S2i image registry, through external network paths. This forced a Pod NAT Gateway Internet Public Load Balancer routing, violating VPC#only traffic requirements and causing timeouts when NAT IPs were not on an allowlist.

This issue is now fixed. Automatic internal routing through host#level DNS redirection is introduced. A new `inject-s2i-host-mapping` init container in the image#puller DaemonSet maps the external S2i registry domain to the internal Istio ingress IP, ensuring that traffic remains inside the VPC.

DSE#54158: Missing soft delete for Jobs to preserve historical executions

Previously, jobs were hard#deleted from the database, leaving dashboard job#run records without valid references and causing historical job details to display an empty workload_crn identifier. This behavior was inconsistent with sessions, applications, and models, which already use soft delete.

This issue is now fixed. Jobs support soft delete through a new deleted_at timestamp. Deleting a job sets this timestamp instead of removing the row, and associated cron entries are handled the same way. Read paths exclude soft#deleted jobs so they are no longer displayed in the UI, while the existing cleanup process still performs hard deletes after the configured retention period. This ensures historical job runs retain a valid workload_crn` identifier even after the job is deleted.

DSE-50549: Application restart not reflecting updated GPU type chosen

Previously, updating the GPU or other resource settings on a stopped application correctly changed the accelerator_label_id key-value pair in the applications table. However, when the application was restarted, the previous GPU configuration was still used because the restart logic was reading stale values from the dashboards table instead of the updated fields in the applications table. This occurred because of incorrect field names in the .pick() call, which used snake_case instead of the expected camelCase. The same issue affected the resource_group_id and resource_profile_id key-value pairs.

This issue is now fixed. The field mappings are corrected so that updated GPU and resource settings now apply properly when applications are restarted.

DSE-47039: Unconfigurable TGT Ticket Lifetime in Cloudera AI

Previously, the Kerberos TGT (Ticket Granting Ticket) lifetime in Cloudera AI was fixed at the default value of 15 minutes and could not be configured. This led to authentication failures (MutualAuthenticationError) during long-running operations such as database queries exceeding the ticket lifetime.

This issue is now fixed and Cloudera AI allows configuring the TGT ticket lifetime. Users can extend the lifetime, for example, to 8 hours, to support long#running workloads without encountering Kerberos reauthentication failures.

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

Previously, restarting Cloudera Embedded Container Service nodes created duplicate entries in the `node_labels` database table for the same GPU type. This resulted in duplicate GPU registrations being displayed in Site Administration Runtime Workload Accelerators and similarly, appeared in the Resource Profile GPU Type drop-down menu when launching a session. Selecting a duplicate GPU entry while launching a session triggered the This accelerator has been removed error.

This issue is now fixed. To prevent duplicate GPU label entries in `node_labels` a UNIQUE constraint is now added and the application uses an UPSERT so existing entries are updated instead of duplicated.

DSE-49776: Ambiguous error message

Previously, workbenches could be created with uppercase characters or names beginning or ending with a hyphen, but their installation failed because Kubernetes resource names must use only lowercase alphanumeric characters, '-', or '.'.

This issue is now fixed by adding validation that rejects invalid workbench names, preventing provisioning attempts with unsupported naming formats.

Quota management

DSE-50976: Describe Workbench API and Workbench details page UI displays different quota value

The Workbench API and the Workbench Details page were displaying different quota values because the `describe#workbench` API returned the quota of the service pool rather than the actual workbench quota.

This issue is now fixed, and the `describe#workbench` API now correctly returns the `workbench#specific` quota.

DSE-50507: Quota alert is missing on UI

Previously, no alert was triggered when configuration did not have sufficient quota. This issue is now resolved. When creating a session, job, or application, an alert notifies the user if the configuration does not have sufficient quota.

DSE-50459: GPU utilization is not tracked in grafana quota management dashboard

GPU utilization metrics were not visible in the Cloudera AI Quota Management dashboard. This issue is now fixed.

DSE#49120: Indirect project collaborators unable to run inference on deployed model endpoints

Previously, users authorized to run model inference were incorrectly denied access due to incomplete permission checks.

This issue is now fixed. Project authorization logic is centralized to ensure consistent validation across services, and affected services are updated to use the new logic.

DSE-50622: Sessions creation API requests fail with RemoteDisconnected Error at scale

When Quota Management is enabled, concurrent session creation API requests fail with RemoteDisconnected errors.

Jobs API

DSE-55055: Dependent job dropdown limited to 10 Jobs

Following the introduction of pagination to the Jobs API, the parent job selection dropdown previously only displayed the first 10 entries when scheduling a dependent job in projects with a high volume of workloads.

This issue is now resolved by replacing the legacy dropdown with an optimized, paginated component that fetches jobs in batches of 10 as a user scrolls down the list, significantly reducing backend API load. Additionally, a search filter has been integrated into the new interface, allowing users to quickly locate and select from the full list of available parent jobs.

DSE-54853: UI faults from paginated Jobs API response

Following the introduction of pagination to the Jobs API, multiple UI interactions failed to process data beyond the first 10 results. This issue is now resolved, the **Project Overview** page now correctly supports pagination, and the Unable To Get License Details error triggered by dependent jobs outside the first page has been eliminated.

DSE-49110: API v2 projects call fails right after login from project quick find

Previously, when a user logged in for the first time, the authorization token was not included in the request header, causing the API call to fail with an error. This issue is now fixed. The authentication token is now correctly passed on initial login, ensuring seamless project discovery.

Previously, a warning was displayed in the browser console immediately after landing on the home page. This warning did not impact the ability to use the Quick Find feature. This issue is now fixed.

UI-related correction**DSE-48653: Inaccurate Project Last Worked On timestamps**

The Last worked on timestamp on the project listing page was previously updated by automated background events like scheduled job executions, making the activity metric unreliable. This issue has been resolved so that the timestamp now exclusively tracks actual user-driven activity, such as creating projects, modifying workloads, manually starting sessions, or performing file operations. Automated system processes and scheduled job runs will no longer trigger an update to this metric.

Upgrade**DSE-56419: Persistent Volume already exists**

Previously, during the retry of a Cloudera AI workbench upgrade to 1.5.5 SP2 or 1.5.5 SP2 CHF1, or during a fresh upgrade from 1.5.5 SP2 to 1.5.5 SP2 CHF1, the process could fail with an error indicating that the preupgrade workspace PersistentVolume (PV) already exists: PersistentVolume pre-upgrade-workspace-pv-[***WORKBENCH NAME***] already exists.

This occurred when a previous failed upgrade attempt left behind a temporary, undeleted PV, causing subsequent runs to fail immediately.

The issue is now resolved by removing the leftover PV before retrying the upgrade.

Known issues for Cloudera AI on premises 1.5.5 SP3

You might run into some known issues while using Cloudera AI on premises 1.5.5 SP3.

Cloudera AI on premises 1.5.5, 1.5.5 SP1 and 1.5.5 SP2 existing known issues are carried into Cloudera AI on premises 1.5.5 SP3.

For more details, see:

- [Known issues for Cloudera AI on premises 1.5.5](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP1](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP2](#)

For known issues fixed since Cloudera AI on premises 1.5.5, see:

- [Fixed issues in 1.5.5 SP1](#)
- [Fixed issues in 1.5.5 SP2](#)

DSE-50448 and DSE-50449 : Cloudera AI API secret exposure for OpenTelemetry (OTel) collectors

To support Observability on Cloudera Data Services on premises, API keys can now be securely exposed as Kubernetes secrets. This enablement allows OpenTelemetry (OTel) collectors to authenticate and access Cloudera APIs directly, ensuring seamless collection of metrics and trace data from the platform.

DSE-55163: Cloudera Data Visualization query with Hive connection fails with error

Queries that use Hive data connections synced from Cloudera Data Warehouse to Cloudera AI fail with the following error: Failed to open new session: Cannot modify applicationName at runtime. It is not in the list of parameters that are allowed to be modified at runtime.

As a result, Hive queries and database listing operations do not work through synced connection.

Workaround: After syncing the connection, open the Edit Data Connection dialog for the failing connection. In the **Advanced** tab, delete the Application name value and leave the field blank.

DSE-49364: Upgrade Hadoop addon's base image to ubuntu 24.04 LTS

Hadoop Runtime add-on images lower than version 2025.01 are not compatible with 20.04-based Runtime images.

DSE-56531: livelog-publisher crashes when the livelog pod restarts

When a large number of workloads are running in the Cloudera AI Workbench, livelog#publisher pods might crash and restart. During periods when the livelog#publisher is unavailable, container logs from active user workloads might be lost. This impacts the content displayed in the **Logs** tab for Sessions and Jobs, as well as the **Container Logs** tab for Applications.

Workaround: There is no workaround.

Cloudera AI Inference service

The gemma2, nemotron#nano#12b, and paddleocr models are planned for deprecation. Cloudera recommends avoiding these models and selecting alternative options available in Model Hub.

DSE-56443: Cloudera AI UI returns 431 Request Header Fields Too Large error for users belonging to many groups

If you are a member of a large number of groups, you might encounter a 431 Request Header Fields Too Large error when accessing the Cloudera AI Registry using the UI. This occurs because the authentication token grows proportionally with group memberships, and when it exceeds the header size limit of 8 KB, Knox rejects the request.

Workaround: Verify the groups that the user belongs to, remove the user from any unnecessary groups, and then synchronize the environments:

1. Open the Management Console User Management .
2. Search for the user and open Resource Roles tab.
3. Check which environments the user is attached to.
4. Remove the user from any unnecessary groups.
5. Run a synchronization of environments.

DSE-58058: Model Endpoint Details page fails to render for Cloudera AI Inference service

If you navigate to the Model Endpoint Details page for model endpoints hosted in lower Cloudera AI Inference service versions, the page might fail to render or display an Unknown status in Cloudera AI 1.5.5 SP3. This issue occurs because the user interface (UI) requires backend fields that are absent from lower backend versions.

Workaround:

Upgrade to Cloudera AI 1.5.5 SP3 and higher releases.

DSE-59376: Updating existing jobs fails in Cloudera AI 1.5.5 SP3 on Cloudera Embedded Container Service

After upgrading to Cloudera AI 1.5.5 SP3 on Cloudera Embedded Container Service, updating an existing Cloudera AI job, such as changing the runtime, can fail with one of the following validation errors: CPU must be greater than 0 vCPU or Memory must be greater than 0 GiB. Additionally, the Resource Profile displayed in the Update Job configuration always defaults to the first profile in the list instead of displaying the profile currently configured for the job.

Workaround:

1. In the left navigation pane, click Jobs.

The **Jobs Overview** page is displayed.

2. Select the relevant job.

The details of the actual job are displayed.

3. Record the Resource Profile currently configured for the job, including the CPU and memory values. You will use this information when updating the job configuration.

4. Click **Update Job**.

5. Select the Resource Profile selector.

- If the configured profile is not the first entry in the list, select the same profile. Verify that the displayed CPU and memory values match the values recorded in step 3.
- If the configured profile is the first entry in the list, select a different profile, then reselect the original profile.

6. Click Save.

The job updates successfully with the correct resource values.

Known issues for Fine-Grained Access Control

Fine-Grained Access Control is the new permission system for inference endpoints that allows Administrators to control who can View, Access (invoke), or Manage each endpoint.

The following known issues represent initial rollout issues:

UI not respecting permission levels (DSE-57690, DSE- 57699, DSE-57737, DSE-57738)

The frontend currently displays all buttons and actions regardless of the user role. When a user attempts an unauthorized action, the backend correctly denies the request, but the resulting errors can be confusing. The UI currently does not hide or disable actions that the user does not have permission to perform.

DSE-57700: Users with Manage permissions can self#demote or lock themselves out

Users with **Manage** permissions on an endpoint can modify their own permission row, allowing them to downgrade or remove themselves entirely. If they are the only users with Manager permission, the endpoint ends up an orphan with no remaining administrator.

DSE-57739: Empty access control creates ghost endpoint and trap state

When an MLAdmin creates an endpoint with Access Control enabled, the Global Access Control list is initially empty. As a result, MLUsers cannot see the endpoint at all (creating a ghost endpoint). Furthermore, after the administrator selects a role in Modify General Access setting, that selection cannot be cleared. This creates a trap state that prevents the access control list from returning an empty state.

DSE-57740: Missing explicit access restrictions (Hide/Deny) to prevent specific MLUsers from viewing endpoints

Currently, an MLAdmin cannot explicitly restrict, hide, or deny access to a specific endpoint for individual users or groups. Because access can only be granted, this prevents multi#tenancy isolation.

Knox token and API key issues

Knox provides API token-based authentication for programmatic access to inference endpoints, as opposed to UI#based access.

DSE-57450: Knox API token returning 403 Forbidden

API requests authenticated with newly generated Knox API tokens fail with an HTTP 403 Forbidden response and an empty body. The issue affects calls to service endpoints for example, /api/describeService, on Cloudera on premises deployments.

DSE-57743: UI allows token generation for users missing from Knox DB, resulting in silent inference failures (403)

The Knox API token UI issues authentication tokens without verifying whether the user exists in the Knox database. Consequently, MLUsers or MLAdmins not registered in Knox can successfully generate a token. However, this token is invalid causing all subsequent inference requests to fail with a silent 403 Forbidden error.

DSE-57702: Generate API key fails to copy to clipboard in Incognito or Private browsing

In Incognito or Private browsing mode, clicking Generate API Key creates the Knox API token but fails to copy it to the clipboard, displaying a Failed to copy: API key message. Browsers block clipboard access in private mode, leaving the clipboard empty or unchanged.

DSE-57742: Open Knox UI button redirects to base cluster homepage instead of Knox token management

On the endpoint UI, clicking the Open Knox UI button — intended to let users manage and view their issued tokens — redirects to a generic base cluster URL instead of deep-linking to the Knox service interface.

Quota Management issues

DSE-60253: Default accelerator quota is not updated when requested value equals max_gpu_count

When updating the default accelerator quota for a heterogeneous GPU type, setting the quota value equal to max_gpu_count, that is the maximum number of that accelerator type on any node in the cluster, could result in the API returning a success response without committing the update to the database.

DSE-60254: No UI error is displayed when user or team quotas exceed the Workbench resource pool quota

When a Workbench resource pool has a configured quota for CPU, memory, or GPU resources, and an administrator attempts to set a user or team quota for any of these resource types to a value that exceeds the resource pool quota, the UI accepts the input without displaying a validation error.

The backend API correctly identifies the violation and returns the following error message: gpu: provided value is greater than max quota value for resource pool



Note: This behaviour applies consistently to CPU, memory, and GPU resource types.

DSE-60256: Quota Management Grafana Dashboard does not display values for workloads using Heterogeneous GPUs

When workloads are run using heterogeneous GPUs, environments configured with multiple distinct GPU types, the **Quota Management Grafana dashboard** fails to display usage values or resource allocation metrics for those workloads. As a result, the administrators cannot use the **Quota Management Grafana dashboard** to track or audit GPU quota consumption in environments running heterogeneous GPU workloads.



Note: The display issue is applicable only to the Quota Management dashboard. Other monitoring dashboards, including Resource Usage and Usage, as well as the Overview page in the UI correctly report heterogeneous GPU counts and workload usage.

Workaround: Use the Resource Usage, Usage dashboards or the Quota Overview page to view accurate workload usage and heterogeneous GPU count information.

DSE-59835: Heterogeneous GPU custom quota is not reset correctly after reverting to default quota

When resetting a custom heterogeneous GPU quota to the default quota for a user, the system could fail to persist the default quota configuration. Upon launching a workload, the quota reverts to the previously configured custom quota values instead of applying the default quota settings.

DSE-59829: Quota usage for a user is not displayed for heterogeneous GPU

When workloads are actively running on heterogeneous GPU setups, the system could fail to display user quota usage correctly. Quota metrics for CPU, memory, heterogeneous GPU resources continuously report zero values (0) despite active resource consumption.



Note: Quota usage is displayed correctly for workloads using only CPU and memory.

DSE-60259: Resource usage details on the Projects Overview do not display Heterogeneous GPU usage for workloads

When a workload is actively running and using heterogeneous GPU resources, the **Projects Overview** page fails to display the corresponding resource consumption details.

Cloudera AI on premises 1.5.5 SP2

Review the features, fixes, and known issues for Cloudera AI on premises 1.5.5 Service Pack 2 release.

What's new in 1.5.5 SP2

Cloudera AI on premises 1.5.5 SP2 delivers a set of new features for Cloudera AI.

Cloudera Generative AI readiness

- The Cloudera AI Inference service is now Generally Available. Upgrading your Cloudera AI Inference service, starting with this release, enables access to new features, enhanced performance, improved security, and updated component versions. For more information, see [Upgrading Cloudera AI Inference service](#).
- To learn more about which LLM architectures are supported for Cloudera AI on premises 1.5.5 SP2, see: [Supported transformer model architectures for vLLM 0.8.5 in Cloudera AI Inference service 1.5.5 SP2](#)
- Registering extra large LLMs is now more reliable with the enhancements that support importing the latest and largest models.
 - The models download process for registering models from third party sources was modified to prevent errors related to massive files. For more information, see [Downloading and uploading Model Repositories for an air-gapped environment](#).
 - The Cloudera AI Registry service can now be set to have only one replica to avoid synchronization and storage constraints related to massive model files. For more information, see [Configuring model import script in air-gapped environment](#).
 - For more information also see [Configuring Traefik timeout values for large file uploads and downloads](#).
- For Cloudera AI on premises 1.5.5 SP2 and higher releases, the Model Hub feature is also available on OpenShift Container Platform.

Usability and performance enhancements

- Reduce waiting times with improved UI performance of Cloudera AI, offering faster page loads for key landing pages, session launches, and listing pages.
- To ensure seamless integration between your Cloudera AI environment and the broader Cloudera, the Hadoop and Spark runtimes are now aligned with Cloudera Data Engineering and Data Hub builds. This alignment allows you to move code between Cloudera AI and other services without concerns about version mismatches. For more information, see [Apache Spark supported versions](#).

- Traffic management in Embedded Container Service-based environments is now more robust, featuring enhanced encryption and deeper observability. Additionally to NGNIX, Istio Gateway API is also available. For more details, see [Gateway API support for Embedded Container Service](#).
- Collecting logs provide comprehensive insights into both past and present system behavior, streamlining troubleshooting processes. For more information, see [Diagnostic bundle support for Cloudera AI Registry](#) and [Diagnostic bundle support for Cloudera AI Inference service](#).
- Several underlying infrastructure enhancements are now implemented to improve performance including disk space optimization, longer retention of critical logs and ring-fencing ability to reserve key infrastructure nodes for critical system functions.

Governance and resource control enhancements

- Workbench Quota Management is in Technical Preview, allowing Administrators to define maximum limits for memory and compute usage per workbench. This feature helps prevent cost overruns and minimizes operational risks. For more information, see [Quota Management overview](#).
- Team quotas can now be applied in mixed-GPU environments, enabling Administrators to precisely allocate access to specific hardware types for each team. For more information, see [Quota for Cloudera AI workloads](#)

Security improvements

- The non-transparent proxy support on Cloudera AI Inference service ensures strict network control by enforcing all outbound communication to pass through the proxy. This approach ensures adherence to internal security policies and mandates for air-gapped environments. For more information see [Non-transparent proxy support on Cloudera AI Inference service](#).
- Routine security improvements are now implemented, addressing Common Vulnerabilities and Exposures (CVE).

Compatibility for Cloudera AI 1.5.5 SP2, Cloudera Runtime components and Data Visualization

Explore Cloudera AI 1.5.5 SP2 and its compatibility with Cloudera Runtime components and with Cloudera Data Visualization across various versions.



Important:

- If you switch from the Red Hat Universal Base Image (Redhat UBI image) to the security hardened image, note that the updated component versions and libraries can impact the Cloudera AI operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

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

For more information, see [Security hardened Spark image migration guide](#).

- From Cloudera Data Services on premises 1.5.5 onwards, Spark 3.4 is no longer available.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported on Spark 3.3.2 and higher.

Table 13: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for Security Hardened image

Cloudera Runtime Version	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.9 SP1	• Spark 3.3.2 • Scala 2.12.15 • Python 3.10.19 • Java 17.0.17	• Spark 3.5.4 • Scala 2.12.18 • Python 3.11.14 • Java 17.0.17	• Airflow 2.11.0 • Python 3.11.14 • Java 17.0.17	Iceberg 1.3.0	1.32	8.0.8-b39

Cloudera Runtime Version	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.3.1	NA	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.14 - Java 17.0.17	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.17	Iceberg 1.5.2	1.32	8.0.8-b39

Table 14: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for Redhat insecure image

Cloudera Runtime Version	Spark 2.4.x	Spark 3.3.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.9 SP1	- Spark 2.4.8 - Scala 2.11.12 - Python 2.7.18 - Python 3.6.8 - Java 1.8.0_472	- Spark 3.3.2 - Scala 2.12.15 - Python 2.7.18 - Python 3.8.17 - Java 11.0.25	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.17	Iceberg 1.3.0	1.32	8.0.8-b39

Supported transformer model architectures for vLLM 0.8.5 in Cloudera AI Inference service 1.5.5 SP2

vLLM 0.8.5 supports the listed transformer model architectures for Cloudera AI Inference service in 1.5.5 SP2 and higher releases.

- aquilaforcausalmlm
- aquilamodel
- arcticforcausalmlm
- ariaforconditionalgeneration
- ayavisionforconditionalgeneration
- baichuanforcausalmlm
- baichuanforcausalmlm
- bambaforcausalmlm
- bartforconditionalgeneration
- bartmodel
- bertforsequenceclassification
- bertmodel
- blip2forconditionalgeneration
- bloomforcausalmlm
- chameleonforconditionalgeneration
- chatglmforconditionalgeneration
- chatglmmodel
- cohere2forcausalmlm
- cohereforcausalmlm
- dbrxforcausalmlm
- decilmforcausalmlm
- deepseekforcausalmlm
- deepseekmtpmodel
- deepseekv2forcausalmlm
- deepseekv2forcausalmlm
- deepseekv3forcausalmlm

- eaglellamaforcausalml
- eaglemodel
- exaoneforcausalml
- fairseq2llamaforcausalml
- falconforcausalml
- falconmambaforcausalml
- florence2forconditionalgeneration
- fuyuforcausalml
- gemma2forcausalml
- gemma2model
- gemma3forcausalml
- gemma3forconditionalgeneration
- gemmaforcausalml
- glm4forcausalml
- glm4vforcausalml
- glmforcausalml
- gpt2lmheadmodel
- gptbigcodeforcausalml
- gptjforcausalml
- gptneoxforcausalml
- graniteforcausalml
- granitemoeforcausalml
- granitemoessharedforcausalml
- gritlm
- grok1modelforcausalml
- h2ovlchatmodel
- idefics3forconditionalgeneration
- internlm2forcausalml
- internlm2forrewardmodel
- internlm2veforcausalml
- internlm3forcausalml
- internlmforcausalml
- internvlchatmodel
- jaislmheadmodel
- jambaforcausalml
- jambaforsequenceclassification
- llama4forconditionalgeneration
- llamaforcausalml
- llamaforcausalml
- llamamodel
- llavaforconditionalgeneration
- llavanextforconditionalgeneration
- llavanextvideoforconditionalgeneration
- lavaonevisionforconditionalgeneration
- mamba2forcausalml
- mambaforcausalml
- mantisforconditionalgeneration
- medusamodel
- minicpm3forcausalml
- minicpmforcausalml

- minicpmo
- minicpmv
- minimaxtext01forcausalml
- mistral3forconditionalgeneration
- mistralforcausalml
- mistralmodel
- mixtralforcausalml
- mlpspeculatorpretrainedmodel
- mllamaforconditionalgeneration
- molmoforcausalml
- mptforcausalml
- mptforcausalml
- nemotronforcausalml
- nvlm_d
- olmoforcausalml
- olmo2forcausalml
- olmoeforcausalml
- optforcausalml
- orionforcausalml
- paligemmaforconditionalgeneration
- persimmonforcausalml
- phi3forcausalml
- phi3smallforcausalml
- phi3vforcausalml
- phi4mmforcausalml
- phiforcausalml
- phimoeforcausalml
- pixtralforconditionalgeneration
- prithvigeospatialmae
- qwen2_5_vlforconditionalgeneration
- qwen2audioforconditionalgeneration
- qwen2forcausalml
- qwen2forprocessrewardmodel
- qwen2forrewardmodel
- qwen2forsequenceclassification
- qwen2model
- qwen2vlforconditionalgeneration
- qwen3forcausalml
- qwen3moeforcausalml
- qwenvlforconditionalgeneration
- quantmixtralforcausalml
- qwenlmheadmodel
- robertaformaskedlm
- robertaforsequenceclassification
- robertamodel
- rwforcausalml
- skyworkr1vchatmodel
- smolvlmforconditionalgeneration
- solarforcausalml
- stablelmepochforcausalml

- stablelmforcausal_lm
- starcoder2forcausal_lm
- telechat2forcausal_lm
- teleflmforcausal_lm
- transformersforcausal_lm
- ultravoxmodel
- whisperforconditionalgeneration
- xlmrobertaforsequenceclassification
- xlmrobertamodel
- xverseforcausal_lm
- zamba2forcausal_lm

Supported vLLM command line arguments

Cloudera AI on premises 1.5.5 SP2 uses vLLM 0.8.5. The command line arguments available in vLLM 0.8.5 are detailed in [vLLM arguments](#).

The following command line arguments are supported with vLLM 0.8.5:

- --block-size
- --calculate-kv-scales
- --cpu-offload-gb
- --disable-cascade-attn
- --disable-chunked-mm-input
- --disable-sliding-window
- --dtype
- --enable-auto-tool-choice
- --enable-chunked-prefill
- --enable-prefix-caching
- --enforce-eager
- --gpu-memory-utilization
- --kv-cache-dtype
- --load-format
- --logprobs-mode
- --long-prefill-token-threshold
- --max-logprobs
- --max-long-partial-prefills
- --max-model-len
- --max-num-batched-tokens
- --max-num-partial-prefills
- --max-num-seqs
- --max-seq-len-to-capture
- --multi-step-stream-outputs
- --no-enable-prefix-caching
- --num-lookahead-slots
- --num-scheduler-steps
- --pipeline-parallel-size, -pp
- --preemption-mode
- --prefix-caching-hash-algo
- --quantization
- --rope-scaling
- --rope-theta

- --scheduling-policy
- --seed
- --tensor-parallel-size, -tp
- --tool-call-parser
- --trust-remote-code

You can find details on the above listed command line arguments here: [vLLM arguments](#).



Note: Only the above listed vLLM command line arguments are supported.

Fixed issues in 1.5.5 SP2

Cloudera AI on premises 1.5.5 SP2 addresses issues previously identified as known issues.

Model serving

DSE-44238: Cannot create Cloudera AI Inference service application deployment using CDP CLI when Ozone credentials are passed

The Cloudera AI Inference service can now be created using CDP CLI as well. This issue is now resolved.

DSE-44141: Failed to delete deployment in executing DeleteML-servingApp

Cloudera AI Inference service did not remove all namespaces if the Cloudera AI Inference service was deleted after an installation failure. This issue is now resolved.

Model training

DSE-41635: Missing cleanup on deletion of registry

The namespace deletion logic is now integrated into the deletion workflows. A new kubeClient client was introduced in the ModelRegistryProvisioner component to enable namespace removal following a Helm release cleanup. The updated code now performs a thorough cleanup process, including the removal of Helm releases, persistent volumes, database records, and Kubernetes namespaces.

DSE-46352: Kserve fails to pull images in air-gapped environments where the Docker registry is not listed in the trusted list

Model endpoints no longer fail in air-gapped environments. This issue is now resolved.

User management

DSE-48606 Team creation and synchronization in Cloudera AI on premises adds default user permission

Cloudera AI on premises did not store the selected user permission, which are viewer, collaborator, or administrator, in the database, when creating and synchronizing a team. Instead, it assigned a default permission, granting write access to all team members. The issue is now fixed, the database is updated correctly and Cloudera AI assigns the proper role to the users.

Usability

DSE-46966: Increase sdx-config-template mount size from 100 MB

The size of the sdx-config-template emptyDir volume is increased from 100 MB to 500 MB in the Helm chart, reducing the frequency of mount size issues.

DSE-47738: Cloudera AI temporary backup pods gets stuck into Pending state

Cloudera AI temporary backup pods got stuck in the Pending state on specific OpenShift Container Platform versions with Umbra enabled. A Seccomp profile has been added for OpenShift Container Platform setups in which the restricted-v2 profiles prevent the backup pod from starting. This issue is now resolved.

Resource usage dashboards

DSE-46977: Dashboards and dashboard_pods table status synchronized

Resource usage discrepancies were identified due to outdated status synchronization between dashboards and the dashboard_pods table. In some cases, the dashboard_pods table displayed active statuses even when the status of the associated dashboards were marked as stopped, failed, or succeeded, resulting in inaccurate resource usage reports.

This issue has been resolved to ensure proper synchronization between dashboard statuses and the dashboard_pods table, providing accurate and consistent resource usage reporting.

Known issues for Cloudera AI on premises 1.5.5 SP2

You might run into some known issues while using Cloudera AI on premises 1.5.5 SP2.

Cloudera AI on premises 1.5.5 and 1.5.5 SP1 existing known issues are carried into Cloudera AI on premises 1.5.5 SP2.

For more details, see:

- [Known issues for Cloudera AI on premises 1.5.5](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP1](#)

For known issues fixed since Cloudera AI on premises 1.5.5, see:

- [Fixed issues in 1.5.5 SP1](#)
- [Fixed issues in 1.5.5 SP2](#)

Upgrade management

DSE-50441: Unable to register or deploy Model with the 1.5.4_h30 # 1.5.5 SP1 # 1.5.5 SP2 upgrade path and DSE-50487: Add workaround for upgrading AI registry from 1.5.4 - 1.5.5

During the Cloudera AI Registry upgrade process, a new Cloudera AI Registry instance is provisioned, and the database is assigned a fresh Persistent Volume (PV). The upgrade workflow is designed to replace this fresh PV with the old Cloudera AI Registry PV to retain historical data.

A race condition occurs between the application startup and storage orchestration. The database schema migration logic might execute on the fresh PV before the volume swap sequence is completed. As a result, when the legacy PV is eventually attached, it lacks the necessary schema updates, leading to query failures caused by schema mismatches.

Workaround

Restart the Cloudera AI Registry v1 pod. This action forces the application to reinitialize, detect the correctly attached legacy PV, and apply the pending schema migrations.

DSE-50712: On premises 1.5.5 SP2 Cloudera AI Registry on OpenShift Container Service: 1.5.4 SP2 to 1.5.5 SP2 upgrade fails with knox Init:ImagePullBackOff failure

An upgrade failure occurs when upgrading the on premises Cloudera AI Registry from 1.5.4 SP2 version to 1.5.5 SP2 version. The failure is caused by an ImagePullBackOff error during Knox initialization.

Workaround

Upgrading directly from 1.5.4 SP2 to 1.5.5 SP2 is not the supported upgrade path. The recommended procedure is to upgrade first to 1.5.5 GA or 1.5.5 SP1 before proceeding to 1.5.5 SP2.

DSE-56419: Persistent Volume already exists

During the retry of a Cloudera AI Workbench upgrade to 1.5.5 SP2 or 1.5.5 SP2 CHF1, or during a fresh upgrade from 1.5.5 SP2 to 1.5.5 SP2 CHF1, the process could fail with an error indicating that the preupgrade workspace PersistentVolume (PV) already exists: persistentvolumes pre-upgrade-workspace-pv-[***WORKBENCH NAME***] already exists.

This can occur when a previous failed upgrade attempt left behind a temporary, undeleted PV, causing subsequent runs to fail immediately.

Workaround

Use the following command to delete the PV:

```
kubectrl delete pv pre-upgrade-workspace-pv-[***WORKBENCH NAME***]
```

Log management

DSE-49031: Add fluent bit sidecar for model endpoints in serving-default namespace and DSE-49032: Add fluent bit sidecar for knative pods

The diagnostic bundle includes live cluster logs from all namespaces, however the archived logs are limited to the cml-serving infra namespace.

Workaround: None.

Quota management

DSE-50530: Error observed when configuring custom user or team quota

When setting custom CPU or memory quotas for a user or team without specifying a GPU quota, the system displays the Request must contain at least one accelerator user quota error message.

Workaround

To resolve this issue, always include a GPU quota value when updating CPU or memory quotas for a user or team.

DSE-49793: Different default quota in heterogeneous GPU for users and teams

In configurations with heterogeneous GPUs, the default GPU quota is common for both users and teams. This limitation requires users to configure a single default GPU quota that applies to both users and teams.

Cloudera recommends setting a single default GPU quota that is suitable for both users and teams.

Workbench management

DSE-49514: Review of the user resources and workbench resources does not display the right data

GPU resources are discovered by the workbench even if no GPUs were added during the creation or the provisioning of the workbench. The workbench automatically discovers all GPUs available in the cluster regardless of whether they were allocated to the workbench. As a result, the User Resources section inaccurately displays the total number of GPUs in the cluster instead of reflecting the GPUs assigned to the workbench.

Workaround: None.

Model training

DSE-48824: Cloudera AI Registry does not work with Google Chrome browser version 142 or higher

Cloudera AI Registry does not function properly when accessed using Google Chrome version 142 or higher. The **Model Endpoints** page fails to load and displays the following error message: Error occurred while communicating with Cloudera AI Registry in environment '<ENVIRONMENT_NAME>'.

Workaround

To resolve this issue, either downgrade Google Chrome to a version lower than 142 or use an alternative browser.

Model import

DSE-60265: NVIDIA GPU Cloud model import failures in non-transparent proxy environments

In Cloudera AI 1.5.5 SP2 and higher releases, importing models from NVIDIA GPU Cloud (NGC) fails when a non-transparent proxy is enabled in the Cloudera AI Registry service. The import request fails immediately with an HTTP 404 Not Found error.

Workaround:

Configure a traffic exclusion patch in the Cloudera AI Registry deployment.

1. Run the following command to add the IP CIDR (Classless Inter-Domain Routing) exclusion annotation:

```
kubectl patch deployment model-registry-v2 -n [***CLOUDERA AI
  REGISTRY NAMESPACE***] --type=json -p '[{"op":"add","path":"/
spec/template/metadata/annotations/traffic.sidecar.istio.io~1
excludeOutboundIPRanges","value":["***PROXY IP***]/32"}]'
```

Site Administration

DSE-36561: Updating Cloudera AI applications fails if the Allow users to use ML runtime addons option is disabled

The Allow users to use ML runtime addons option is enabled by default in Cloudera AI on premises. However, if this option is disabled for any configuration activity, updating a Cloudera AI application can fail with the Whoops, there was an unexpected error HTTP 400 error message.

Workaround:

To address this issue, navigate to the [Site Administration Settings](#) page and enable the Allow users to use ML runtime addons option.

Model serving

DSE-50375: Remove provisioning to update S3 bucket field in update storage configuration for Cloudera AI Inference service instance

The Cloudera AI Inference service does not use the S3 bucket option, consequently the secret does not contain S3 fields either.

Workaround: None.

DSE-48823: Cloudera AI Inference does not work with Google Chrome browser version 142 or higher

Cloudera AI Inference service does not function properly when accessed using Google Chrome version 142 or higher. The **Model Endpoints** page fails to load and displays the following error message: Error occurred while communicating with Cloudera AI Inference service in environment '<ENVIRONMENT_NAME>'.

Workaround

To resolve this issue, either downgrade Google Chrome to a version lower than 142 or use an alternative browser.

Cloudera AI Inference service Known issues

- Updating the description after a model has been added to a model endpoint will lead to a UI mismatch in the model builder for models listed by the model builder and the deployed models.
- Embedding models function in two modes: query or passage. This has to be specified when interacting with the models in one of the following ways:
 - Suffixing the model ID in the payload by either -query or -passage
 - Specifying the input_type parameter in the request payload.

For more information, see [NVIDIA documentation](#).

- Embedding models only accept strings as input. Token stream input is currently not supported.
- Llama 3.2 Vision models are not supported on AWS on A10G and L40S GPUs.

Livelog**DSE-56531: livelog-go fails during reconnect**

The livelog#publisher might crash with a concurrent write to websocket error when its WebSocket connection drops, resulting in temporary loss of container logs.

Cloudera AI on premises 1.5.5 SP1

Review the features, fixes, and known issues for Cloudera AI on premises 1.5.5 Service Pack 1 release.

What's new in 1.5.5 SP1

Cloudera AI on premises 1.5.5 SP1 delivers a set of new features for Cloudera AI.

Job Retry feature is available on Cloudera AI on premises

The Job Retry feature enables automatic retries of jobs based on their terminal execution states, that is, failed, timed out or skipped. It also supports concurrent execution of job retry runs, ensuring that the scheduled job runs remain unaffected and are not blocked by retry processes. Users have the flexibility to configure various options to define the retry behavior. These retries are fully automated, eliminating the need for manual intervention.

For details on the Job Retry feature, see the instructions in [Configuring Job Retry settings for the Administrator](#) and references in [Job Retry parameters](#).

Apache Spark defaults and custom configuration details available

Custom Apache Spark settings can now be configured at workbench level. When set, the custom Spark configuration provided by the Administrator will be merged with the default Spark configuration used in Cloudera AI sessions. These settings will automatically apply to all newly launched Spark sessions within the workbench. The configuration option is available under Site Administration Runtimes . If Spark pushdown is enabled, Spark configuration details can be configured at workbench-level and are visible on Base cluster-level. For more details, see [Setting custom Spark configurations at workbench-level](#) and [Modifying Project settings](#).

Support added for NVIDIA Inference Microservice (NIM) CLI

Support for NIM CLI in the Cloudera AI Registry is provided to import the latest offerings from NVIDIA.

Cloudera AI Inference service

The performance of the Cloudera AI Inference service has been significantly enhanced by implementing token caching, which improves UI responsiveness and reduces network load.

Cloudera AI Registry

Support for the following models are now added for Cloudera AI on premises 1.5.5 SP1:

- llama-3.3-nemotron-super-49b-v1
- llama-3.1-nemotron-nano-v1
- starcoder-2
- deepseek-r1-distill-llama
- llama-3.2-instruct
- nemoretriever-parse
- nemoretriever-graphic-elements-v1
- nemoretriever-page-elements-v2
- nemoretriever-table-structure-v1
- paddleocr
- llama-3.2-nv-embedqa-v2
- llama-3.2-nv-rerankqa-v2

- riva-asr-whisper-large-v3
- boltz2
- gpt-oss

Bucket name validation during creating Cloudera AI Registry

When creating a Cloudera AI Registry, the system now performs a validation to ensure the S3 bucket name meets the required standards.

As part of this validation, Cloudera AI Registry creation will be blocked if the bucket name contains a backslash (/), as this character is not permitted in S3 bucket names.

Increased default resource requests for infrastructure pods

Resource requests for several core Cloudera AI services are now increased, adding 1350 millicores of CPU and 1094 MiB of memory. This enhancement is aimed at improving performance and stability, providing a smoother and more reliable experience without requiring any user intervention.

Configurable Livelog retention period in Cloudera AI

Cloudera AI now provides enhanced flexibility with a configurable livelog retention period. This new feature addresses the challenges caused by the hardcoded retention period in Helm charts, which often lead to system crashes due to the insufficient default livelog storage size of 100 GB for many users. With this update, Administrators can define the livelog retention period in days, offering more granular control compared to the previous monthly configuration. The retention period can be easily customized in the Site Administration Settings, with a default value of 180 days, ensuring improved adaptability to diverse storage requirements. For details, see the instructions in [Historical workload cleanup settings](#)

Periodic workload records cleanup from the Cloudera AI Workbench PostgreSQL database

A new Cleanup DB Entries checkbox in the Site Administration Settings allows Administrators to enable periodic cleanup of database tables, including dashboards, dashboard_pods, model_deployments, and user_events, for entries older than the livelog retention period. The livelog cleaner triggers a gRPC call for DeleteDBWorkloads action, which performs the cleanup when the enable_clean_db_workloads setting is activated.

Additionally, the cleanup logic for the usageview table is now configurable directly through the UI, providing greater control and ease of management.



Note:

The automatic cleanup process does not include the Dashboards Archive tables. If you use the Dashboards Archive feature, you will need to manually clean up the dashboards_archive table.

For details on the Dashboard Archive feature, see [Optimized queries with Dashboards Archive table](#).

Improved stability of web deployments

The stability of web deployments is improved by optimizing Kubernetes liveness probes to prevent unnecessary pod restarts during high load scenarios.

New model build options introduced

`model_root_dir`: This option allows you to set a custom build root directory, enabling deployment even when a .git structure is nested.

`build_script_path`: This option provides the ability to specify a custom path for the build script.

UI displays skipped jobs with skipped status tag

Previously, when a job was already running and a new job run was triggered by a cron job or an API call, the new run would be skipped and displayed as **Failed** in the UI. This update introduces a **Skipped** status, and any skipped job runs now appear with the **Skipped** status in the UI.

Compatibility for Cloudera AI 1.5.5 SP1, Cloudera Runtime components and Data Visualization

Explore Cloudera AI 1.5.5 SP1 and its compatibility with Cloudera Runtime components and with Cloudera Data Visualization across various versions.



Important:

- If you switch from the Red Hat Universal Base Image (Redhat UBI image) to the security hardened image, note that the updated component versions and libraries can impact the Cloudera AI operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

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

For more information, see [Security hardened Spark image migration guide](#).

- From Cloudera Data Services on premises 1.5.5 onwards, Spark 3.4 is no longer available.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported on Spark 3.3.2 and higher.

Table 15: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for Security Hardened image

Cloudera Runtime Version	Spark 3.2.x	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.7 SP 3	- Spark 3.2.3 - Scala 2.12.15 - Python 3.9.23 - Java 11.0.27	NA	NA	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.16	Iceberg 0.14.1	1.31	8.0.6-b93
7.1.9 SP1	- Spark 3.2.3 - Scala 2.12.15 - Python 3.9.23 - Java 11.0.27	- Spark 3.3.2 - Scala 2.12.15 - Python 3.10.18 - Java 17.0.16	NA	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.16	Iceberg 1.3.0	1.31	8.0.6-b93
7.3.1	NA	NA	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.13 - Java 17.0.16	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.16	Iceberg 1.5.2	1.31	8.0.6-b93

Table 16: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for Redhat insecure image

Cloudera Runtime Version	Spark 2.4.x	Spark 3.2.x	Spark 3.3.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.7 SP3	- Spark 2.4.7 - Scala 2.11.12 - Python 2.7.18 - Python 3.6.8 - Java 1.8.0_452	- Spark 3.2.3 - Scala 2.12.15 - Python 2.7.18 - Python 3.6.8 - Java 11.0.25	NA	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.16	Iceberg 0.14.1	1.31	8.0.6-b93
7.1.9 SP1	- Spark 2.4.7 - Spark 2.4.8 - Scala 2.11.12 - Python 2.7.18 - Python 3.6.8 - Java 1.8.0_452	- Spark 3.2.3 - Scala 2.12.15 - Python 2.7.18 - Python 3.6.8 - Java 11.0.25	- Spark 3.3.2 - Scala 2.12.15 - Python 2.7.18 - Python 3.8.17 - Java 11.0.25	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.16	Iceberg 1.3.0	1.31	8.0.6-b93

Supported transformer model architectures for vLLM 0.8.4 in Cloudera AI Inference service 1.5.5 SP1

vLLM 0.8.4 supports the listed transformer model architectures for Cloudera AI Inference service in 1.5.5 SP1.

- aquilaforcausalml
- aquilamodel
- arcticforcausalml
- ariaforconditionalgeneration
- ayavisionforconditionalgeneration
- baichuanforcausalml
- bambaforcausalml
- bartforconditionalgeneration
- bartmodel
- bertforsequenceclassification
- bertmodel
- blip2forconditionalgeneration
- bloomforcausalml
- chameleonforconditionalgeneration
- chatglmforconditionalgeneration
- chatglmmodel
- cohere2forcausalml
- cohereforcausalml
- dbrxforcausalml
- decilmforcausalml
- deepseekforcausalml
- deepseekmtpmodel
- deepseekv2forcausalml
- deepseekv3forcausalml

- deepseekvlv2forcausalmlm
- eaglellamaforcausalmlm
- eaglemodel
- exaoneforcausalmlm
- falconforcausalmlm
- falconmambaforcausalmlm
- fairseq2llamaforcausalmlm
- florence2forconditionalgeneration
- fuyuforcausalmlm
- gemma2forcausalmlm
- gemma2model
- gemma3forcausalmlm
- gemma3forconditionalgeneration
- gemmaforcausalmlm
- glm4forcausalmlm
- glm4vforcausalmlm
- glmforcausalmlm
- gpt2lmheadmodel
- gptbigcodeforcausalmlm
- gptjforcausalmlm
- gptneoxforcausalmlm
- graniteforcausalmlm
- granitemoeforcausalmlm
- granitemoessharedforcausalmlm
- gritlm
- grok1modelforcausalmlm
- h2ovlchatmodel
- idefics3forconditionalgeneration
- internlm2forcausalmlm
- internlm2forrewardmodel
- internlm2veforcausalmlm
- internlm3forcausalmlm
- internlmforcausalmlm
- internvlchatmodel
- jaislmheadmodel
- jambaforcausalmlm
- jambaforsequenceclassification
- llama4forconditionalgeneration
- llamaforcausalmlm
- llamamodel
- llavaforconditionalgeneration
- llavanextforconditionalgeneration
- llavanextvideoforconditionalgeneration
- lavaonevisionforconditionalgeneration
- mambaforcausalmlm
- mamba2forcausalmlm
- mantisforconditionalgeneration
- medusamodel
- minicpm3forcausalmlm
- minicpmforcausalmlm

- minicpmo
- minicpmv
- minimaxtext01forcausalml
- mistral3forconditionalgeneration
- mistralforcausalml
- mistralmodel
- mixtralforcausalml
- mlpspeculatorpretrainedmodel
- mllamaforconditionalgeneration
- molmoforcausalml
- mptforcausalml
- nemotronforcausalml
- nvlm_d
- olmoforcausalml
- olmo2forcausalml
- olmoeforcausalml
- optforcausalml
- orionforcausalml
- paligemmaforconditionalgeneration
- persimmonforcausalml
- phi3forcausalml
- phi3smallforcausalml
- phi3vforcausalml
- phi4mmforcausalml
- phiforcausalml
- phimoeforcausalml
- pixtralforconditionalgeneration
- prithvigeospatialmae
- qwen2_5_vlforconditionalgeneration
- qwen2audioforconditionalgeneration
- qwen2forcausalml
- qwen2forprocessrewardmodel
- qwen2forrewardmodel
- qwen2forsequenceclassification
- qwen2model
- qwen2moeforcausalml
- qwen2vlforconditionalgeneration
- qwen3forcausalml
- qwen3moeforcausalml
- qwenvlforconditionalgeneration
- qwenlmheadmodel
- quantmixtralforcausalml
- robertaformaskedlm
- robertaforsequenceclassification
- robertamodel
- rwforcausalml
- skyworkr1vchatmodel
- smolvlmforconditionalgeneration
- solarforcausalml
- stablelmepochforcausalml

- stablelmforcausal_lm
- starcoder2forcausal_lm
- telechat2forcausal_lm
- teleflmforcausal_lm
- transformersforcausal_lm
- ultravoxmodel
- whisperforconditionalgeneration
- xlmrobertaforsequenceclassification
- xlmrobertamodel
- xverseforcausal_lm
- zamba2forcausal_lm

Supported vLLM command line arguments

Cloudera AI on premises 1.5.5 SP1 uses vLLM 0.8.4. The command line arguments available in vLLM 0.8.4 are detailed in [vLLM arguments](#).

The following command line arguments are supported with vLLM 0.8.4:

- --block-size
- --calculate-kv-scales
- --cpu-offload-gb
- --disable-cascade-attn
- --disable-chunked-mm-input
- --disable-sliding-window
- --dtype
- --enable-auto-tool-choice
- --enable-chunked-prefill
- --enable-prefix-caching
- --enforce-eager
- --gpu-memory-utilization
- --kv-cache-dtype
- --load-format
- --logprobs-mode
- --long-prefill-token-threshold
- --max-logprobs
- --max-long-partial-prefills
- --max-model-len
- --max-num-batched-tokens
- --max-num-partial-prefills
- --max-num-seqs
- --max-seq-len-to-capture
- --multi-step-stream-outputs
- --no-enable-prefix-caching
- --num-lookahead-slots
- --num-scheduler-steps
- --pipeline-parallel-size, -pp
- --prefix-caching-hash-algo
- --preemption-mode
- --quantization
- --rope-scaling
- --rope-theta

- --scheduling-policy
- --seed
- --tensor-parallel-size, -tp
- --tool-call-parser
- --trust-remote-code

You can find details on the above listed command line arguments here: [vLLM arguments](#).



Note: Only the above listed vLLM command line arguments are supported.

Fixed issues in 1.5.5 SP1

Cloudera AI on premises 1.5.5 SP1 addresses issues previously identified as known issues.

DOCS-27138: Possible occurrence for incorrect status for successful workloads

A potential race condition in the reconciler service could result in the status of successful workloads having been incorrectly updated to unknown or failed status.

The issue occurred due to high system load, which led to incorrect status reporting after pod deletion. This issue is now resolved.

DSE-46395: Validation of roles for CAIR against UMS not working

This fix addressed an authorization bug for AI Registry APIs in the control plane. This issue is now resolved.

DSE-45793: Disable zero initial scale for autoscaler configuration

Previously, due to a known KServe issue (kservice/kservice#4471), all newly created model endpoints would initially deploy with a single replica, regardless of the specified configuration. This issue is now resolved.

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

Previously, when rotating or updating the TLS certificate used by Cloudera AI, the Cloudera AI did not automatically retrieve the new certificate from the Cloudera Control Pane. This issue is now resolved. You can upload CA certificates directly through the Cloudera AI workbench UI and refresh certificates to apply and propagate trust across all relevant Cloudera AI services seamlessly.

DSE-44238: Cannot create Cloudera AI Inference service application deployment using CDP CLI when ozone credentials are passed

Cloudera AI Inference service could not be created using CDP CLI. This issue is now resolved.

DSE-44141: Failed to delete deployment in executing DeleteML-servingApp

Cloudera AI Inference service failed to remove all namespaces if the Cloudera AI Inference service was deleted after an installation failure. This issue is now resolved.

DSE-46352: Kserve fails to pull images in air-gapped environments where the docker registry is not listed in the trusted list

In air-gapped environments, model endpoints failed with an error message due to the Docker registry not being included in the trusted list. This issue is now resolved.

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

The Kubernetes client failure error during the StartJobRun API was not properly reported, resulting in misleading success messages. The fix ensures that engine start failures are now accurately reported with detailed error information, eliminating misleading success notifications.

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

Web requests to other services were generating malformed UUIDs in logs. The issue is now resolved by increasing the contextId length to 36 characters, ensuring proper UUID formatting.

DSE-44088: Operator pod start failure log missing ID

The operator pod start failure logs were missing critical identifiers, such as engineId and request UUID, which made error tracking challenging. The issue is now resolved by ensuring these identifiers are included in the operator logs.

DSE-44936: Pause job removes the assigned service-account

Pausing a Cloudera AI job caused the assigned service account to be removed. The issue arose when a job was paused and the page refreshed, unintentionally clearing the service account. The resolution involved updating the job update logic to modify the run_as field only when explicitly specified, ensuring that the service account remains intact.

DSE-44563: Heterogeneous GPU support not available in APIv2 for models deployment

The lack of heterogeneous GPU support in APIv2 for model deployments is now resolved. Updates to the frontend and API code now include the accelerator label ID, enabling GPU support for model deployments.

DSE-44564: Model Deployment fails to use selected GPU during deployment

Model deployments were failing to utilize the selected GPU during deployment due to the accelerator_label_id property not being properly persisted in the database. The issue is now resolved, ensuring that the correct GPU is utilized during model deployment.

DSE-41700: Team owner must be able to change team members privileges or role

The requirement for each team to have at least one Administrator is now removed for both local and synchronized teams. Team Administrators, when present, can now manage membership and roles through Team Settings. If no Team Administrator is assigned, Site Administrators can manage membership and roles instead.

DSE-25966: Memory leak detected in model proxy

A memory leak issue in the model proxy component was causing unplanned outages due to Kubernetes OOM (Out of Memory) terminations. The issue is now resolved by replacing the custom cache with a library already utilized by APIv2.

DSE-13708: Main page -> Active workload counter problems (Session, Experiment, Job, Application)

Previously, active workload counters were incorrectly categorizing workloads, counting experiments as sessions, jobs as applications, and entirely omitting applications. Additionally, the counters displayed workspace-wide active workloads for jobs, experiments, and applications instead of filtering them by the logged-in user.

This issue is now fixed, by correcting the counting logic and ensuring that the counters now display workloads specific to the logged-in user.

DSE-45166: New model build options have been introduced

- `model_root_dir`: This option allows users to set a custom build root directory, enabling deployment even when a .git structure is nested.
- `build_script_path`: This option provides the ability to specify a custom path for the build script.

DSE-40393: Horizontal window or browser resize is not working

Previously, the horizontal window resize functionality was not working as expected. The issue is now resolved by updating margin handling and width calculations, ensuring that the layout restores correctly when the browser window is resized.

DSE-40619: Allow emails to be updated through the UpdateJob API

The UpdateJob API previously did not allow email recipients to be modified. As a result, users were required to delete and recreate jobs to update email notifications. Additionally, the API did not support modifying email criteria, such as Success, Failure, Stopped, or Timeout notifications.

This issue is now resolved by implementing logic in the UpdateJob APIv2 to allow updates to all recipient types and attachments. The behavior now aligns with the CreateJob API, enabling seamless notification management without the need for job recreation.

DSE-41734: Livelog cleaner granularity needs to be increased

The livelog cleaner retention period was previously limited to being specified in months. Since the livelog cleaner Kubernetes job runs daily, increasing the granularity to days provides greater flexibility and control over the livelog cleanup policy.

This improvement updates the livelog cleaner retention period to be configurable in days, with a default value of 180 days, allowing for more precise management of log retention.

DSE-18866: Jupyter notebook editor initially redirects to OpenShift application not available page

Previously, the Jupyter notebook editor redirected to an OpenShift Application not available page due to the iframe loading before the OpenShift route was fully created, causing a 503 error.

The issue, caused by delays in OpenShift route creation after ingress and pod readiness, is now resolved by adding a readiness probe in API v1. This ensures the editor loads only when fully ready. Periodic readiness checks for up to 10 minutes address scaling challenges and prevent CORS issues and race conditions.

DSE-44009: Feature announcement request causes homepage loading timeout in air-gapped setup

A bug caused the homepage to hang indefinitely in air-gapped environments due to a failed feature announcement request. The issue occurred because the news feed API call could not complete, causing the homepage to remain in a loading state until the browser default timeout of 300 seconds was reached.

This issue is now fixed by ensuring that the loading spinner is limited to the feature announcement section. Additionally, feeds are now loaded from a local JSON file instead of external sources, resolving the timeout issue and ensuring proper homepage functionality in air-gapped setups.

DSE-44784: Web crashing when very large files selected for job email attachment

Previously, job email notifications with file attachments could crash the web service in the following scenarios:

- When the specified file was large enough, for example 10 GB, to exceed the 60-second timeout.
- When the Virtual File System (VFS) service became unresponsive during file streaming.
- When the VFS service experienced artificial delays.

The issue is now resolved to ensure the web service remains stable in these scenarios too.

DSE-46572: Project Owner field in Project Setting page is showing user ID instead of name

Previously, the Project Owner field on the Project Settings page was displaying user IDs instead of user names.

The issue was caused by the Project Owner field incorrectly referencing user IDs. The issue is now resolved by extracting the project owner name from the project object and implementing logic to fetch users by their names, ensuring the correct display of user names in the Project Owner field.

DSE-46860, DSE-46859, DSE-44806, DSE-43876 - Cloudera AI Workbench web and postgres database performance optimizations

Improved postgres database caching and indexing of frequently queried data. Improved Cloudera AI Web auto refresh intervals. Optimized the open websocket livelog connections used by web service.

DSE-46220, DSE-46219, DSE-46218 - Improved error handling in Cloudera AI Workbench web service

Better error handling for web services that could potentially result in service crashes.

DSE-44023 - Fixed issue with stopped applications getting started after Cloudera AI Workbench upgrades

Previously the reconciler service was restarting applications that were intentionally stopped by users, upon workspace upgrades. After the fix, stopped applications will not be restarted after workbench upgrade or when the reconciler service is restarted on its own.

DSE-43704 - Rename Custom Tee Binary to cml-tee

Previously, certain vulnerability scanners could incorrectly flag Cloudera AI as using a vulnerable version of the coreutils package, specifically the tee command.

Cloudera AI services included a custom tee binary, developed entirely in-house by Cloudera, which was not based on the open-source coreutils library. The custom tee command, version 0.9, was mistakenly identified by some scanners as the tee command from coreutils, which contains known vulnerabilities.

This issue is now resolved by renaming the Cloudera custom tee binary to cml-tee, ensuring that it is no longer misidentified.

DSE-44367 - Buildkitd pod CrashLoopBackOff due to port conflicts

During the creation or upgrade of Cloudera AI Workbench, buildkitd pods could occasionally enter a CrashLoopBackOff state. This issue occurred when the port used by BuildKit was not properly released during pod restarts or was occupied by another process. Users might have encountered errors, such as the following error:

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

This issue is now resolved to prevent port conflict and to ensure stable functioning of Cloudera AI Workbench.

DSE-44682 - Model deployment failing at build stage due to TLS issues

Previously, TLS-related issues could occasionally arise during the model build process in Cloudera AI Workbench when using Cloudera Embedded Container Service clusters. These issues occurred specifically when pulling images from the container registry and were caused by missing registry certificates on the worker nodes. The required certificates were expected to be located at the `/etc/docker/certs.d/` path.

This issue is now resolved to ensure successful model deployment and to eliminate TLS-related errors during the build stage.

DSE-40756: Resource Profile field displays empty on Job Settings page

After creating or upgrading a job and saving a **Resource Profile**, the **Resource Profile** field on the **Job Settings** pages might have appeared empty.

The **Resource Profile** field now correctly displays the previously selected value after loading the **Job Settings** page.

DSE-47904: Resource Profile is not visible for jobs and applications

After creating or upgrading an application and saving a **Resource Profile**, the **Resource Profile** field on the **Application Settings** pages might have appeared empty.

The **Resource Profile** field now correctly displays the previously selected value after loading the **S** page.

Known issues for Cloudera AI on premises 1.5.5 SP1

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

Cloudera AI on premises 1.5.5 existing known issues are carried into Cloudera AI on premises 1.5.5 SP1. For more details, see [Known issues for Cloudera AI on premises 1.5.5](#). For known issues fixed since Cloudera AI on premises 1.5.5, see [Fixed issues in 1.5.5 SP1](#).

DSE-47608: Update Hive on Base to use the correct SSL configuration value

SSL is always set to true when discovering the Hive on TEZ data connection, regardless of whether SSL is enabled.

Workaround:

As a workaround, the JDBC URL can be modified by editing the data connection settings.

DSE-47480: MLUsers are able to delete other users' deployed endpoints

Cloudera AI Inference service lacks fine-grained authorization, allowing endpoints created by one user to be deleted by another user.

DSE-47517: Parallel backup or restore during other ongoing Platform actions ends up workbench in failed state

Backup and restore actions lack validation checks for ongoing platform actions, leading to conflicts when triggered simultaneously with other platform actions.

No workaround exists for this issue. However, Cloudera recommends avoiding the initiation of any other workbench actions while a platform action is in progress on the same workbench.

DSE-46826: Registry upgrade is incorrectly returning 500 even though upgrade flow is being triggered

The registry upgrade process returns an HTTP 500 error in the UI and CLI, despite successfully completing in the background. The issue is linked to missing records and Docker secrets, likely due to recent changes in secrets handling.

Workaround:

Recreating the Cloudera AI Registry with valid Ozone credentials resolves the error and ensures that the upgrade completes successfully.

DSE-47916: Update storage configuration option overwrites the S3 Region parameter

The S3 Region parameter with an empty value overwrites the existing s3Region value when using the Update storage configuration option in the UI or the updateMIServingAppS3Creds API. This causes model deployment failures, as updating only the mandatory fields clears the s3Region value, leading to authorization errors.

Workaround:

Always include the S3 Region parameter when using either the updateMIServingAppS3Creds API or the Update storage configuration option in the UI.

DSE-47973: Update Storage Configuration disabled in Update Creds failed state

If the credentials update fails, the S3 storage configuration cannot be updated. Additionally, the UI does not display a warning to inform the user that the S3 storage configuration has been disabled.

DSE-48821: Dashboards referenced by applications must not be cleaned up

In Cloudera AI, after cleaning up outdated dashboards in the database, certain applications might disappear from the Application page of the Cloudera AI. Also, you might receive the following error message: Cannot read properties of null (reading 'creator').

The deletion of older records in the dashboards table was carried out to address web-to-database query performance issues caused by an excessive number of entries. However, if an application continues to reference an old dashboard and the application is not marked as deleted, the referenced dashboard ID in the applications table is automatically set to NULL value due to a trigger defined on the constraints.

This issue occurs only when the Automated Database Cleanup feature is enabled or when a Manual Cleanup script is used. For more information, see [Historical workload cleanup settings](#) and [Cleaning up and backing up the Cloudera AI Workbench database manually](#).

Workaround:

1. Identify the affected applications using the following query:

```
current_dashboard_id is set to null but not deleted apps
```

```
sense=# select a.current_dashboard_id,a.id,a.name, a.description,u.username,u.email from applications a, users u where a
.creator_id=u.id and a.current_dashboard_id is null and a.deleted_at is null;
```

current_dashboard_id	id	name	description	username	email
	75	Recreate Application	Recreate A	admin	admin@cdp.example

(1 row)

2. Update the dashboard reference (current_dashboard_id) to a fixed existing one.

After this update, the application logs might point to a different application. This issue will be resolved once you restart the application using [Step 4](#), as Cloudera AI will automatically assign the correct dashboard ID.

```
sense=# select id,application_id from dashboards where application_id is not null limit 1;
```

id	application_id
268uw5z5fui1iypl	69

- a. Update the current dashboard to the existing one of all affected applications using the following command:

```
update applications set current_dashboard_id='268uw5z5fui1iypl' where applications.current_dashboard_id is null and applications.deleted_at is null;
```

At this stage, you can access the Applications page, however, unexpected logs might appear as the application is temporarily linked to an unrelated dashboard.

3. Restart the affected applications using the details from [Step 1](#), including the application name and the email address of the owner.

Perform the following steps to restart the application:

- a. In the Cloudera console, click the Cloudera AI tile.

The Home page displays.

- b. Select the required Workbench.

The AI Workbench page displays.

- c. Open the Cloudera AI console.

- d. Click Projects in the left navigation pane and select the required project.

- e. Go to the project Overview page.

- f. Click Applications.

- g. Find the required application by its Name value.

- h. Click the  icon and select Application Details Settings Resource Profile CPU/Memory

- i. Update the application and restart it.

4. Update the dashboard cleanup procedure to include the following where condition:

```
// backup the dashboards table as usual
```

```
// and modify the delete where condition to have
delete from dashboards where finished_at < NOW() - INTERVAL
'1 month' and id not in (select distinct current_dashboard_id
from applications where applications.deleted_at is null) ;
```

This ensures that the referenced dashboards of all active and non-deleted, applications are excluded from the cleanup process.

DSE-48606 Team creation and synchronization in Cloudera AI on premises adds default user permission

When creating and synchronizing a team, Cloudera AI on premises does not store the selected user permission, viewer, collaborator, or administrator, in the database. Instead, it assigns a default permission, granting write access to all team members.

There is no workaround.

DSE#56081: Users cannot pass custom headers in application

A Cross-Origin Resource Sharing (CORS) regression might prevent applications from sending custom headers, such as refresh-token and access-token, during cross-origin inter-application communication (browser-based calls). The regression also causes Browser Accessible Service (BAS) to incorrectly set credentialed CORS headers for all subdomains, creating a potential security exposure.

As a result:

- Applications cannot pass custom headers without encountering CORS errors.
- Browser-based frontend to backend application communication might fail when relying on custom headers.

Cumulative Hotfixes: Cloudera AI on premises

Review the cumulative hotfixes for Cloudera on premises 1.5.5.

Cloudera AI on premises 1.5.5 SP3 CHF3

Review the features, fixes, and known issues in the Cloudera AI on premises 1.5.5 SP3 Cumulative hotfix 3 release.

Fixed issues in Cloudera AI on premises 1.5.5 SP3 CHF3

Cloudera AI on premises 1.5.5 SP3 CHF3 resolves livelog performance bottlenecks, while fixing persistent volume workbench upgrade failures.

Livelog fixed issues

DSE-57386: livelog: RocksDB memory resource constraints delay data access in livelog workloads

Previously, the in#memory settings for RocksDB were too small for the livelog workload, which led to delays in accessing data on Ceph platform and slowed down all workspace sessions under load.

This issue is now fixed. The block cache is increased to 256 MB, the Memtable is increased to 256 MB and the RocksDB statistics collection is improved. When enabled, RocksDB logs internal counters, the enhanced statistics provide better visibility into cache behavior and background operations.

DSE-57348: livelog: Single global lock remained held during slow disk operations causes sessions to pile up

Previously, various operations on livelog topics used a single global lock that remained held during slow disk operations, causing active sessions to pile up behind one delayed request. Under heavy load, this created significant delays for a large number of topic lookups.

This issue is now fixed. Only requests for the same topic wait on each other. All other lookups continue independently, eliminating performance bottlenecks and improving overall responsiveness.

DSE-57349: Livelog performance parameters and timeouts are not configurable

Previously, the livelog and livelog#publisher pods used hardcoded operational parameters, preventing users from adjusting performance. In some cases, these defaults caused storage slowdowns that affected all workspace sessions.

This issue is now resolved. Performance and timeout settings are now fully configurable through Kubernetes, and memory tuning is improved to provide smoother, more reliable performance.

DSE-56531: livelog-go fails during reconnect

Previously, the livelog#publisher could crash with a concurrent write to websocket error when its WebSocket connection dropped, resulting in temporary loss of container logs.

This issue is now fixed.

DSE-52116: Livelog cleanup is slow

Previously, the livelog service deleted topic entries one key at a time, which made topic cleanup slow.

This issue is now fixed. Iterative deletion is replaced with a single bulk delete operation, making topic cleanup significantly faster and more reliable. Related cleanup code was simplified and corrected, resulting in smoother maintenance behavior without changing the external API.

DSE-57347: Jupyter UI was while backend session is running

Previously, Jupyter session UIs could freeze permanently when livelog was slow or unresponsive because the system stopped sending status updates and never recovered on its own.

The issue is now fixed. A timeout and safe retry mechanism is added so delayed updates fail quickly and automatically recover. The timeout is set to 30 seconds by default and can be adjusted as needed.

Cloudera AI Workbench**DSE-60031: Team synchronization assigned incorrect team permissions**

In Cloudera AI Workbench on premises, the team synchronization process could incorrectly assign owner permissions to users and service accounts. When a workspace administrator was processed during synchronization, the **owner** permission could persist for subsequent team members in the same synchronization operation. As a result, some service accounts lost their intended administrator or operator role and became unusable.

This issue is now fixed. Team synchronization now assigns permissions independently for each member. Administrators receive the correct administrator permission, while non-administrator users and service accounts retain their configured team roles.

This fix applies only to teams created after upgrading to Cloudera AI 1.5.5 SP3 CHF3 and does not change member permissions for existing teams.

DSE-59376: Updating only the Runtime in Job Settings failed with validation errors

In Cloudera AI on premises 1.5.5 SP3, updating an existing Workbench Job to change only the Runtime could fail with CPU and memory validation errors. The update request incorrectly sent 0 value for CPU and memory, even when the resource profile was unchanged. In addition, the Job Settings page in the UI could display the wrong resource profile.

This issue was caused by a UI timing issue that reset the saved resource profile values before resource groups finished loading.

The issue is now fixed. Runtime updates now preserve the configured resource profile and correctly display the selected profile in Job Settings.

DSE-58654: GPU accelerator quota values could not be edited in the Workbench Quotas page of the UI

In a quota-enabled Workbench with heterogeneous GPU mode active, editing per-accelerator quota values in Site Administration Quotas # did not work correctly.

For example, when updating the quota value for an accelerator, such as NVIDIA-A100, the **Maximum Accelerator Count** did not reflect the entered value consistently. In addition, the **Total Accelerator** value was inconsistent because the individual accelerator quota values were not summed correctly. The **Total Accelerator** value must be the sum of the quota values for all individual accelerators.

This issue affected both the **Default Quota** and **Custom Quota** edit modals. **CPU** and **Memory** quota fields were not affected.

This issue is now fixed.

Cloudera AI Registry**DSE-54908: The AI Registry page appeared blank after upgrade**

After upgrading to Cloudera AI 1.5.5 SP3, the **AI Registry** page could appear blank and prevent users from viewing registered models.

This issue was caused by incorrect client-side filtering and limited pagination, which excluded some models from display, including restored models, public models, and models owned by system or service accounts.

This issue is now fixed. The **AI Registry** page now correctly displays public and restored models and uses improved pagination to ensure models are loaded properly.

Cloudera AI Inference service**DSE-58058: Model Endpoint Details page failed to render for Cloudera AI Inference service**

The Model Endpoint Details page failed to render for model endpoints hosted in older Cloudera AI Inference service versions and displayed an Unknown status for Cloudera AI 1.5.5 SP3. This issue has been resolved, and the UI now renders correctly regardless of your backend version.

This issue is now resolved, and the UI renders correctly regardless of your backend version.

Known issues for Cloudera AI on premises 1.5.5 SP3 CHF3

You might run into some known issues while using Cloudera AI on premises 1.5.5 SP3.

Cloudera AI on premises 1.5.5, 1.5.5 SP1 and 1.5.5 SP2 existing known issues are carried into Cloudera AI on premises 1.5.5 SP3.

For more details, see:

- [Known issues for Cloudera AI on premises 1.5.5](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP1](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP2](#)
- [Known issues for Cloudera AI on premises 1.5.5 SP3](#)

For known issues fixed since Cloudera AI on premises 1.5.5, see:

- [Fixed issues in 1.5.5](#)
- [Fixed issues in 1.5.5 SP1](#)
- [Fixed issues in 1.5.5 SP2](#)
- [Fixed issues in 1.5.5 SP3](#)

DSE-60234: Service Accounts are assigned incorrect roles when created using synchronized teams

In Cloudera AI on premises, when a synchronized group is used to create a team, the service accounts (SAs) associated with that team inherit the same permission level selected for the team,

unless the permission is MLAdmin. If the team permission is set to Contributor role, the service accounts are also created with the Contributor role instead of the required Administrator or Operator role.

Because service accounts require the Administrator or Operator role to run workloads, users cannot use these service accounts for Run As operations.

Workaround

Create the team with either the Administrator or Operator role instead of Contributor.

Cloudera AI on premises 1.5.5 SP3 CHF2

Cloudera AI on premises 1.5.5 Service Pack 3 Cumulative Hotfix 2 release contains no new features, bug fixes, or known issues.

Cloudera AI on premises 1.5.5 SP2 CHF1 HF1

Review the features, fixes, and known issues in the Cloudera AI on premises 1.5.5 SP2 Cumulative hotfix 1 hotfix 1 release.

Repository Locations for 1.5.5 SP2 CHF1 HF1

The URLs for Cloudera AI on premises 1.5.5 SP2 CHF1 HF1 are listed in the following table:

URL Type	Repository Location
Index	<code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2101/</code>
Manifest	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2101/manifest.json</code>
Parcels	Repository: <code>https://username:password@archive.cloudera.com/p/cdp-pvc-ds/1.5.5-h2101/parcels/</code>

Fixed issues in Cloudera AI on premises 1.5.5 SP2 CHF1 HF1

Cloudera AI on premises 1.5.5 SP2 CHF1 HF1 resolves livelog performance bottlenecks, while fixing persistent volume workbench upgrade failures.

Livelog fixed issues

DSE-57386: livelog: RocksDB memory resource constraints delay data access in livelog workloads

Previously, the `in#memory` settings for RocksDB were too small for the livelog workload, which led to delays in accessing data on Ceph platform and slowed down all workspace sessions under load.

This issue is now fixed. The block cache is increased to 256 MB, the Memtable is increased to 256 MB and the RocksDB statistics collection is improved. When enabled, RocksDB logs internal counters, the enhanced statistics provide better visibility into cache behavior and background operations.

DSE-57348: livelog: Single global lock remained held during slow disk operations causes sessions to pile up

Previously, various operations on livelog topics used a single global lock that remained held during slow disk operations, causing active sessions to pile up behind one delayed request. Under heavy load, this created significant delays for a large number of topic lookups.

This issue is now fixed. Only requests for the same topic wait on each other. All other lookups continue independently, eliminating performance bottlenecks and improving overall responsiveness.

DSE-57349: Livelog performance parameters and timeouts are not configurable

Previously, the livelog and livelog#publisher pods used hardcoded operational parameters, preventing users from adjusting performance. In some cases, these defaults caused storage slowdowns that affected all workspace sessions.

This issue is now resolved. Performance and timeout settings are now fully configurable through Kubernetes, and memory tuning is improved to provide smoother, more reliable performance.

DSE-56531: livelog-go fails during reconnect

Previously, the livelog#publisher could crash with a concurrent write to websocket error when its WebSocket connection dropped, resulting in temporary loss of container logs.

This issue is now fixed.

DSE-52116: Livelog cleanup is slow

Previously, the livelog service deleted topic entries one key at a time, which made topic cleanup slow.

This issue is now fixed. Iterative deletion is replaced with a single bulk delete operation, making topic cleanup significantly faster and more reliable. Related cleanup code was simplified and corrected, resulting in smoother maintenance behavior without changing the external API.

DSE-57347: Jupyter UI was while backend session is running

Previously, Jupyter session UIs could freeze permanently when livelog was slow or unresponsive because the system stopped sending status updates and never recovered on its own.

The issue is now fixed. A timeout and safe retry mechanism is added so delayed updates fail quickly and automatically recover. The timeout is set to 30 seconds by default and can be adjusted as needed.

Upgrade

DSE-56419: Persistent Volume already exists

Previously, during the retry of a Cloudera AI workbench upgrade to 1.5.5 SP2 or 1.5.5 SP2 CHF1, or during a fresh upgrade from 1.5.5 SP2 to 1.5.5 SP2 CHF1, the process could fail with an error indicating that the preupgrade workspace PersistentVolume (PV) already exists: persistentvolumes/pre-upgrade-workspace-pv-["***WORKBENCH NAME***"] already exists.

This occurred when a previous failed upgrade attempt left behind a temporary, undeleted PV, causing subsequent runs to fail immediately.

The issue is now resolved by removing the leftover PV before retrying the upgrade.

Cloudera AI on premises 1.5.5 SP2 CHF1

Review the features, fixes, and known issues in the Cloudera AI on premises 1.5.5 SP2 Cumulative hotfix 1 release.

What's new in Cloudera AI on premises 1.5.5 SP2 CHF1

Cloudera AI on premises 1.5.5 SP2 cumulative hotfix 1 delivers a set of new features for Cloudera AI.

Spark pushdown

Spark pushdown for 7.3.2 Cloudera Base is supported for Cloudera AI on premises 1.5.5 SP2 CHF1.

APIv2 endpoint for registering ML Runtime add-ons

Cloudera AI provides an APIv2 endpoint that allows Administrators to upload and register ML Runtime add-ons. The API accepts the JSON ML Runtime add-on repository file, validates its contents, and then initiates asynchronous loading of the add-ons. Administrators can upload add-ons through the Swagger UI or by using command-line tools such as cURL.

Hadoop CLI

Hadoop CLI 7.3.2.0 and higher Hadoop CLI versions mount Java 17 into workloads.

Compatibility for Cloudera AI 1.5.5 SP2 CHF1, Cloudera Runtime components and Data Visualization

Explore Cloudera AI 1.5.5 SP2 CHF1 and its compatibility with Cloudera Runtime components and with Cloudera Data Visualization across various versions.



Important:

- If you switch from the Red Hat Universal Base Image (Redhat UBI image) to the security hardened image, note that the updated component versions and libraries can impact the Cloudera AI operation. Cloudera strongly recommends testing the new implementation in a non-production environment first.

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

For more information, see [Security hardened Spark image migration guide](#).

- From Cloudera Data Services on premises 1.5.5 onwards, Spark 3.4 is no longer available.
- Hive Warehouse Connector (HWC) is supported on all Spark 2.x versions. However, on Spark 3.x versions, HWC is supported on Spark 3.3.2 and higher.

Table 17: Cloudera AI compatibility with Cloudera Runtime component details and with Cloudera Data Visualization for security hardened image and for Redhat insecure image

Cloudera Runtime Version	Spark 2.4.x ³	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.1.9 SP1	- Spark 2.4.8³ - Scala 2.11.12³ - Python 2.7.18³ - Python 3.6.8³ - Java 1.8.0_472³	- Spark 3.3.2 - Scala 2.12.15 - Python 2.7.18³ - Python 3.8.17³ - Python 3.10.19 - Java 11.0.25³ - Java 17.0.17	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.14 - Java 17.0.17	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.17	Iceberg 1.3.0	1.32	8.0.8-b39

Cloudera Runtime Version	Spark 2.4.x ³	Spark 3.3.x	Spark 3.5.x	Airflow	Iceberg	Kubernetes	Cloudera Data Visualization (Data tab)
7.3.1		NA	- Spark 3.5.4 - Scala 2.12.18 - Python 3.11.14 - Java 17.0.17	- Airflow 2.11.0 - Python 3.11.14 - Java 17.0.17	Iceberg 1.5.2	1.32	8.0.8-b39
7.3.2		NA	- Spark 3.5.4 - Scala 2.12.19 - Python 3.11.15 - Java 17.0.18	- Airflow 2.11.0 - Python 3.11.15 - Java 17.0.18	Iceberg 1.5.2	1.32	8.0.8-b39

Supported transformer model architectures for vLLM 0.8.5 in Cloudera AI Inference service 1.5.5 SP2

vLLM 0.8.5 supports the listed transformer model architectures for Cloudera AI Inference service in 1.5.5 SP2 and higher releases.

- aquilaforcausallm
- aquilamodel
- arcticforcausallm
- ariaforconditionalgeneration
- ayavisionforconditionalgeneration
- baichuanforcausallm
- baichuanforcausallm
- bambaforcausallm
- bartforconditionalgeneration
- bartmodel
- bertforsequenceclassification
- bertmodel
- blip2forconditionalgeneration
- bloomforcausallm
- chameleonforconditionalgeneration
- chatglmforconditionalgeneration
- chatglmmodel
- cohere2forcausallm
- cohereforcausallm
- dbrxforcausallm
- decilmforcausallm
- deepseekforcausallm
- deepseekmtpmodel
- deepseekv2forcausallm
- deepseekv2forcausallm
- deepseekv3forcausallm
- eaglellamaforcausallm

- eaglemodel
- exaoneforcausalml
- fairseq2llamaforcausalml
- falconforcausalml
- falconmambaforcausalml
- florence2forconditionalgeneration
- fuyuforcausalml
- gemma2forcausalml
- gemma2model
- gemma3forcausalml
- gemma3forconditionalgeneration
- gemmaforcausalml
- glm4forcausalml
- glm4vforcausalml
- glmforcausalml
- gpt2lmheadmodel
- gptbigcodeforcausalml
- gptjforcausalml
- gptneoxforcausalml
- graniteforcausalml
- granitemoeforcausalml
- granitemoessharedforcausalml
- gritlm
- grok1modelforcausalml
- h2ovlchatmodel
- idefics3forconditionalgeneration
- internlm2forcausalml
- internlm2forrewardmodel
- internlm2veforcausalml
- internlm3forcausalml
- internlmforcausalml
- internvlchatmodel
- jaislmheadmodel
- jambaforcausalml
- jambaforsequenceclassification
- llama4forconditionalgeneration
- llamaforcausalml
- llamaforcausalml
- llamamodel
- llavaforconditionalgeneration
- llavanextforconditionalgeneration
- llavanextvideooforconditionalgeneration
- lavaonevisionforconditionalgeneration
- mamba2forcausalml
- mambaforcausalml
- mantisforconditionalgeneration
- medusamodel
- minicpm3forcausalml
- minicpmforcausalml
- minicpmo

- minicpmv
- minimaxtext01forcausalml
- mistral3forconditionalgeneration
- mistralforcausalml
- mistralmodel
- mixtralforcausalml
- mlpspeculatorpretrainedmodel
- mllamaforconditionalgeneration
- molmoforcausalml
- mptforcausalml
- mptforcausalml
- nemotronforcausalml
- nvlm_d
- olmoforcausalml
- olmo2forcausalml
- olmoeforcausalml
- optforcausalml
- orionforcausalml
- paligemmaforconditionalgeneration
- persimmonforcausalml
- phi3forcausalml
- phi3smallforcausalml
- phi3vforcausalml
- phi4mmforcausalml
- phiforcausalml
- phimoeforcausalml
- pixtralforconditionalgeneration
- prithvigeospatialmae
- qwen2_5_vlforconditionalgeneration
- qwen2audioforconditionalgeneration
- qwen2forcausalml
- qwen2forprocessrewardmodel
- qwen2forrewardmodel
- qwen2forsequenceclassification
- qwen2model
- qwen2vlforconditionalgeneration
- qwen3forcausalml
- qwen3moeforcausalml
- qwenvlforconditionalgeneration
- quantmixtralforcausalml
- qwenlmheadmodel
- robertaformaskedlm
- robertaforsequenceclassification
- robertamodel
- rwforcausalml
- skyworkr1vchatmodel
- smolvlmforconditionalgeneration
- solarforcausalml
- stablelmepochforcausalml
- stablelmforcausalml

- starcoder2forcausal_lm
- telechat2forcausal_lm
- telefilmforcausal_lm
- transformersforcausal_lm
- ultravoxmodel
- whisperforconditionalgeneration
- xlmrobertaforsequenceclassification
- xlmrobertamodel
- xverseforcausal_lm
- zamba2forcausal_lm

Supported vLLM command line arguments

Cloudera AI on premises 1.5.5 SP2 uses vLLM 0.8.5. The command line arguments available in vLLM 0.8.5 are detailed in [vLLM arguments](#).

The following command line arguments are supported with vLLM 0.8.5:

- --block-size
- --calculate-kv-scales
- --cpu-offload-gb
- --disable-cascade-attn
- --disable-chunked-mm-input
- --disable-sliding-window
- --dtype
- --enable-auto-tool-choice
- --enable-chunked-prefill
- --enable-prefix-caching
- --enforce-eager
- --gpu-memory-utilization
- --kv-cache-dtype
- --load-format
- --logprobs-mode
- --long-prefill-token-threshold
- --max-logprobs
- --max-long-partial-prefills
- --max-model-len
- --max-num-batched-tokens
- --max-num-partial-prefills
- --max-num-seqs
- --max-seq-len-to-capture
- --multi-step-stream-outputs
- --no-enable-prefix-caching
- --num-lookahead-slots
- --num-scheduler-steps
- --pipeline-parallel-size, -pp
- --preemption-mode
- --prefix-caching-hash-algo
- --quantization
- --rope-scaling
- --rope-theta
- --scheduling-policy

- --seed
- --tensor-parallel-size, -tp
- --tool-call-parser
- --trust-remote-code

You can find details on the above listed command line arguments here: [vLLM arguments](#).



Note: Only the above listed vLLM command line arguments are supported.

Fixed issues in Cloudera AI on premises 1.5.5 SP2 CHF1

This section lists the issues that have been fixed since the last release of Cloudera AI on premises.

Cloudera AI Workbench

DSE-51975: Some of the Model Metrics charts do not show data

Several Model Metrics charts, such as model replica memory usage, request and response size, and CPU usage, did not display data, due to changes to Kubernetes-level metric names that were not reflected in the corresponding queries. This issue is now fixed by updating query configurations to ensure accurate data collection and proper chart display on the Model Metrics page.

DSE-44227: Save As in the Cloudera AI editor does not switch to a new file

Previously, the **Save As** functionality in the Cloudera AI editor did not switch the view to a newly saved file. The problem occurred because the editor failed to run the open command after saving, leaving users on the original file.

This issue is now fixed. The **Save As** feature now correctly switches the editor view to the newly saved file, ensuring a seamless user experience.

DSE-53131: Expose RocksDB configurations for customer-specific tuning

Previously, essential RocksDB configuration parameters for the Livelog service used hardcoded defaults. Because RocksDB is used as the internal database for the Livelog service, and its performance characteristics can significantly influence livelog service behavior, these hardcoded settings limited performance optimization.

This issue is now resolved. Essential RocksDB configuration parameters can be tuned through ConfigMaps providing greater flexibility for performance optimization while ensuring that existing data and session histories remain unaffected.

The following Key RocksDB tuning parameters can now be adjusted:

- max_bytes_for_level_base
- max_bytes_for_level_multiplier
- Additional recommended options from the RocksDB Tuning Guide

This improvement enables safer, more controlled performance tuning aligned with workload requirements.

DSE-43983: Excessive memory usage in livelog resource during scale testing

Previously, the livelog service exhibited excessive memory usage during scale testing and deployments. This is now resolved by enabling RocksDB Direct I/O, replacing glibc with jemalloc for more efficient memory allocation, and capping the t.last cache in Go to prevent staircase-style memory growth. These improvements significantly reduce memory pressure and improve livelog service performance under load.

DSE#56081: Users cannot pass custom headers in application

Previously, applications could not send custom headers during cross-origin inter-application communication because of a CORS regression. Additionally, Browser Accessible Service (BAS)

incorrectly set credentialed CORS headers for all subdomains. The issue is now fixed. Custom header support and cross-origin inter-application communication are now fully restored, and CORS handling in BAS is corrected to eliminate security vulnerabilities.

The fix includes the following key corrections and improvements :

- Replaced the global CORS middleware with targeted handling scoped specifically to GET requests from the main Cloudera AI domain only.
- Exempted OPTIONS requests from authentication to allow preflight checks to pass.
- Restricted Access-Control-Allow-Origin and Access-Control-Allow-Credentials headers to main#domain GET requests only.
- Added strict origin validation, requiring exact hostname matches and preventing subdomain spoofing or attacker prefix or suffix attacks.
- Restored support for custom headers, such as refresh-token and access-token, in frontend to backend application communication.

Securing applications on Cloudera AI Workbench is a shared responsibility between Cloudera and the users. While the platform enforces secure defaults and strict CORS behavior for core components, users are responsible for defining correct CORS policies and header handling for their own applications. Misconfigured or overly permissive CORS settings in user workloads can result in functional issues or security vulnerabilities.

Jobs and pipelines

DSE-51041: Job Retry changes are not persisting

Previously, job retry settings were saved but failed to be displayed correctly at the job level in the UI because of missing react properties. The issue is now fixed by restoring the missing properties. The UI now properly displays the saved job retry settings.

DSE-50513: Unable to terminate cron job

A bug prevented users from stopping stuck or problematic recurring job instances through the UI. Previously, a recurring job could be marked as **Failed** even if an existing instance was still running. In such cases, the Stop button would not be displayed, leaving users unable to terminate the stuck instances.

This issue is now fixed by adding a Stop button to the job history page, enabling users to stop non-terminal job runs directly from the interface. The Stop button is displayed only for job runs in non-terminal states and includes a confirmation modal along with permission checks to ensure safe and secure operation.

DSE-48646: Web pods crashing (OOM) when job emails are sent with larger attachments

Previously, web pods could crash with out-of-memory (OOM) errors when job notification emails included large attachments or excessive console logs embedded directly in the email body. This issue is now fixed. Only the last 50 lines of console output are included in the email body, while larger logs are sent as attachments when within allowed limits, preventing large logs from being inlined as HTML content.

The fix also introduces configurable attachment size limits (25 MB by default, adjustable between 10–100 MB) using site administration settings. Administrators can control the maximum total attachment size, and if the combined size exceeds this configured limit, attachments are omitted and replaced with platform access links in the email body.

Additionally, email delivery is now more resilient to SMTP constraints. If attachments (even within the configured limit) are rejected by the mail server due to payload size restrictions, the system automatically falls back to sending a links-only email instead of failing the notification.



Note: While this update resolves one major cause of web pod crashes related to large email body payloads and improves overall email reliability, high memory usage persists when fetching large logs from the livelog service to the web service.

ML Runtime and Runtime add-ons

DSE-49301: Missing old topic cleanup for livelog container

Previously, the livelog cleaner service did not remove old topics efficiently. As a result, stale data could accumulate due to earlier livelog cleaner errors, manual deletions from the dashboard table, or prior livelog database compaction. This issue is now resolved. A new configurable cleanup endpoint is introduced to delete topics older than a specified retention period. The solution supports multiple timestamp formats across PBJ, WB, and Engine workloads and can remove workload#related topics that the livelog cleaner failed to delete, including topics whose primary references are missing.

DSE-51856: Jobs ran with APIv 2 are missing Ozone Runtime add-ons

Previously, jobs run through APIv 2 were missing the Ozone runtime add-ons because API v2 did not load the latest Ozone runtime add-ons at the start of the workload.

This issue is now fixed. APIv2 now correctly validates and includes the latest Ozone runtime add-ons in workloads.

DSE-36561: The Update application operation fails when enable_runtime_addons is disabled

Previously, updating an application failed when the **Allow users to use ML runtime addons** feature flag was disabled on the Site Administration page in the UI. The issue occurred because the backend expected an add-ons argument was always present, leading to validation errors when the feature flag was turned off.

This issue is now fixed by modifying the backend handler to validate add-ons only when the feature flag is enabled.

Upgrade

DSE-50712: On premises 1.5.5 SP2 Cloudera AI Registry on OpenShift Container Service: 1.5.4 SP2 to 1.5.5 SP2 upgrade fails with Knox Init:ImagePullBackOff failure

Previously, upgrading the on premises Cloudera AI Registry from 1.5.4 SP2 to 1.5.5 SP2 failed with an ImagePullBackOff error during Knox initialization. This issue occurred only on OpenShift Container Service 4.19 and higher versions due to strict requirements on the secret type used for the imagePullSecret.

The issue is now fixed. The secret creation process now explicitly specifies the dockerconfigjson secret type or replaces it with a similar API from the Cloudera AI Inference service codebase to ensure compatibility and seamless functionality.

DSE-50441: Unable to register or deploy Model with the 1.5.4_h30 # 1.5.5 SP1 # 1.5.5 SP2 upgrade path and DSE-50487: Add workaround for upgrading AI registry from 1.5.4 - 1.5.5

Previously, an issue in the Cloudera AI Registry upgrade path (1.5.4_h30 # 1.5.5 SP1 # 1.5.5 SP2) could prevent model registration or deployment. During the upgrade, the Cloudera AI Registry provisioned a new instance of Cloudera AI Registry and assigned a fresh Persistent Volume (PV), which was later replaced with the legacy PV to preserve existing data. A race condition allowed the database schema migration to run on the fresh PV before the volume swap completed. When the legacy PV was attached afterward, it did not contain the required schema updates, resulting in query failures and deployment errors.

This issue is now fixed. The upgrade process is updated to automatically restart the v1 pod after the PV transfer so that schema migration runs against the correct volume, ensuring consistent storage state and successful model operations.

Cloudera AI on premises 1.5.5 CHF1

Review the features, fixes, and known issues in the Cloudera AI on premises 1.5.5 Cumulative hotfix 1 release.

For information on the Repository locations, see [Repository Locations for Data Services on premises 1.5.5 CHF1](#).

Fixed issues in 1.5.5 CHF1

Review the fixed issues in the Cloudera AI 1.5.5 Cumulative hotfix 1 release.

DSE-37313: Dashboards and user_events tables scalability to avoid web pods OOM and performance issues

Large dashboards and user event tables containing millions of records previously caused web pod out-of-memory (OOM) restarts and UI performance degradation.

This issue is now resolved by archiving old records into the dashboards_archive table, which reduces the size of the dashboards table. As a result, active and archived data are now managed separately to enhance overall performance and query execution for active dashboards is faster, providing a more consistent user experience. For more details, see [Optimized queries with Dashboards Archive table](#).

DSE-42826: Creator filter shows a user multiple times

The creator filter in the Project job list displayed duplicate usernames. This issue occurred when multiple users shared the same full name but had different usernames, resulting in duplicates in the creator filter dropdown. The problem is now resolved by implementing a Map-based method to ensure that each creator appears only once in the dropdown. This fix eliminates redundancy and improves the usability of the creator filter.

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

The job timeout field could not be cleared once set, leading to update errors. This issue is now resolved by converting null timeout updates to 0, which the backend interprets as no timeout restriction. This fix ensures that users can reset the timeout value directly through the UI, providing greater flexibility and control over job timeout configurations.

DSE-45485: Update tenant if the tenant is an empty string in model registry table

If the tenant field in the model registry table was empty in 1.5.4, it led to query failures after upgrading to version 1.5.5. This issue is now resolved by implementing a migration script that sets the empty tenant fields to the appropriate values, ensuring smooth operation after the upgrade.

DSE-25966: Memory leak detected in model proxy

A memory leak in the model proxy caused Kubernetes out-of-memory (OOM) termination and brief model serving outages. This issue is now resolved by replacing the custom cache with the APIv2 cache library, properly closing database calls, eliminating heavy synchronous cache calls, and implementing daily cache pruning. These changes ensure improved stability and performance of the model proxy.

DSE-45050: Large file job attachment crashes web

Large file attachments, ranging from approximately 500MB to 10GB, caused the web container to crash due to unhandled Google Remote Procedure Calls (gRPC) deadline exceeded errors. This issue is now resolved by implementing file size limitation and graceful error handling. The system now refuses Virtual File System (VFS) calls for files exceeding a predefined size limit, preventing the processing of excessively large attachments. Errors related to large file attachments are now handled gracefully, ensuring that the web container remains stable and operational even when such files are encountered.

DSE-41316: Fix Add model resource profile dropdown

The **Add Model Resource Profile** dropdown option in the UI was not functioning correctly due to the broadcast function not being passed to the view layer. As a result, POST arguments failed to update despite changes made in the UI.

This issue is now resolved, ensuring that the dropdown properly updates POST arguments in response to UI changes, restoring its intended functionality.

DSE-45141: Project creation fail - Git Clone

Project creation was failing due to Git clone permission errors on filesystem mounted on Network File System (NFS). The root cause was identified as Git attempting to read temporary pack files directly on NFS, which triggered permission issues.

This issue is now resolved by modifying the process to first clone Git repositories to the local filesystem. The cloned files are then copied to the NFS path, effectively avoiding NFS-related permission problems and ensuring successful project creation.

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

Web requests to other services were logging malformed Universally Unique Identifier (UUIDs), which complicated debugging efforts. The issue was traced back to request IDs from web pods being trimmed in operator logs due to a 30-character limit on the `contextId` element.

This issue is now resolved by increasing the length of the contextId element to 36 characters, ensuring that full UUIDs are preserved in the logs. This fix improves log accuracy and facilitates more effective debugging.

DSE-44088: Operator pod start failure log missing ID

Operator pod failure logs lacked critical identifiers, such as the engineId identifier and request UUID, which complicated error identification and traceability. This issue is now resolved by enhancing the failure logs to include both the engineId and request UUID identifiers for engine creation by the operator pod. This improvement provides better error traceability and simplifies debugging.

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

During the StartJobRun process, the Kubernetes client failure errors were not correctly reported, leading to misleading success messages. The issue stemmed from failure responses from the Kubernetes client not being properly translated, resulting in false indications of success. The root cause was identified in the error handling of the GetFailureResponse function, which lacked robustness. This issue is now resolved, and engine start failures are accurately reported with detailed error data, eliminating misleading success messages and improving error transparency.