Machine Learning 1.5.1

Machine Learning Runtimes Release Notes

Date published: 2020-07-16
Date modified: 2023-01-31

CLOUDERA

Legal Notice

Contents

ML Runtimes What's New

Major features and updates for ML Runtimes.

ML Runtime Releases included in CML PVC versions

- Runtime Release 2022.11 is included in CML PVC 1.5.0, 1.5.1
- Runtime Release 2021.12 is included in CML PVC 1.4.0, 1.4.1

Related Information
ML Runtimes Nvidia GPU Edition
ML Runtimes 2021.02
Upgrading R and Python Packages
ML Runtimes Environment Variables

ML Runtimes Version 2022.11.2

Major features and updates for ML Runtimes.

This release is available with ML Runtimes version 2022.11.2.

Note: New ML Runtime releases are automatically added to the deployment, if internet connection is available.

Version 2022.11.2

Fixed Issues

- Multiple CVEs have been fixed.

ML Runtimes Version 2022.11

Major features and updates for ML Runtimes.

This release is available with ML Runtimes version 2022.11

Note: New ML Runtime releases are automatically added to the deployment, if internet connection is available.

Version 2022.11

New features

- PBJ (Powered by Jupyter) Runtime GA: PBJ Runtimes were previously available as a Tech Preview only. In ML Runtimes 2022.11 the PBJ Runtimes are available to all users (GA). The following PBJ Runtimes are included:

  - Python 3.7
  - Python 3.8
  - Python 3.9
  - R 3.6
  - R 4.0
  - R 4.1

  These PBJ Runtimes are available on Cloudera Public Cloud (from 2022.10-1) and Cloudera Private Cloud (from 1.5.0) form factors, but are not available on CDSW.

  The new PBJ Runtimes appear on the Runtime Selector elements when users start new workloads. Their usage is the same as the non-PBJ Runtimes, apart from Models. Models require the usage of a new library called cml for

Python and R, that was introduced in Cloudera Public Cloud (2022.10-1) and the upcoming Private Cloud (1.5.0) versions.

Usage of cml with regard to models is detailed here: PBJ Runtimes and Models

Examples can be found here: Example models with PBJ Runtimes.

Fixed Issues

- DSE-22433 Upgrade numpy version in Python runtimes to resolve issues.
- DSE-22374 Reduced log level to WARN log level can be manipulated through the HADOOP_ROOT_LOGGER environment variable. This is set to WARN,console in order to keep console logging, while reducing log level from INFO to WARN.
- DSE-22251 Add flextable dependencies to R Runtimes Added flextable dependencies (libfontconfig1-dev, libcairo2-dev) to R runtimes. Added two OS libraries so flextable can be installed and used in CML.

Known issues

- The CUDA version (11.4.1) being used to build Nvidia GPU Runtimes is not supported by newer Torch versions (1.13+).

  Workaround: Use Torch versions up to 1.12.1.

  Cloudera Bug: DSE-24038
- When installing sparklyr for R, the installation will succeed, but importing the library in the same session will fail. This happens, because earlier versions of dependent packages (vctrs, lifecycle, rlang, cli) are already loaded in the R session, therefore newer versions cannot be loaded.

  Workaround: Start a new session, import and use sparklyr there.

  Cloudera Bug: DSE-23762

ML Runtimes Version 2022.04

Major features and updates for ML Runtimes.

This release is available with ML Runtimes version 2022.04.

Version 2022.04

New features

- *Powered By Jupyter (PBJ) Workbench* (Preview) - The Powered By Jupyter (PBJ) Workbench features the classic workbench UI backed by the open-source Jupyter protocol pre-packaged in an ML Runtime image. Users can easily choose this Runtime when launching a session. The open-source Jupyter infrastructure eliminates the dependency on proprietary CML code when building a Docker image, allowing runtime images to be built with open standards. The first PBJ Workbench Runtimes are available as Tech Preview.
- The Global Registry for npm is set to https://registry.npmjs.org/ in all of our runtimes. With this change, customers can use npm install in their Dockerfiles, when they build custom runtimes on top of ours.

  Note: RAPIDS Runtimes are now an optional extension of the Cloudera ML Runtime distribution. The RAPIDS images will no longer be distributed automatically. Administrators can Register them in the Runtime Catalog.

Fixed issues

- DSE-2064 Upgrade of Python Kernels across of all ML Runtimes:
  - Python 3.7.12 -> 3.7.13
  - Python 3.8.12 -> 3.8.13
  - Python 3.9.7 -> Python 3.9.11

ML Runtimes Version 2021.12

Major features and updates for ML Runtimes.

This release is available with ML Runtimes version 2021.12.

Version 2021.12

New features

- New versions provided for all ML Runtimes.
- Added R 4.1 Runtimes
- RAPIDS base image was updated to use 21.12 RAPIDS release (DSE-19745)

Note: Python 3.6 ML Runtimes are not longer provided due to Python 3.6 end of life

Fixed issues

- DSE-17817 psycopg2 now can be installed for Models.
- DSE-17547 Fixed a matplotlib warning in RAPIDS Runtimes.
- DSE-10146 Fixed issue causing R workbench to crash due to an illegal character.
- DSE-17657 Upgrade jupyterlab and notebook for CVE-2021-32797 and CVE-2021-32798.
- Various security and library version updates
- DSE-19805 Fixed CVE-2022-22817 critical vulnerability in Python Runtimes

ML Runtimes Version 2021.09.02

Major features and updates for ML Runtimes.

Version 2021.09.02

Fixed issues

- DSE-19496 The log4j build dependency is removed for the Scala ML Runtime. Once this ML Runtimes version is deployed, the user interface will always point to this latest version of the 2021.09.02 Scala Runtime to run workloads using the Scala kernel (until a newer version is released). Users will not be able to select the originally released 2021.09 Scala Runtime.

ML Runtimes Version 2021.09

Major features and updates for ML Runtimes.

Version 2021.09

New features

- Python 3.9 - ML Runtimes now support Python 3.9 kernels.
- Scala - Scala kernel is supported for running Sessions and Jobs.
- CUDA 11.4 - Nvidia GPU Editions are using CUDA 11.4.

Fixed issues

- DSE-17126 Starting a worker from Runtime session will create a worker with engine image

  This issue is resolved in ML Runtimes 2021.09 when used with the latest ML Workspace versions.

  For workers to function properly with ML Runtimes, please use ML Runtimes 2021.09 or later with CML Workspace version of 2.0.22 or later.

ML Runtimes Version 2021.06

Major features and updates for ML Runtimes.

Version 2021.06

New features

- Impyla - Impyla is now preinstalled in standard Python Runtimes.

  Cloudera bug: DSE-14293
- Runtimes support for Models - Runtimes now support running models.

  Cloudera bug: DSE-14294
- JupyterLab shutdown_no_activity_timout configurable - JupyterLab shutdown_no_activity_timout is now configurable through environment variable.

  Cloudera bug: DSE-14817

Fixed issues

- DSE-15191 - Fixed installation issue with thrift-sasl python package.
- DSE-14886 - Fixed an issue preventing matplotlib installation into project.
- DSE-14737 - Fixed warnings appearing in the log due to using JupyterLab 3.
- DSE-15215 - Fixed an issue due to decorators that could not be used with Python 3.8 and Workbench editor.

ML Runtimes Version 2021.04

Major features and updates for ML Runtimes.

Version 2021.04

New features

- RAPIDS Runtimes - The RAPIDS Edition Runtimes are built on top of community built RAPIDS docker images. The RAPIDS suite of software libraries relies on NVIDIA CUDA primitives for low-level compute optimization, but exposes that GPU parallelism and high-bandwidth memory speed through user-friendly Python interfaces.

Fixed issues

The following fixed issus relate only to RAPIDS Runtimes.

- DSE-13743 - Idle JupyterLab sessions are now ended after around   IDLE_MAXIMUM_MINUTES. (See *ML Runtimes Environment Variables Environment Variables*).

  > **Note:** These sessions may run for an additional 5 minutes after IDLE_MAXIMUM_MINUTES. (DSE-13743)

- DSE-14979 - matplotlib figures restore the styling used in engine:13 (MLRuntimes 2020.11 used the matplotlib defaults)
- DSE-12881 - For Python runtimes, py4j is now installed.
- Security fixes, python library version updates

ML Runtimes Version 2021.02

Major features and updates for ML Runtimes.

Version 2021.02

New features

- Nvidia GPU support - Runtimes supporting Nvidia GPUs are available with preinstalled CUDA software for both Workbench and JupyterLab editors. (See *ML Runtimes Nvidia GPU Edition*.)
- Runtimes supporting R 3.6 and R 4.0 available for Workbench editor

  > **Note:** User R libraries are installed in /home/cdsw/.local/lib/R rather than /home/cdsw/R, which is where they were installed in engine:13. This change was made to support simultaneous use of R 3.6 and R 4.0. Because of this change you will need to reinstall user R packages when migrating an existing project from engine:13 to MLRuntimes 2021.02, or from R 3.6 to R 4.0

  > Cloudera bug: DSE-3686

- Support for JupyterLab runtimes - The JupyterLab runtimes now use JupyterLab 3.0 (was 2.2) and are considered generally available using Standard Edition ML Runtimes. See JupyterLab blog for notes on this upgrade.
- Python runtimes include a C++ compiler - Python runtimes now include a C++ compiler (g++), which enables the installation and use of certain Python libraries such as impyla and pystan.

  Cloudera bug: DSE-14492

- Preinstalled Python libraries (See *ML Runtimes 2021.02*) are now installed in /usr/local/lib rather than /var/lib/ cdsw, which is where they were installed in engine:13 and runtimes 2020.11 (DSE-12177). This means that you can upgrade these packages more easily, but there are some packages you should not upgrade. (See *Upgrading R and Python Packages*)
- ML runtimes are supported on CDSW (version 1.9.0 or higher).

Fixed issues

- DSE-13743 - Idle JupyterLab sessions are now ended after around   IDLE_MAXIMUM_MINUTES. (See *ML Runtimes Environment Variables Environment Variables*).

  > **Note:** These sessions may run for an additional 5 minutes after IDLE_MAXIMUM_MINUTES. (DSE-13743)

- DSE-14979 - matplotlib figures restore the styling used in engine:13 (MLRuntimes 2020.11 used the matplotlib defaults)
- DSE-12881 - For Python runtimes, py4j is now installed.
- Security fixes, python library version updates

ML Runtimes Version 2020.11

Release notes and fixed issues

Version 2020.11

New features

- New Runtimes versions added

  > **Note:** Nvdia GPU Edition comes with CUDA 11.1 preinstalled.

Fixed issues

- DSE-13904 - Fixed an issue where CML workspace installation may take up to 10 minutes longer when the Autoscaling Group minimum is set to 0.
- DSE-13898 - Updated TGT image to fix an issue so that the freeIPA HA release retries on failure.
- CDPSDX-2673 - Added a Retry step to the login context to reduce the chance of PRE_AUTH failures.

ML Runtimes Known Issues and Limitations

You might run into some known issues while using ML Runtimes.

DSE-25143 Assembling plots in PBJ R Runtimes does not work

When trying to plot additional content on already existing plots, PBJ R Runtimes throw an error. Plots can only be created using the plot function.

DSE-32839 Extra configuration needed when using Spark in a PBJ Workbench-based R Runtime

When using Spark in R workloads that are running PBJ Workbench Runtimes, the environmental variable R_LIBS_USER must be passed to the Spark executors with the value set to "/home/cdsw/.local/lib/R/<R_VERSION>/library".

E.g. when using `sparklyr` with a PBJ Workbench R 4.3 Runtime, the correct way to set up a `sparklyr` connection is:

```
library(sparklyr)
config <- spark_config()
config$spark.executorEnv.R_LIBS_USER="/home/cdsw/.local/lib/R/4.3/library"
sc <- spark_connect(config = config)
```

DSE-27222 Workbench editor for Python Runtimes and PBJ Runtimes cannot parse multiline strings properly

Workbench editor for Python 3.7 Runtime cannot parse multiline strings properly. Trying to evaluate multiline strings in some cases result in a "SyntaxError: EOF while scanning triple-quoted string literal" error message. This has been fixed in Python 3.8 and higher.

Workaround: Transform multiline strings to single line strings in code that is entered into the workbench directly.

Packages can fail to load in a session

When installing R or Python packages in a Session, the kernel might not be able to load the package in the same session, if a previous version of the package or its newly installed dependencies have been loaded in the same Session. Such issues are observable more often in PBJ R Runtimes, which automatically load basic R packages like `vctrs`, `lifecycle`, `rlang`, `cli` at session startup.

Workaround: Start a new session, import and use the newly installed package there.

Python Runtimes in CML fail to import the setuptools Python library and can fail installing some Python packages

Python Runtimes in CML fail to import the setuptools Python library and therefore can fail installing some Python packages when the library setuptools is present on the Runtime or is installed into the CML project with version 60.0.0 or higher.

Python 3.10 Runtimes from the 2023.05 Runtime release ship with a newer version of setuptools, so customers can run into this issue when they are using that Runtime. Also they can run into this issue when they are using Custom Runtimes that has a newer setuptools library version or when they install a new setuptools version into their project (regardless of what Runtime they use).

Workaround: Set the environmental variable SETUPTOOLS_USE_DISTUTILS=stdlib either on a project level under Project Settings -> Advanced or on a workspace level under Site Administration -> Runtime -> Environment variables.

Version of jupyter-client Python package must be less than version 8 for PBJ Runtimes

Upgrading the Python package jupyter-client with a version greater than 7.4.9 can temporarily break a Project. Workloads using PBJ Runtimes will not be able to start Projects if the jupyter-client version is greater than 7.4.9.

Workaround: Launch the same version of Python, but not on a PBJ Runtime (either Workbench or JupyterLab). Open a Terminal window and uninstall the jupyter-client package from the Project by executing pip3 uninstall jupyter-client. Verify your change by running pip3 list and checking that the version of the jupyter-client package is less than version 8.

DSE-9818 JupyterLab Conda Tech Preview Runtime

Sessions

When starting a Notebook or a Console for a specific environment, the installed packages will be available and the interpreter used to evaluate the contents of the Notebook or Console will be the one installed in the environment. However, the Conda environment is not "activated" in these sessions, therefore commands like !which python will return with the base Python 3.10 interpreter on the Runtime. The recommended ways to modify a Conda environments or install packages are the following:

- conda commands must be used with the -n or --name argument to specify the environment, for example conda -n myenv install pandas
- When installing packages with pip, use the %pip magic to install packages in the active kernel's environment, for example %pip install pandas

Applications and Jobs

To start an Application or Job, first create a launcher Python script containing the following line: ! source activate <conda_env_name> && python <job / application script.py>

When starting the Application or Job, select the launcher script as the "Script".

Models

Models are currently not supported for the Conda Runtime.

Spark

Spark is not supported in JupyterLab Notebooks and Consoles.

Spark workloads are supported in activated Conda environments in JupyterLab Terminals, or in Jobs or Applications.

The CDSW libraries for Python and R are not available for the Conda Runtimes.

DSE-27222 Workbench editor for Python Runtimes and PBJ Runtimes cannot parse multiline strings properly

Workbench editor for Python 3.7 Runtime cannot parse multiline strings properly. Trying to evaluate multiline strings in some cases result in a "SyntaxError: EOF while scanning triple-quoted string literal" error message.

Workaround: Transform multiline strings to single line strings in code that is entered into the workbench directly.

Adding a new ML Runtimes when using a custom root certificate might generate error messages

When trying to add new ML Runtimes, a number of error messages might appear in various places when using a custom root certificate. For example, you might see: "Could not fetch the image metadata" or "certificate signed by unknown authority". This is caused by the runtime-puller pods not having access to the custom root certificate that is in use.

Workaround:

1. Create a directory at any location on the master node:

   For example:

   mkdir -p /certs/

2. Copy the full server certificate chain into this folder. It is usually easier to create a single file with all of your certificates (server, intermediate(s), root):

   ```
   # copy all certificates into a single file:
   cat server-cert.pem intermediate.pem root.pem > /certs/cert-chain.crt
   ```

3. (Optional) If you are using a custom docker registry that has its own certificate, you need to copy this certificate chain into this same file:

```
cat docker-registry-cert.pem >> /certs/cert-chain.crt
```

4. Copy the global CA certificates into this new file:

```
# cat /etc/ssl/certs/ca-bundle.crt >> /certs/cert-chain.crt
```

5. Edit your deployment of runtime manager and add the new mount.

   Do not delete any existing objects.

   #kubectl edit deployment   runtime-manager

6. Under VolumeMounts, add the following lines.

   Note that the text is white-space sensitive - use spaces and not tabs.

```
- mountPath: /etc/ssl/certs/ca-certificates.crt
    name: mycert
    subPath: cert-chain.crt #this should match the new file name created in
 step 4
```

   Under Volumes add the following text in the same edit:

```
- hostPath:
    path: /certs/  #this needs to match the folder created in step 1
    type: ""
name: mycert
```

7. Save your changes:

   wq!

   Once saved, you will receive the message "deployment.apps/runtime-manager edited" and the pod will be restarted with your new changes.

8. To persist these changes across cluster restarts, use the following Knowledge Base article to create a kubernetes patch file for the runtime-manager deployment: https://community.cloudera.com/t5/Customer/Patching-CDSW-Kubernetes-deployments/ta-p/90241

Cloudera Bug: DSE-20530

Spark Runtime Add-on required for Spark 2 integration with Scala Runtimes

Scala Runtimes on CML require Spark Runtime Addon to enable Spark2 integration. Spark3 is not supported with the Scala Runtime.

DSE-17981 - Disable Scala runtimes in models, experiments and applications runtime selection

Scala Runtimes should not appear as an option for Models, Experiments, and Applications in the user interface. Currently Scala Runtimes only support Session and Jobs.