# PBJ Workbench (Preview)

Date published: 2022-04-21
Date modified: 2022-04-28

# Legal Notice

# Contents

# PBJ Workbench

The PBJ Workbench features the classic workbench UI backed by the open-source Jupyter protocol pre-packaged in a runtime image. Users can easily choose this runtime image 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 more quickly.

Note: Before constructing a new PBJ Workbench runtime image, make sure to upgrade your workspace to CDSW-2.0.29-b61 or higher.

## Accessing the PBJ Workbench

The PBJ Workbench is available by default, but you have to select it when you launch a session.

1. Click **New Session**.
2. In **Runtime** > **Editor**, select **PBJ Workbench**.
3. Click **Start Session**.

Now you can use the PBJ Workbench as you would the normal workbench.

# Constructing a PBJ Workbench runtime image

You can also construct a custom runtime image that includes the PBJ Workbench, by starting from any base image (including arbitrary non-Cloudera base images) and to use them with the CDSW/CML workbench.

This guide documents the requirements for such an image to be compatible with the workbench and the wider Cloudera data platform.

**Install Jupyter dependencies**

First, the Jupyter kernel gateway, version 2.5.1 must be installed into the Docker image. This example command may need to be modified depending on the filename and path of the pip executable in the image.

```
RUN pip3 install "jupyter-kernel-gateway==2.5.1"
```

The path to the `jupyter` executable installed by pip should be noted, and the command to run the Jupyter kernel gateway on IP address 127.0.0.1 and port 8888

```
/path/to/jupyter kernelgateway --ip='127.0.0.1' --port=8888 --debug
```

should be incorporated into the runtime metadata.

Each instance of the PBJ workbench communicates with a Jupyter kernel via the Jupyter protocol. Kernels are [available](#) for a wide variety of languages and versions. Install the kernel of your choice to the image by following its installation instructions. Then, note the name of the kernel, ie its directory name under Jupyter's kernel spec path. For example, if the kernel is installed to `jupyter/kernels/python3` then its name, for CML's purposes, is `python3`. The name should be incorporated into the runtime metadata under the field `kernel`.

**Add the cdsw user (recommended)**

User code will be run in the image under user and group 8536.8536 . To enhance the readability of file ownership data and shell output, we recommend associating these id's with the name cdsw in the image by adding the following command to the dockerfile:

```
RUN groupadd --gid 8536 cdsw && \
    useradd -c "CDSW User" --uid 8536 -g cdsw -m -s /bin/bash cdsw
```

**Relax permissions so that Cloudera client config can be written**

All code in the runtime container, including initial setup, will be run as the cdsw user. The initial setup includes linking client files for Cloudera data services out to their standard paths. To facilitate this, permissions to the following paths should be set so that user 8536 can write to them:

- /etc
- /usr/share/java

**Full example Dockerfile**

This Dockerfile produces a runtime image that is compatible with the PBJ workbench from a third-party base image.

```
FROM jupyter/datascience-notebook
```

```
# Install the Jupyter kernel gateway.
# The IPython kernel is automatically installed to
# /opt/conda/share/jupyter/kernels/python3
# so below we set the kernel name to python3.
#
# Note that this base image has python3 and pip3
# preinstalled. If it did not, we would need to
# add steps to install them.
RUN pip3 install "jupyter-kernel-gateway==2.5.1"


# Associate uid and gid 8536 with username cdsw
RUN \
  addgroup --gid 8536 cdsw && \
  adduser --disabled-password --gecos "CDSW User" --uid 8536 --gid
8536 cdsw



# Relax permissions to facilitate installation of Cloudera
# client files at startup
RUN chown -R 8536.8536 /etc && chown -R 8536.8536 /usr/share/java

# Configure the Kernel Gateway CommandLabel the image
LABEL com.cloudera.ml.runtime.editor=PBJWorkbench
LABEL com.cloudera.ml.runtime.kernel=python3
LABEL
com.cloudera.ml.runtime.kernel-gateway-command="/opt/conda/bin/jupyter
kernelgateway −debug −ip='127.0.0.1' −port=8888"

# Set Runtime label and environment variables metadata
ENV ML_RUNTIME_EDITOR="PBJ Workbench" \
    ML_RUNTIME_KERNEL="Python 3.7" \
    ML_RUNTIME_EDITION="Custom" \
    ML_RUNTIME_SHORT_VERSION="1.0" \
    ML_RUNTIME_MAINTENANCE_VERSION="1" \


ML_RUNTIME_FULL_VERSION="$ML_RUNTIME_SHORT_VERSION.$ML_RUNTIME_MAINTEN
ANCE_VERSION" \
```

```
 ML_RUNTIME_DESCRIPTION="This runtime includes telnet and sklearn and
upgraded packages" \
ML_RUNTIME_METADATA_VERSION="2"

LABEL com.cloudera.ml.runtime.editor=$ML_RUNTIME_EDITOR \
      com.cloudera.ml.runtime.kernel=$ML_RUNTIME_EDITOR \
      com.cloudera.ml.runtime.edition=$ML_RUNTIME_EDITOR \
      com.cloudera.ml.runtime.full-version=$ML_RUNTIME_FULL_VERSION \
      com.cloudera.ml.runtime.short-version=$ML_RUNTIME_SHORT_VERSION \
      com.cloudera.ml.runtime.maintenance-version=$ML_RUNTIME_MAINTENAN
CE_VERSION \
      com.cloudera.ml.runtime.description=$ML_RUNTIME_DESCRIPTION \
      com.cloudera.ml.runtime.runtime-metadata-version=$ML_RUNTIME_MET
ADATA_VERSION
```

# Limitations

Workers are not supported by the PBJ Workbench.