



Red Hat OpenShift Data Science 1

データサイエンスプロジェクトでの作業

OpenShift Data Science ノートブックでのプロジェクトの作成および連携

Red Hat OpenShift Data Science 1 データサイエンスプロジェクトでの作業

OpenShift Data Science ノートブックでのプロジェクトの作成および連携

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

OpenShift Data Science ノートブック環境で、ノートブックを作成し、連携する方法を説明します。

目次

第1章 ノートブックの作成およびインポート	3
1.1. 新規ノートブックの作成	3
1.2. ローカルストレージからの既存のノートブックのアップロード	4
1.3. JUPYTERLAB を使用した GIT リポジトリからの既存ノートブックファイルのアップロード	5
1.4. コマンドラインインターフェイスを使用した GIT リポジトリから既存のノートブックファイルのアップロード	5
1.5. 関連情報	6
第2章 GIT を使用したノートブックでの連携	7
2.1. JUPYTERLAB を使用した GIT リポジトリからの既存ノートブックファイルのアップロード	7
2.2. コマンドラインインターフェイスを使用した GIT リポジトリから既存のノートブックファイルのアップロード	7
2.3. リモート GIT リポジトリからの変更を含むプロジェクトの更新	8
2.4. プロジェクトの変更を GIT リポジトリにプッシュ	9
第3章 データサイエンスプロジェクトでの作業	10
3.1. データサイエンスプロジェクトの使用	10
3.2. プロジェクトワークベンチの使用	12
3.3. データ接続の使用	15
3.4. クラスタストレージの設定	18
3.5. データサイエンスパイプラインの設定	20
3.6. モデルサーバーの設定	23
3.7. データサイエンスプロジェクトへのアクセスの設定	30
3.8. ノートブックサーバーにインストールされている PYTHON パッケージの表示	34
3.9. ノートブックサーバーへの PYTHON パッケージのインストール	35
3.10. サーバーの再起動によるノートブックサーバー設定の更新	36
第4章 データサイエンスパイプラインの操作	37
4.1. データサイエンスパイプラインの管理	37
4.2. パイプライン実行の管理	44
4.3. JUPYTERLAB でのパイプラインの操作	52
4.4. 関連情報	62
第5章 OPENSIFT DATA SCIENCE のモデルサービス	63
5.1. モデルサーバーの設定	63
5.2. デプロイされたモデルの使用	69
第6章 JUPYTER での一般的な問題の管理者向けトラブルシューティング	74
6.1. ユーザーが JUPYTER にログインすると、404: PAGE NOT FOUND エラーが表示される	74
6.2. ユーザーのノートブックサーバーが起動しない	74
6.3. ノートブックセルの実行時に DATABASE OR DISK IS FULL エラーまたは NO SPACE LEFT ON DEVICE エラーが表示される	75
第7章 JUPYTER での一般的な問題のトラブルシューティング	77
7.1. JUPYTER にログインすると 403: FORBIDDEN エラーが表示される	77
7.2. ノートブックサーバーが起動しない	77
7.3. ノートブックセルの実行時に DATABASE OR DISK IS FULL エラーまたは NO SPACE LEFT ON DEVICE エラーが表示される	77

第1章 ノートブックの作成およびインポート

空のノートブックを作成したり、さまざまなソースからノートブックをインポートしたりできます。

1.1. 新規ノートブックの作成

既存のノートブックコンテナイメージから新規の Jupyter ノートブックを作成し、そのリソースおよびプロパティにアクセスできます。Notebook server control panelには、単一ユーザーのノートブックサーバーとして実行できる利用可能なコンテナイメージの一覧が含まれます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- ノートブックサーバーを起動し、Jupyter にログインしていることを確認している。
- ノートブックイメージがレジストリー、イメージストリームに存在し、アクセス可能である。

手順

1. File → New → Notebook の順にクリックします。
2. プロンプトが表示されたら、リストからノートブックのカーネルを選択します。カーネルを使用する場合は、Select をクリックします。カーネルを使用しない場合は、No Kernel をクリックします。

検証

- ノートブックファイルが JupyterLab に表示されることを確認します。

1.1.1. データサイエンティストのノートブックイメージ

Red Hat OpenShift Data Science には、データサイエンスの作業に必要な業界をリードするツールやライブラリーで最適化された Jupyter ノートブックイメージが含まれます。モデル開発に一貫性のある安定したプラットフォームを提供するために、すべてのノートブックイメージには、同じバージョンの Python が含まれます。Red Hat OpenShift Data Science で利用可能なノートブックイメージは事前に構築されており、OpenShift Data Science のインストールまたはアップグレード後すぐに使用できるようになっています。

ノートブックイメージは、少なくとも1年間サポートされます。事前に設定されたノートブックイメージへのメジャー更新は、約6か月ごとに行われます。したがって、サポートされている2つのノートブックイメージはいつでも利用できます。このサポート期間を使用して、利用可能な最新のノートブックイメージからコンポーネントを使用するようにコードを更新できます。

必要な場合は、サポートされなくなった場合でも、レジストリーから古いノートブックイメージにアクセスできます。次に、古いノートブックイメージをカスタムノートブックイメージとして追加し、プロジェクトの特定の要件に対応できます。

Red Hat OpenShift Data Science には、デフォルトでインストールされる以下のノートブックイメージが含まれます。

表1.1 デフォルトのノートブックイメージ

イメージ名	説明
CUDA	GPU サポートを必要とするコンピュート集約型データサイエンスモデルを使用している場合は、Compute Unified Device Architecture (CUDA) ノートブックイメージを使用して NVIDIA CUDA Toolkit にアクセスします。このツールキットを使用すると、GPU アクセラレーションライブラリーおよび最適化ツールを使用して作業を最適化できます。
Standard Data Science	TensorFlow または PyTorch を必要としないモデルには、Standard Data Science ノートブックイメージを使用します。このイメージには、機械学習モデルの開発に役立つ、一般的に使用されるライブラリーが含まれています。
TensorFlow	TensorFlow は、機械学習のためのオープンソースプラットフォームです。TensorFlow を使用すると、マシンラーニングモデルを構築し、トレーニングし、デプロイできます。TensorFlow には、計算グラフの視覚化など、高度なデータ可視化機能が含まれています。また、モデルの進捗を簡単に監視し、追跡することもできます。
PyTorch	Pytorch は、ディープラーニング向けに最適化されたオープンソース機械学習ライブラリーです。コンピュータービジョンまたは自然言語処理モデルを使用している場合は、Pytorch ノートブックイメージを使用してください。
Minimal Python	高度な機械学習機能や、計算量の多いデータサイエンス作業のための追加のリソースが必要ない場合は、Minimal Python イメージを使用してモデルを開発できます。
TrustyAI	TrustyAI ノートブックイメージを使用して、モデルの説明可能性、トレースと説明可能性、およびランタイム監視を使用したデータサイエンスの作業を活用します。

関連情報

- [ノートブックサーバーへの Python パッケージのインストール](#)
- [ノートブックサーバー環境のオプション](#)


1.2. ローカルストレージからの既存のノートブックのアップロード

ローカルストレージから JupyterLab に既存のノートブックを読み込んで作業を継続したり、新しいユースケースに合わせてプロジェクトを調整したりできます。

前提条件

- Jupyter にログインするための認証情報。
- 起動済みで実行中のノートブックサーバー。
- ノートブックファイルがローカルストレージにある。

手順

1. JupyterLab インターフェイスの左サイドバーにある **File Browser** で、**Upload Files** () をクリックします。
2. ノートブックファイルを見つけ、これを選択して **Open** をクリックします。

ファイルは **File Browser** に表示されます。

検証

- ノートブックファイルは、JupyterLab インターフェイスの左側のサイドバーの **File Browser** に表示されます。
- JupyterLab でノートブックファイルを開くことができます。



1.3. JUPYTERLAB を使用した GIT リポジトリからの既存ノートブックファイルのアップロード

JupyterLab ユーザーインターフェイスを使用して Git リポジトリのクローンをワークスペースに作成し、そのまま作業を継続したり、外部プロジェクトのファイルを統合したりできます。

前提条件

- 起動済みで実行中の Jupyter サーバー。
- クローンを作成する Git リポジトリの読み取りアクセス。

手順

1. Git リポジトリの HTTPS URL をコピーします。
 - GitHub で **Code** → **HTTPS** をクリックし、Clipboard ボタンをクリックします。
 - GitLab で **Clone** をクリックし、**Clone with HTTPS** の下にある Clipboard ボタンをクリックします。
2. JupyterLab インターフェイスで、**Git Clone** ボタン () をクリックします。
メニューで **Git** → **Clone a repository** をクリックするか、Git アイコン () をクリックして **Clone a repository** ボタンをクリックすることもできます。
Clone a repo ダイアログが表示されます。
3. ノートブックを格納するリポジトリの HTTPS URL を入力します。
4. **CLONE** をクリックします。
5. プロンプトが表示されたら、Git リポジトリのユーザー名とパスワードを入力します。

検証

- リポジトリの内容が JupyterLab のファイルブラウザーに表示されるか、ターミナルで **ls** コマンドを実行して、リポジトリがディレクトリとして表示されることを確認します。

1.4. コマンドラインインターフェイスを使用した GIT リポジトリから既存のノートブックファイルのアップロード

コマンドラインインターフェイスを使用して Git リポジトリをワークスペースにクローンし、そのまま作業を継続したり、外部プロジェクトのファイルを統合したりできます。

前提条件

- 起動済みで実行中の Jupyter サーバー。

手順

1. Git リポジトリの HTTPS URL をコピーします。
 - GitHub で **Code** → **HTTPS** をクリックし、Clipboard ボタンをクリックします。
 - GitLab で **Clone** をクリックし、**Clone with HTTPS** の下にある Clipboard ボタンをクリックします。
2. JupyterLab で **File** → **New** → **Terminal** の順にクリックして、ターミナルウィンドウを開きます。
3. **git clone** コマンドを入力します。

```
git clone <git-clone-URL>
```

`<git-clone-URL>` は HTTPS URL に置き換えます。以下に例を示します。

```
[1234567890@jupyter-nb-jdoe ~]$ git clone https://github.com/example/myrepo.git
Cloning into myrepo...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 2821 (delta 1), reused 5 (delta 1), pack-reused 2810
Receiving objects: 100% (2821/2821), 39.17 MiB | 23.89 MiB/s, done.
Resolving deltas: 100% (1416/1416), done.
```

検証

- リポジトリの内容が JupyterLab のファイルブラウザーに表示されるか、ターミナルで **ls** コマンドを実行して、リポジトリがディレクトリとして表示されることを確認します。

1.5. 関連情報

- [Git を使用したノートブックでの連携](#)

第2章 GIT を使用したノートブックでの連携

ノートブックまたはその他のファイルが Git バージョン制御に保存される場合は、Git リポジトリからノートブックサーバーにインポートして JupyterLab で使用できます。準備が整ったら、変更を Git リポジトリに再びプッシュし、他のユーザーがモデルをレビューしたり、使用したりできるようにします。



2.1. JUPYTERLAB を使用した GIT リポジトリからの既存ノートブックファイルのアップロード

JupyterLab ユーザーインターフェイスを使用して Git リポジトリのクローンをワークスペースに作成し、そのまま作業を継続したり、外部プロジェクトのファイルを統合したりできます。

前提条件

- 起動済みで実行中の Jupyter サーバー。
- クローンを作成する Git リポジトリの読み取りアクセス。

手順

1. Git リポジトリの HTTPS URL をコピーします。
 - GitHub で **Code** → **HTTPS** をクリックし、Clipboard ボタンをクリックします。
 - GitLab で **Clone** をクリックし、**Clone with HTTPS** の下にある Clipboard ボタンをクリックします。
2. JupyterLab インターフェイスで、**Git Clone** ボタン () をクリックします。
メニューで **Git** → **Clone a repository** をクリックするか、Git アイコン () をクリックして **Clone a repository** ボタンをクリックすることもできます。
Clone a repo ダイアログが表示されます。
3. ノートブックを格納するリポジトリの HTTPS URL を入力します。
4. **CLONE** をクリックします。
5. プロンプトが表示されたら、Git リポジトリのユーザー名とパスワードを入力します。

検証

- リポジトリの内容が JupyterLab のファイルブラウザーに表示されるか、ターミナルで `ls` コマンドを実行して、リポジトリがディレクトリとして表示されることを確認します。

2.2. コマンドラインインターフェイスを使用した GIT リポジトリから既存のノートブックファイルのアップロード

コマンドラインインターフェイスを使用して Git リポジトリをワークスペースにクローンし、そのまま作業を継続したり、外部プロジェクトのファイルを統合したりできます。

前提条件

- 起動済みで実行中の Jupyter サーバー。

手順

1. Git リポジトリの HTTPS URL をコピーします。
 - GitHub で **Code** → **HTTPS** をクリックし、Clipboard ボタンをクリックします。
 - GitLab で **Clone** をクリックし、**Clone with HTTPS** の下にある Clipboard ボタンをクリックします。
2. JupyterLab で **File** → **New** → **Terminal** の順にクリックして、ターミナルウィンドウを開きます。
3. **git clone** コマンドを入力します。

```
git clone <git-clone-URL>
```

`<git-clone-URL>` は HTTPS URL に置き換えます。以下に例を示します。

```
[1234567890@jupyter-nb-jdoe ~]$ git clone https://github.com/example/myrepo.git
Cloning into myrepo...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 2821 (delta 1), reused 5 (delta 1), pack-reused 2810
Receiving objects: 100% (2821/2821), 39.17 MiB | 23.89 MiB/s, done.
Resolving deltas: 100% (1416/1416), done.
```

検証

- リポジトリの内容が JupyterLab のファイルブラウザーに表示されるか、ターミナルで **ls** コマンドを実行して、リポジトリがディレクトリとして表示されることを確認します。

2.3. リモート GIT リポジトリからの変更を含むプロジェクトの更新

他のユーザーが作成した変更をリモート Git リポジトリからデータサイエンスプロジェクトにプルできます。

前提条件

- リモート Git リポジトリを設定している。
- Git リポジトリを JupyterLab にインポートしており、リポジトリの内容が JupyterLab のファイルブラウザーに表示される。
- リモート Git リポジトリからローカルリポジトリにファイルをプルするパーミッションがある。
- Jupyter にログインするための認証情報がある。
- Jupyter サーバーを起動して実行している。

手順

1. JupyterLab インターフェイスで、**Git** ボタン () をクリックします。
2. **Pull latest changes** ボタン () をクリックします。

検証

- Git ペインの **History** タブで、リモートリポジトリからプルした変更を表示できます。


2.4. プロジェクトの変更を GIT リポジトリにプッシュ

実稼働環境でアプリケーションをビルドし、デプロイするには、作業をリモート Git リポジトリにアップロードします。

前提条件

- JupyterLab インターフェイスでノートブックを開いている。
- 関連する Git リポジトリがノートブックサーバーに追加されている。
- 関連する Git リポジトリに変更をプッシュするパーミッションがある。
- Git バージョン制御拡張がインストールされている。

手順

1. **File** → **Save All** をクリックして、保存していない変更を保存します。
2. Git アイコン () をクリックし、JupyterLab インターフェイスで Git ペインを開きます。
3. 変更したファイルが **Changed** に表示されることを確認します。
変更したファイルが **Untracked** にある場合は、**Git** → **Simple Staging** をクリックして簡素化された Git プロセスを有効にします。
4. 変更をコミットします。
 - a. **Changed** の下にあるすべてのファイルに、青いチェックマークが付いていることを確認します。
 - b. **Summary** フィールドに、加えた変更の簡単な説明を入力します。
 - c. **Commit** をクリックします。
5. **Git** → **Push to Remote** をクリックして、変更をリモートリポジトリにプッシュします。
6. プロンプトが表示されたら、Git 認証情報を入力し、**OK** をクリックします。

検証

- 最近プッシュされた変更は、リモート Git リポジトリに表示されます。

第3章 データサイエンスプロジェクトでの作業

データサイエンティストは、データサイエンスの作業を1つのプロジェクトにまとめることができます。OpenShift Data Science のデータサイエンスプロジェクトは、以下のコンポーネントで設定できます。

ワークベンチ

ワークベンチを作成すると、Jupyter ノートブックをプロジェクトに追加できます。

クラスターストレージ

データを保持する必要があるデータサイエンスプロジェクトの場合は、クラスターストレージをプロジェクトに追加できます。

データ接続

プロジェクトにデータ接続を追加すると、データ入力をワークベンチに接続できます。

Pipelines

機械学習ワークフローを標準化および自動化し、データサイエンスモデルをさらに強化してデプロイメントできるようにします。

モデルおよびモデルサーバー

トレーニング済みのデータサイエンスモデルをデプロイして、インテリジェントなアプリケーションを提供します。モデルは、アプリケーションがモデルにリクエストを送信できるようにするエンドポイントを使用してデプロイされます。



重要

OpenShift Data Science ユーザーインターフェイスの外で OpenShift プロジェクトを作成した場合、そのプロジェクトは **Data science projects** ページに表示されません。また、標準の OpenShift プロジェクトでワークベンチやモデル提供などの OpenShift Data Science 専用機能は使用できません。

OpenShift プロジェクトをデータサイエンスプロジェクトとして分類し、OpenShift Data Science 専用機能を使用可能にするには、プロジェクト namespace に **opendatahub.io/dashboard: 'true'** ラベルを追加する必要があります。このラベルを追加すると、プロジェクトが **Data science projects** ページに表示されます。

3.1. データサイエンスプロジェクトの使用

3.1.1. データサイエンスプロジェクトの作成

データサイエンスの作業を開始するには、データサイエンスプロジェクトを作成します。プロジェクトを作成すると、作業を1カ所で整理できます。ワークベンチの追加、プロジェクトのクラスターへのストレージの追加、データ接続の追加、およびモデルサーバーの追加によって、データサイエンスプロジェクトの機能を強化することもできます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。
Data science projects のページが開きます。
2. **Create data science project** をクリックします。
Create a data science project ダイアログが開きます。
3. データサイエンスプロジェクトの **name** を入力します。
4. オプション: データサイエンスプロジェクトの **リソース名** を編集します。リソース名は、小文字の英数字 - で設定され、英数字で開始および終了する必要があります。
5. データサイエンスプロジェクトの **説明** を入力します。
6. **Create** をクリックします。
プロジェクトの詳細ページが開きます。ここから、ワークベンチを作成し、クラスターストレージを追加し、プロジェクトにデータ接続を追加できます。

検証

- 作成したデータサイエンスプロジェクトが **Data science projects** ページに表示されます。

3.1.2. データサイエンスプロジェクトの更新

プロジェクトの名前と説明テキストを変更することで、データサイエンスプロジェクトの詳細を更新できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- データサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。
Data science projects のページが開きます。
2. 詳細を更新するプロジェクトの横にあるアクションメニュー (⋮) をクリックし、**Edit project** をクリックします。
Edit data science project ダイアログが開きます。
3. オプション: データサイエンスプロジェクトの **name** を更新します。
4. オプション: データサイエンスプロジェクトの **説明** を更新します。
5. **Update** をクリックします。

検証

- 更新したデータサイエンスプロジェクトが **Data science projects** ページに表示されます。

3.1.3. データサイエンスプロジェクトの削除

不要になったプロジェクトが OpenShift Data Science **Data science projects** ページに表示されないように、データサイエンスプロジェクトを削除できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users** など) に属している。
- データサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. 削除するプロジェクトの横にあるアクションメニュー (⋮) をクリックし、**Delete project** をクリックします。**Delete project** ダイアログが開きます。
3. テキストフィールドにプロジェクトの名前を入力して、プロジェクトを削除することを確認します。
4. **プロジェクトの削除** をクリックします。

検証

- 削除したデータサイエンスプロジェクトは、**Data science projects** ページに表示されなくなります。
- データサイエンスプロジェクトを削除すると、関連するすべてのワークベンチ、クラスターストレージ、およびデータ接続が削除されます。このデータは完全に削除され、復元できません。

3.2. プロジェクトワークベンチの使用

3.2.1. プロジェクトワークベンチの作成

孤立した領域でデータモデルを調べて操作するために、ワークベンチを作成できます。このワークベンチを使用すると、既存のノートブックコンテナイメージから新しい Jupyter ノートブックを作成して、そのリソースとプロパティにアクセスできます。データの保持が必要なデータサイエンスプロジェクトの場合は、作成中のワークベンチにコンテナストレージを追加できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを追加できるデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. ワークベンチを追加するプロジェクトの名前をクリックします。プロジェクトの **Details** ページが開きます。
3. **Workbenches** セクションで **Create workbench** をクリックします。**Create workbench** ページが開きます。
4. 作成するワークベンチのプロパティを設定します。
 - a. ワークベンチの **name** を入力します。
 - b. ワークベンチの **description** を入力します。
 - c. ワークベンチサーバーに使用する **notebook image** を選択します。
 - d. サーバーの **container size** を選択します。
 - e. オプション: 新規の **環境変数** の値を選択し、指定します。
 - f. OpenShift Data Science クラスターのストレージを設定します。
 - i. OpenShift Data Science からログアウトした後に保持されるストレージを作成するには、**Create new persistent storage** を選択します。関連するフィールドに入力してストレージを定義します。
 - ii. 既存のストレージを再利用するには、**Use existing persistent storage** を選択し、**Persistent storage** リストからストレージを選択します。
5. **Create workbench** をクリックします。

検証

- 作成したワークベンチがプロジェクトの **Details** ページに表示されます。
- 作成プロセス中にワークベンチに関連付けたクラスターストレージは、プロジェクトの **Details** ページに表示されます。
- **Details** ページの **Workbenches** セクションにある **Status** 列には、ワークベンチサーバーが起動している場合は **Starting** のステータスが表示され、ワークベンチが正常に起動した場合は **Running** のステータスが表示されます。

3.2.2. ワークベンチの開始

プロジェクトの **Details** ページから、データサイエンスプロジェクトのワークベンチを手動で開始できます。デフォルトでは、ワークベンチは作成後すぐに起動します。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. ワークベンチを開始するプロジェクトの名前をクリックします。プロジェクトの **Details** ページが開きます。
3. 関連するワークベンチの **Status** 列のトグルをクリックして、実行されていないワークベンチを起動します。
起動したワークベンチのステータスが **Stopped** から **Running** に変わります。ワークベンチが起動したら、**Open** をクリックしてワークベンチのノートブックを開きます。

検証

- 起動したワークベンチは、プロジェクトの **Details** ページに **Running** のステータスで表示されます。

3.2.3. プロジェクトワークベンチの更新

データサイエンスの作業で、ワークベンチのノートブックイメージ、コンテナサイズ、または識別情報を変更する必要がある場合は、プロジェクトのワークベンチのプロパティを変更できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. ワークベンチを更新するプロジェクトの名前をクリックします。プロジェクトの **Details** ページが開きます。
3. **Workbenches** セクションで更新するワークベンチの横にあるアクションメニュー (⋮) をクリックし、**Edit workbench** をクリックします。**Edit workbench** ページが開きます。
4. ワークベンチのプロパティを更新します。
 - a. 該当する場合は、ワークベンチの **name** を更新します。
 - b. 該当する場合は、ワークベンチの **description** を更新します。
 - c. 該当する場合は、ワークベンチサーバーに使用する新しい **notebook image** を選択します。
 - d. 該当する場合は、サーバーの新しい **container size** を選択します。
5. **Update workbench** をクリックします。

検証

- 更新したワークベンチがプロジェクトの **Details** ページに表示されます。

3.2.4. データサイエンスプロジェクトからのワークベンチの削除

データサイエンスプロジェクトからワークベンチを削除して、作業に関連しなくなった Jupyter ノートブックを削除できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを使用してデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. ワークベンチを削除するプロジェクトの名前をクリックします。プロジェクトの **Details** ページが開きます。
3. **Workbenches** セクションで削除するワークベンチの横にあるアクションメニュー (⋮) をクリックし、**Delete workbench** をクリックします。**Delete workbench** ダイアログが開きます。
4. ワークベンチの名前をテキストフィールドに入力して、削除することを確認します。
5. **Delete workbench** をクリックします。

検証

- 削除したワークベンチは、プロジェクトの **Details** ページの **Workbenches** セクションに表示されなくなります。
- ワークベンチの Jupyter ノートブックに関連付けられているカスタムリソース (CR) が削除されます。

3.3. データ接続の使用

3.3.1. データサイエンスプロジェクトへのデータ接続の追加

データソースへの接続を追加することで、データサイエンスプロジェクトを強化できます。非常に大きなデータセットを使用する場合は、データを Amazon Web Services (AWS) Simple Storage Service (S3) バケットに保存し、ローカルストレージがいっぱいにならないようにできます。まだ接続されていない既存のワークベンチにデータ接続を関連付けるオプションもあります。

前提条件

- Red Hat OpenShift Data Science にログインしている。

- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- データ接続を追加できるデータサイエンスプロジェクトを作成している。
- 既存のワークベンチにデータ接続を追加する場合は、作業が失われないように、データをワークベンチに保存している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. データ接続を追加するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. ページの **Data connections** セクションで、**Add data connection** をクリックします。**Add data connection** ダイアログが開きます。
4. データ接続の **name** を入力します。
5. **AWS_ACCESS_KEY_ID** フィールドに、Amazon Web Services のアクセスキー ID を入力します。
6. **AWS_SECRET_ACCESS_KEY_ID** フィールドで指定したアカウントのシークレットアクセスキーを入力します。
7. **AWS_S3_ENDPOINT** フィールドに AWS S3 ストレージのエンドポイントを入力します。
8. **AWS_DEFAULT_REGION** フィールドに、AWS アカウントのデフォルトリージョンを入力します。
9. **AWS_S3_BUCKET** フィールドに AWS S3 バケットの名前を入力します。
10. **Add data connection** をクリックします。

検証

- 追加したデータ接続は、プロジェクトの **Details** ページの **Data connections** セクションに表示されます。
- ワークベンチを選択した場合、データ接続はデータサイエンスプロジェクトページの **Workbenches** セクションに表示されます。

3.3.2. データ接続の削除

データサイエンスプロジェクトからデータ接続を削除して、作業に関連しなくなった接続を削除できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。

- データ接続を使用してデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. データ接続を削除するプロジェクトの名前をクリックします。プロジェクトの **Details** ページが開きます。
3. **Data connections** セクションで削除するデータ接続の横にあるアクションメニュー (⋮) をクリックし、**Delete data connection** をクリックします。**Delete data connection** ダイアログが開きます。
4. テキストフィールドにデータ接続の名前を入力して、削除することを確認します。
5. **Delete data connection** をクリックします。

検証

- 削除したデータ接続は、プロジェクトの **Details** ページの **Data connections** セクションに表示されなくなります。

3.3.3. 接続されたデータソースの更新

別のワークベンチで既存のデータソースを使用するには、プロジェクトのワークベンチに接続されているデータソースを変更できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- データサイエンスプロジェクトを作成し、ワークベンチを作成し、データ接続を定義している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. データソースを変更するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Data connections** セクションで変更するデータソースの横にあるアクションメニュー (⋮) をクリックし、**Change connected workbenches** をクリックします。**Update connected workbenches** ダイアログが開きます。
4. データソースを接続する既存の **workbench** をリストから選択します。
5. **Update connected workbenches** をクリックします。

検証

- 変更したデータ接続は、プロジェクトの **Details** ページの **Data connections** セクションに表示されます。
- 接続されたワークベンチで環境変数を使用して S3 データソースにアクセスできます。

3.4. クラスタストレージの設定

3.4.1. データサイエンスプロジェクトへのクラスタストレージの追加

データを保持する必要があるデータサイエンスプロジェクトの場合は、クラスタストレージをプロジェクトに追加できます。さらに、クラスタストレージを特定のプロジェクトのワークベンチに接続することもできます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- クラスタストレージを追加できるデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. クラスタストレージを追加するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. ページの **Cluster storage** セクションで、**Add cluster storage** をクリックします。**Add storage** ダイアログが開きます。
4. クラスタストレージの **name** を入力します。
5. クラスタストレージの **description** を入力します。
6. **Persistent storage size** で、新しいサイズをギビバイト単位で入力します。最小サイズは 1 GiB、最大サイズは 16384 GiB です。
7. オプション: リストから **ワークベンチ** を選択して、クラスタストレージを既存のワークベンチに接続します。
8. ストレージを接続するワークベンチを選択した場合は、**Mount folder** フィールドにストレージディレクトリーを入力します。
9. **Add storage** をクリックします。

検証

- 追加したクラスタストレージは、プロジェクトの **Details** ページの **Cluster storage** セクションに表示されます。
- 定義したストレージサイズで、新しい永続ボリューム要求 (PVC) が作成されます。

- 永続ボリューム要求 (PVC) は、プロジェクトの **Details** ページの **Workbenches** セクションに接続されたストレージとして表示されます。

3.4.2. クラスタストレージの更新

データサイエンスの作業で、プロジェクトのクラスタストレージまたはストレージが接続されているワークベンチの識別情報を変更する必要がある場合は、プロジェクトのクラスタストレージを更新してこれらのプロパティを変更できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- クラスタストレージを含むデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. ストレージを更新するプロジェクトの名前をクリックします。プロジェクトの **Details** ページが開きます。
3. **Cluster storage** セクションで更新するストレージの横にあるアクションメニュー (⋮) をクリックし、**Edit storage** をクリックします。**Edit storage** ページが開きます。
4. ストレージのプロパティを更新します。
 - a. 該当する場合は、ストレージの **name** を更新します。
 - b. 該当する場合は、ストレージの **description** を更新します。
 - c. 必要に応じて、ストレージの **Persistent storage size** を増やします。
増加できるのはストレージサイズのみであることに注意してください。ストレージサイズを更新すると、ワークベンチが再起動され、通常はサイズの変更に比例して一定期間使用できなくなります。
 - d. 該当する場合は、ストレージが接続されている **ワークベンチ** を更新します。
 - e. ストレージを接続する新しいワークベンチを選択した場合は、**Mount folder** フィールドにストレージディレクトリーを入力します。
5. **Update storage** をクリックします。

ストレージサイズを増やした場合、ワークベンチが再起動され、通常はサイズの変更に比例して一定期間使用できなくなります。

検証

- 更新したストレージは、プロジェクトの **Details** ページの **Cluster storage** セクションに表示されます。

3.4.3. データサイエンスプロジェクトからのクラスターストレージの削除

データサイエンスプロジェクトからクラスターストレージを削除すると、リソースを解放し、不要なストレージスペースを削除できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- クラスターストレージを使用してデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. ストレージを削除するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Cluster storage** セクションで、削除するストレージの横にあるアクションメニュー (⋮) をクリックし、**Delete storage** をクリックします。**Delete storage** ダイアログが開きます。
4. テキストフィールドにストレージの名前を入力して、削除することを確認します。
5. **Delete storage** をクリックします。

検証

- 削除したストレージは、プロジェクトの **Details** ページの **Cluster storage** セクションに表示されなくなります。
- クラスターストレージに関連付けられている永続ボリューム (PV) と永続ボリューム要求 (PVC) は両方とも完全に削除されます。このデータは回復できません。

3.5. データサイエンスパイプラインの設定

3.5.1. パイプラインサーバーの設定

OpenShift Data Science でパイプラインを正常に作成するには、パイプラインサーバーを設定する必要があります。これには、パイプラインアーティファクトとデータの保存場所の設定が含まれます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを追加できるデータサイエンスプロジェクトが作成されている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。
Data science projects のページが開きます。
2. パイプラインサーバーを設定するプロジェクトの名前をクリックします。
プロジェクトの詳細ページが開きます。
3. **Pipelines** セクションで、**Create a pipeline server** をクリックします。
Configure pipeline server ダイアログが表示されます。
4. **Object storage connection** セクションで、パイプラインアーティファクトを保存するための S3 互換データ接続を指定するには、次のアクションセットのいずれかを選択します。



注記

パイプラインサーバーの作成後、**/metadata** フォルダと **/artifacts** フォルダがデフォルトの **root** フォルダに自動的に作成されます。したがって、パイプラインサーバーのデータ接続を設定するときに、ストレージディレクトリーを指定する必要はありません。

- **Existing data connection** を選択して、以前に定義したデータ接続を使用します。このオプションを選択した場合は、**Name** リストから関連するデータ接続の名前を選択し、手順 6 に進みます。
 - **Create new data connection** を選択して、パイプラインサーバーがアクセスできる新しいデータ接続を追加します。
5. **Create new data connection** を選択した場合は、次の手順を実行します。
 - a. **Name** フィールドに、データ接続の名前を入力します。
 - b. **AWS_ACCESS_KEY_ID** フィールドに、Amazon Web Services のアクセスキー ID を入力します。
 - c. **AWS_SECRET_ACCESS_KEY_ID** フィールドで指定したアカウントのシークレットアクセスキーを入力します。
 - d. オプション: **AWS_S3_ENDPOINT** フィールドに、AWS S3 ストレージのエンドポイントを入力します。
 - e. オプション: **AWS_DEFAULT_REGION** フィールドに、AWS アカウントのデフォルトのリージョンを入力します。
 - f. **AWS_S3_BUCKET** フィールドに AWS S3 バケットの名前を入力します。



重要

新しいデータ接続を作成する場合は、指定された他の必須フィールドに加えて、**AWS_S3_BUCKET** フィールドも必須です。間違ったデータ接続設定を指定した場合は、同じパイプラインサーバー上でこれらの設定を更新できません。したがって、パイプラインサーバーを削除し、別のパイプラインサーバーを設定する必要があります。

6. **Databases** セクションで、**Show advanced database options** をクリックしてパイプラインサーバー

- b. **Database** セクションで、**snw advanced database options** をクリックしてハイノファイルータを保存するデータベースを指定し、次の一連のアクションのいずれかを選択します。
 - プロジェクトに MariaDB データベースをデプロイするには、**Use default database stored on your cluster** を選択します。
 - **Connect to external MySQL database** を選択して、パイプラインサーバーがアクセスできる外部データベースに新しい接続を追加します。
 - i. **Host** フィールドにデータベースのホスト名を入力します。
 - ii. **Port** フィールドにデータベースのポートを入力します。
 - iii. **Username** フィールドに、データベースに接続するデフォルトのユーザー名を入力します。
 - iv. **Password** フィールドに、デフォルトのユーザーアカウントのパスワードを入力します。
 - v. **Database** フィールドにデータベース名を入力します。
7. **Configure** をクリックします。

検証

- 設定したパイプラインサーバーは、Project details ページの **Pipelines** セクションに表示されません。
- **Import pipeline** ボタンは、Project details ページの **Pipelines** セクションで使用できます。

3.5.2. パイプラインの定義

Kubeflow Pipelines SDK を使用すると、エンドツーエンドの機械学習とデータパイプラインを定義できます。Kubeflow Pipelines SDK を使用して、Python コードでデータサイエンスパイプラインを構築します。パイプラインを構築したら、kfp-tekton SDK (バージョン 1.5.x のみ) を使用してパイプラインを Tekton 形式の YAML コードにコンパイルします。パイプラインを定義した後、YAML ファイルを OpenShift Data Science ダッシュボードにインポートして、その実行設定を設定できるようにします。Tekton への Kubeflow Pipelines SDK のインストールおよび使用に関する詳細は、[Kubeflow Pipelines SDK for Tekton](#) を参照してください。

Elyra JupyterLab 拡張機能を使用して、JupyterLab 内でデータサイエンスパイプラインを作成して実行することもできます。JupyterLab でのパイプラインの作成の詳細は、[JupyterLab でのパイプラインの操作](#) を参照してください。Elyra JupyterLab 拡張機能の詳細は、[Elyra ドキュメンテーション](#) を参照してください。

関連情報

- [Kubeflow Pipelines SDK for Tekton](#)
- [KFP Tekton samples and compiler samples](#)
- [Kubeflow Pipelines v1 Documentation](#)
- [Elyra ドキュメンテーション](#)

3.5.3. データサイエンスパイプラインのインポート

OpenShift Data Science で簡単にデータサイエンスパイプラインの操作を開始できるように、パイプラインのコードを含むYAML ファイルをアクティブなパイプラインサーバーにインポートできます。このファイルには、Tekton コンパイラーでコンパイルされたKubeflowパイプラインが含まれています。パイプラインをパイプラインサーバーにインポートしたら、パイプライン実行を作成してパイプラインを実行できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、パイプラインをインポートするプロジェクトを選択します。
3. **Import pipeline** をクリックします。
Import pipeline ダイアログが開きます。
4. インポートするパイプラインの詳細を入力します。
 - a. **Pipeline name** フィールドに、インポートするパイプラインの名前を入力します。
 - b. **Pipeline description** フィールドに、インポートするパイプラインの説明を入力します。
 - c. **Upload** をクリックします。または、ローカルマシンのファイルシステムからファイルをドラッグし、**Import pipeline** ダイアログの指定された領域にドロップします。
ファイルブラウザーが開きます。
 - d. パイプラインコードを含むファイルに移動し、**Select** をクリックします。
 - e. **Import pipeline** をクリックします。

検証

- インポートしたパイプラインが **Pipelines** ページに表示されます。

OpenShift Data Science でのパイプラインの使用の詳細は、[データサイエンスパイプラインの使用](#) を参照してください。

3.6. モデルサーバーの設定

3.6.1. データサイエンスプロジェクトにモデルサーバーを追加する

OpenShift Data Science にデータサイエンスモデルを正常にデプロイするには、モデルサーバーを設定する必要があります。これには、デプロイされるレプリカの数、サーバーのサイズ、トークンの承認、およびプロジェクトへのアクセス方法の設定が含まれます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- モデルサーバーを追加できるデータサイエンスプロジェクトを作成している。
- モデルサーバーにカスタムのモデル提供ランタイムを使用する場合は、ランタイムを追加して有効にしている。[カスタムモデル提供ランタイムの追加](#) を参照してください。
- モデルサーバーでグラフィックスプロセッシングユニット (GPU) を使用する場合は、OpenShift Data Science で GPU サポートを有効にしている。[OpenShift Data Science での GPU サポートの有効化](#) を参照してください。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. モデルサーバーを設定するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Models and model servers** セクションで、**Add server** をクリックします。**Add model server** ダイアログが開きます。
4. **Model sever name** フィールドに、モデルサーバーの一意的名前を入力します。
5. **Serving runtime** リストから、OpenShift Data Science デプロイメントにインストールされ有効になっているモデル提供ランタイムを選択します。
6. **Number of model replicas to deploy** フィールドに値を指定します。
7. **Model server size** リストから、次のいずれかのサーバーサイズを選択します。
 - Small
 - Medium
 - Large
 - Custom
8. オプション: 前の手順で **Custom** を選択した場合は、**Model server size** セクションで次を設定して、モデルサーバーをカスタマイズします。
 - a. **CPUs requested** フィールドで、モデルサーバーで使用する CPU の数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。
 - b. **CPU limit** フィールドに、モデルサーバーで使用する CPU の最大数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。

- c. **Memory requested** フィールドに、モデルサーバーに要求されたメモリーをギビバイト (Gi) 単位で指定します。
 - d. **Memory limit** フィールドに、モデルサーバーの最大メモリー制限をギビバイト (Gi) 単位で指定します。
9. オプション: **Model server GPUs** フィールドで、モデルサーバーで使用する GPU の数を指定します。



重要

OpenShift Data Science には、デフォルトで2つのバージョンの OpenVINO Model Server (OVMS) ランタイムが含まれています。GPU をサポートするバージョンとサポートしないバージョン。GPU を使用するには、**Serving runtime** リストから、表示名に **Supports GPUs** が含まれるバージョンを選択する必要があります。

モデルサーバーで **カスタム** モデル提供ランタイムを使用している場合は、カスタムランタイムが GPU をサポートし、GPU を使用するように適切に設定されていることを確認する必要があります。

10. オプション: **Model route** セクションで、**Make deployed models available through an external route** チェックボックスをオンにして、デプロイされたモデルを外部クライアントが利用できるようにします。
11. オプション: **Token authorization** セクションで、**Require token authentication** チェックボックスをオンにして、モデルサーバーにトークン認証を要求します。トークン認証の設定を完了するには、次のアクションを実行します。
 - a. **Service account name** フィールドに、トークンが生成されるサービスアカウント名を入力します。生成されたトークンは、モデルサーバーの設定時に作成され、**Token secret** フィールドに表示されます。
 - b. 追加のサービスアカウントを追加するには、**Add a service account** をクリックし、別のサービスアカウント名を入力します。
12. **Add** をクリックします。

検証

- 設定したモデルサーバーは、プロジェクトの詳細ページの **Models and model servers** セクションに表示されます。

3.6.2. カスタムのモデル提供ランタイムの追加

モデル提供ランタイムは、指定されたモデルサーバーおよびそれがサポートするモデルフレームワークとの統合を提供します。デフォルトでは、Red Hat OpenShift Data Science には OpenVINO Model Server ランタイムが含まれています。ただし、このランタイムがニーズを満たさない場合 (たとえば、特定のモデルフレームワークをサポートしていない場合) は、独自のカスタムランタイムを追加することもできます。

管理者は、OpenShift Data Science インターフェイスを使用して、カスタムのモデル提供ランタイムを追加および有効化できます。新しいモデルサーバーを作成するときに、有効なランタイムから選択できます。

前提条件

- OpenShift Data Science に管理者としてログインしている。
- [モデルサーバーをプロジェクトに追加する](#) 方法を理解していること。カスタムのモデル提供ランタイムを追加した場合は、そのランタイムを使用するように新しいモデルサーバーを設定する必要があります。
- [kserve/modelmesh-serving](#) リポジトリ内のサンプルランタイムを確認している。これらの例を **開始点** として使用できます。ただし、各ランタイムを OpenShift Data Science にデプロイするには、さらにいくつかの変更が必要です。必要な変更は、次の手順で説明します。



注記

OpenShift Data Science には、OpenVINO Model Server モデル提供ランタイムがデフォルトで含まれています。このランタイムを OpenShift Data Science に追加する必要はありません。

手順

1. OpenShift Data Science ダッシュボードから、**Settings > Serving runtimes** をクリックします。
Serving runtimes ページが開き、OpenShift Data Science デプロイメントにすでにインストールされ有効になっているモデル提供ランタイムが表示されます。デフォルトでは、OpenVINO Model Server ランタイムがプリインストールされ、OpenShift Data Science で有効になっています。
2. 新しいカスタムランタイムを追加するには、**Add serving runtime** をクリックします。
Add serving runtime ページが開きます。
3. 新しいランタイムの追加を開始するには、次の一連のアクションのいずれかを実行します。
 - **YAML ファイルをアップロードするには**
 - a. **Upload files** をクリックします。
ファイルブラウザが開きます。
 - b. ファイルブラウザで、コンピューター上の YAML ファイルを選択します。このファイルは、[kserve/modelmesh-serving](#) リポジトリからダウンロードしたサンプルランタイムの1つである可能性があります。
埋め込み YAML エディターが開き、アップロードしたファイルの内容が表示されます。
 - **エディターに YAML コードを直接入力するには**
 - a. **Start from scratch** をクリックします。
埋め込み YAML エディターがコンテンツなしで開きます。
 - b. 埋め込みエディターに YAML コードを直接入力または貼り付けます。貼り付ける YAML は、[kserve/modelmesh-serving](#) リポジトリ内のサンプルランタイムの1つからコピーされる場合があります。
4. オプション: [kserve/modelmesh-serving](#) リポジトリにサンプルランタイムの1つを追加する場合は、次の変更を実行します。
 - a. YAML エディターで、ランタイムの **kind** フィールドを見つけます。このフィールドの値を **ServingRuntime** に更新します。

- b. YAML エディターで、ランタイムの **containers.image** フィールドを見つけます。追加するランタイムに基づいて、フィールドを次のいずれかに更新します。

Nvidia Triton Inference Server

image: nvcr.io/nvidia/tritonserver:21.06.1-py3

Seldon Python MLServer

image: seldonio/mlserver:0.5.2

TorchServe

image: pytorch/torchserve:0.6.0-cpu

5. **metadata.name** フィールドで、追加するランタイムの値が一意であることを確認します (つまり、値がすでに追加したランタイムの値と同じではありません)。
6. オプション: 追加するランタイムのカスタム表示名を設定するには、次の例に示すように、**metadata.annotations.openshift.io/display-name** フィールドを追加し、値を指定します。

```
apiVersion: serving.kserve.io/v1alpha1
kind: ServingRuntime
metadata:
  name: mlserver-0.x
  annotations:
    openshift.io/display-name: MLServer
```



注記

ランタイムのカスタム表示名を設定しないと、OpenShift Data Science には **metadata.name** フィールドの値が表示されます。

7. **Add** をクリックします。
Serving runtimes ページが開き、インストールされているランタイムの更新されたリストが表示されます。追加したランタイムが自動的に有効になることを確認します。
8. オプション: カスタムランタイムを編集するには、アクションメニュー (:) をクリックし、**Edit** を選択します。



注記

デフォルトで OpenShift Data Science に含まれる OpenVINO Model Server ランタイムを直接編集できません。ただし、このランタイムを複製し、複製したバージョンの編集は可能です。その後、編集したクローンを新しいカスタムランタイムとして追加できます。これを行うには、OpenVINO Model Server の横にあるアクションメニューをクリックし、**Clone** を選択します。

検証

- 追加したモデル提供ランタイムは、**サービス提供ランタイム** ページに有効な状態で表示されません。

関連情報

- 追加したカスタムモデルサービスランタイムを使用するモデルサーバーを設定する方法については、[データサイエンスプロジェクトへのモデルサーバーの追加](#) を参照してください。

3.6.3. モデルサーバーの更新

デプロイされたレプリカの数、サーバーのサイズ、トークン認証、プロジェクトへのアクセス方法などの詳細を変更することで、データサイエンスプロジェクトのモデルサーバーを更新できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- モデルサーバーが割り当てられたデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. モデルサーバーの詳細を更新するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Models and model servers** セクションで、更新するモデルサーバーを見つけます。アクションメニュー (⋮) をクリックし、**Edit model server** を選択します。**Configure model server** ダイアログが開きます。
4. 次のように、モデルサーバーのプロパティを更新します。



注記

すでに設定されているモデルサーバーの **提供ランタイム** の選択を変更できません。これにより、すでにデプロイされたモデルをサポートしないランタイムへの変更が防止されます。

- a. **Model server name** フィールドに、モデルサーバーの新しい一意の名前を入力します。
- b. **Number of model replicas to deploy** フィールドに値を指定します。
- c. **Model server size** リストから、次のいずれかのサーバーサイズを選択します。
 - Small
 - Medium
 - Large
 - Custom
- d. オプション: 前の手順で **Custom** を選択した場合は、**Model server size** セクションで次を設定して、モデルサーバーをカスタマイズします。
 - i. **CPUs requested** フィールドで、モデルサーバーで使用する CPU の数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。
 - ii. **CPU limit** フィールドに、モデルサーバーで使用する CPU の最大数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。

- iii. **Memory requested** フィールドに、モデルサーバーに要求されたメモリーをギビバイト (Gi) 単位で指定します。
- iv. **Memory limit** フィールドに、モデルサーバーの最大メモリー制限をギビバイト (Gi) 単位で指定します。
- e. オプション: **Model server GPUs** フィールドで、モデルサーバーで使用する GPU の数を指定します。



重要

OpenShift Data Science には、デフォルトで 2 つのバージョンの OpenVINO Model Server (OVMS) ランタイムが含まれています。GPU をサポートするバージョンとサポートしないバージョン。GPU を使用するには、**Serving runtime** リストから、表示名に **Supports GPUs** が含まれるバージョンを選択する必要があります。

モデルサーバーで **カスタム** モデル提供ランタイムを使用している場合は、カスタムランタイムが GPU をサポートし、GPU を使用するように適切に設定されていることを確認する必要があります。

- f. オプション: **Model route** セクションで、**Make deployed models available through an external route** チェックボックスをオンにして、デプロイされたモデルを外部クライアントが利用できるにします。
 - g. オプション: **Token authorization** セクションで、**Require token authentication** チェックボックスをオンにして、モデルサーバーにトークン認証を要求します。トークン認証の設定を完了するには、次のアクションを実行します。
 - i. **Service account name** フィールドに、トークンが生成されるサービスアカウント名を入力します。生成されたトークンは、モデルサーバーの設定時に作成され、**Token secret** フィールドに表示されます。
 - ii. 追加のサービスアカウントを追加するには、**Add a service account** をクリックし、別のサービスアカウント名を入力します。
5. **Configure** をクリックします。

検証

- 更新したモデルサーバーは、Project Detail ページの **Models and model servers** セクションに表示されます。

3.6.4. モデルサーバーの削除

モデルをホストするためにモデルサーバーが必要なくなった場合は、データサイエンスプロジェクトからモデルサーバーを削除できます。



注記

モデルサーバーを削除すると、そのモデルサーバーでホストされているモデルも削除されます。その結果、アプリケーションはモデルを利用できなくなります。

前提条件

- データサイエンスプロジェクトと関連するモデルサーバーを作成している。
- モデルにアクセスするアプリケーションのユーザーに、モデルが利用できなくなることを通知している。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. モデルサーバーを削除するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Models and model servers** セクションでモデルサーバーを削除するプロジェクトの横にあるアクションメニュー (⋮) をクリックし、**Delete model server** をクリックします。**Delete model server** ダイアログが開きます。
4. テキストフィールドにモデルサーバーの名前を入力して、削除することを確認します。
5. **Delete model server** をクリックします。

検証

- 削除したモデルサーバーは、プロジェクトの Details ページの **Models and model servers** セクションに表示されなくなります。

モデル提供の詳細は、**OpenShift Data Science** での**モデル提供**を参照してください。

3.7. データサイエンスプロジェクトへのアクセスの設定

3.7.1. データサイエンスプロジェクトへのアクセスの設定

データサイエンスプロジェクトで他のユーザーと共同作業できるようにするために、プロジェクトへのアクセスを共有できます。プロジェクトを作成した後、OpenShift Data Science ユーザーインターフェイスから適切なアクセス許可を設定できます。

次のアクセス許可レベルをデータサイエンスプロジェクトに割り当てることができます。

- Admin: 詳細 (プロジェクト名と説明)、コンポーネント、アクセス許可など、プロジェクトのすべての領域を変更できます。
- Edit: ワークベンチなどのプロジェクトのコンポーネントを変更できますが、プロジェクトのアクセス権限や詳細 (プロジェクト名と説明) を編集できません。


3.7.2. データサイエンスプロジェクトへのアクセス共有

組織が共同作業できるようにするために、データサイエンスプロジェクトへのアクセスを他のユーザーやグループと共有できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- データサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. データサイエンスプロジェクトのリストから、アクセスを共有するデータサイエンスプロジェクトの名前をクリックします。
プロジェクトの詳細ページが開きます。
3. **Permissions** タブをクリックします。
プロジェクトの **Permissions** ページが開きます。
4. 1人以上のユーザーにプロジェクトへのアクセス権を付与します。
 - a. **Users** セクションで、**Add plan** をクリックします。
 - b. **Name** フィールドに、プロジェクトへのアクセスを許可するユーザーのユーザー名を入力します。
 - c. **Permissions** リストから、次のいずれかのアクセス権レベルを選択します。
 - **管理者**: このアクセスレベルを持つユーザーは、プロジェクトの詳細を編集し、プロジェクトへのアクセスを管理できます。
 - **編集**: このアクセスレベルを持つユーザーは、ワークベンチ、データ接続、ストレージなどのプロジェクトコンポーネントを表示および編集できます。
 - d. 入力内容を確認するには、**Confirm** () をクリックします。
 - e. オプション: さらにユーザーを追加するには、**Add user** をクリックしてプロセスを繰り返します。
5. 1つ以上の OpenShift グループにプロジェクトへのアクセスを提供します。
 - a. **Groups** セクションで、**Add group** をクリックします。
 - b. **Name** リストから、プロジェクトへのアクセスを提供するグループを選択します。



注記

cluster-admin 権限または **dedicated-admin** 権限がない場合は、**名前** リストが表示されません。代わりに、グループ権限を設定できる入力フィールドが表示されます。

- c. **Permissions** リストから、次のいずれかのアクセス権レベルを選択します。
 - **管理者**: このアクセス許可レベルを持つグループは、プロジェクトの詳細を編集し、プロジェクトへのアクセスを管理できます。

- **編集:** このアクセス許可レベルを持つグループは、ワークベンチ、データ接続、ストレージなどのプロジェクトコンポーネントを表示および編集できます。

d. 入力内容を確認するには、**Confirm** () をクリックします。

e. オプション: さらにグループを追加するには、**グループの追加** をクリックしてプロセスを繰り返します。

検証

- プロジェクトへのアクセスを許可されたユーザーは、アクセス許可レベルで許可されたアクションのみを実行できます。
- **Permissions** タブの **Users** と **Groups** セクションには、プロジェクトへのアクセス権を付与したそれぞれのユーザーとグループが表示されます。

3.7.3. データサイエンスプロジェクトへのアクセスの更新

データサイエンスプロジェクトのコラボレーションのレベルを変更するには、プロジェクトにアクセスできるユーザーとグループのアクセス権限を更新します。


前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- データサイエンスプロジェクトを作成している。
- 以前にプロジェクトへのアクセスを他のユーザーまたはグループと共有している。
- 管理者権限を持っているか、プロジェクト所有者である。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. アクセス権限を変更するプロジェクト名をクリックします。プロジェクトの詳細ページが開きます。
3. **Permissions** タブをクリックします。プロジェクトの **Permissions** ページが開きます。
4. プロジェクトに対するユーザーのアクセス権限を更新します。
 - a. **Name** フィールドで、プロジェクトへのアクセスを許可するユーザーのユーザー名を更新します。
 - b. **Permissions** リストから、次のいずれかを選択してユーザーのアクセス権限を更新します。
 - **管理者:** このアクセスレベルを持つユーザーは、プロジェクトの詳細を編集し、プロジェクトへのアクセスを管理できます。

- **編集:** このアクセスレベルを持つユーザーは、ワークベンチ、データ接続、ストレージなどのプロジェクトコンポーネントを表示および編集できます。

c. エントリーの更新を確認するには、**Confirm** () をクリックします。

5. プロジェクトへの OpenShift グループのアクセス権限を更新します。

a. **Name** リストから別のグループを選択して、プロジェクトにアクセスできるグループを更新します。




注記

cluster-admin 権限または **dedicated-admin** 権限がない場合は、**名前** リストが表示されません。代わりに、グループ権限を設定できる入力フィールドが表示されます。

b. **Permissions** リストから、次のいずれかを選択してグループのアクセス権限を更新します。

- **管理者:** このアクセス許可レベルを持つグループは、プロジェクトの詳細を編集し、プロジェクトへのアクセスを管理できます。
- **編集:** このアクセス許可レベルを持つグループは、ワークベンチ、データ接続、ストレージなどのプロジェクトコンポーネントを表示および編集できます。

c. エントリーの更新を確認するには、**Confirm** () をクリックします。

検証

- **Permissions** タブの **Users** と **Groups** セクションには、プロジェクトのアクセス権限を変更したそれぞれのユーザーとグループが表示されます。

3.7.4. データサイエンスプロジェクトへのアクセスの削除

データサイエンスプロジェクトで共同作業する必要がなくなった場合は、以前にプロジェクトへのアクセスを許可したユーザーとグループを削除することで、プロジェクトへのアクセスを制限できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- データサイエンスプロジェクトを作成している。
- 以前にプロジェクトへのアクセスを他のユーザーまたはグループと共有している。
- 管理者権限を持っているか、プロジェクト所有者である。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。

2. アクセス権を変更するプロジェクト名をクリックします。
プロジェクトの詳細ページが開きます。
3. **Permissions** タブをクリックします。
プロジェクトの **Permissions** ページが開きます。
4. アクセス権を取り消すユーザーまたはグループの横にあるアクションメニュー (⋮) をクリックし、**Delete** をクリックします。

検証

- アクセスを取り消されたユーザーは、アクセス許可レベルで許可されていたアクションを実行できなくなります。

3.8. ノートブックサーバーにインストールされている PYTHON パッケージの表示

ノートブックセルで **pip** ツールを実行して、ノートブックサーバーにインストールされている Python パッケージおよびパッケージのバージョンを確認できます。

前提条件

- Jupyter にログインし、ノートブックを開いている。

手順

1. ノートブックの新しいセルに、以下のコマンドを入力します。

```
!pip list
```

2. セルを実行します。

検証

- 出力には、インストールされているすべての Python パッケージとそのバージョンがアルファベット順に表示されます。たとえば、**Minimal** イメージを使用してノートブックサーバーを作成した後すぐにこのコマンドを使用する場合に、表示される最初のパッケージは以下のようになります。

```
Package          Version
-----
aiohttp          3.7.3
alembic          1.5.2
appdirs          1.4.4
argo-workflows   3.6.1
argon2-cffi      20.1.0
async-generator   1.10
async-timeout    3.0.1
attrdict         2.0.1
attrs            20.3.0
backcall         0.2.0
```

関連情報

- ノートブックサーバーへの Python パッケージのインストール

3.9. ノートブックサーバーへの PYTHON パッケージのインストール

パッケージとバージョンを **requirements.txt** ファイルに追加し、ノートブックセルで **pip install** コマンドを実行すると、デフォルトのノートブックサーバーイメージには含まれない Python パッケージをインストールできます。



注記

パッケージを直接インストールすることもできますが、Red Hat は、**requirements.txt** ファイルを使用して、ファイルに記載されているパッケージを異なるノートブック間で簡単に再利用できるようにすることが推奨されます。さらに、**requirements.txt** ファイルを使用すると、S2I ビルドを使用してモデルをデプロイする場合にも役立ちます。

前提条件

- Jupyter にログインし、ノートブックを開いている。

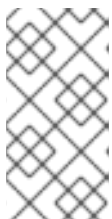
手順

1. 以下の方法のいずれかを使用して、新規テキストファイルを作成します。
 - + をクリックして新規ランチャーを開き、**Text file** をクリックします。
 - **File** → **New** → **Text File** の順にクリックします。
2. テキストファイルの名前を **requirements.txt** に変更します。
 - a. ファイルの名前を右クリックし、**Rename Text** をクリックします。**Rename File** ダイアログが開きます。
 - b. **New Name** フィールドに **requirements.txt** を入力し、**Rename** をクリックします。
3. **Requirements.txt** ファイルにインストールするパッケージを追加します。

```
altair
```

== (に等しい) 演算子を使用して、インストールするバージョンを指定できます。以下はその例です。

```
altair==4.1.0
```



注記

Red Hat は、時間が経過してもノートブックサーバーの高い安定性を確保するために、正確なパッケージバージョンを指定することが推奨されます。新しいパッケージバージョンは、お使いの環境における動作に対し、望ましくない/予期しない変更を加える可能性があります。

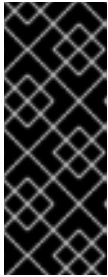
複数のパッケージを同時にインストールする場合は、各パッケージを別々の行に配置します。

4. ノートブックセルを使用して、**requirements.txt** のパッケージをサーバーにインストールします。

- a. ノートブックに新しいセルを作成し、以下のコマンドを入力します。

```
!pip install -r requirements.txt
```

- b. Shift と Enter を押してセルを実行します。



重要

このコマンドでは、ノートブックサーバーにパッケージがインストールされますが、コードセルで **import** ディレクティブを実行してコード内のパッケージを使用する必要があります。

```
import altair
```

検証

- **requirements.txt** のパッケージが、ノートブックサーバーにインストールされているパッケージ一覧に表示されることを確認します。詳細は、[ノートブックサーバーにインストールされている Python パッケージの表示](#) を参照してください。

3.10. サーバーの再起動によるノートブックサーバー設定の更新

ノートブックサーバーを停止して再起動することで、ノートブックサーバーの設定を更新できます。たとえば、サーバーのメモリーが不足する場合は、サーバーを再起動してコンテナのサイズを大きくできます。

前提条件

- 実行中のノートブックサーバー。
- Jupyter にログインしている。

手順

1. **File** → **Hub Control Panel** をクリックします。
Notebook server control panel が開きます。
2. **Stop notebook server** ボタンをクリックします。
Stop server ダイアログが開きます。
3. **Stop server** をクリックして、決定内容を確認します。
Start a notebook server ページが表示されます。
4. 関連するノートブックサーバー設定を更新し、**Start server** をクリックします。

検証

- ノートブックサーバーが起動し、更新された設定が含まれます。

関連情報

- [Jupyter の起動およびノートブックサーバーの起動](#)

第4章 データサイエンスパイプラインの操作

データサイエンティストは、Docker コンテナを使用して、データサイエンスパイプラインを備えたポータブル機械学習 (ML) ワークフローを構築することで、OpenShift Data Science でデータサイエンスプロジェクトを強化できます。これにより、機械学習ワークフローを標準化および自動化し、データサイエンスモデルを開発およびデプロイできるようになります。

たとえば、機械学習ワークフローのステップには、データ抽出、データ処理、機能抽出、モデルトレーニング、モデル検証、モデル提供などの項目が含まれる場合があります。これらのアクティビティを自動化すると、組織は新しく受信したデータに基づいてモデルを再トレーニングおよび更新する継続的なプロセスを開発できるようになります。これは、統合された機械学習デプロイメントの構築と実稼働環境の継続的な運用に関連する課題に対処するのに役立ちます。

Elyra JupyterLab 拡張機能を使用して、JupyterLab 内でデータサイエンスパイプラインを作成して実行することもできます。詳細は、[JupyterLab でのパイプラインの操作](#) を参照してください。

OpenShift Data Science のデータサイエンスパイプラインは、次のコンポーネントで設定されます。

- **パイプラインサーバー:** データサイエンスプロジェクトに接続され、データサイエンスパイプラインをホストするサーバー。
- **パイプライン:** パイプラインは、機械学習ワークフローの設定と、ワークフロー内の各コンポーネント間の関係を定義します。
 - **パイプラインコード:** Tekton 形式の YAML ファイル内のパイプラインの定義。
 - **パイプライングラフ:** パイプライン実行で実行されるステップとステップ間の関係をグラフィカルに示したもの。
- **パイプラインの実行:** パイプラインの実行
 - **トリガーされた実行:** 以前実行されたパイプラインの実行
 - **スケジュールされた実行:** 少なくとも 1 回実行するようにスケジュールされたパイプライン実行

この機能は Kubeflow Pipelines v1 に基づいています。Kubeflow Pipelines SDK を使用して、Python コードでデータサイエンスパイプラインを構築します。パイプラインを構築したら、kfp-tekton SDK (バージョン 1.5.x のみ) を使用してパイプラインを Tekton 形式の YAML コードにコンパイルします。OpenShift Data Science ユーザーインターフェイスを使用すると、パイプラインとパイプライン実行を追跡および管理できます。

データサイエンスパイプラインを使用する前に、OpenShift Pipelines Operator をインストールする必要があります。OpenShift Pipelines Operator の互換性のあるバージョンのインストールの詳細は、[Red Hat OpenShift Pipelines リリースノート](#) および [Red Hat OpenShift Data Science: Supported Configurations](#) を参照してください。

ローカルストレージを消費しないように、パイプラインアーティファクトを Amazon Web Services (AWS) シンプルストレージサービス (S3) バケットに保存できます。これを行うには、まず AWS アカウントで S3 バケットへの書き込みアクセスを設定する必要があります。

4.1. データサイエンスパイプラインの管理

4.1.1. パイプラインサーバーの設定

OpenShift Data Science でパイプラインを正常に作成するには、パイプラインサーバーを設定する必要があります。これには、パイプラインアーティファクトとデータの保存場所の設定が含まれます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを追加できるデータサイエンスプロジェクトが作成されている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. パイプラインサーバーを設定するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Pipelines** セクションで、**Create a pipeline server** をクリックします。**Configure pipeline server** ダイアログが表示されます。
4. **Object storage connection** セクションで、パイプラインアーティファクトを保存するための S3 互換データ接続を指定するには、次のアクションセットのいずれかを選択します。



注記

パイプラインサーバーの作成後、**/metadata** フォルダと **/artifacts** フォルダがデフォルトの **root** フォルダに自動的に作成されます。したがって、パイプラインサーバーのデータ接続を設定するときに、ストレージディレクトリーを指定する必要はありません。

- **Existing data connection** を選択して、以前に定義したデータ接続を使用します。このオプションを選択した場合は、**Name** リストから関連するデータ接続の名前を選択し、手順 6 に進みます。
 - **Create new data connection** を選択して、パイプラインサーバーがアクセスできる新しいデータ接続を追加します。
5. **Create new data connection** を選択した場合は、次の手順を実行します。
 - a. **Name** フィールドに、データ接続の名前を入力します。
 - b. **AWS_ACCESS_KEY_ID** フィールドに、Amazon Web Services のアクセスキー ID を入力します。
 - c. **AWS_SECRET_ACCESS_KEY_ID** フィールドで指定したアカウントのシークレットアクセスキーを入力します。
 - d. オプション: **AWS_S3_ENDPOINT** フィールドに、AWS S3 ストレージのエンドポイントを入力します。

- e. オプション: `AWS_DEFAULT_REGION` フィールドに、AWS アカウントのデフォルトのリージョンを入力します。
- f. `AWS_S3_BUCKET` フィールドに AWS S3 バケットの名前を入力します。



重要

新しいデータ接続を作成する場合は、指定された他の必須フィールドに加えて、`AWS_S3_BUCKET` フィールドも必須です。間違ったデータ接続設定を指定した場合は、同じパイプラインサーバー上でこれらの設定を更新できません。したがって、パイプラインサーバーを削除し、別のパイプラインサーバーを設定する必要があります。

6. **Database** セクションで、**Show advanced database options** をクリックしてパイプラインデータを保存するデータベースを指定し、次の一連のアクションのいずれかを選択します。
 - プロジェクトに MariaDB データベースをデプロイするには、**Use default database stored on your cluster** を選択します。
 - **Connect to external MySQL database** を選択して、パイプラインサーバーがアクセスできる外部データベースに新しい接続を追加します。
 - i. **Host** フィールドにデータベースのホスト名を入力します。
 - ii. **Port** フィールドにデータベースのポートを入力します。
 - iii. **Username** フィールドに、データベースに接続するデフォルトのユーザー名を入力します。
 - iv. **Password** フィールドに、デフォルトのユーザーアカウントのパスワードを入力します。
 - v. **Database** フィールドにデータベース名を入力します。
7. **Configure** をクリックします。

検証

- 設定したパイプラインサーバーは、Project details ページの **Pipelines** セクションに表示されます。
- **Import pipeline** ボタンは、Project details ページの **Pipelines** セクションで使用できます。

4.1.2. パイプラインの定義

Kubeflow Pipelines SDK を使用すると、エンドツーエンドの機械学習とデータパイプラインを定義できます。Kubeflow Pipelines SDK を使用して、Python コードでデータサイエンスパイプラインを構築します。パイプラインを構築したら、kfp-tekton SDK (バージョン 1.5.x のみ) を使用してパイプラインを Tekton 形式の YAML コードにコンパイルします。パイプラインを定義した後、YAML ファイルを OpenShift Data Science ダッシュボードにインポートして、その実行設定を設定できるようにします。Tekton への Kubeflow Pipelines SDK のインストールおよび使用に関する詳細は、[Kubeflow Pipelines SDK for Tekton](#) を参照してください。

Elyra JupyterLab 拡張機能を使用して、JupyterLab 内でデータサイエンスパイプラインを作成して実行することもできます。JupyterLab でのパイプラインの作成の詳細は、[JupyterLab でのパイプラインの操作](#) を参照してください。Elyra JupyterLab 拡張機能の詳細は、[Elyra ドキュメンテーション](#) を参照

してください。

関連情報

- [Kubeflow Pipelines SDK for Tekton](#)
- [KFP Tekton samples and compiler samples](#)
- [Kubeflow Pipelines v1 Documentation](#)
- [Elyra ドキュメンテーション](#)

4.1.3. データサイエンスパイプラインのインポート

OpenShift Data Science で簡単にデータサイエンスパイプラインの操作を開始できるように、パイプラインのコードを含む YAML ファイルをアクティブなパイプラインサーバーにインポートできます。このファイルには、Tekton コンパイラーでコンパイルされた Kubeflow パイプラインが含まれています。パイプラインをパイプラインサーバーにインポートしたら、パイプライン実行を作成してパイプラインを実行できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、パイプラインをインポートするプロジェクトを選択します。
3. **Import pipeline** をクリックします。
Import pipeline ダイアログが開きます。
4. インポートするパイプラインの詳細を入力します。
 - a. **Pipeline name** フィールドに、インポートするパイプラインの名前を入力します。
 - b. **Pipeline description** フィールドに、インポートするパイプラインの説明を入力します。
 - c. **Upload** をクリックします。または、ローカルマシンのファイルシステムからファイルをドラッグし、**Import pipeline** ダイアログの指定された領域にドロップします。
ファイルブラウザーが開きます。
 - d. パイプラインコードを含むファイルに移動し、**Select** をクリックします。
 - e. **Import pipeline** をクリックします。

検証

- インポートしたパイプラインが **Pipelines** ページに表示されます。


4.1.4. データサイエンスパイプラインのダウンロード

以前に OpenShift Data Science にアップロードしたデータサイエンスパイプラインにさらに変更を加えるには、ユーザーインターフェイスからパイプラインのコードをダウンロードできます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインを作成し、ダウンロードできるアクティブなパイプラインサーバーにインポートしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、パイプラインをダウンロードするプロジェクトを選択します。
3. **Pipeline name** 列で、ダウンロードするパイプラインの名前をクリックします。
Pipeline details ページが開き、**Graph** タブが表示されます。
4. **YAML** タブをクリックします。
ページがリロードされ、パイプラインコードを示す埋め込み YAML エディターが表示されます。
5. **Download** ボタン () をクリックして、パイプラインのコードを含む YAML ファイルをローカルマシンにダウンロードします。

検証

- パイプラインコードは、ブラウザのダウンロードファイル用のデフォルトディレクトリーにダウンロードされます。

4.1.5. データサイエンスパイプラインの削除

て、OpenShift Data Science の **Pipelines** ページに表示されないように、データサイエンスパイプラインを削除できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- **Pipelines** ページにアクティブなパイプラインが表示される。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、削除するパイプラインを含むプロジェクトを選択します。
3. 削除するパイプラインの横にあるアクションメニュー (⋮) をクリックし、**Delete pipeline** の削除をクリックします。
Delete pipeline ダイアログが開きます。
4. テキストフィールドにパイプライン名を入力して、削除することを確認します。
5. **Delete pipeline** をクリックします。

検証

- 削除したデータサイエンスパイプラインは、**Pipelines** ページに表示されなくなります。

4.1.6. パイプラインサーバーの削除

データサイエンスパイプラインの実行が終了したら、パイプラインサーバーを削除できます。パイプラインサーバーを削除すると、関連付けられているすべてのパイプラインと実行が自動的に削除されます。パイプラインデータがデータベースに保存されている場合は、データベースもそのメタデータとともに削除されます。さらに、パイプラインサーバーを削除した後は、別のパイプラインサーバーを作成するまで、新しいパイプラインを作成したり、パイプラインを実行したりできません。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、パイプラインサーバーを削除するプロジェクトを選択します。

3. **Pipeline server actions** リストから、**Delete pipeline server** を選択します。**Delete pipeline server** ダイアログが開きます。
4. パイプラインサーバーの名前をテキストフィールドに入力して、削除することを確認します。
5. **Delete** をクリックします。

検証

- 削除されたパイプラインサーバーに以前に割り当てられていたパイプラインは、関連するデータサイエンスプロジェクトの **Pipelines** ページに表示されなくなります。
- 削除されたパイプラインサーバーに以前に割り当てられていたパイプライン実行は、関連するデータサイエンスプロジェクトの **Runs** ページに表示されなくなります。

4.1.7. パイプラインサーバーの詳細の表示

パイプラインのデータ接続の詳細やそのデータの保存場所など、OpenShift Data Science で設定されたパイプラインサーバーの詳細を表示できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 以前に、アクティブで使用可能なパイプラインサーバーを含むデータサイエンスプロジェクトを作成している。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、パイプラインサーバーを表示するプロジェクトを選択します。
3. **Pipeline server actions** リストから、**View pipeline server configuration** を選択します。
4. パイプラインサーバーの詳細の検査が終了したら、**Done** をクリックします。

検証

- **View pipeline server** ダイアログで、関連するパイプラインサーバーの詳細を表示できます。


4.1.8. 既存のパイプラインの表示

Red Hat OpenShift Data Science にインポートしたパイプラインの詳細 (パイプラインの最後の実行、作成日、パイプラインの実行済実行など) を表示できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインがアクティブで利用可能なパイプラインサーバーにインポートされている。
- インポートしたパイプラインが利用可能であるか、以前にインポートされ利用可能な他のパイプラインが存在するかを表示できる。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. **Project** リストから、パイプラインを表示する関連プロジェクトを選択します。
3. リストにあるパイプラインを調べてください。
4. オプション: 関連する行の **Expand** () をクリックして、パイプラインの実行済み実行を表示します。パイプラインに実行が含まれていない場合は、**Create run** をクリックしてを作成します。

検証

- 以前に作成したデータサイエンスパイプラインのリストが **Pipelines** ページに表示されます。

4.2. パイプライン実行の管理

4.2.1. パイプライン実行の概要

パイプラインの実行は、データサイエンスパイプラインの1回の実行です。データサイエンティストは、OpenShift Data Science を使用して、データサイエンスパイプラインの実行を定義、管理、追跡できます。OpenShift Data Science ユーザーインターフェイスの **Runs** ページから、データサイエンスプロジェクトで以前に実行されたものおよびスケジュールされている実行の記録を表示できます。

実行は移植性を目的としています。したがって、パイプライン実行のクローンを作成して、必要に応じて再作成およびスケールしたり、必要がなくなった場合には削除したりできます。実行は、作成直後に1回だけか、定期的かに設定できます。定期的な実行は、すべてのパラメーター値を含むパイプラインのコピーと実行トリガーで設定されます。実行トリガーは、定期的な実行のタイミングを示します。次の実行トリガーを定義できます。

- 定期的: 一定の間隔で実行するようにスケジュールを設定するために使用されます。
- Cron: cron ジョブとして実行をスケジュールするために使用されます。

実行すると、OpenShift Data Science ユーザーインターフェイスの実行の **Details** ページから実行の進行状況を追跡できます。ここから、実行のグラフと出力アーティファクトを表示できます。

パイプラインの実行は次のように分類できます。

- スケジュールされた実行: 少なくとも1回実行するようにスケジュールされたパイプライン実行。
- トリガーされた実行: 以前実行されたパイプラインの実行

4.2.2. パイプライン実行のスケジュール設定


パイプラインの実行をスケジュールすることで、パイプラインの単一の実行をインスタンス化できます。OpenShift Data Science では、特定の時間に実行するようにスケジュールを設定したり、作成後すぐに実行したりできます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインをアクティブなパイプラインサーバーにインポートしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。
2. 関連するパイプラインの横にあるアクションメニュー (⋮) をクリックし、**Create run** をクリックします。
Create run ページが開きます。
3. **Project** リストから、実行を作成するパイプラインを含むプロジェクトを選択します。
4. **Name** フィールドには、実行の名前を入力します。
5. **Description** フィールドには、実行の説明を入力します。
6. **Pipeline** リストから、実行を作成するパイプラインを選択します。または、新しいパイプラインをアップロードするには、**Upload new pipeline** をクリックし、**Import pipeline** ダイアログの関連フィールドに入力します。
7. 次の一連のアクションのいずれかを実行して、実行タイプを設定します。
 - **Run once immediately after creation** を選択して、実行を作成直後に1回実行するように指定します。
 - 繰り返し実行をスケジュールするには、**Schedule recurring run** を選択します。
 - i. 実行のトリガータイプを設定します。
 - A. **Periodic** を選択し、リストから実行頻度を選択します。
 - B. **cron** 形式で実行スケジュールを指定するには、**Cron** を選択します。これにより、

実行用の cron ジョブが作成されます。Copy ボタン () をクリックして、cron ジョブスケジュールをクリップボードにコピーします。一番左のフィールドは秒を表します。サポートされている cron 形式を使用したタスクのスケジュールの詳細は、[Cron Expression Format](#) を参照してください。

- ii. 実行時間を設定します。
 - A. **Start date** チェックボックスをオンにして、実行の開始日を指定します。Calendar を使用して実行の開始日を選択し、時間のリストから開始時間を選択します。
 - B. **End date** チェックボックスをオンにして、実行の終了日を指定します。Calendar を使用して実行の終了日を選択し、時間のリストから終了時間を選択します。
8. リストからパラメーターを選択して、実行の入力パラメーターを設定します。
9. **Create** をクリックします。

検証

- 作成したパイプライン実行は、Runs ページの **Scheduled** タブに表示されます。


4.2.3. スケジュールされたパイプライン実行のクローン作成


パイプライン設定の一部として実行するように実行をスケジュールしやすくするために、スケジュールされた既存の実行をクローンして複製できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインをアクティブなパイプラインサーバーにインポートしている。
- クローンを作成できる実行を以前にスケジュールしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Runs** をクリックします。
Runs ページが開きます。
2. 関連する実行の横にあるアクションメニュー () をクリックし、**Clone** をクリックします。
Clone ページが開きます。
3. **Project** リストから、クローンを作成する実行のパイプラインを含むプロジェクトを選択します。

4. **Name** フィールドに、クローンを作成する実行の名前を入力します。
5. **Description** フィールドに、クローンを作成する実行の説明を入力します。
6. **Pipeline** リストから、クローンを作成する実行を含むパイプラインを選択します。
7. 複製する実行のタイプを設定するには、**Run type** セクションで、次の一連のアクションのいずれかを実行します。
 - **Run once immediately after create** を選択して、クローンを作成する実行が作成直後に1回実行するように指定します。このオプションを選択した場合は、ステップ10に進みません。
 - **Schedule recurring run** を選択して、クローンを作成する実行を繰り返し実行するようにスケジュールします。
8. 前の手順で **Schedule recurring run** を選択した場合は、実行のトリガータイプを設定し、次のいずれかのアクションを実行します。
 - **Periodic** を選択し、**Run every** リストから実行頻度を選択します。
 - **cron** 形式で実行スケジュールを指定するには、**Cron** を選択します。これにより、実行用の cron ジョブが作成されます。**Copy** ボタン () をクリックして、cron ジョブスケジュールをクリップボードにコピーします。一番左のフィールドは秒を表します。サポートされている **cron** 形式を使用したタスクのスケジュールの詳細は、[Cron Expression Format](#) を参照してください。
9. 手順7で **Schedule recurring run** を選択した場合は、クローンを作成する実行の期間を設定します。
 - a. **Start date** チェックボックスをオンにして、実行の開始日を指定します。カレンダーツールを使用して開始日を選択し、時間のリストから開始時間を選択します。
 - b. **End date** チェックボックスをオンにして、実行の終了日を指定します。カレンダーツールを使用して終了日を選択し、時間のリストから終了時刻を選択します。
10. **Parameters** セクションで、リストから適切なパラメーターを選択して、クローンを作成する実行の入力パラメーターを設定します。
11. **Create** をクリックします。

検証

- クローンを作成したパイプラインの実行は、**Runs** ページの **Scheduled** タブに表示されます。

4.2.4. トリガーされたパイプライン実行の停止

トリガーされたパイプライン実行の継続が必要なくなった場合は、定義された終了日の前に実行を停止できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。

- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを含む、以前に作成されたデータサイエンスプロジェクトが利用できるようになっている。
- パイプラインがアクティブで利用可能なパイプラインサーバーにインポートされている。
- パイプラインの実行をトリガーしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Runs** をクリックします。
Runs ページが開きます。
2. **Project** リストから、パイプラインの実行を停止するプロジェクトを選択します。
3. **Triggered** タブをクリックします。
4. 表の **Name** 列で、停止する実行の名前をクリックします。
Run details ページが開きます。
5. **Actions** リストから **Stop run** を選択します。
実行が停止するまでに短い遅延が発生する可能性があります。

検証

- 以前にトリガーされた実行のリストが、**Runs** ページの **Triggered** タブに表示されます。

4.2.5. スケジュールされたパイプライン実行の削除

そのパイプライン実行を削除して、**Runs** ページに表示されないように、以前にスケジュールしたものの不要になったパイプライン実行を破棄できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインをアクティブなパイプラインサーバーにインポートしている。
- 削除可能な実行をスケジュールしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Runs** をクリックします。
Runs ページが開きます。

2. **Project** リストから、スケジュールされた実行を削除するパイプラインを含むプロジェクトを選択します。
ページが更新されて、パイプラインのスケジュールされた実行が **Scheduled** タブに表示されません。
3. 削除するスケジュールされた実行の横にあるアクションメニュー (⋮) をクリックし、**Delete** をクリックします。
Delete scheduled run ダイアログが開きます。
4. テキストフィールドに実行の名前を入力して、プロジェクトを削除することを確認します。
5. **Delete scheduled run** をクリックします。

検証

- 削除した実行は、**Scheduled** タブに表示されなくなります。

4.2.6. トリガーされたパイプライン実行の削除

以前に実行したが記録が不要になったパイプライン実行を破棄して、**Runs** ページの **Triggered** タブに表示されないように削除できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- 設定されたパイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインをアクティブなパイプラインサーバーにインポートしている。
- 削除可能な実行を実行している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Runs** をクリックします。
Runs ページが開きます。
2. **Project** リストから、トリガーされた実行を削除するパイプラインを含むプロジェクトを選択します。
ページが更新されて、パイプラインのトリガーされた実行が **Triggered** タブに表示されます。
3. 削除するトリガーされた実行の横にあるアクションメニュー (⋮) をクリックし、**Delete** をクリックします。
Delete triggered run ダイアログが開きます。
4. テキストフィールドに実行の名前を入力して、プロジェクトを削除することを確認します。
5. **Delete triggered run** をクリックします。

検証

- 削除した実行は、**Triggered** タブに表示されなくなります。

4.2.7. スケジュールされたパイプライン実行の表示

OpenShift Data Science で実行がスケジュールされているパイプライン実行のリストを表示できます。このリストから、実行が属するパイプラインなど、パイプラインの実行に関連する詳細を表示できます。実行のステータス、実行頻度、スケジュールを表示することもできます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- OpenShift Pipelines Operator がインストールされている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインがアクティブで利用可能なパイプラインサーバーにインポートされている。
- パイプライン実行を作成してスケジュールしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Runs** をクリックします。
Runs ページが開きます。
2. **Project** リストから、スケジュールされたパイプライン実行を表示するプロジェクトを選択します。
3. **Scheduled** タブをクリックします。
4. スケジュールされた実行のリストを示す表を調べてください。
実行がスケジュールされると、表の **Status** 列に実行のステータスが表示され、実行の準備ができていないか、実行できないかを示します。以前にインポートしたノートブックイメージを有効または無効にするには、関連するノートブックイメージを含む行で、**Enabled** 列のトグルをクリックします。

検証

- スケジュールされた実行のリストは、**Runs** ページの **Scheduled** タブに表示されます。

4.2.8. トリガーされたパイプライン実行の表示

OpenShift Data Science で以前に実行されたパイプライン実行のリストを表示できます。このリストから、実行が属するパイプライン、実行のステータス、期間、実行開始時間など、パイプラインの実行に関連する詳細を表示できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。

- OpenShift Pipelines Operator がインストールされている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインがアクティブで利用可能なパイプラインサーバーにインポートされている。
- パイプラインの実行をトリガーしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Runs** をクリックします。
Runs ページが開きます。
2. **Project** リストから、以前に実行されたパイプライン実行を表示するプロジェクトを選択します。
Run details ページが開きます。
3. **Triggered** タブをクリックします。
トリガーされた実行のリストを示すテーブルが開きます。実行が完了すると、表の **Status** 列に実行のステータスが表示され、実行が成功したか失敗したかが示されます。

検証

- 以前にトリガーされた実行のリストが、**Runs** ページの **Triggered** タブに表示されます。

4.2.9. パイプライン実行の詳細の表示


パイプライン実行をより明確に理解するために、グラフ、実行の詳細、実行出力など、以前にトリガーされたパイプライン実行の詳細を表示できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- パイプラインサーバーを含む利用可能なデータサイエンスプロジェクトを作成している。
- パイプラインがアクティブで利用可能なパイプラインサーバーにインポートされている。
- パイプラインの実行をトリガーしている。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Pipelines** → **Pipelines** をクリックします。
Pipelines ページが開きます。

2. **Project** リストから、パイプライン実行を表示するプロジェクトを選択します。
3. 実行の詳細を表示するパイプラインの場合は、**Expand** () をクリックします。
4. **Runs** セクションで、詳細を表示する実行の名前をクリックします。
Run details ページが開きます。

検証

- **Run details** ページでは、実行のグラフ、実行の詳細、入力パラメーター、および実行出力を表示できます。

4.3. JUPYTERLAB でのパイプラインの操作

4.3.1. JupyterLab のパイプラインの概要

Elyra を使用して、JupyterLab で視覚的なエンドツーエンドのパイプラインワークフローを作成できます。Elyra は、OpenShift Data Science で実行できるパイプラインワークフローを作成するための Pipelines Editor を提供する JupyterLab の拡張機能です。

JupyterLab でパイプラインを使用する前に、OpenShift Pipelines Operator をインストールする必要があります。OpenShift Pipelines Operator の互換性のあるバージョンのインストールの詳細は、[Red Hat OpenShift Pipelines リリースノート](#) および [Red Hat OpenShift Data Science: Supported Configurations](#) を参照してください。

以下のノートブックイメージのいずれかの最新バージョンを作成すると、JupyterHub 内の Elyra 拡張にアクセスできます。

- Standard Data Science
- PyTorch
- TensorFlow
- TrustyAI

Pipeline Editor を使用してパイプラインを視覚的に設計できるため、パイプラインを作成および実行するのに最小限のコーディングが必要になります。Elyra の詳細は、[Elyra ドキュメンテーション](#) を参照してください。Pipeline Editor の詳細は、[Visual Pipeline Editor](#) を参照してください。パイプラインを作成したら、JupyterLab でローカルに実行することも、OpenShift Data Science のデータサイエンスパイプラインを使用してリモートで実行することもできます。

パイプライン作成プロセスは次のタスクで設定されます。

- ワークベンチを含むデータサイエンスプロジェクトを作成する。
- パイプラインサーバーを作成する。
- JupyterLab の Pipeline Editor で新規パイプラインを作成する。
- Python ノートブックまたは Python スクリプトを追加し、それらのランタイムプロパティを定義してパイプラインを開発する。
- 実行の依存関係を定義する。
- Pipeline を実行するか、エクスポートする。

JupyterLab でパイプラインを実行するには、パイプラインインスタンスにランタイム設定が含まれている必要があります。ランタイム設定は、パイプラインインスタンスと S3 互換のクラウドストレージの接続情報を定義します。

データサイエンスプロジェクトの一部としてワークベンチを作成すると、デフォルトのランタイム設定が自動的に作成されます。ただし、OpenShift Data Science ダッシュボードの Jupyter タイルからノートブックを作成する場合は、JupyterLab でパイプラインを実行する前にランタイム設定を作成する必要があります。ランタイム設定の詳細は、[Runtime Configuration](#) を参照してください。前提条件として、ワークベンチを作成する前に、ワークベンチと同じデータサイエンスプロジェクト内にパイプラインサーバーを作成および設定していることを確認します。

S3 互換のクラウドストレージを使用すると、ノートブックやスクリプトの実行中にデータを利用できるようになります。クラウドストレージには、JupyterLab を実行するデプロイメント内のマシン、および Data Science Pipelines をホストするクラスターからアクセスする必要があります。JupyterLab でパイプラインを作成して実行する前に、s3 互換ストレージの認証情報がすぐに利用できることを確認してください。

関連情報

- [Elyra ドキュメンテーション](#)
- [Visual Pipeline Editor](#)
- [Runtime Configuration](#)

4.3.2. パイプラインエディターへのアクセス

Elyra を使用して、JupyterLab で視覚的なエンドツーエンドのパイプラインワークフローを作成できます。Elyra は、OpenShift Data Science で実行できるパイプラインワークフローを作成するための Pipelines Editor を提供する JupyterLab の拡張機能です。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。
- S3 互換ストレージにアクセスできる。

手順

1. JupyterLab を開くと、JupyterLab ランチャーが自動的に表示されることを確認します。
2. JupyterLab ランチャーの **Elyra** セクションで、**Pipeline Editor** タイルをクリックします。Pipeline Editor が開きます。

検証

- JupyterLab で Pipeline Editor を表示できます。



4.3.3. ランタイム設定の作成

データサイエンスプロジェクトの一部としてワークベンチを作成すると、デフォルトのランタイム設定が自動的に作成されます。ただし、OpenShift Data Science ダッシュボードの Jupyter タイルからノートブックを作成する場合は、JupyterLab でパイプラインを実行する前にランタイム設定を作成する必要があります。これにより、パイプラインインスタンスと S3 互換のクラウドストレージの接続情報を指定できます。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- S3 互換のクラウドストレージにアクセスできる。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。

手順

1. JupyterLab の左側のサイドバーで、**Runtimes** () をクリックします。
2. **Create new runtime configuration** ボタン () をクリックします。
Add new Data Science Pipelines runtime configuration ページが開きます。
3. 関連フィールドに入力してランタイム設定を定義します。
 - a. **Display Name** フィールドに、ランタイム設定の名前を入力します。
 - b. オプション: **Description** フィールドに、ランタイム設定を定義する説明を入力します。
 - c. オプション: **Tags** フィールドで、**Add Tag** をクリックして、パイプラインインスタンスのカテゴリを定義します。タグの名前を入力し、Enter キーを押します。
 - d. データサイエンスパイプラインの認証情報を定義します。
 - i. **Data Science Pipelines API Endpoint** フィールドに、データサイエンスパイプラインの API エンドポイントを入力します。このフィールドにはパイプラインの namespace を指定しないでください。
 - ii. **Public Data Science Pipelines API Endpoint** フィールドに、データサイエンスパイプラインのパブリック API エンドポイントを入力します。



重要

ダッシュボードの **Data Science Pipelines → Runs** ページから Data Science Pipelines API エンドポイントを取得できます。関連するエンドポイントをコピーし、**Public Data Science Pipelines API Endpoint** フィールドに入力します。

- iii. オプション: **Data Science Pipelines User Namespace** フィールドに、パイプラインを実行するための関連するユーザー namespace を入力します。
- iv. **Data Science Pipelines engine** リストから、**Tekton** を選択します。
- v. **Authentication Type** リストから、パイプラインの認証に必要な認証タイプを選択します。



重要

ダッシュボードの Jupyter タイルから直接ノートブックを作成した場合は、**Authentication Type** リストから **EXISTING_BEARER_TOKEN** を選択します。

- vi. **Data Science Pipelines API Endpoint Username** フィールドに、認証タイプに必要なユーザー名を入力します。
- vii. **Data Science Pipelines API Endpoint Password Or Token** に、認証タイプに必要なパスワードまたはトークンを入力します。



重要

Data Science Pipelines API エンドポイントトークンを取得するには、OpenShift Web コンソールの右上隅にあるユーザー名をクリックし、**Copy login command** を選択します。ログインしたら、**Display token** をクリックし、**Log in with this token** コマンドから **--token=** の値をコピーします。


- e. S3 互換ストレージの接続情報を定義します。
 - i. **Cloud Object Storage Endpoint** フィールドに、S3 互換ストレージのエンドポイントを入力します。Amazon s3 エンドポイントの詳細は、[Amazon Simple Storage Service エンドポイントおよびクォータ](#) を参照してください。
 - ii. オプション: **Public Cloud Object Storage Endpoint** フィールドに、S3 互換ストレージの URL を入力します。
 - iii. **Cloud Object Storage Bucket Name** フィールドに、パイプラインアーティファクトが保存されているバケットの名前を入力します。バケット名が存在しない場合は、自動的に作成されます。
 - iv. **Cloud Object Storage Authentication Type** リストから、S3 互換クラウドストレージへのアクセスに必要な認証タイプを選択します。AWS S3 バケットを使用する場合は、一覧から **KUBERNETES_SECRET** を選択します。
 - v. **Cloud Object Storage Credentials Secret** フィールドに、ストレージのユーザー名とパスワードを含むシークレットを入力します。このシークレットは、該当する場合、関連するユーザー namespace で定義されます。さらに、パイプラインランタイムをホス

トするクラスターに保存する必要があります。

- vi. **Cloud Object Storage Username** フィールドに、S3 互換クラウドストレージに接続するためのユーザー名を入力します (該当する場合)。AWS S3 バケットを使用する場合は、AWS Secret Access Key ID を入力します。
- vii. **Cloud Object Storage Password** フィールドに、S3 互換クラウドストレージに接続するためのパスワードを入力します (該当する場合)。AWS S3 バケットを使用する場合は、AWS Secret Access Key を入力します。

f. **Save & Close** をクリックします。

検証

- 作成したランタイム設定は、JupyterLab の左側のサイドバーにある **Runtimes** タブ () をクリックします。



4.3.4. ランタイム設定の更新

ランタイム設定が正確で更新されていることを確認するには、既存のランタイム設定の設定を変更します。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- S3 互換ストレージにアクセスできる。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- 以前に作成したランタイム設定が JupyterLab インターフェイスで利用できる。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。

手順

1. JupyterLab の左側のサイドバーで、**Runtimes** () をクリックします。
2. 更新するランタイム設定の上にカーソルを置き、**Edit** ボタン () をクリックします。
Data Science Pipelines runtime configuration ページが開きます。
3. 関連フィールドに入力してランタイム設定を更新します。
 - a. 必要に応じて、**Display Name** フィールドでランタイム設定の名前を更新します。

- b. オプション: 必要に応じて、**Description** フィールドでランタイム設定の説明を更新します。
- c. オプション: **Tags** フィールドで、**Add Tag** をクリックして、パイプラインインスタンスのカテゴリーを定義します。タグの名前を入力し、Enter キーを押します。
- d. データサイエンスパイプラインの認証情報を定義します。
 - i. **Data Science Pipelines API Endpoint** フィールドで、データサイエンスパイプラインの API エンドポイントを更新します (該当する場合)。このフィールドにはパイプラインの namespace を指定しないでください。
 - ii. **Public Data Science Pipelines API Endpoint** フィールドで、データサイエンスパイプラインの API エンドポイントを更新します (該当する場合)。
 - iii. オプション: 該当する場合、**Data Science Pipelines User Namespace** フィールドで、パイプラインを実行するために関連するユーザー namespace を更新します。
 - iv. **Data Science Pipelines engine** リストから、**Tekton** を選択します。
 - v. 該当する場合は、**Authentication Type** リストから、パイプラインの認証に必要な新しい認証タイプを選択します。



重要

ダッシュボードの Jupyter タイルから直接ノートブックを作成した場合は、**Authentication Type** リストから **EXISTING_BEARER_TOKEN** を選択します。

- vi. **Data Science Pipelines API Endpoint Username** フィールドで、認証タイプに必要なユーザー名を更新します (該当する場合)。
- vii. **Data Science Pipelines API Endpoint Password Or Token** で、認証タイプに必要なパスワードまたはトークンを更新します (該当する場合)。




重要

Data Science Pipelines API エンドポイントトークンを取得するには、OpenShift Web コンソールの右上隅にあるユーザー名をクリックし、**Copy login command** を選択します。ログインしたら、**Display token** をクリックし、**Log in with this token** コマンドから **--token=** の値をコピーします。

- e. S3 互換ストレージの接続情報を定義します。
 - i. 該当する場合は、**Cloud Object Storage Endpoint** フィールドで、S3 互換ストレージのエンドポイントを更新します。Amazon s3 エンドポイントの詳細は、[Amazon Simple Storage Service エンドポイントおよびクォータ](#) を参照してください。
 - ii. オプション: **Public Cloud Object Storage Endpoint** フィールドで、S3 互換ストレージの URL を更新します (該当する場合)。
 - iii. 必要に応じて、**Cloud Object Storage Bucket Name** フィールドで、パイプラインアーティファクトが保存されているバケットの名前を更新します。バケット名が存在しない場合は、自動的に作成されます。

- iv. 該当する場合は、**Cloud Object Storage Authentication Type** リストから、S3 互換クラウドストレージへのアクセスに必要な認証タイプを更新します。AWS S3 バケットを使用する場合は、リストから **USER_CREDENTIALS** を選択する必要があります。
 - v. オプション: **Cloud Object Storage Credentials Secret** フィールドで、ストレージのユーザー名とパスワードを含むシークレットを更新します (該当する場合)。このシークレットは、関連するユーザー namespace で定義されます。パイプラインランタイムをホストするクラスターにシークレットを保存する必要があります。
 - vi. オプション: **Cloud Object Storage Username** フィールドで、S3 互換クラウドストレージに接続するためのユーザー名を更新します (該当する場合)。AWS S3 バケットを使用する場合は、AWS Secret Access Key ID を更新します。
 - vii. オプション: **Cloud Object Storage Password** フィールドで、S3 互換のクラウドストレージに接続するためのパスワードを更新します (該当する場合)。AWS S3 バケットを使用する場合は、AWS Secret Access Key を更新します。
- f. **Save & Close** をクリックします。

検証

- 更新したランタイム設定は、JupyterLab の左側のサイドバーにある **Runtimes** タブ () をクリックします。


4.3.5. ランタイム設定の削除


ランタイム設定の使用が終了したら、JupyterLab インターフェイスからランタイム設定を削除できます。ランタイム設定を削除した後は、別のランタイム設定を作成するまで JupyterLab でパイプラインを実行できません。

前提条件


- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- 以前に作成したランタイム設定が JupyterLab インターフェイスに表示されている。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。

手順

1. JupyterLab の左側のサイドバーで、**Runtimes** () をクリックします。

- 削除するランタイム設定の上にカーソルを置き、**Delete Item** ボタン () をクリックします。
実行時設定の削除を確認するダイアログボックスが表示されます。
- OK** をクリックします。

検証

- 削除したランタイム設定は、JupyterLab の左側のサイドバーにある **Runtimes** タブ () に表示されなくなります。

4.3.6. ランタイム設定の複製

全体的に同様の値を持つランタイム設定が再作成されないようにするために、JupyterLab インターフェイスで既存のランタイム設定を複製できます。


前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- 以前に作成したランタイム設定が JupyterLab インターフェイスに表示されている。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。

手順

- JupyterLab の左側のサイドバーで、**Runtimes** () をクリックします。
- 複製するランタイム設定の上にカーソルを置き、**Duplicate** ボタン () をクリックします。

検証

- 複製したランタイム設定が、JupyterLab の左側のサイドバーにある **Runtimes** タブ () をクリックします。

4.3.7. JupyterLab でパイプラインの実行


Pipeline Editor ユーザーインターフェイスから JupyterLab で作成したパイプラインを実行できます。パイプラインを実行する前に、データサイエンスプロジェクトとパイプラインサーバーを作成する必要

があります。パイプラインサーバーを作成したら、パイプラインサーバーと同じプロジェクト内にワークベンチを作成する必要があります。JupyterLab のパイプラインインスタンスにはランタイム設定が含まれている必要があります。データサイエンスプロジェクトの一部としてワークベンチを作成すると、デフォルトのランタイム設定が自動的に作成されます。ただし、OpenShift Data Science ダッシュボードの Jupyter タイルからノートブックを作成する場合は、JupyterLab でパイプラインを実行する前にランタイム設定を作成する必要があります。ランタイム設定は、パイプラインインスタンスと S3 互換のクラウドストレージの接続情報を定義します。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- S3 互換ストレージにアクセスできる。
- JupyterLab でパイプラインを作成している。
- JupyterLab の Pipeline Editor でパイプラインを開いている。
- パイプラインインスタンスにはランタイム設定が含まれている。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。

手順

1. Pipeline Editor ユーザーインターフェイスで **Run Pipeline** () をクリックします。**Run Pipeline** ダイアログが表示されます。**Pipeline Name** フィールドには、パイプラインファイル名が自動的に入力されます。



重要

一意のパイプライン名を入力する必要があります。入力するパイプライン名は、以前に実行されたパイプラインの名前と一致してはなりません。

2. パイプライン実行の設定を定義します。
 - a. **Runtime Configuration** リストから、パイプラインを実行するための関連するランタイム設定を選択します。
 - b. オプション: 必要に応じて、パイプラインパラメーターを設定します。パイプラインにパイプラインパラメーターを参照するノードが含まれている場合は、デフォルトのパラメーター値を変更できます。パラメーターが必須でデフォルト値がない場合は、値を入力する必要があります。
3. **OK** をクリックします。

検証

- パイプライン実行の出力アーティファクトを表示できます。アーティファクトは、指定されたオブジェクトストレージバケットに保存されます。

4.3.8. JupyterLab でのパイプラインのエクスポート


JupyterLab で作成したパイプラインをエクスポートできます。パイプラインをエクスポートすると、パイプラインは後で実行できるように準備されますが、すぐにアップロードまたは実行されるわけではありません。エクスポートプロセス中に、パッケージの依存関係はすべて S3 互換ストレージにアップロードされます。また、ターゲットランタイム用にパイプラインコードが生成されます。

パイプラインをエクスポートする前に、データサイエンスプロジェクトとパイプラインサーバーを作成する必要があります。パイプラインサーバーを作成したら、パイプラインサーバーと同じプロジェクト内にワークベンチを作成する必要があります。さらに、JupyterHub のパイプラインインスタンスにはランタイム設定が含まれている必要があります。データサイエンスプロジェクトの一部としてワークベンチを作成すると、デフォルトのランタイム設定が自動的に作成されます。ただし、OpenShift Data Science ダッシュボードの Jupyter タイルからノートブックを作成する場合は、JupyterLab でパイプラインをエクスポートする前にランタイム設定を作成する必要があります。ランタイム設定は、パイプラインインスタンスと S3 互換のクラウドストレージの接続情報を定義します。

前提条件

- OpenShift Pipelines Operator がインストールされている。
- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- ワークベンチを含むデータサイエンスプロジェクトを作成している。
- ワークベンチを含むデータサイエンスプロジェクト内にパイプラインサーバーを作成して設定している。
- S3 互換ストレージにアクセスできる。
- JupyterLab でパイプラインを作成している。
- JupyterLab の Pipeline Editor でパイプラインを開いている。
- パイプラインインスタンスにはランタイム設定が含まれている。
- Elyra 拡張機能 (Standard データサイエンス、TensorFlow、TrustyAI、または PyTorch) を含むノートブックイメージから Jupyter サーバーを作成し、起動している。

手順

1. Pipeline Editor ユーザーインターフェイスで、**Export Pipeline** () をクリックします。**Export Pipeline** ダイアログが表示されます。**Pipeline Name** フィールドには、パイプラインファイル名が自動的に入力されます。
2. パイプラインをエクスポートするための設定を定義します。

- a. **Runtime Configuration** リストから、パイプラインをエクスポートする関連するランタイム設定を選択します。
 - b. **Export Pipeline as** から適切なファイル形式を選択します。
 - c. **Export Filename** フィールドに、エクスポートされたパイプラインのファイル名を入力します。
 - d. **Replace if file already exists** チェックボックスをオンにして、エクスポートするパイプラインと同じ名前の既存のファイルを置換します。
 - e. オプション: 必要に応じて、パイプラインパラメーターを設定します。パイプラインにパイプラインパラメーターを参照するノードが含まれている場合は、デフォルトのパラメーター値を変更できます。パラメーターが必須でデフォルト値がない場合は、値を入力する必要があります。
3. **OK** をクリックします。

検証

- 指定したオブジェクトストレージバケットにエクスポートしたパイプラインを含むファイルを表示できます。

4.4. 関連情報

- [Kubeflow Pipelines v1 Documentation](#)
- [JupyterLab でのパイプラインの操作](#)

第5章 OPENSIFT DATA SCIENCE のモデルサービス

データサイエンティストとして、トレーニングされた機械学習モデルをデプロイし、インテリジェントなアプリケーションを実稼働環境で提供できます。モデルをデプロイした後、アプリケーションはデプロイされた API エンドポイントを使用してモデルにリクエストを送信できます。

5.1. モデルサーバーの設定

5.1.1. データサイエンスプロジェクトにモデルサーバーを追加する

OpenShift Data Science にデータサイエンスモデルを正常にデプロイするには、モデルサーバーを設定する必要があります。これには、デプロイされるレプリカの数、サーバーのサイズ、トークンの承認、およびプロジェクトへのアクセス方法の設定が含まれます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- モデルサーバーを追加できるデータサイエンスプロジェクトを作成している。
- モデルサーバーにカスタムのモデル提供ランタイムを使用する場合は、ランタイムを追加して有効にしている。[カスタムモデル提供ランタイムの追加](#) を参照してください。
- モデルサーバーでグラフィックスプロセッシングユニット (GPU) を使用する場合は、OpenShift Data Science で GPU サポートを有効にしている。[OpenShift Data Science での GPU サポートの有効化](#) を参照してください。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. モデルサーバーを設定するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Models and model servers** セクションで、**Add server** をクリックします。**Add model server** ダイアログが開きます。
4. **Model sever name** フィールドに、モデルサーバーの一意の名前を入力します。
5. **Serving runtime** リストから、OpenShift Data Science デプロイメントにインストールされ有効になっているモデル提供ランタイムを選択します。
6. **Number of model replicas to deploy** フィールドに値を指定します。
7. **Model server size** リストから、次のいずれかのサーバーサイズを選択します。
 - Small
 - Medium
 - Large

- Custom
8. オプション: 前の手順で **Custom** を選択した場合は、**Model server size** セクションで次を設定して、モデルサーバーをカスタマイズします。
 - a. **CPUs requested** フィールドで、モデルサーバーで使用する CPU の数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。
 - b. **CPU limit** フィールドに、モデルサーバーで使用する CPU の最大数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。
 - c. **Memory requested** フィールドに、モデルサーバーに要求されたメモリーをギビバイト (Gi) 単位で指定します。
 - d. **Memory limit** フィールドに、モデルサーバーの最大メモリー制限をギビバイト (Gi) 単位で指定します。
 9. オプション: **Model server GPUs** フィールドで、モデルサーバーで使用する GPU の数を指定します。



重要

OpenShift Data Science には、デフォルトで 2 つのバージョンの OpenVINO Model Server (OVMS) ランタイムが含まれています。GPU をサポートするバージョンとサポートしないバージョン。GPU を使用するには、**Serving runtime** リストから、表示名に **Supports GPUs** が含まれるバージョンを選択する必要があります。

モデルサーバーで **カスタム** モデル提供ランタイムを使用している場合は、カスタムランタイムが GPU をサポートし、GPU を使用するように適切に設定されていることを確認する必要があります。

10. オプション: **Model route** セクションで、**Make deployed models available through an external route** チェックボックスをオンにして、デプロイされたモデルを外部クライアントが利用できるようにします。
11. オプション: **Token authorization** セクションで、**Require token authentication** チェックボックスをオンにして、モデルサーバーにトークン認証を要求します。トークン認証の設定を完了するには、次のアクションを実行します。
 - a. **Service account name** フィールドに、トークンが生成されるサービスアカウント名を入力します。生成されたトークンは、モデルサーバーの設定時に作成され、**Token secret** フィールドに表示されます。
 - b. 追加のサービスアカウントを追加するには、**Add a service account** をクリックし、別のサービスアカウント名を入力します。
12. **Add** をクリックします。

検証

- 設定したモデルサーバーは、プロジェクトの詳細ページの **Models and model servers** セクションに表示されます。

5.1.2. カスタムのモデル提供ランタイムの追加

モデル提供ランタイムは、指定されたモデルサーバーおよびそれがサポートするモデルフレームワークとの統合を提供します。デフォルトでは、Red Hat OpenShift Data Science には OpenVINO Model Server ランタイムが含まれています。ただし、このランタイムがニーズを満たさない場合 (たとえば、特定のモデルフレームワークをサポートしていない場合) は、独自のカスタムランタイムを追加することもできます。

管理者は、OpenShift Data Science インターフェイスを使用して、カスタムのモデル提供ランタイムを追加および有効化できます。新しいモデルサーバーを作成するときに、有効なランタイムから選択できます。

前提条件

- OpenShift Data Science に管理者としてログインしている。
- [モデルサーバーをプロジェクトに追加する](#) 方法を理解していること。カスタムのモデル提供ランタイムを追加した場合は、そのランタイムを使用するように新しいモデルサーバーを設定する必要があります。
- [kserve/modelmesh-serving](#) リポジトリ内のサンプルランタイムを確認している。これらの例を **開始点** として使用できます。ただし、各ランタイムを OpenShift Data Science にデプロイするには、さらにいくつかの変更が必要です。必要な変更は、次の手順で説明します。



注記

OpenShift Data Science には、OpenVINO Model Server モデル提供ランタイムがデフォルトで含まれています。このランタイムを OpenShift Data Science に追加する必要はありません。

手順

1. OpenShift Data Science ダッシュボードから、**Settings > Serving runtimes** をクリックします。
Serving runtimes ページが開き、OpenShift Data Science デプロイメントにすでにインストールされ有効になっているモデル提供ランタイムが表示されます。デフォルトでは、OpenVINO Model Server ランタイムがプリインストールされ、OpenShift Data Science で有効になっています。
2. 新しいカスタムランタイムを追加するには、**Add serving runtime** をクリックします。
Add serving runtime ページが開きます。
3. 新しいランタイムの追加を開始するには、次の一連のアクションのいずれかを実行します。
 - **YAML ファイルをアップロードするには**
 - a. **Upload files** をクリックします。
ファイルブラウザが開きます。
 - b. ファイルブラウザで、コンピューター上の YAML ファイルを選択します。このファイルは、[kserve/modelmesh-serving](#) リポジトリからダウンロードしたサンプルランタイムの1つである可能性があります。
埋め込み YAML エディターが開き、アップロードしたファイルの内容が表示されます。
 - **エディターに YAML コードを直接入力するには**
 - a. **Start from scratch** をクリックします。

埋め込み YAML エディターがコンテンツなしで開きます。

- b. 埋め込みエディターに YAML コードを直接入力または貼り付けます。貼り付ける YAML は、[kserve/modelmesh-serving](#) リポジトリ内のサンプルランタイムの1つからコピーされる場合があります。
4. オプション: [kserve/modelmesh-serving](#) リポジトリにサンプルランタイムの1つを追加する場合は、次の変更を実行します。
 - a. YAML エディターで、ランタイムの **kind** フィールドを見つけます。このフィールドの値を **ServingRuntime** に更新します。
 - b. YAML エディターで、ランタイムの **containers.image** フィールドを見つけます。追加するランタイムに基づいて、フィールドを次のいずれかに更新します。

Nvidia Triton Inference Server

image: nvcr.io/nvidia/tritonserver:21.06.1-py3

Seldon Python MLServer

image: seldonio/mlserver:0.5.2

TorchServe

image: pytorch/torchserve:0.6.0-cpu

5. **metadata.name** フィールドで、追加するランタイムの値が一意であることを確認します (つまり、値がすでに追加したランタイムの値と同じではありません)。
6. オプション: 追加するランタイムのカスタム表示名を設定するには、次の例に示すように、**metadata.annotations.openshift.io/display-name** フィールドを追加し、値を指定します。

```
apiVersion: serving.kserve.io/v1alpha1
kind: ServingRuntime
metadata:
  name: mlserver-0.x
  annotations:
    openshift.io/display-name: MLServer
```



注記

ランタイムのカスタム表示名を設定しないと、OpenShift Data Science には **metadata.name** フィールドの値が表示されます。

7. **Add** をクリックします。
Serving runtimes ページが開き、インストールされているランタイムの更新されたリストが表示されます。追加したランタイムが自動的に有効になることを確認します。
8. オプション: カスタムランタイムを編集するには、アクションメニュー (:) をクリックし、**Edit** を選択します。



注記

デフォルトで OpenShift Data Science に含まれる OpenVINO Model Server ランタイムを直接編集できません。ただし、このランタイムを複製し、複製したバージョンの編集は可能です。その後、編集したクローンを新しいカスタムランタイムとして追加できます。これを行うには、OpenVINO Model Server の横にあるアクションメニューをクリックし、**Clone** を選択します。

検証

- 追加したモデル提供ランタイムは、**サービス提供ランタイム** ページに有効な状態で表示されません。

関連情報

- 追加したカスタムモデルサービスランタイムを使用するモデルサーバーを設定する方法については、[データサイエンスプロジェクトへのモデルサーバーの追加](#) を参照してください。

5.1.3. モデルサーバーの更新

デプロイされたレプリカの数、サーバーのサイズ、トークン認証、プロジェクトへのアクセス方法などの詳細を変更することで、データサイエンスプロジェクトのモデルサーバーを更新できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。
- モデルサーバーが割り当てられたデータサイエンスプロジェクトを作成している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. モデルサーバーの詳細を更新するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Models and model servers** セクションで、更新するモデルサーバーを見つけます。アクションメニュー (⋮) をクリックし、**Edit model server** を選択します。**Configure model server** ダイアログが開きます。
4. 次のように、モデルサーバーのプロパティを更新します。



注記

すでに設定されているモデルサーバーの **提供ランタイム** の選択を変更できません。これにより、すでにデプロイされたモデルをサポートしないランタイムへの変更が防止されます。

- a. **Model server name** フィールドに、モデルサーバーの新しい一意の名前を入力します。

- b. **Number of model replicas to deploy** フィールドに値を指定します。
- c. **Model server size** リストから、次のいずれかのサーバーサイズを選択します。
 - Small
 - Medium
 - Large
 - Custom
- d. オプション: 前の手順で **Custom** を選択した場合は、**Model server size** セクションで次を設定して、モデルサーバーをカスタマイズします。
 - i. **CPUs requested** フィールドで、モデルサーバーで使用する CPU の数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。
 - ii. **CPU limit** フィールドに、モデルサーバーで使用する CPU の最大数を指定します。このフィールドの横にあるリストを使用して、値をコアまたはミリコアで指定します。
 - iii. **Memory requested** フィールドに、モデルサーバーに要求されたメモリーをギビバイト (Gi) 単位で指定します。
 - iv. **Memory limit** フィールドに、モデルサーバーの最大メモリー制限をギビバイト (Gi) 単位で指定します。
- e. オプション: **Model server GPUs** フィールドで、モデルサーバーで使用する GPU の数を指定します。



重要

OpenShift Data Science には、デフォルトで 2 つのバージョンの OpenVINO Model Server (OVMS) ランタイムが含まれています。GPU をサポートするバージョンとサポートしないバージョン。GPU を使用するには、**Serving runtime** リストから、表示名に **Supports GPUs** が含まれるバージョンを選択する必要があります。

モデルサーバーで **カスタム** モデル提供ランタイムを使用している場合は、カスタムランタイムが GPU をサポートし、GPU を使用するように適切に設定されていることを確認する必要があります。

- f. オプション: **Model route** セクションで、**Make deployed models available through an external route** チェックボックスをオンにして、デプロイされたモデルを外部クライアントが利用できるようにします。
- g. オプション: **Token authorization** セクションで、**Require token authentication** チェックボックスをオンにして、モデルサーバーにトークン認証を要求します。トークン認証の設定を完了するには、次のアクションを実行します。
 - i. **Service account name** フィールドに、トークンが生成されるサービスアカウント名を入力します。生成されたトークンは、モデルサーバーの設定時に作成され、**Token secret** フィールドに表示されます。
 - ii. 追加のサービスアカウントを追加するには、**Add a service account** をクリックし、別のサービスアカウント名を入力します。

5. **Configure** をクリックします。

検証

- 更新したモデルサーバーは、Project Detail ページの **Models and model servers** セクションに表示されます。

5.1.4. モデルサーバーの削除

モデルをホストするためにモデルサーバーが必要なくなった場合は、データサイエンスプロジェクトからモデルサーバーを削除できます。



注記

モデルサーバーを削除すると、そのモデルサーバーでホストされているモデルも削除されます。その結果、アプリケーションはモデルを利用できなくなります。

前提条件

- データサイエンスプロジェクトと関連するモデルサーバーを作成している。
- モデルにアクセスするアプリケーションのユーザーに、モデルが利用できなくなることを通知している。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users**、**rhods-admins** など) に属している。

手順

1. OpenShift Data Science ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. モデルサーバーを削除するプロジェクトの名前をクリックします。プロジェクトの詳細ページが開きます。
3. **Models and model servers** セクションでモデルサーバーを削除するプロジェクトの横にあるアクションメニュー (⋮) をクリックし、**Delete model server** をクリックします。**Delete model server** ダイアログが開きます。
4. テキストフィールドにモデルサーバーの名前を入力して、削除することを確認します。
5. **Delete model server** をクリックします。

検証

- 削除したモデルサーバーは、プロジェクトの Details ページの **Models and model servers** セクションに表示されなくなります。

5.2. デプロイされたモデルの使用

5.2.1. OpenShift Data Science でのモデルのデプロイ

OpenShift Data Science にトレーニングされたモデルをデプロイし、それらをテストしてインテリジェントなアプリケーションに実装できます。モデルをデプロイすると、API を使用してアクセスできるサービスとして利用可能になります。これにより、データ入力に基づく予測を返すことができます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users** など) に属している。
- 関連するモデルサーバーを含むデータサイエンスプロジェクトを作成している。
- モデルがアクセスするデータ接続のフォルダーパスを把握している。

手順

1. {product-short} ダッシュボードから、**Data Science Projects** をクリックします。**Data science projects** のページが開きます。
2. デプロイするモデルが含まれるプロジェクトの名前をクリックします。プロジェクトの詳細が開きます。
3. **Models and model servers** セクションで、モデルサーバーの名前の横にある **Deploy model** をクリックします。**Deploy model** ダイアログが開きます。
4. モデルをデプロイするためのプロパティを次のように設定します。
 - a. **Model Name** フィールドに、デプロイするモデルの一意の名前を入力します。
 - b. **Model framework** リストから、モデルのフレームワークを選択します。



注記

Model framework リストには、モデルサーバーの設定時に指定したモデル提供ランタイムによってサポートされるフレームワークのみが表示されます。

- c. モデルの場所を指定するには、次の一連のアクションのいずれかを実行します。
 - **既存のデータ接続を使用するには**
 - i. **Existing data connection** を選択します。
 - ii. **Name** リストから、以前に定義したデータ接続を選択します。
 - iii. **Folder path** フィールドに、指定したデータソース内のモデルを含むフォルダーパスを入力します。
 - **新しいデータ接続を使用するには**
 - i. モデルがアクセスできる新しいデータ接続を定義するには、**New data connection** を選択します。
 - ii. **Name** フィールドに、データ接続の一意の名前を入力します。

- iii. **AWS_ACCESS_KEY_ID** フィールドに、Amazon Web Services (AWS) のアクセスキー ID を入力します。
 - iv. **AWS_SECRET_ACCESS_KEY** フィールドで指定した AWS アカウントのシークレットアクセスキーを入力します。
 - v. **AWS_S3_ENDPOINT** フィールドに AWS S3 ストレージのエンドポイントを入力します。
 - vi. **AWS_DEFAULT_REGION** フィールドに、AWS アカウントのデフォルトリージョンを入力します。
 - vii. **AWS_S3_BUCKET** フィールドに AWS S3 バケットの名前を入力します。
 - viii. **Folder path** フィールドに、データファイルが含まれる AWS S3 バケット内のフォルダーパスを入力します。
- d. **Deploy** をクリックします。

検証

- デプロイしたモデルは、ダッシュボードの **Model Serving** ページに表示されます。


5.2.2. デプロイされたモデルの表示

作業の結果を分析するために、Red Hat OpenShift Data Science にデプロイされたモデルのリストを表示できます。デプロイされたモデルとそのエンドポイントの現在のステータスも表示できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users** など) に属している。
- **Model Serving** ページに、アクティブでデプロイ済みのデータサイエンスモデルがある。

手順

1. OpenShift Data Science ダッシュボードで、**Model Serving** をクリックします。**Model Serving** ページが開きます。
2. デプロイされたモデルのリストを確認します。
推論エンドポイントは、**Deployed models** テーブルの **Inference endpoint** 列に表示されません。
3. オプション: 関連する行で **Copy** ボタン () をクリックし、モデルの推論エンドポイントをクリップボードにコピーします。

検証

- 以前にデプロイされたデータサイエンスモデルのリストが **Model Serving** ページに表示されません。

5.2.3. デプロイされたモデルのデプロイメントプロパティの更新

以前にデプロイされたモデルのデプロイメントプロパティを更新できます。これにより、モデルのデータ接続および名前を変更できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users** など) に属している。
- OpenShift Data Science にモデルをデプロイしている。

手順

1. OpenShift Data Science ダッシュボードで、**Model serving** をクリックします。
Model Serving ページが開きます。
2. 更新するデプロイメントプロパティを持つモデルの横にあるアクションメニュー (⋮) をクリックし、**Edit** をクリックします。
Deploy model ダイアログが開きます。
3. 次のようにモデルのデプロイメントプロパティを更新します。
 - a. **Model Name** フィールドに、モデルの一意の名前を新たに入力します。
 - b. **Model framework** リストから、モデルのフレームワークを選択します。



注記

Model framework リストには、モデルサーバーの設定時に指定したモデル提供ランタイムによってサポートされるフレームワークのみが表示されます。

- c. モデルの場所の指定方法を更新するには、次の一連のアクションのいずれかを実行します。
 - **既存のデータ接続を指定した場合**
 - i. **Folder path** フィールドに、指定したデータソース内のモデルを含むフォルダーパスを更新します。
 - **以前に新しいデータ接続を指定した場合**
 - i. **Name** フィールドで、データ接続の名前を更新します。
 - ii. **AWS_ACCESS_KEY_ID** フィールドで Amazon Web Services (AWS) のアクセスキー ID を更新します。
 - iii. **AWS_SECRET_ACCESS_KEY** フィールドで指定した AWS アカウントのシークレットアクセスキーを更新します。
 - iv. **AWS_S3_ENDPOINT** フィールドで AWS S3 ストレージのエンドポイントを更新します。

- v. **AWS_DEFAULT_REGION** フィールドで AWS アカウントのデフォルトリージョンを更新します。
 - vi. **AWS_S3_BUCKET** フィールドで AWS S3 バケットの名前を更新します。
 - vii. **Folder path** フィールドに、データファイルが含まれる AWS S3 バケット内のフォルダーパスを更新します。
- d. **Configure** をクリックします。

検証

- デプロイメントプロパティを更新したモデルが **Model Serving** ページに表示されます。

5.2.4. デプロイされたモデルの削除

以前にデプロイしたモデルを削除できます。これにより、不要になったデプロイ済みのモデルを削除できます。

前提条件

- Red Hat OpenShift Data Science にログインしている。
- 特殊な OpenShift Data Science グループを使用している場合は、OpenShift のユーザーグループ、または、管理者グループ (**rhods-users** など) に属している。
- モデルをデプロイしている。

手順

1. OpenShift Data Science ダッシュボードで、**Model serving** をクリックします。**Model Serving** ページが開きます。
2. 削除するデプロイモデルの横にあるアクションメニュー (⋮) をクリックし、**Delete** をクリックします。**Delete deployed model** ダイアログが開きます。
3. テキストフィールドにデプロイしたモデルの名前を入力し、削除することを確認します。
4. **Delete deployed model** をクリックします。

検証

- 削除したモデルは、**Model Serving** ページに表示されなくなります。

第6章 JUPYTER での一般的な問題の管理者向けトラブルシューティング

Jupyter、そのノートブック、またはノートブックサーバーに関連する Red Hat OpenShift Data Science でエラーが発生した場合は、このセクションを読み、問題の原因と解決方法を確認してください。

該当する問題が、このドキュメントやリリースノートに記載されていない場合は、Red Hat サポートに連絡してください。

6.1. ユーザーが JUPYTER にログインすると、404: PAGE NOT FOUND エラーが表示される

問題

専用の OpenShift Data Science ユーザーグループを設定している場合は、ユーザー名が、OpenShift Data Science のデフォルトのユーザーグループに追加されていない可能性があります。

診断

ユーザーがデフォルトユーザーグループの一部であるかどうかを確認します。

1. Jupyter へのアクセスが許可されるグループ名を見つけます。
 - a. OpenShift Dedicated Web コンソールにログインします。
 - a. **User Management** → **Groups** をクリックします。
 - b. ユーザーグループの名前 (**rhods-users** など) をクリックします。
そのグループの **Group details** ページが表示されます。
 1. グループの **Details** タブをクリックし、関連するグループの **Users** セクションに、Jupyter へのアクセス権限があるユーザーが含まれていることを確認します。

解決方法

- ユーザーが Jupyter へのアクセスが許可されるグループに追加されていない場合は、[Adding users for OpenShift Data Science](#) に従ってユーザーを追加します。
- Jupyter へのアクセスが許可されるグループにユーザーがすでに追加されている場合は、Red Hat サポートにお問い合わせください。

6.2. ユーザーのノートブックサーバーが起動しない

ユーザーのノートブックサーバーをホストする OpenShift Dedicated クラスタが十分なリソースにアクセスできないか、Jupyter Pod に障害が発生した可能性があります。

1. OpenShift Dedicated Web コンソールにログインします。
1. このユーザーのノートブックサーバー Pod を削除して再起動します。
 - a. **Workloads** → **Pods** をクリックし、**Project** を **rhods-notebooks** に設定します。
 - b. このユーザーに属するノートブックサーバー Pod (例: **jupyter-nb-<username>-***) を検索します。

ノートブックサーバー Pod が存在する場合は、ノートブックサーバー Pod で障害が断続的に発生した可能性があります。

ユーザーのノートブックサーバー Pod が存在しない場合は、診断を続行します。

2. 選択したノートブックサーバーイメージで必要なリソースに対して、OpenShift クラスターで現在利用可能なリソースを確認します。
クラスター内でのスケジューリングに CPU および RAM が十分にあるワーカーノードが利用可能な場合は、診断を続行します。
3. Jupyter Pod の状態を確認します。

解決方法

- ノートブックサーバー Pod で断続的に障害が発生した場合は、以下を行います。
 - a. ユーザーに属するノートブックサーバー Pod を削除します。
 - b. ユーザーにノートブックサーバーを再度開始するよう依頼します。
- ノートブックサーバーに、選択したノートブックサーバーイメージを実行するのに十分なリソースがない場合は、OpenShift クラスターにリソースを追加するか、小規模なイメージサイズを選択します。
- Jupyter Pod が **FAILED** 状態にある場合は、以下を実行します。
 - a. **jupyter-nb-*** Pod のログを取得し、詳細な評価のために Red Hat サポートに送信します。
 - b. **jupyter-nb-*** Pod を削除します。
- 以前の解決策がない場合は、Red Hat サポートにお問い合わせください。

6.3. ノートブックセルの実行時に DATABASE OR DISK IS FULL エラーまたは NO SPACE LEFT ON DEVICE エラーが表示される

問題

ノートブックサーバーのストレージ領域を使い果たした可能性があります。

診断

1. Jupyter にログインし、問題のあるユーザーに属するノートブックサーバーを起動します。
ノートブックサーバーが起動しない場合は、以下の手順を実行して、ユーザーのストレージ領域が不足しているかどうかを確認します。
 - a. OpenShift Dedicated Web コンソールにログインします。
 - a. **Workloads** → **Pods** をクリックし、**Project** を **rhods-notebooks** に設定します。
 - b. このユーザーに属するノートブックサーバー Pod (例: **jupyter-nb-`<idp>-<username>-*`**) をクリックします。
 - c. **Logs** をクリックします。以下のような行が表示された場合は、ユーザーが利用可能な容量を超えています。


```
Unexpected error while saving file: XXXX database or disk is full
```

```
Unexpected error while saving file: XXXX database or disk is full
```

解決方法

- 永続ボリュームを拡張し、ユーザーで利用可能なストレージを拡張します ([永続ボリュームの拡張](#))。
- ユーザーと連携し、ノートブックサーバーの `/opt/app-root/src` ディレクトリーから削除できるファイルを特定し、既存のストレージ領域を解放します。

第7章 JUPYTER での一般的な問題のトラブルシューティング

Jupyter、ノートブック、またはノートブックサーバーに関連する Red Hat OpenShift Data Science のエラーが表示される場合は、このセクションを読み、問題が発生する可能性があるかどうかを確認します。

その問題がこのドキュメントまたはリリースノートに記載されていない場合は、Red Hat サポートに連絡してください。

7.1. JUPYTER にログインすると 403: FORBIDDEN エラーが表示される

問題

管理者が専用の OpenShift Data Science ユーザーグループを設定している場合は、ユーザー名が、OpenShift Data Science のデフォルトのユーザーグループまたはデフォルトの管理者グループに追加されていない可能性があります。

解決方法

Contact your administrator so that they can add you to the correct group/s.

7.2. ノートブックサーバーが起動しない

ノートブックサーバーをホストする OpenShift Dedicated クラスタが十分なリソースにアクセスできないか、Jupyter Pod に障害が発生した可能性があります。

解決方法

OpenShift Container Platform の **Events** セクションのログで、問題に関連するエラーメッセージの有無を確認します。以下に例を示します。

```
Server requested
2021-10-28T13:31:29.830991Z [Warning] 0/7 nodes are available: 2 Insufficient memory,
2 node(s) had taint {node-role.kubernetes.io/infra: }, that the pod didn't tolerate, 3 node(s) had taint
{node-role.kubernetes.io/master: },
that the pod didn't tolerate.
```

追加のチェックを実行できるように、関連するエラーメッセージの詳細を管理者に連絡してください。

7.3. ノートブックセルの実行時に DATABASE OR DISK IS FULL エラーまたは NO SPACE LEFT ON DEVICE エラーが表示される

問題

ノートブックサーバーでストレージ領域が不足している可能性があります。

解決方法

詳細を確認できるように管理者に問い合わせてください。