



Red Hat OpenShift AI Self-Managed 2.10

在断开连接的环境中安装和卸载 OpenShift AI Self-Managed

在断开连接的环境中安装和卸载 OpenShift AI Self-Managed

Red Hat OpenShift AI Self-Managed 2.10 在断开连接的环境中安装和卸载 OpenShift AI Self-Managed

在断开连接的环境中安装和卸载 OpenShift AI Self-Managed

法律通告

Copyright © 2024 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.

摘要

在断开连接的环境中，在 OpenShift Container Platform 集群上安装并卸载 OpenShift AI Self-Managed。

目录

前言	3
第 1 章 OPENSIFT AI SELF-MANAGED 的架构	4
第 2 章 了解更新频道	5
第 3 章 在断开连接的环境中部署 OPENSIFT AI	6
3.1. OPENSIFT AI 自我管理的要求	6
3.2. 为 OPENSIFT CONTAINER PLATFORM 添加管理用户	8
3.3. 为断开连接的安装将镜像镜像到私有 REGISTRY	8
3.4. 安装 RED HAT OPENSIFT AI OPERATOR	12
3.5. 安装和管理 RED HAT OPENSIFT AI 组件。	16
3.6. 配置 OPENSIFT AI OPERATOR 日志记录器	23
第 4 章 准备 OPENSIFT AI 用于 IBM CLOUD PAK FOR DATA	27
4.1. 使用 CLI 安装 RED HAT OPENSIFT AI OPERATOR	27
4.2. 使用 CLI 管理 RED HAT OPENSIFT AI 组件	30
4.3. 编辑模型推断配置	33
第 5 章 使用证书	36
5.1. 了解 OPENSIFT AI 中的证书	36
5.2. 添加 CA 捆绑包	38
5.3. 删除 CA 捆绑包	39
5.4. 从命名空间中删除 CA 捆绑包	41
5.5. 管理证书	41
5.6. 在 OPENSIFT AI 组件中使用自签名证书	43
第 6 章 在 OPENSIFT AI 中启用 GPU 支持	47
第 7 章 访问仪表板	49
第 8 章 卸载 RED HAT OPENSIFT AI SELF-MANAGED	50
8.1. 了解卸载过程	50
8.2. 使用 CLI 卸载 OPENSIFT AI 自我管理	51

前言

了解如何使用 OpenShift 命令行界面和 Web 控制台在断开连接的环境中在 OpenShift Container Platform 集群上安装 Red Hat OpenShift AI Self-Managed。要卸载产品，了解如何使用推荐的命令行界面 (CLI) 方法。



注意

红帽建议在集群中只安装一个 OpenShift AI 实例。

不建议在与 Red Hat OpenShift AI Add-on 相同的集群中安装 Red Hat OpenShift AI Operator。

第 1 章 OPENSIFT AI SELF-MANAGED 的架构

Red Hat OpenShift AI Self-Managed 是一个 Operator，可在自我管理的环境（如 Red Hat OpenShift Container Platform）上提供。

OpenShift AI 集成了以下组件和服务：

- 在服务层：

OpenShift AI 仪表盘

此面向客户的仪表盘会显示 OpenShift AI 环境的可用和已安装的应用程序，以及学习资源，如教程、快速启动示例和文档等。管理用户可访问管理用户、集群、笔记本镜像、加速器配置集和模型服务运行时的功能。数据科学家可以使用控制面板创建项目来整理其数据科学工作。

模型服务

数据科学家可以部署受培训的机器学习模型，为生产环境中的智能应用程序提供服务。部署后，应用程序可以使用其部署的 API 端点向模型发送请求。

数据科学项目(data Science)管道

数据科学家可以使用 Docker 容器通过 Data Science Pipelines 2.0 构建可移植机器学习(ML)工作流。通过数据科学管道，数据科学家可以在开发其数据科学模型时自动化工作流。

Jupyter（自助管理的）

一个自我管理的应用程序，允许数据科学家自行配置笔记本服务器环境并在 JupyterLab 中开发机器学习模型。

分布式工作负载

数据科学家可以并行使用多个节点来更迅速地培训机器学习模型或处理数据。这种方法可显著减少任务完成时间，并允许使用更大的数据集和更复杂的模型。

- 在管理层：

Red Hat OpenShift AI Operator

一个 meta-operator，它部署和维护属于 OpenShift AI 的所有组件和子 Operator。

监控服务

Prometheus 从 OpenShift AI 收集指标数据以进行监控。

在 OpenShift Container Platform 集群中安装 Red Hat OpenShift AI Operator 时，会创建以下新项目：

- **redhat-ods-operator** 项目包含 Red Hat OpenShift AI Operator。
- **redhat-ods-applications** 项目将安装仪表盘和 OpenShift AI 的其他必要组件。
- **redhat-ods-monitoring** 项目包含用于监控的服务。
- **rhods-notebooks** 项目是默认部署笔记本环境的位置。

您或您的数据科学家必须为将使用机器学习模式的应用创建额外的项目。

不要在与 OpenShift AI 关联的命名空间中安装独立软件供应商 (ISV) 的应用程序。

第 2 章 了解更新频道

您可以使用更新频道来指定您要将在 Operator 更新至的 Red Hat OpenShift AI 次版本。更新频道还允许您通过 **fast**, **stable**, **stable -x.y**, **eus-x.y**, 和 **alpha** 频道选项选择您的更新的时间和级别。

已安装的 Operator 的订阅指定更新频道，用于跟踪和接收 Operator 的更新。您可以更改更新频道，以开始跟踪并从更新频道接收更新。有关发行频率以及与每个可用更新频道关联的生命周期的更多信息，请参阅 [Red Hat OpenShift AI Self-Managed Life Cycle](#)。

Channel	支持	发行频率	推荐的环境
fast	一个月的完全支持	每个月	可访问最新产品功能的生产环境。 使用自动更新选择此流频道，以避免每月手动升级。
stable	三个完全支持	每三个月	生产环境，其稳定性优先于新功能可用性。 使用自动更新选择此流频道来访问最新的稳定版本，并避免手动升级。
stable -x.y	七月的完全支持	每三个月	生产环境，其稳定性优先于新功能可用性。 选择编号的 stable 频道（如 stable-2.10 ）来计划和升级到下一个稳定版本，同时保持您的部署获得全面支持。
eus-x.y	持续 7 个月的完全支持后，延长的更新支持将持续 7 个月	每 9 个月	在 7 个月内无法升级的企业级环境。 如果您优先选择对新功能可用性的稳定性，请选择此流频道。
alpha	一个月的完全支持	每个月	具有早期访问功能的开发环境可能无法正常工作。 选择此频道以使用早期访问功能来测试功能并在开发过程中提供反馈。红帽产品服务等级协议 (SLA) 不支持 early-access 的功能。 有关红帽技术预览功能支持范围的更多信息，请参阅 技术预览功能支持范围 。 有关 Red Hat Developer 技术预览功能支持范围的更多信息，请参阅 开发人员技术预览功能支持范围 。



注意

embedded 和 **beta** 频道是将在以后的版本中删除的传统频道。不要为新的 Operator 安装选择 **内嵌** 或 **beta** 频道。

第 3 章 在断开连接的环境中部署 OPENSIFT AI

请阅读本节，了解如何在断开连接的环境中为数据科学家部署 Red Hat OpenShift AI 作为开发和测试环境。断开连接的集群位于受限网络中，通常在防火墙后面。在这种情况下，集群无法访问红帽提供的 OperatorHub 源所在的远程 registry。相反，Red Hat OpenShift AI Operator 可以使用私有 registry 镜像到断开连接的环境中。

在断开连接的环境中安装 OpenShift AI 涉及以下高级别任务：

1. 确认 OpenShift Container Platform 集群满足所有要求。请参阅 [OpenShift AI Self-Managed 的要求](#)。
2. 为 OpenShift Container Platform 添加管理用户。请参阅 [为 OpenShift Container Platform 添加管理用户](#)。
3. 将镜像镜像(mirror)到私有 registry。对于 [断开连接的安装](#)，请参阅 [将镜像镜像到私有 registry](#)。
4. 安装 Red Hat OpenShift AI Operator。请参阅 [安装 Red Hat OpenShift AI Operator](#)。
5. 安装 OpenShift AI 组件。请参阅 [安装和管理 Red Hat OpenShift AI 组件](#)。
6. 配置用户和管理员组，为用户提供 OpenShift AI 的访问权限。请参阅 [添加用户](#)。
7. 为您的用户提供部署 OpenShift AI 的 OpenShift Container Platform 集群的 URL。请参阅 [访问 OpenShift AI 仪表盘](#)。

3.1. OPENSIFT AI 自我管理的要求

在断开连接的环境中，您必须在 Red Hat OpenShift Container Platform 集群上安装 Red Hat OpenShift AI：

产品订阅

- 您必须有 Red Hat OpenShift AI Self-Managed 订阅。

请联系您的红帽客户经理购买新的订阅。如果您还没有帐户管理器，请在 <https://www.redhat.com/en/contact> 中填写表单以请求一个。

集群管理员对 OpenShift 集群的访问权限

- 您必须有一个具有集群管理员访问权限的 OpenShift 集群。按照 OpenShift Container Platform 文档使用现有集群或创建集群：[断开连接的安装镜像](#)。
- 安装集群后，按照 OpenShift Container Platform 文档配置 Cluster Samples Operator：[为受限集群配置 Samples Operator](#)。
- 在安装 Operator 时，您的集群必须至少有 2 个 worker 节点，至少有 8 个 CPU 和 32 GiB RAM 用于 OpenShift AI。为确保 OpenShift AI 可用，除了最低要求外，还需要额外的集群资源。
- 您的集群配置了可动态置备的默认存储类。
运行 `oc get storageclass` 命令确认配置了默认存储类。如果名称旁边没有被标记为 **(默认)** 的存储类，请按照 OpenShift Container Platform 文档配置默认存储类：[更改默认存储类](#)。有关动态置备的更多信息，请参阅[动态置备](#)。
- Open Data Hub 不能安装在集群中。

有关管理组成 OpenShift 集群的机器的更多信息，请参阅[机器管理概述](#)。

为 OpenShift Container Platform 配置身份提供程序

- Red Hat OpenShift AI 使用与 Red Hat OpenShift Container Platform 相同的身份验证系统。有关配置身份提供程序的更多信息，请参阅[了解身份提供程序配置](#)。
- 使用具有 **cluster-admin** 角色的用户访问集群；不允许 **kubeadmin** 用户。

镜像机器上的互联网访问

除了互联网访问外，还必须访问以下域来镜像 OpenShift AI Self-Managed 安装所需的镜像：

- cdn.redhat.com
- subscription.rhn.redhat.com
- registry.access.redhat.com
- registry.redhat.io
- quay.io
 - 对于基于 CUDA 的镜像，必须访问以下域：
- ngc.download.nvidia.cn
- developer.download.nvidia.com

数据科学项目准备

- Data Science Pipelines (DSP) 2.0 包含 Argo 工作流的安装。如果一个现有 Argo 工作流安装没有在集群中的 DSP 安装，则安装 OpenShift AI 后将禁用数据科学管道。在安装 OpenShift AI 前，请确保集群没有现有由 DSP 安装的 Argo 工作流安装，或删除 Argo 工作流的独立安装。
- 在断开连接的环境中执行管道前，您必须将镜像上传到私有 registry。如需更多信息，请参阅[镜像以在受限环境中运行管道](#)。
- 您可以将管道工件存储在 S3 兼容对象存储桶中，以便不使用本地存储。要做到这一点，您必须首先对存储帐户上的 S3 存储桶配置写入访问权限。

安装 KServe 依赖项

- 要支持 KServe 组件（由 single-model 服务平台用来为大型模型提供服务），还必须为 Red Hat OpenShift Serverless 和 Red Hat OpenShift Service Mesh 安装 Operator 并执行额外的配置。如需更多信息，请参阅[Serving 大模型](#)。
- 如果要为 single-model 服务平台添加授权供应商，您必须安装 **Red Hat - Authorino** Operator。如需更多信息，请参阅[单模式服务平台添加授权供应商](#)。

访问对象存储

- OpenShift AI 组件需要或可以使用 S3 兼容对象存储，如 AWS S3、MinIO、Ceph 或 IBM Cloud Storage。对象存储是一种数据存储机制，允许用户作为对象或文件访问其数据。S3 API 是基于 HTTP 访问对象存储服务的可识别标准。

- 对象存储必须可以被 OpenShift Container Platform 集群访问。在与集群相同的断开连接的网络中部署对象存储。
- 以下组件需要对象存储：
 - 单一或多模式服务平台，以部署存储模型。请参阅 [在单模式服务平台上部署模型或使用多型号服务平台部署模型](#)。
 - 数据科学管道、存储工件、日志和中间结果。请参阅 [配置管道服务器](#) 和 [关于管道日志](#)。
- 以下组件可以使用对象存储：
 - 工作台，用于访问大型数据集。请参阅 [在数据科学项目中添加数据连接](#)。
 - 分布式工作负载，用于从和推送结果中提取输入数据。请参阅 [从数据科学管道运行分布式数据科学工作负载](#)。
 - 管道中执行的代码。例如，要将生成的模型存储在对象存储中。请参阅 [Jupyterlab 中的管道概述](#)。

3.2. 为 OPENSIFT CONTAINER PLATFORM 添加管理用户

在您可以为数据科学家用户安装和配置 OpenShift AI 前，您必须定义管理用户。只有具有集群管理员特权的用户才能安装和配置 OpenShift AI。

有关创建集群管理员用户的更多信息，请参阅 OpenShift Container Platform 文档中的创建集群管理员。https://access.redhat.com/documentation/zh-cn/openshift_container_platform/4.15/html/authentication_and_authorization/using-rbac#creating-cluster-admin-using-rbac

3.3. 为断开连接的安装将镜像镜像到私有 REGISTRY

您可以通过将所需的容器镜像镜像到私有容器 registry，在断开连接的环境中将 Red Hat OpenShift AI Operator 安装到 OpenShift 集群。将镜像镜像到容器 registry 后，您可以使用 OperatorHub 安装 Red Hat OpenShift AI Operator。

您可以将 *mirror registry 用于 Red Hat OpenShift*，它是一个小型容器 registry，用作在断开连接的环境中为 OpenShift AI 镜像所需的容器镜像的目标。如果安装环境中已有另一个容器 registry，则使用 mirror registry for Red Hat OpenShift 是可选的。

先决条件

- 集群管理员可以访问正在运行的 OpenShift Container Platform 集群，版本 4.12 或更高版本。
- 您有 Red Hat OpenShift Cluster Manager 的凭证(<https://console.redhat.com/openshift/>)。
- 您的镜像机器正在运行 Linux，有 100 GB 的可用空间，并可访问互联网，以便它可以获取镜像来填充镜像存储库。
- 已安装 OpenShift CLI (**oc**)。
- 如果您计划使用 NVIDIA GPU，已镜像并部署了 NVIDIA GPU Operator。请参阅 OpenShift Container Platform 文档中的配置 [NVIDIA GPU Operator](#) 部分。
- 如果您计划使用数据科学管道，已镜像 OpenShift Pipelines operator。

- 如果您计划使用 single-model 服务平台为大型模型提供服务，您已对 Red Hat OpenShift Serverless 和 Red Hat OpenShift Service Mesh 的 Operator 进行镜像。如需更多信息，请参阅 [Serving 大模型](#)。
- 如果您计划使用分布式工作负载组件，已镜像 Ray 集群镜像。

流程

1. 创建镜像 registry。请参阅 OpenShift Container Platform 文档中的使用 [Red Hat OpenShift 创建带有 mirror registry 的镜像 registry](#)。
2. 要镜像 registry 镜像，请在运行 Linux 的镜像机器上安装 **oc-mirror** OpenShift CLI 插件（版本 4.12 或更高版本）。请参阅 [OpenShift Container Platform 文档中的安装 oc-mirror OpenShift CLI 插件](#)。



重要

在 4.12 版本前 **oc-mirror** 版本不允许镜像提供的完整镜像集配置。

3. 创建容器镜像 registry 凭证文件，允许将镜像从红帽镜像到您的镜像。请参阅 OpenShift Container Platform [文档中的配置允许镜像镜像的凭证](#)。
4. 从 [断开连接的安装程序帮助程序](#) 存储库打开示例镜像设置配置文件 (`rhoai-<version>.md`) 并检查其内容。
5. 使用示例镜像设置配置文件，创建一个名为 **imageset-config.yaml** 的文件，并使用部署中镜像设置配置的值填充。
 - 要查看可用的 OpenShift 版本列表，请运行以下命令：这可能需要几分钟。如果命令返回错误，请重复 [配置允许镜像镜像的凭证](#) 的步骤。

```
oc-mirror list operators
```

- 要查看特定版本的 OpenShift Container Platform 中的软件包的可用频道（如 4.15），请运行以下命令：

```
oc-mirror list operators --catalog=registry.redhat.io/redhat/redhat-operator-index:v4.15 --package=<package-name>
```

- 有关订阅更新频道的详情，[请参阅了解更新频道](#)。



重要

示例镜像设置配置仅用于演示目的，可能需要根据部署进行进一步更改。

要识别最适合您的部署的属性，请使用 **oc-mirror** 插件为 [断开连接的安装镜像文档和用例](#)。

您的 **imageset-config.yaml** 应该类似以下示例，其中数据科学项目需要 **openshift-pipelines-operator-rh**，KServe 组件需要 **serverless-operator** 和 **servicemeshoperator**。

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
archiveSize: 4
```

```

storageConfig:
  registry:
    imageURL: registry.example.com:5000/mirror/oc-mirror-metadata
    skipTLS: false
  mirror:
    operators:
      - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.15
        packages:
          - name: rhods-operator
            channels:
              - name: stable
                minVersion: 2.10.0
                maxVersion: 2.10.0
          - name: openshift-pipelines-operator-rh
            channels:
              - name: stable
          - name: serverless-operator
            channels:
              - name: stable
          - name: servicemeshoperator
            channels:
              - name: stable

```

6. 将指定的镜像设置配置下载到镜像机器上的本地文件中：

- 将 **mirror-rhoai** 替换为您要输出镜像集文件的目标目录。
- 目标目录路径必须以 **file://** 开头。
- 下载可能需要几分钟时间。

```
$ oc mirror --config=./imageset-config.yaml file://mirror-rhoai
```

提示

如果 **tls: failed to verify certificate: x509: certificate signed by unknown authority** 错误被返回，并且您想要忽略它，请在镜像设置配置文件中将 **skipTLS** 设置为 **true**，然后再次运行命令。

7. 验证镜像设置 **.tar** 文件是否已创建：

```
$ ls mirror-rhoai
mirror_seq1_000000.tar mirror_seq1_000001.tar
```

如果在镜像设置配置文件中指定了 **archiveSize** 值，则镜像集可能会划分为多个 **.tar** 文件。

8. 可选：验证镜像集 **.tar** 文件的总大小大约为 75 GB：

```
$ du -h --max-depth=1 ./mirror-rhoai/
```

如果镜像集的总大小明显低于 75 GB，请再次运行 **oc mirror** 命令。

9. 将生成的镜像集的内容上传到目标镜像 registry 中：

- 将 **mirror-rhoai** 替换为包含镜像设置 **.tar** 文件的目录。

- 将 **registry.example.com:5000** 替换为您的镜像 registry。

```
$ oc mirror --from=../mirror-rhoai docker://registry.example.com:5000
```

提示

如果 **tls: 无法验证 certificate: x509: certificate signed by unknown authority** 错误, 并且您要忽略它, 请运行以下命令:

```
$ oc mirror --dest-skip-tls --from=../mirror-rhoai docker://registry.example.com:5000
```

10. 以具有 **cluster-admin** 角色的用户身份登录到目标 OpenShift 集群。
11. 验证 **ImageContentSourcePolicy** 和 **CatalogSource** 资源是否存在 YAML 文件:
 - 将 **<results-directory>** 替换为您的结果目录的名称。

```
$ ls oc-mirror-workspace/<results-directory>/
catalogSource-cs-redhat-operator-index.yaml
charts
imageContentSourcePolicy.yaml
mapping.txt
release-signatures
```

12. 将生成的 **ImageContentSourcePolicy** 和 **CatalogSource** 资源安装到集群中:
 - 将 **<results-directory>** 替换为您的结果目录的名称。

```
$ oc apply -f ./oc-mirror-workspace/<results-directory>/imageContentSourcePolicy.yaml
$ oc apply -f ./oc-mirror-workspace/<results-directory>/catalogSource-cs-redhat-operator-index.yaml
```

验证

- 验证 **CatalogSource** 和 pod 是否已成功创建:

```
$ oc get catalogsource,pod -n openshift-marketplace
```

这应该至少返回一个目录和两个 pod。

- 检查 OperatorHub 中是否存在 Red Hat OpenShift AI Operator:
 - a. 登录到 OpenShift Container Platform 集群 Web 控制台。
 - b. 点 **Operators → OperatorHub**。
OperatorHub 页面将打开。
 - c. 确认显示了 Red Hat OpenShift AI Operator。
- 如果您镜像了额外的 Operator, 如 OpenShift Pipelines、Red Hat OpenShift Serverless 或 Red Hat OpenShift Service Mesh, 请检查这些 Operator 是否存在 OperatorHub。

其他资源

断开连接的安装镜像

3.4. 安装 RED HAT OPENSIFT AI OPERATOR

本节介绍如何使用命令行界面(CLI)和 OpenShift Web 控制台在 OpenShift Container Platform 集群上安装 Red Hat OpenShift AI Operator。



注意

如果要从以前的 OpenShift AI 版本升级，而不是执行新安装，请参阅 [在断开连接的环境中升级 OpenShift AI](#)。



注意

如果 OpenShift 集群使用代理访问互联网，您可以配置 Red Hat OpenShift AI Operator 的代理设置。如需更多信息，请参阅 [覆盖 Operator 的代理设置](#)。

3.4.1. 使用 CLI 安装 Red Hat OpenShift AI Operator

以下流程演示了如何使用 OpenShift 命令行界面(CLI)在 OpenShift Container Platform 集群上安装 Red Hat OpenShift AI Operator。您必须在集群中安装 OpenShift AI 组件前安装 Operator。

先决条件

- 您有一个正在运行的 OpenShift Container Platform 集群，版本 4.12 或更高版本，带有可动态置备的默认存储类。
- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅 [安装 OpenShift CLI](#)。
- 您已将所需的容器镜像镜像到私有 registry。对于 [断开连接的安装](#)，请参阅 [将镜像镜像到私有 registry](#)。

流程

1. 打开一个新的终端窗口。
2. 按照以下步骤，以集群管理员身份登录到 OpenShift Container Platform 集群：
 - a. 在 OpenShift Web 控制台右上角，单击您的用户名，然后选择 **Copy login command**。
 - b. 登录后，单击 **Display token**。
 - c. **使用此令牌命令复制登录，并将它粘贴到 OpenShift 命令行界面(CLI)中。**

```
$ oc login --token=<token> --server=<openshift_cluster_url>
```

3. 通过执行以下操作作为 Operator 安装创建一个命名空间：

- a. 创建一个命名空间 YAML 文件，如 **rhods-operator-namespace.yaml**。

```
apiVersion: v1
```

```
kind: Namespace
metadata:
  name: redhat-ods-operator ❶
```

- ❶ **redhat-ods-operator** 是 Operator 推荐的命名空间。

- b. 在 OpenShift Container Platform 集群中创建命名空间。

```
$ oc create -f rhods-operator-namespace.yaml
```

您会看到类似如下的输出：

```
namespace/redhat-ods-operator created
```

4. 通过执行以下操作为 Operator 安装创建一个 operator 组：

- a. 创建一个 **OperatorGroup** 对象自定义资源 (CR) 文件，如 **rhods-operator-group.yaml**。

```
apiVersion: operators.coreos.com/v1
kind: OperatorGroup
metadata:
  name: rhods-operator
  namespace: redhat-ods-operator ❶
```

- ❶ 您必须指定您此流程中创建的同命名空间。

- b. 在 OpenShift Container Platform 集群中创建 **OperatorGroup** 对象。

```
$ oc create -f rhods-operator-group.yaml
```

您会看到类似如下的输出：

```
operatorgroup.operators.coreos.com/rhods-operator created
```

5. 通过执行以下操作为 Operator 安装创建订阅：

- a. 创建一个 **Subscription** 对象 CR 文件，如 **rhods-operator-subscription.yaml**。

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: rhods-operator
  namespace: redhat-ods-operator ❶
spec:
  name: rhods-operator
  channel: stable ❷
  source: cs-redhat-operator-index
  sourceNamespace: openshift-marketplace
```

- ❶ 您必须指定您此流程中创建的同命名空间。

❷

对于 **频道**，请选择 **fast**, **stable**, **stable -x.y** 或 **alpha** 的值。有关选择更新频道的更多信息，[请参阅了解更新频道](#)。

- b. 在 OpenShift Container Platform 集群中创建 **Subscription** 对象来安装 Operator。

```
$ oc create -f rhods-operator-subscription.yaml
```

您会看到类似如下的输出：

```
subscription.operators.coreos.com/rhods-operator created
```

验证

- 在 OpenShift Container Platform Web 控制台中，点 **Operators** → **Installed Operators**，并确认 Red Hat OpenShift AI Operator 显示了以下状态之一：
 - **Installing** - 安装正在进行中；等待它变为 **Succeeded**。这可能需要几分钟。
 - **Succeeded** - 安装成功。
- 在 Web 控制台中，点 **Home** → **Projects** 并确认以下项目命名空间可见并列为 **Active**：
 - **redhat-ods-applications**
 - **redhat-ods-monitoring**
 - **redhat-ods-operator**

其他资源

- [安装和管理 Red Hat OpenShift AI 组件](#)。
- [添加用户](#)
- [在集群中添加 Operator](#)

3.4.2. 使用 Web 控制台安装 Red Hat OpenShift AI Operator

以下流程演示了如何使用 OpenShift Container Platform Web 控制台在集群中安装 Red Hat OpenShift AI Operator。您必须在集群中安装 OpenShift AI 组件前安装 Operator。

先决条件

- 您有一个正在运行的 OpenShift Container Platform 集群，版本 4.12 或更高版本，带有可动态置备的默认存储类。
- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 您已将所需的容器镜像镜像到私有 registry。对于 [断开连接的安装](#)，[请参阅将镜像镜像到私有 registry](#)。

流程

1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。

2. 在 Web 控制台中，点 **Operators** → **OperatorHub**。
3. 在 **OperatorHub** 页面中，通过滚动浏览可用的 Operator 或在 **Filter by keyword** 框中输入 *Red Hat OpenShift AI* 来找到 Red Hat OpenShift AI。
4. 点 **Red Hat OpenShift AI** 标题。**Red Hat OpenShift AI Information** 窗格将打开。
5. 选择一个频道。有关订阅更新频道的详情，[请参阅了解更新频道](#)。
6. 选择 **版本**。
7. 点 **Install**。此时会打开 **Install Operator** 页面。
8. 根据需要查看或更改所选频道和版本。
9. 对于 **Installation 模式**，请注意唯一可用的值为 **All namespaces on the cluster (default)**。此安装模式使 Operator 可供集群中的所有命名空间使用。
10. 对于 **Installed Namespace**，选择 **Operator recommended Namespace: redhat-ods-operator**。
11. 对于 **更新批准**，请选择以下更新策略之一：
 - **自动**：您的环境会尝试根据镜像的内容在可用时安装新的更新。
 - **手动**：集群管理员必须在安装开始前批准任何新更新。



重要

默认情况下，Red Hat OpenShift AI Operator 遵循一个后续更新过程。这意味着，如果当前版本和目标版本之间存在多个版本，Operator Lifecycle Manager (OLM) 会将 Operator 升级到每个中间版本，然后再将其升级到最终的目标版本。

如果配置自动升级，OLM 会自动将 Operator 升级到 *最新的可用版本*。如果配置手动升级，集群管理员必须手动批准当前版本和最终目标版本之间的每个后续更新。

有关支持的版本的详情，请查看 [Red Hat OpenShift AI 生命周期](#)。

12. 点 **Install**。
此时会出现 **Installing Operators** 窗格。安装完成后，Operator 名称旁边会出现一个复选标记。

验证

- 在 OpenShift Container Platform Web 控制台中，点 **Operators** → **Installed Operators**，并确认 Red Hat OpenShift AI Operator 显示了以下状态之一：
 - **Installing** - 安装正在进行中；等待它变为 **Succeeded**。这可能需要几分钟。
 - **Succeeded** - 安装成功。
- 在 Web 控制台中，点 **Home** → **Projects** 并确认以下项目命名空间可见并列为 **Active**：
 - **redhat-ods-applications**
 - **redhat-ods-monitoring**

- **redhat-ods-operator**

其他资源

- [安装和管理 Red Hat OpenShift AI 组件。](#)
- [添加用户](#)
- [在集群中添加 Operator](#)

3.5. 安装和管理 RED HAT OPENSIFT AI 组件。

以下流程演示了如何使用命令行界面(CLI)和 OpenShift Container Platform Web 控制台在 OpenShift Container Platform 集群中安装和管理 Red Hat OpenShift AI 的组件。

3.5.1. 使用 CLI 安装 Red Hat OpenShift AI 组件

以下流程演示了如何使用 OpenShift 命令行界面(CLI)在 OpenShift Container Platform 集群上安装 Red Hat OpenShift AI 的特定组件。



重要

以下流程描述了如何创建和配置 **DataScienceCluster** 对象，以作为新安装的一部分安装 Red Hat OpenShift AI 组件。但是，如果您从 OpenShift AI 版本 1（以前为 OpenShift Data Science）升级，升级过程会自动创建默认的 **DataScienceCluster** 对象。如果您从以前的次版本升级，升级过程将使用之前版本的 **DataScienceCluster** 对象中的设置。要检查 **DataScienceCluster** 对象并更改 Red Hat OpenShift AI 组件的安装状态，请参阅使用 [Web 控制台更新 Red Hat OpenShift AI 组件的](#) 安装状态。

先决条件

- Red Hat OpenShift AI Operator 已安装在 OpenShift Container Platform 集群中。请参阅[安装 Red Hat OpenShift AI Operator](#)。
- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅[安装 OpenShift CLI](#)。

流程

1. 打开一个新的终端窗口。
2. 按照以下步骤，以集群管理员身份登录到 OpenShift Container Platform 集群：
 - a. 在 OpenShift Web 控制台右上角，单击您的用户名，然后选择 **Copy login command**。
 - b. 登录后，单击 **Display token**。
 - c. 使用此令牌命令复制登录，并将它粘贴到 OpenShift 命令行界面(CLI)中。

```
$ oc login --token=<token> --server=<openshift_cluster_url>
```

3. 创建 **DataScienceCluster** 对象自定义资源 (CR) 文件，如 **rhods-operator-dsc.yaml**。

```

apiVersion: datasciencecluster.opendatahub.io/v1
kind: DataScienceCluster
metadata:
  name: default-dsc
spec:
  components:
    codeflare:
      managementState: Removed
    dashboard:
      managementState: Removed
    datasciencepipelines:
      managementState: Removed
    kserve:
      managementState: Removed ❶ ❷
    kueue:
      managementState: Removed
    modelmeshserving:
      managementState: Removed
    ray:
      managementState: Removed
    workbenches:
      managementState: Removed

```

- ❶ 要完全安装 KServe 组件（由单模型服务平台用来提供大型模型），您必须为 Red Hat OpenShift Service Mesh 和 Red Hat OpenShift Serverless 安装 Operator 并执行额外的配置。请参阅 [Serving 大型模型](#)。
- ❷ 如果您还没有启用 KServe 组件（即，将 **managementState** 字段设置为 **Removed**），还必须禁用依赖 Service Mesh 组件以避免错误。请参阅 [禁用 KServe 依赖项](#)。

4. 在 CR 的 **spec.components** 部分中，对于所示的每个 OpenShift AI 组件，将 **managementState** 字段的值设置为 **Managed** 或 **Removed**。这些值定义如下：

受管

Operator 会主动管理组件，安装它，并尝试保持其活跃。只有在组件安全时，Operator 才会升级组件。

删除

Operator 会主动管理组件，但不安装它。如果组件已安装，Operator 将尝试将其删除。



重要

- 要了解如何完全安装 KServe 组件，该组件由单模式服务平台用来提供大型模型，请参阅 [Serving 大模型](#)。
- 要了解如何配置使用 CodeFlare、KubeRay 和 Kueue 组件的 [分布式工作负载功能](#)，请参阅 [配置分布式工作负载](#)。
- 要了解如何在断开连接的环境中运行分布式工作负载，请参阅 [在断开连接的环境中运行分布式数据科学工作负载](#)。

5. 在 OpenShift Container Platform 集群中创建 **DataScienceCluster** 对象来安装指定的 OpenShift AI 组件。

```
$ oc create -f rhods-operator-dsc.yaml
```

您会看到类似如下的输出：

```
datasciencecluster.datasciencecluster.opendatahub.io/default created
```

验证

- 确认每个组件都有一个正在运行的 pod：
 1. 在 OpenShift Container Platform web 控制台中点 **Workloads** → **Pods**。
 2. 在页面顶部的 **Project** 列表中，选择 **redhat-ods-applications**。
 3. 在 applications 命名空间中，确认您安装的每个 OpenShift AI 组件都有运行 pod。
- 确认所有安装的组件的状态：
 1. 在 OpenShift Container Platform web 控制台中，点击 **Operators** → **Installed Operators**。
 2. 点 Red Hat OpenShift AI Operator。
 3. 点 **Data Science Cluster** 选项卡，再选择名为 **default-dsc** 的 **DataScienceCluster** 对象。
 4. 选择 **YAML** 选项卡。
 5. 在 **installedComponents** 部分中，确认您安装的组件的状态为 **true**。



注意

如果组件显示 CR 的 **spec.components** 部分中的 **component-name: {}** 格式，则不会安装该组件。

3.5.2. 使用 Web 控制台安装 Red Hat OpenShift AI 组件

以下流程演示了如何使用 OpenShift Container Platform Web 控制台在集群中安装 Red Hat OpenShift AI 的特定组件。



重要

以下流程描述了如何创建和配置 **DataScienceCluster** 对象，以作为新安装的一部分安装 Red Hat OpenShift AI 组件。但是，如果您从 OpenShift AI 版本 1（以前为 OpenShift Data Science）升级，升级过程会自动创建默认的 **DataScienceCluster** 对象。如果您从以前的次版本升级，升级过程将使用之前版本的 **DataScienceCluster** 对象中的设置。要检查 **DataScienceCluster** 对象并更改 Red Hat OpenShift AI 组件的安装状态，请参阅使用 [Web 控制台更新 Red Hat OpenShift AI 组件的安装状态](#)。

先决条件

- Red Hat OpenShift AI Operator 已安装在 OpenShift Container Platform 集群中。请参阅 [安装 Red Hat OpenShift AI Operator](#)。
- 具有 OpenShift Container Platform 集群的集群管理员特权。

流程

1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。
2. 在 Web 控制台中，点 **Operators** → **Installed Operators**，然后点 Red Hat OpenShift AI Operator。
3. 通过执行以下操作，创建一个 **DataScienceCluster** 对象来安装 OpenShift AI 组件：
 - a. 点 **Data Science Cluster** 选项卡。
 - b. 点 **Create DataScienceCluster**。
 - c. 对于 **Configure via**，选择 **YAML view**。
此时会打开嵌入的 YAML 编辑器，显示 **DataScienceCluster** 对象的默认自定义资源 (CR)。
 - d. 在 CR 的 **spec.components** 部分中，对于所示的每个 OpenShift AI 组件，将 **managementState** 字段的值设置为 **Managed** 或 **Removed**。这些值定义如下：

受管

Operator 会主动管理组件，安装它，并尝试保持其活跃。只有在组件安全时，Operator 才会升级组件。

删除

Operator 会主动管理组件，但不安装它。如果组件已安装，Operator 将尝试将其删除。



重要

- 要了解如何安装 KServe 组件，该组件由单模式服务平台用来提供大型模型，请参阅 [Serving 大模型](#)。
- 如果您还没有启用 KServe 组件（即，将 **managementState** 字段设置为 **Removed**），还必须禁用依赖 Service Mesh 组件以避免错误。请参阅 [禁用 KServe 依赖项](#)。
- 要了解如何配置使用 CodeFlare 和 KubeRay 组件的 [分布式工作负载功能](#)，请参阅 [配置分布式工作负载](#)。

4. 点 **Create**。

验证

- 确认每个组件都有一个正在运行的 pod：
 1. 在 OpenShift Container Platform web 控制台中点 **Workloads** → **Pods**。
 2. 在页面顶部的 **Project** 列表中，选择 **redhat-ods-applications**。
 3. 在 applications 命名空间中，确认您安装的每个 OpenShift AI 组件都有运行 pod。
- 确认所有安装的组件的状态：
 1. 在 OpenShift Container Platform web 控制台中，点击 **Operators** → **Installed Operators**。
 2. 点 Red Hat OpenShift AI Operator。
 3. 点 **Data Science Cluster** 选项卡，再选择名为 **default-dsc** 的 **DataScienceCluster** 对象。

4. 选择 **YAML** 选项卡。
5. 在 **installedComponents** 部分中，确认您安装的组件的状态为 **true**。



注意

如果组件显示 CR 的 **spec.components** 部分中的 **component-name: {}** 格式，则不会安装该组件。

3.5.3. 使用 Web 控制台更新 Red Hat OpenShift AI 组件的安装状态

以下流程演示了如何使用 OpenShift Container Platform Web 控制台更新 OpenShift Container Platform 集群上的 Red Hat OpenShift AI 组件的安装状态。



重要

如果您从版本 1 升级到 OpenShift AI，升级过程会自动创建默认 **DataScienceCluster** 对象并启用多个 OpenShift AI 组件。如果您从以前的次版本升级，升级过程将使用之前版本的 **DataScienceCluster** 对象中的设置。

以下流程描述了如何编辑 **DataScienceCluster** 对象来执行以下操作：

- 更改现有 Red Hat OpenShift AI 组件的安装状态
- 在以前的 OpenShift AI 版本中没有可用的 **DataScienceCluster** 对象添加额外的组件。

先决条件

- Red Hat OpenShift AI Operator [安装在](#) OpenShift Container Platform 集群中。
- 具有 OpenShift Container Platform 集群的集群管理员特权。

流程

1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。
2. 在 Web 控制台中，点 **Operators** → **Installed Operators**，然后点 Red Hat OpenShift AI Operator。
3. 点 **Data Science Cluster** 选项卡。
4. 在 **DataScienceClusters** 页面中，点 **default** 对象。
5. 点 **YAML** 标签。
此时会打开嵌入的 YAML 编辑器，显示 **DataScienceCluster** 对象的默认自定义资源 (CR)。
6. 在 CR 的 **spec.components** 部分中，对于所示的每个 OpenShift AI 组件，将 **managementState** 字段的值设置为 **Managed** 或 **Removed**。这些值定义如下：



注意

如果组件显示 CR 的 **spec.components** 部分中的 **component-name: {}** 格式，则不会安装该组件。

受管

Operator 会主动管理组件，安装它，并尝试保持其活跃。只有在组件安全时，Operator 才会升级组件。

删除

Operator 会主动管理组件，但不安装它。如果组件已安装，Operator 将尝试将其删除。



重要

- 要了解如何安装 KServe 组件，该组件由单模式服务平台用来提供大型模型，请参阅 [Serving 大模型](#)。
- 如果您还没有启用 KServe 组件（即，将 **managementState** 字段设置为 **Removed**），还必须禁用依赖 Service Mesh 组件以避免错误。请参阅 [禁用 KServe 依赖项](#)。
- 如果 CR 文件中还没有它们，您可以通过将 **codeflare**、Ray 和 Kueue 添加到 CR 的 **spec.components** 部分，安装 CodeFlare、KubeRay 和 Kueue 组件。
- 要了解如何配置使用 CodeFlare、KubeRay 和 Kueue 组件的 [分布式工作负载功能](#)，请参阅 [配置分布式工作负载](#)。
- 要了解如何在断开连接的环境中运行分布式工作负载，请参阅 [在断开连接的环境中运行分布式数据科学工作负载](#)。

7.

点击 **Save**。

对于您更新的任何组件，OpenShift AI 会启动一个推出影响所有 Pod 来使用更新的镜像。

验证

●

确认每个组件都有一个正在运行的 pod：

1.

在 OpenShift Container Platform web 控制台中点 **Workloads** → **Pods**。

2.

在页面顶部的 **Project** 列表中，选择 **redhat-ods-applications**。

3.

在 **applications** 命名空间中，确认您安装的每个 OpenShift AI 组件都有运行 pod。

- 确认所有安装的组件的状态：
 1. 在 OpenShift Container Platform web 控制台中，点击 **Operators** → **Installed Operators**。
 2. 点 **Red Hat OpenShift AI Operator**。
 3. 点 **Data Science Cluster** 选项卡，再选择名为 **default-dsc** 的 **DataScienceCluster** 对象。
 4. 选择 **YAML** 选项卡。
 5. 在 **installedComponents** 部分中，确认您安装的组件的状态为 **true**。



注意

如果组件显示 CR 的 **spec.components** 部分中的 **component-name: {}** 格式，则不会安装该组件。

3.5.4. 禁用 KServe 依赖项

如果您还没有启用 KServe 组件（即，将 **managementState** 字段设置为 **Removed**），还必须禁用依赖 **Service Mesh** 组件以避免错误。

先决条件

- 您已使用 OpenShift 命令行界面(CLI)或 Web 控制台来禁用 KServe 组件。

流程

1. 以集群管理员身份登录 OpenShift Web 控制台。
2. 在 Web 控制台中，点 **Operators** → **Installed Operators**，然后点 **Red Hat OpenShift AI Operator**。

3. 禁用 OpenShift Service Mesh 组件，如下所示：

a. 点 DSC 初始化选项卡。

b. 点 default-dsci 对象。

c. 点 YAML 标签。

d. 在 spec 部分中，添加 serviceMesh 组件（如果还没有存在），并配置 managementState 字段，如下所示：

```
spec:
  serviceMesh:
    managementState: Removed
```

e. 点击 Save。

验证

1. 在 Web 控制台中，点 Operators → Installed Operators，然后点 Red Hat OpenShift AI Operator。

Operator 详情页面将打开。

2. 在 Conditions 部分中，确认没有 ReconcileComplete 条件，状态为 Unknown。

3.6. 配置 OPENSIFT AI OPERATOR 日志记录器

您可以调整 OpenShift AI Operator 组件的日志级别，以增加或减少日志详细程度以满足您的用例。

3.6.1. 配置 OpenShift AI Operator 日志记录器

您可以通过在运行时为 DSC Initialization/DSCI 自定义资源设置 .spec.devFlags.logmode 标志来更

改 OpenShift AI Operator 组件的日志级别。如果没有设置 `logmode` 值，则日志记录器默认使用 `INFO` 日志级别。

使用 `.spec.devFlags.logmode` 设置的日志级别适用于所有组件，而不仅仅是处于 `Managed` 状态的组件。

下表显示了可用的日志级别：

日志级别	堆栈追踪级别	详细程度	输出	时间戳类型
devel 或 development	WARN	INFO	控制台 (Console)	epoch timestamp
"" (或没有设置 logmode 值)	ERROR	INFO	JSON	人类可读的时间戳
prod 或 production	ERROR	INFO	JSON	人类可读的时间戳

以纯文本控制台格式设置为 `devel` 或 `development` 生成的日志。设置为 `prod`、`production` 或没有 `JSON` 格式的级别集的日志。

先决条件

- 您有 `admin` 访问权限，访问 OpenShift Container Platform 集群中的 `DSCInitialization` 资源。
- 如 [CLI 入门](#) 中所述，已安装 OpenShift 命令行界面(oc)。

流程

1. 以集群管理员身份登录 OpenShift Container Platform。
2. 点 `Operators` → `Installed Operators`，然后点 `Red Hat OpenShift AI Operator`。
3. 点 `DSC 初始化选项卡`。

4. 点 **default-dsci** 对象。
5. 点 **YAML** 标签。
6. 在 **spec** 部分中，使用您要设置的日志级别更新 **.spec.devFlags.logmode** 标志。

```
apiVersion: dscinitialization.opendatahub.io/v1
kind: DSCInitialization
metadata:
  name: default-dsci
spec:
  devFlags:
    logmode: development
```

7. 点击 **Save**。

您还可以使用以下命令，将 **logmode** 值设置为您想要的日志级别，从 **OpenShift CLI** 配置日志级别。

```
oc patch dsci default-dsci -p '{"spec":{"devFlags":{"logmode":"development"}}}' --type=merge
```

验证

- 如果将组件日志级别设置为 **devel** 或 **development**，日志会更频繁地生成，并在 **WARN** 级别及更高级别包含日志。
- 如果将组件日志级别设置为 **prod** 或 **production**，或者没有设置日志级别，日志会更频繁地生成，并在 **ERROR** 级别或更高级别包含日志。

3.6.1.1. 查看 OpenShift AI Operator 日志

1. 登录到 **OpenShift CLI**。
2. 运行以下命令：

```
oc get pods -l name=rhods-operator -o name -n redhat-ods-operator | xargs -l {} oc logs -f {} -n redhat-ods-operator
```

-

Operator pod 日志将打开。

您还可以在 **OpenShift Console** 中查看 **Operator pod** 日志，在 **Workloads > Deployments > Pods > redhat-ods-operator > Logs** 下。

第 4 章 准备 OPENSIFT AI 用于 IBM CLOUD PAK FOR DATA

本节中的步骤演示了如何准备 Red Hat OpenShift AI，以便在 IBM Cloud Pak for Data 版本 5.0 或更高版本中使用。这些版本的云 Pak for Data 包括 watsonx.ai。



重要

当您要准备 *新的* Red Hat OpenShift AI 安装以便在 IBM Cloud Pak for Data 版本 5.0 或更高版本中使用时，会应用本节中的步骤。该流程演示了如何在 *raw* 模式中安装 OpenShift AI 的 KServe 组件，这意味着 KServe 没有任何其他组件作为依赖项。但是，如果您已有 OpenShift AI 部署带有 KServe 及其依赖项是否已安装并启用了，*则不需要* 修改在 Cloud Pak for Data 中使用的配置。

4.1. 使用 CLI 安装 RED HAT OPENSIFT AI OPERATOR



重要

只有在准备 Red Hat OpenShift AI 在 IBM Cloud Pak for Data 版本 5.0 或更高版本中使用时，才按照以下步骤操作。 这些版本的云 Pak for Data 包括 watsonx.ai。如果这个用例不适用于您的机构，请参阅 [在断开连接的环境中安装和部署 OpenShift AI](#)。

此流程演示了如何使用 OpenShift 命令行界面(CLI)在 OpenShift Container Platform 集群上安装 Red Hat OpenShift AI Operator。您必须先安装 Operator，然后才能管理 OpenShift AI 组件的安装。

先决条件

- 您有一个正在运行的 OpenShift Container Platform 集群，版本 4.12 或更高版本，带有可动态置备的默认存储类。
- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅[安装 OpenShift CLI](#)。
- 您已将所需的容器镜像镜像到私有 registry。对于 [断开连接的安装](#)，请参阅[将镜像镜像到私有 registry](#)。

流程

1. 打开一个新的终端窗口。
2. 在 OpenShift 命令行界面 (CLI) 中，以集群管理员身份登录到 OpenShift Container Platform 集群，如下例所示：

```
$ oc login <openshift_cluster_url> -u <admin_username> -p <password>
```

3. 创建包含以下内容的新 YAML 文件：

```
apiVersion: v1
kind: Namespace 1
metadata:
  name: redhat-ods-operator
---
apiVersion: operators.coreos.com/v1
kind: OperatorGroup 2
metadata:
  name: rhods-operator
  namespace: redhat-ods-operator
---
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription 3
metadata:
  name: rhods-operator
  namespace: redhat-ods-operator
spec:
  name: rhods-operator
  channel: eus-2.8 4
  source: cs-redhat-operator-index
  sourceNamespace: openshift-marketplace
config:
  env:
    - name: "DISABLE_DSC_CONFIG"
```

1

定义用于安装 Operator 的推荐 redhat-ods-operator 命名空间。

2

定义用于安装 Operator 的 Operator 组。您必须指定之前在文件中定义的同命名空间。另外，其他 Operator 组不能存在于命名空间中。

3

为 Operator 定义订阅。您必须指定之前在文件中定义的同命名空间。

4

对于 `channel`，指定 `eus-2.8` 值。此频道名称表示 OpenShift AI 2.8 是一个延长更新支持(EUS)版本。如需更多信息，请参阅 [Red Hat OpenShift AI Self-Managed Life Cycle](#)。

4.

部署 YAML 文件以创建您定义的命名空间、Operator 组和订阅。

```
$ oc create -f <file_name>.yaml
```

您会看到类似如下的输出：

```
namespace/redhat-ods-operator created
operatorgroup.operators.coreos.com/rhods-operator created
subscription.operators.coreos.com/rhods-operator created
```

5.

创建包含以下内容的另一个新 YAML 文件：

```
apiVersion: dscinitialization.opendatahub.io/v1
kind: DSCInitialization 1
metadata:
  name: default-dsci
spec:
  applicationsNamespace: redhat-ods-applications
  monitoring:
    managementState: Managed
    namespace: redhat-ods-monitoring
  serviceMesh:
    managementState: Removed 2
  trustedCABundle:
    managementState: Managed
    customCABundle: ""
```

1

定义名为 `default-dsci` 的 `DSCInitialization` 对象。`DSCInitialization` 对象供 Operator 用于管理适用于所有产品组件的资源。

2

指定 *没有安装* `serviceMesh` 组件（对于某些设置中的 `KServe` 的依赖项）。当准备 OpenShift AI 在 IBM 产品中使用，需要此设置。

- 在 OpenShift CLI 中，通过执行以下操作来检查 Operator 是否正在运行 pod：
 - 检查 `redhat-ods-operator` 项目中的 pod。
- 确认有一个带有 **Running** 状态的 `rhods-operator channel pod`，如下例所示：

```
$ oc get pods -n redhat-ods-operator
```

```
NAME                                READY STATUS  RESTARTS  AGE
rhods-operator-56c85d44c9-vtk74    1/1   Running   0         3h57m
```

- 在 OpenShift CLI 中，通过执行以下操作来检查您创建的 `DSCInitialization` 对象是否正在运行：
 - 检查集群是否有 `DSCInitialization` 对象。
- 确认有一个带有 **Ready** 状态的 `default-dsci` 对象，如下例所示：

```
$ oc get dscinitialization
```

```
NAME      AGE   PHASE
default-dsci 4d18h Ready
```

其他资源

- [在集群中添加 Operator](#)

4.2. 使用 CLI 管理 RED HAT OPENSIFT AI 组件

以下步骤演示了如何使用 OpenShift 命令行界面(CLI)管理 OpenShift Container Platform 集群上特定组件的安装。



重要

只有在准备 Red Hat OpenShift AI 在 IBM Cloud Pak for Data 版本 5.0 或更高版本中使用时，才按照以下步骤操作。这些版本的云 Pak for Data 包括 watsonx.ai。如果这个用例不适用于您的机构，请参阅 [安装和管理 Red Hat OpenShift AI 组件](#)。

先决条件

- Red Hat OpenShift AI Operator 已安装在 OpenShift Container Platform 集群中。请参阅 [使用 CLI 安装 Red Hat OpenShift AI Operator](#)。
- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅 [安装 OpenShift CLI](#)。

流程

1. 打开一个新的终端窗口。
2. 在 OpenShift 命令行界面 (CLI) 中，以集群管理员身份登录到 OpenShift Container Platform 集群，如下例所示：

```
$ oc login <openshift_cluster_url> -u <admin_username> -p <password>
```

3. 创建包含以下内容的新 YAML 文件：

```
kind: DataScienceCluster 1
apiVersion: datasciencecluster.opendatahub.io/v1
metadata:
  name: default-dsc
  labels:
    app.kubernetes.io/name: datasciencecluster
    app.kubernetes.io/instance: default-dsc
    app.kubernetes.io/part-of: rhods-operator
    app.kubernetes.io/managed-by: kustomize
    app.kubernetes.io/created-by: rhods-operator
spec:
  components:
    codeflare:
      managementState: Removed
  dashboard:
```

```

managementState: Removed
datasciencepipelines:
  managementState: Removed
kserve:
  managementState: Managed
  defaultDeploymentMode: RawDeployment 2
serving:
  managementState: Removed 3
  name: knative-serving
kueue:
  managementState: Removed
modelmeshserving:
  managementState: Removed
ray:
  managementState: Removed
workbenches:
  managementState: Removed

```

1

定义一个名为 `default-dsc` 的新 `DataScienceCluster` 对象。Operator 使用 `DataScienceCluster` 对象来管理 OpenShift AI 组件。

2

指定 `KServe` 组件在 `raw` 模式中由 Operator 安装和管理，这意味着 `KServe` 没有任何其他组件作为依赖项。

3

指定服务组件（这是某些设置中的 `KServe` 的依赖项）不会被安装。当准备 OpenShift AI 在 `Cloud Pak for Data` 中使用时，需要此设置。

另外，观察到所有其他 OpenShift AI 组件的 `managementState` 字段的值是否设置为 `Removed`。这意味着不会安装这些组件。

通常，`Managed` 和 `Removed` 值如下所述：

受管

Operator 会主动管理组件，安装它，并尝试保持其活跃。只有在组件安全时，Operator 才会升级组件。

删除

Operator 会主动管理组件，但不安装它。如果组件已安装，Operator 将尝试将其删除。

4. 在 OpenShift 集群中创建 DataScienceCluster 对象。

```
$ oc create -f <file_name>.yaml
```

您会看到类似如下的输出：

```
datasciencecluster.datasciencecluster.opendatahub.io/default-dsc created
```

验证

- 在 OpenShift CLI 中，通过执行以下操作来检查是否有 KServe 控制器正在运行的 pod：

- 检查 redhat-ods-applications 项目中的 pod。

```
$ oc get pods -n redhat-ods-applications
```

- 确认有一个 kserve-controller-manager fluentd pod 带有 Running 状态，如下例所示：

NAME	READY	STATUS	RESTARTS	AGE
kserve-controller-manager-57796d5b44-sh9n5	1/1	Running	0	4m57s

4.3. 编辑模型推断配置

IBM Cloud Pak for Data 版本 5.0 或更高版本（包括 watsonx.ai）中的特殊用例可能需要对 Red Hat OpenShift AI 使用的模型推断配置进行自定义。在进行此类自定义之前，您需要将 inferencing 配置文件置于可编辑状态。另外，您必须进行特定的配置更新，以防止在 Cloud Pak for Data 和 watsonx.ai 中使用 OpenShift AI 时出现错误。



重要

只有在准备 Red Hat OpenShift AI 在 IBM Cloud Pak for Data 版本 5.0 或更高版本中使用，才按照以下步骤操作。这些版本的云 Pak for Data 包括 watsonx.ai。

先决条件

-

Red Hat OpenShift AI Operator 已安装在 OpenShift Container Platform 集群中。请参阅使用 [CLI 安装 Red Hat OpenShift AI Operator](#)。

- 具有 OpenShift Container Platform 集群的集群管理员特权。

流程

1. 以集群管理员身份登录 OpenShift Container Platform Web 控制台。
2. 在 Web 控制台中，点击 Workloads → ConfigMaps。
3. 在 Project 列表中，单击 redhat-ods-applications。
4. 在 ConfigMap 资源列表中，点 inferenceservice-config 资源，然后点 YAML 选项卡。
5. 在文件的 metadata.annotations 部分，添加 opendatahub.io/managed: 'false'，如下所示：

```
metadata:
  annotations:
    internal.config.kubernetes.io/previousKinds: ConfigMap
    internal.config.kubernetes.io/previousNames: inferenceservice-config
    internal.config.kubernetes.io/previousNamespaces: opendatahub
    opendatahub.io/managed: 'false'
```

额外的注解使推断的配置文件可编辑。

6. 要防止在 Cloud Pak for Data（包括 watsonx.ai）中使用 OpenShift AI 时更新配置，请按如下所示更新配置：
 - a. 在 YAML 文件中，找到以下条目：


```
"domainTemplate": "{{ .Name }}-{{ .Namespace }}.{{ .IngressDomain }}"
```
 - b. 更新 domainTemplate 字段的值，如下所示：

```
"domainTemplate": "example.com"
```

这个新的、明确指定的值可确保 **OpenShift AI** 无法为超过最大长度的 **domainTemplate** 字段生成值，并导致错误。在准备 **OpenShift AI** 以便在 **Cloud Pak for Data** 中使用，为 **KServe** 组件配置的 **raw** 部署模式不使用 **domainTemplate** 字段。

7.

点击 **Save**。

第 5 章 使用证书

证书供 **OpenShift Container Platform** 中的不同组件用来验证对集群的访问。对于依赖自签名证书的集群，您可以将这些自签名证书添加到集群范围的证书颁发机构(CA)捆绑包中，并使用 Red Hat OpenShift AI 中的 CA 捆绑包。您还可以在与集群范围捆绑包分开的自定义 CA 捆绑包中使用自签名证书。管理员可以添加 CA 捆绑包，从所有命名空间中删除 CA 捆绑包，从单个命名空间中删除 CA 捆绑包，或者手动管理证书更改，而不是系统。

5.1. 了解 OPENS SHIFT AI 中的证书

对于依赖自签名证书的 **OpenShift Container Platform** 集群，您可以将这些自签名证书添加到集群范围的证书颁发机构(CA)捆绑包(ca-bundle.crt)，并使用 Red Hat OpenShift AI 中的 CA 捆绑包。您还可以在与集群范围捆绑包分开的自定义 CA 捆绑包(odh-ca-bundle.crt)中使用自签名证书。

5.1.1. 如何注入 CA 捆绑包

安装 OpenShift AI 后，Red Hat OpenShift AI Operator 会自动创建一个空的 odh-trusted-ca-bundle 配置文件(ConfigMap)，Cluster Network Operator (CNO)将集群范围的 CA 捆绑包注入 odh-trusted-ca-bundle configMap，标签为 "config.openshift.io/inject-trusted-cabundle"。在受影响命名空间中部署的组件负责将此 configMap 挂载为部署 pod 中的卷。

```
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/part-of: opendatahub-operator
    config.openshift.io/inject-trusted-cabundle: 'true'
name: odh-trusted-ca-bundle
```

在 CNO operator 注入捆绑包后，它会使用包含证书的 ca-bundle.crt 文件更新 ConfigMap。

```
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/part-of: opendatahub-operator
    config.openshift.io/inject-trusted-cabundle: 'true'
name: odh-trusted-ca-bundle
data:
  ca-bundle.crt: |
    <BUNDLE OF CLUSTER-WIDE CERTIFICATES>
```

5.1.2. ConfigMap 的管理方式

默认情况下，Red Hat OpenShift AI Operator 管理 odh-trusted-ca-bundle ConfigMap。如果要管理或删除 odh-trusted-ca-bundle ConfigMap，或者添加与集群范围 CA 捆绑包（ca-bundle.crt）分开的自定义 CA 捆绑包（odh-ca-bundle.crt），您可以在 Operator 的 DSC Initialization (DSCI) 对象中使用 trustedCABundle 属性。

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: ""
```

在 Operator 的 DSCI 对象中，您可以将 spec.trustedCABundle.managementState 字段设置为以下值：

- Managed:** Red Hat OpenShift AI Operator 管理 odh-trusted-ca-bundle ConfigMap，并将其添加到所有非保留现有命名空间和新命名空间中（ConfigMap 不会添加到任何保留或系统命名空间中，如 default, openshift-* 或 kube-*）。ConfigMap 会自动更新，以反映对 customCABundle 字段所做的任何更改。这是安装 Red Hat OpenShift AI 后的默认值。
- Removed:** Red Hat OpenShift AI Operator 删除 odh-trusted-ca-bundle ConfigMap（如果存在），并禁用在命名空间中创建 ConfigMap。如果将此字段从 Managed 改为 Removed，则 odh-trusted-ca-bundle ConfigMap 也会从命名空间中删除。这是将 Red Hat OpenShift AI 从 2.7 或更早版本升级到 2.10 后的默认值。
- Unmanaged:** Red Hat OpenShift AI Operator 不管理 odh-trusted-ca-bundle ConfigMap，允许管理员管理它。将 managementState 从 Managed 改为 Unmanaged 不会删除 odh-trusted-ca-bundle ConfigMap，但如果对 customCABundle 字段进行更改，则不会更新 ConfigMap。

在 Operator 的 DSCI 对象中，您可以将自定义证书添加到 spec.trustedCABundle.customCABundle 字段中。这会将包含证书的 odh-ca-bundle.crt 文件添加到 odh-trusted-ca-bundle ConfigMap 中，如下例所示：

```
apiVersion: v1
kind: ConfigMap
metadata:
  labels:
    app.kubernetes.io/part-of: opendatahub-operator
    config.openshift.io/inject-trusted-cabundle: 'true'
  name: odh-trusted-ca-bundle
data:
  ca-bundle.crt: |
    <BUNDLE OF CLUSTER-WIDE CERTIFICATES>
  odh-ca-bundle.crt: |
    <BUNDLE OF CUSTOM CERTIFICATES>
```

5.2. 添加 CA 捆绑包

将证书颁发机构(CA)捆绑包添加到 OpenShift AI 中有两种方法。您可以使用以下任一方法：

- 对于依赖自签名证书的 OpenShift Container Platform 集群，您可以将这些自签名证书添加到集群范围的证书颁发机构(CA)捆绑包(ca-bundle.crt)，并使用 Red Hat OpenShift AI 中的 CA 捆绑包。要使用此方法，请以集群管理员身份登录到 OpenShift Container Platform，并按照在 [安装过程中配置集群范围代理](#)中所述的步骤进行操作。
- 您可以在与集群范围捆绑包分开的自定义 CA 捆绑包中使用自签名证书(odh-ca-bundle.crt)。要使用这个方法，请按照本节中的步骤操作。

先决条件

- 您有 admin 访问权限，访问 OpenShift Container Platform 集群中的 DSCInitialization 资源。
- 如 [CLI 入门](#) 中所述，已安装 OpenShift 命令行界面(oc)。
- 您在新的 Red Hat OpenShift AI 安装中工作。如果您升级了 Red Hat OpenShift AI，请参[阅升级后添加 CA 捆绑包](#)。

流程

1. 登录到 OpenShift Container Platform。
2. 点 Operators → Installed Operators，然后点 Red Hat OpenShift AI Operator。
3. 点 DSC 初始化选项卡。
4. 点 default-dsci 对象。
5. 点 YAML 标签。

6.

在 `spec` 部分中，将自定义证书添加到 `trustedCABundle` 的 `customCABundle` 字段中，如下例所示：

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: |
      -----BEGIN CERTIFICATE-----
      examplebundle123
      -----END CERTIFICATE-----
```

7.

点击 **Save**。

验证

•

如果使用集群范围的 CA 捆绑包，请运行以下命令验证所有非保留命名空间是否包含 `odh-trusted-ca-bundle ConfigMap`：

```
$ oc get configmaps --all-namespaces -l app.kubernetes.io/part-of=opendatahub-operator |
grep odh-trusted-ca-bundle
```

•

如果您使用自定义 CA 捆绑包，请运行以下命令来验证非保留命名空间是否包含 `odh-trusted-ca-bundle ConfigMap`，并且 `ConfigMap` 包含您的 `customCABundle` 值。在以下命令中，`example-namespace` 是非保留的命名空间，`examplebundle123` 是 `customCABundle` 值。

```
$ oc get configmap odh-trusted-ca-bundle -n example-namespace -o yaml | grep
examplebundle123
```

5.3. 删除 CA 捆绑包

您可以从 OpenShift AI 中的所有非保留命名空间中删除证书颁发机构(CA)捆绑包。此流程更改了默认配置，并禁用创建 `odh-trusted-ca-bundle` 配置文件(ConfigMap)，如了解 *OpenShift AI 中的证书* 中所述。



注意

只有当您为 `trustedCABundle` 的 `managementState` 设置为 `Removed` 时，`odh-trusted-ca-bundle ConfigMap` 才会从命名空间中删除；删除 DSC 初始化不会删除 `ConfigMap`。

要只从单个命名空间中删除 CA 捆绑包，请参阅[从命名空间中删除 CA 捆绑包](#)。

先决条件

- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 如 [CLI 入门](#) 中所述，已安装 OpenShift 命令行界面(oc)。

流程

1. 在 OpenShift Container Platform web 控制台中，点 Operators → Installed Operators，然后点 Red Hat OpenShift AI Operator。
2. 点 DSC 初始化选项卡。
3. 点 default-dsci 对象。
4. 点 YAML 标签。
5. 在 spec 部分中，将 trustedCABundle 的 managementState 字段的值改为 Removed :

```
spec:  
  trustedCABundle:  
    managementState: Removed
```

6. 点击 **Save**。

验证

- 运行以下命令，以验证 odh-trusted-ca-bundle ConfigMap 是否已从所有命名空间中移除 :

```
$ oc get configmaps --all-namespaces | grep odh-trusted-ca-bundle
```

该命令不应返回任何 **ConfigMap**。

5.4. 从命名空间中删除 CA 捆绑包

您可以从 OpenShift AI 中的独立命名空间中删除自定义证书颁发机构(CA)捆绑包。这个过程禁用只为指定命名空间创建 **odh-trusted-ca-bundle** 配置文件(ConfigMap)。

要从所有命名空间中删除证书捆绑包，请参阅 [删除 CA 捆绑包](#)。

先决条件

- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 如 [CLI 入门](#) 中所述，已安装 OpenShift 命令行界面(oc)。

流程

- 运行以下命令以从命名空间中删除 CA 捆绑包。在以下命令中，**example-namespace** 是非保留命名空间。

```
$ oc annotate ns example-namespace security.opendatahub.io/inject-trusted-ca-bundle=false
```

验证

- 运行以下命令，以验证已从命名空间中删除 CA 捆绑包。在以下命令中，**example-namespace** 是非保留命名空间。

```
$ oc get configmap odh-trusted-ca-bundle -n example-namespace
```

该命令应该返回 **configmaps "odh-trusted-ca-bundle" not found**。

5.5. 管理证书

安装 OpenShift AI 后，Red Hat OpenShift AI Operator 会创建 **odh-trusted-ca-bundle** 配置文件(ConfigMap)，其中包含可信 CA 捆绑包，并将其添加到集群中的所有新和非保留命名空间中。默认情况下，Red Hat OpenShift AI Operator 管理 **odh-trusted-ca-bundle** ConfigMap，并在对 CA 捆绑包进行

任何更改时自动更新。您可以选择管理 `odh-trusted-ca-bundle ConfigMap`，而不是允许 Red Hat OpenShift AI Operator 管理它。

先决条件

- 具有 OpenShift Container Platform 集群的集群管理员特权。

流程

1. 在 OpenShift Container Platform web 控制台中，点 **Operators** → **Installed Operators**，然后点 **Red Hat OpenShift AI Operator**。
2. 点 **DSC 初始化选项卡**。
3. 点 **default-dsci 对象**。
4. 点 **YAML 标签**。
5. 在 **spec** 部分中，将 **trustedCABundle** 的 **managementState** 字段的值改为 **Unmanaged**，如下所示：

```
spec:
  trustedCABundle:
    managementState: Unmanaged
```

6. 点击 **Save**。

请注意，将 **managementState** 从 **Managed** 改为 **Unmanaged** 不会删除 `odh-trusted-ca-bundle ConfigMap`，但如果对 `customCABundle` 字段进行更改，则不会更新 `ConfigMap`。

验证

1. 在 **spec** 部分中，为 **trustedCABundle** 设置或更改 **customCABundle** 字段的值，例如：

```
spec:
  trustedCABundle:
    managementState: Unmanaged
```

1. `customCABundle: example123`
2. 点击 **Save**。
3. 点 **Workloads** → **ConfigMaps**。
4. 从项目列表中选择个项目。
5. 点 **odh-trusted-ca-bundle ConfigMap**。
6. 点 **YAML** 选项卡，并验证 **customCABundle** 字段的值是否没有更新。

5.6. 在 OPENSIFT AI 组件中使用自签名证书

有些 OpenShift AI 组件为自签名证书有额外的选项或所需的配置。

5.6.1. 使用带有数据科学项目管道的证书

如果要使用自签名证书，请将它们添加到中央证书颁发机构(CA)捆绑包中，如使用 [证书](#)（用于断开连接的环境）中所述，[请参阅使用证书](#)。

在数据科学项目管道中使用这些证书不需要额外的配置。

5.6.1.1. 仅为数据科学项目提供 CA 捆绑包

执行以下步骤仅为数据科学管道提供证书颁发机构(CA)捆绑包。

流程

1. 登录 **OpenShift Container Platform**。
2. 在 **Workloads** → **ConfigMaps** 中，在与目标数据科学项目管道相同的数据科学项目或命名空间中创建一个所需捆绑包的 **ConfigMap**：

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: custom-ca-bundle
data:
  ca-bundle.crt: |
    # contents of ca-bundle.crt
```

3.

将以下代码片段添加到底层 **Data Science Pipelines Application (DSPA)** 的 **.spec.apiserver.caBundle** 字段中：

```
apiVersion: datasciencepipelinesapplications.opendatahub.io/v1alpha1
kind: DataSciencePipelinesApplication
metadata:
  name: data-science-dspa
spec:
  ...
  apiServer:
  ...
  caBundle:
    configMapName: custom-ca-bundle
    configMapKey: ca-bundle.crt
```

Pipeline 服务器 pod 使用更新的捆绑包重新部署，并在新创建的管道 pod 中使用它。

验证

执行以下步骤确认您的 **CA 捆绑包** 已被成功挂载。

1. **登录到 OpenShift Container Platform 控制台。**
2. **进入与数据科学项目对应的 OpenShift Container Platform 项目。**
3. **点 Pods 选项卡。**
4. **点带有 ds-pipeline-dspa-`<hash >` 前缀的管道服务器 pod。**
5. **点 Terminal。**

6. 输入 `cat /dsp-custom-certs/dsp-ca.crt`。
7. 验证您的 CA 捆绑包是否存在于此文件中。

您还可以使用 CLI 确认您的 CA 捆绑包已被成功挂载：

1. 在终端窗口中，登录部署了 OpenShift AI 的 OpenShift 集群。

```
oc login
```

2. 设置 `dspa` 值：

```
dspa=dspa
```

3. 设置 `dsProject` 值，将 `$YOUR_DS_PROJECT` 替换为您的数据科学项目的名称：

```
dsProject=$YOUR_DS_PROJECT
```

4. 设置 `pod` 值：

```
pod=$(oc get pod -n ${dsProject} -l app=ds-pipeline-${dspa} --no-headers | awk '{print $1}')
```

5. 显示 `/dsp-custom-certs/dsp-ca.crt` 文件的内容：

```
oc -n ${dsProject} exec $pod -- cat /dsp-custom-certs/dsp-ca.crt
```

6. 验证您的 CA 捆绑包是否存在于此文件中。

5.6.2. 使用带有工作台的证书



重要

自签名证书适用于您在集中配置自签名证书后创建的工作台，如 [使用证书](#) 中所述，请参阅 [使用证书](#)。在配置自签名证书前，不会更改您创建的工作台。

5.6.2.1. 使用 Elyra 和自签名证书创建数据科学管道

要使用包含 Elyra 扩展以及使用自签名证书的工作台创建管道，请参阅 [在断开连接的环境中使用 Elyra 执行管道的 Workbench 临时解决方案](#)。

第 6 章 在 OPENSIFT AI 中启用 GPU 支持

另外，为了确保数据科学家可以在其模型中使用计算密集型工作负载，您可以在 OpenShift AI 中启用图形处理单元(GPU)。

先决条件

- 已登陆到 OpenShift Container Platform 集群。
- 在 OpenShift Container Platform 集群中具有 cluster-admin 角色。

流程

1. 要在断开连接的或 airgapped 环境中的 OpenShift 集群上启用 GPU 支持，请按照 NVIDIA 文档中的 [断开连接的或 airgapped 环境中部署 GPU Operator](#) 的说明。
2. 删除 migration-gpu-status ConfigMap。
 - a. 在 OpenShift Container Platform web 控制台中切换到 Administrator 视角。
 - b. 将项目设置为 All Projects 或 redhat-ods-applications，以确保您可以看到适当的 ConfigMap。
 - c. 搜索 migration-gpu-status ConfigMap。
 - d. 点操作菜单 (⋮)，并从列表中选择 Delete ConfigMap。

此时会出现 Delete ConfigMap 对话框。
 - e. 检查对话框，并确认您删除正确的 ConfigMap。
 - f. 点击 Delete。

3. 重启仪表盘 replicaset.
 - a. 在 OpenShift Container Platform web 控制台中切换到 Administrator 视角。
 - b. 点 Workloads → Deployments。
 - c. 将项目设置为 All Projects 或 redhat-ods-applications，以确保您可以看到适当的部署。
 - d. 搜索 rhods-dashboard 部署。
 - e. 点操作菜单(HBAC)，然后从列表中选择 Restart Rollout。
 - f. 等待 Status 列指出 rollout 中的所有 pod 都完全重启。

验证

- NVIDIA GPU Operator 会出现在 OpenShift Container Platform Web 控制台的 Operators → Installed Operators 页面中。
- 重置 migration-gpu-status 实例存在于 AcceleratorProfile 自定义资源定义 (CRD) 详情页面上的 Instances 选项卡中。

安装 NVIDIA GPU Operator 后，创建一个加速器配置集，如 [使用加速器配置集](#) 中所述。

第 7 章 访问仪表板

安装 OpenShift AI 并添加用户后，您可以访问 OpenShift AI 控制台的 URL，并与用户共享 URL，以便他们登录并处理其模型。

先决条件

- 您已在 OpenShift Container Platform 集群中安装了 OpenShift AI。
- 您至少向 OpenShift AI 的用户组添加了一个用户。

流程

1. 登录到 OpenShift Container Platform Web 控制台。
2. 点应用程序启动程序()。
3. 右键点 Red Hat OpenShift AI，再复制 OpenShift AI 实例的 URL。
4. 为您的数据科学家提供这个实例 URL，以便他们登录到 OpenShift AI。

验证

- 确认您和您的用户可以使用实例 URL 登录 OpenShift AI。

其他资源

- [登录到 OpenShift AI](#)
- [添加用户](#)

第 8 章 卸载 RED HAT OPENSIFT AI SELF-MANAGED

本节介绍如何使用 OpenShift 命令行界面(CLI)卸载 Red Hat OpenShift AI Operator 以及由 Operator 安装和管理的任何 OpenShift AI 组件。



注意

使用 CLI 是卸载 Operator 的建议方法。根据您的 OpenShift Container Platform 版本，使用 Web 控制台执行卸载可能无法提示您卸载所有相关组件。这可能会导致您不知道集群的最终状态。

8.1. 了解卸载过程

为 OpenShift AI 的不同组件安装 Red Hat OpenShift AI 在 OpenShift Container Platform 集群中创建多个自定义资源实例。安装后，用户可能会在使用 OpenShift AI 时创建多个其他资源。卸载 OpenShift AI 会删除 Operator 创建的资源，但保留用户创建的资源，以防止意外删除您可能需要的信息。

要删除的内容

卸载 OpenShift AI 从 OpenShift Container Platform 集群中删除以下资源：

- **DataScienceCluster 自定义资源实例**
- **DSCInitialization 自定义资源实例**
- **FeatureTracker 自定义资源实例在安装过程中或安装后创建**
- **在 Operator 期间或安装后创建的 ServiceMesh 自定义资源实例**
- **Operator 在安装过程中或安装后创建的 KnativeServing 自定义资源实例**
- **redhat-ods-applications, redhat-ods-monitoring, 和 rhods-notebooks 命名空间由 Operator 创建**

- rhods-notebooks 命名空间中的工作负载
- Subscription, ClusterServiceVersion, 和 InstallPlan 对象
- KfDef 对象（仅限版本 1 Operator）

可能保留什么

卸载 OpenShift AI 在 OpenShift Container Platform 集群中保留以下资源：

- 用户创建的数据科学项目
- 用户创建的自定义资源实例
- 由用户或 Operator 创建的自定义资源定义 (CRD)

虽然这些资源可能仍然保留在 OpenShift Container Platform 集群中，但它们无法正常工作。卸载后，红帽建议您查看 OpenShift Container Platform 集群中的数据科学项目和自定义资源，并删除不再用于防止潜在的问题（如无法运行的管道、无法取消部署）或无法取消部署的模型。

其他资源

[Operator Lifecycle Manager \(OLM\) 卸载文档](#)

8.2. 使用 CLI 卸载 OPENSIFT AI 自我管理

以下流程演示了如何使用 OpenShift 命令行界面(CLI) 卸载 Red Hat OpenShift AI Operator 以及由 Operator 安装和管理的任何 OpenShift AI 组件。

先决条件

- 具有 OpenShift Container Platform 集群的集群管理员特权。

- 您已下载并安装 **OpenShift 命令行界面 (CLI)**。请参阅[安装 OpenShift CLI](#)。
- 您已备份了持久性卷声明 (PVC) 使用的持久性卷或卷。

流程

1. 打开一个新的终端窗口。
2. 在 **OpenShift 命令行界面 (CLI)** 中，以集群管理员身份登录到 **OpenShift Container Platform** 集群，如下例所示：

```
$ oc login <openshift_cluster_url> -u system:admin
```

3. 创建 **ConfigMap** 对象以删除 **Red Hat OpenShift AI Operator**。

```
$ oc create configmap delete-self-managed-odh -n redhat-ods-operator
```

4. 要删除 **rhods-operator**，请将 **addon-managed-odh-delete** 标签设置为 **true**。

```
$ oc label configmap/delete-self-managed-odh api.openshift.com/addon-managed-odh-delete=true -n redhat-ods-operator
```

5. 删除与 **Operator** 关联的所有对象时，删除 **redhat-ods-operator** 项目。

- a. 为 **redhat-ods-applications** 项目设置环境变量。

```
$ PROJECT_NAME=redhat-ods-applications
```

- b. 等待 **redhat-ods-applications** 项目已被删除。

```
$ while oc get project $PROJECT_NAME && /dev/null; do  
echo "The $PROJECT_NAME project still exists"  
sleep 1  
done  
echo "The $PROJECT_NAME project no longer exists"
```

删除 **redhat-ods-applications** 项目后，您会看到以下输出：

```
The redhat-ods-applications project no longer exists
```

c.

删除 **redhat-ods-applications** 项目后，删除 **redhat-ods-operator** 项目。

```
$ oc delete namespace redhat-ods-operator
```

验证

- 确认 **rhods-operator** 订阅不再存在。

```
$ oc get subscriptions --all-namespaces | grep rhods-operator
```

- 确认以下项目不再存在。
 - **redhat-ods-applications**
 - **redhat-ods-monitoring**
 - **redhat-ods-operator**
 - **rhods-notebooks**

```
$ oc get namespaces | grep -e redhat-ods* -e rhods*
```



注意

只有在安装了 **OpenShift AI** 的工作台组件时，才会创建 **rhods-notebooks** 项目。请参阅 [安装和管理 Red Hat OpenShift AI 组件](#)。