Red Hat OpenShift AI Cloud Service 1

安装和卸载 OpenShift AI 云服务

作为 OpenShift 集群的附加组件安装和卸载 OpenShift AI

Last Updated: 2024-10-22

Red Hat OpenShift AI Cloud Service 1 安装和卸载 OpenShift AI 云服务

作为 OpenShift 集群的附加组件安装和卸载 OpenShift AI

法律通告

摘要

在 Red Hat OpenShift Dedicated 或 Red Hat OpenShift Service on Amazon Web Services (ROSA)上安装和使用 OpenShift AI 作为附加组件。

目录

第 1 章 OPENSHIFT AI 架构

Red Hat OpenShift AI 是一个完全托管的云服务，可作为 Red Hat OpenShift Dedicated 和 Amazon Web Services (ROSA Classic)上的 Red Hat OpenShift Service 的附加组件提供。

OpenShift AI 集成了以下组件和服务：

- 在服务层：

  OpenShift AI 仪表板

  一个面向客户的仪表板，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。您还可以从仪表板访问管理功能，如用户管理、集群设置、加速器配置集和笔记本镜像设置。此外，数据科学家可以从控制面板创建自己的项目。这使得他们能够将其数据科学组织到一个项目中。

  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 容器构建带有数据科学管道 2.0 的可移植机器学习(ML)工作流。借助数据科学项目，数据科学家可以在开发其数据科学项目模型时自动化工作流。

  Jupyter （红帽管理的）

  一个由红帽管理的应用程序，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

Alertmanager、OpenShift Telemetry 和 Prometheus 协同工作，从 OpenShift AI 和组织收集指标，并以实用的方式显示这些指标。来自 Alertmanager 的警报发送到 PagerDuty，负责通知红帽与受管云服务相关的问题。

在 Cluster Manager 中安装 Red Hat OpenShift AI Add-on 时，会创建以下新项目：

- **redhat-ods-operator** 项目包含 Red Hat OpenShift AI Operator。
- **redhat-ods-applications** 项目将安装仪表板和 OpenShift AI 的其他必要组件。
- **redhat-ods-monitoring** 项目包含用于监控和计费的服务。
- **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。

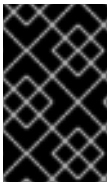
不要在与 OpenShift AI 附加组件关联的命名空间中安装独立软件供应商(ISV)应用程序，除非您特别定向到仪表板上的应用程序标题。

第 2 章 安装并部署 OPENSHIFT AI

Red Hat OpenShift AI 是人工智能(AI)应用程序的数据科学家和开发人员的平台。它提供了一个完全支持的环境，可让您在公共云中快速开发、培训、测试和部署机器学习模型。

OpenShift AI 作为受管云服务附加组件提供给 Red Hat OpenShift，或作为自我管理的软件提供，您可以在 OpenShift 上安装内部或公有云。

有关在连接的或断开连接的环境中将 OpenShift AI 作为自我管理的软件安装的详情，请参考 Red Hat OpenShift AI Self-Managed 产品文档。



重要

Data Science pipelines 2.0 包含 Argo 工作流的安装。OpenShift AI 不支持直接客户使用此 Argo 工作流安装。要安装带有数据科学管道 2.0 的 OpenShift AI，请确保集群中没有单独的安装 Argo 工作流。

Red Hat OpenShift AI 有两个部署选项作为受管云服务附加组件：

- **在 Amazon Web Services 或 Google Cloud Platform 上使用客户云订阅的 OpenShift Dedicated**
  OpenShift Dedicated 是一个完整的 OpenShift Container Platform 集群，作为云服务提供，配置为高可用性，专用于单个客户。OpenShift Dedicated 主要由红帽管理，托管在 Amazon Web Services (AWS)或 Google Cloud Platform (GCP)上。客户云订阅(CCS)模型允许红帽将集群部署和管理到客户的 AWS 或 GCP 帐户中。请联系您的红帽客户经理，通过 CCS 获取 OpenShift Dedicated。

- **Red Hat OpenShift Service on AWS (ROSA Classic)**
  ROSA 是一个完全被管理的应用平台，它使您可以专注于通过构建和部署应用程序来为客户创造价值。您直接从 AWS 帐户订阅该服务。

将 OpenShift AI 安装为受管云服务涉及以下高级别任务：

1. 确认 OpenShift 集群满足所有要求。
2. 为 OpenShift 集群配置身份提供程序。
3. 为 OpenShift 集群添加管理用户。
4. 订阅 Red Hat OpenShift AI Add-on。
   对于带有 CCS for AWS 或 GCP 的 OpenShift Dedicated，通过红帽获取订阅。

   对于 ROSA Classic，通过 AWS Marketplace 获取订阅。
5. 安装 Red Hat OpenShift AI Add-on。
6. 访问 OpenShift AI 仪表板。
7. 另外，还可在 OpenShift AI 中启用图形处理单元(GPU)，以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。

2.1. OPENSHIFT AI 的要求

在 Red Hat OpenShift Dedicated 或 Red Hat OpenShift Service on Amazon Web Services (ROSA Classic)集群上安装 OpenShift AI 前，您必须满足以下要求：

- **Red Hat OpenShift Dedicated 或 ROSA 订阅的订阅**
  您可以使用 AWS 上的客户云订阅或 GCP 模型上的 客户云订阅，在 Amazon Web Services (AWS)或 Google Cloud Platform (GCP)帐户上部署 Red Hat OpenShift Dedicated。https://docs.openshift.com/dedicated/osd_planning/gcp-ccs.html请注意，如果想要安装 OpenShift AI，但红帽提供了一个在 Red Hat 云帐户上安装 OpenShift Dedicated 的选项，但您必须在您自己的云帐户中安装 OpenShift Dedicated。

  请联系您的红帽客户经理，购买新的 Red Hat OpenShift Dedicated 订阅。如果您还没有帐户管理器，请在 https://cloud.redhat.com/products/dedicated/contact/ 上填写表单以请求一个。

  您可以直接从 AWS 帐户(ROSA Classic)直接订阅 Red Hat OpenShift Service on AWS (ROSA Classic)，或联系您的红帽帐户经理。

- **红帽客户帐户**
  进入 OpenShift Cluster Manager(http://console.redhat.com/openshift)，并登录或注册新帐户。

- **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。使用现有集群，或按照相关文档中的步骤创建集群：

  - 创建 OpenShift Dedicated 集群
  - 安装 ROSA Classic 集群

- **满足以下要求的 OpenShift Dedicated 或 ROSA 集群配置：**

- 安装附加组件时，至少有 2 个 worker 节点至少有 8 个 CPU 和 32 GiB RAM 用于 OpenShift AI。
  如果没有满足这个要求，安装过程无法启动并显示错误。
  在创建新集群时，为计算机节点实例类型选择 **m6a.2xlarge** 来满足要求。

  对于现有的 ROSA Classic 集群，您可以使用以下命令获取计算节点实例类型：

  rosa list machinepools --cluster=cluster-name

  您无法更改集群的计算节点实例类型，但您可以添加额外机器池或修改默认池来满足最低要求。但是，集群中单个机器池必须满足最低资源要求。

  如需更多信息，请参阅相关文档：

  - 在 OpenShift Dedicated 中创建机器池
  - OpenShift AI 服务定义
  - 在 ROSA Classic 中创建机器池
  - 准备您的环境 (ROSA Classic)

- **对于 ROSA 集群，选择访问管理策略**
  要在 ROSA Classic 集群上安装 OpenShift AI，决定是否要在使用 AWS 安全令牌服务(STS)的 ROSA 集群上安装，或使用 AWS Identity and Access Management (IAM)凭证。有关部署使用或不使用 AWS STS 的 ROSA 集群的建议，请参阅安装 ROSA Classic 集群。

- **安装 KServe 依赖项**

- 要支持 KServe 组件，该组件由单模式服务平台用来服务大型模型，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，请参阅关于单模式服务平台。
- 如果要为单模式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，请参阅为 单模式服务平台添加授权供应商。

安装模型 registry 依赖项（技术预览功能）

+ ** 要使用模型 registry 组件，还必须为 Red Hat Authorino、Red Hat OpenShift Serverless 和 Red Hat OpenShift Service Mesh 安装 Operator。有关配置模型 registry 组件的更多信息，请参阅配置模型 registry 组件。

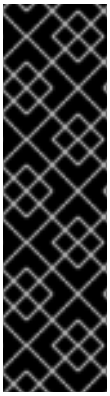
- **访问对象存储**
  - OpenShift AI 组件需要或可以使用 S3 兼容对象存储，如 AWS S3、MinIO、Ceph 或 IBM Cloud Storage。对象存储是一种数据存储机制，允许用户作为对象或文件访问其数据。S3 API 是基于 HTTP 访问对象存储服务的可识别标准。
  - 以下组件需要对象存储：
    - 单一或多模式服务平台，以部署存储模型。请参阅 在单模式服务平台上部署 模型或使用多型号服务平台部署模型。
    - 数据科学管道、存储工件、日志和中介结果。请参阅配置管道服务器和 关于管道日志。
  - 以下组件可以使用对象存储：
    - 工作台，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。

2.2. 为 OPENSHIFT 集群配置身份提供程序

为 OpenShift Dedicated 或 Red Hat OpenShift Service on Amazon Web Services (ROSA)集群配置身份提供程序来管理用户和组。

Red Hat OpenShift AI 支持与 Red Hat OpenShift Dedicated 和 ROSA 相同的身份验证系统。如需更多信息，请参阅相应的集群文档。

- OpenShift Dedicated 支持的身份提供程序
- 在 ROSA 上支持的身份提供程序



重要

当多个提供程序中存在相同用户名时，添加多个 OpenShift 身份提供程序可能会造成问题。

当 **mappingMethod** 设为 **claim** （身份提供程序的默认映射方法），并且多个供应商有与同一用户名关联的凭证，用于登录到 OpenShift 的第一个供应商是该用户适用的供应商，无论配置身份提供程序的顺序是什么。

有关映射方法的更多信息，请参阅 ROSA 中的 OpenShift Dedicated 或身份提供程序参数中的身份提供程序参数。

先决条件

- OpenShift Cluster Manager 的凭证(https://console.redhat.com/openshift/)。
- 现有的 OpenShift Dedicated 或 ROSA 集群。

步骤

1. 登录到 OpenShift Cluster Manager (https://console.redhat.com/openshift/) 。
2. 点 **Clusters**。**Clusters** 页面将打开。
3. 点击要配置的集群名称。
4. 点 **Access control** 选项卡。
5. 点 **Identity provider**。
6. 点 **Add identity provider**。
   a. 从 **Identity Provider** 列表中选择您的供应商。
   b. 完成与您选择的身份提供程序相关的剩余字段。如需更多信息，请参阅在 OpenShift Dedicated 中 配置身份提供程序 或 在 ROSA 中配置身份提供程序。
7. 单击 **Confirm**。

验证

- 配置的身份提供程序可以在 **Cluster details** 页面的 **Access control** 选项卡中看到。

其他资源

- 在 OpenShift Dedicated 中配置身份提供程序
- 在 ROSA 中配置身份提供程序

2.3. 在 OPENSHIFT 中添加管理用户

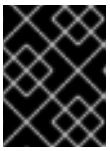
在您可以为数据科学家用户安装和配置 OpenShift AI 前，您必须获取 OpenShift 集群管理员(**cluster-admin**)特权。

先决条件

- Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)的凭证。
- 配置了身份提供程序的现有 OpenShift Dedicated 或 Red Hat OpenShift Service on AWS (ROSA Classic)集群。

流程

1. 登录到 OpenShift Cluster Manager (https://console.redhat.com/openshift/) 。
2. 点 **Clusters**。**Clusters** 页面将打开。
3. 点击要配置的集群名称。
4. 点 **Access control** 选项卡。
5. 单击 **Cluster Roles 和 Access**。
6. 在**集群管理用户**下，点**添加用户**按钮。
   此时会出现 **Add cluster user**弹出窗口。
7. 在 **User ID** 字段中输入用户名。
8. 为用户选择一个适当的组。



重要

如果此用户需要使用身份提供程序中的现有组来控制 OpenShift AI 访问，请选择 **cluster-admins**。

有关这些用户类型的更多信息，请参阅 ROSA 文档中的 管理角色 和用户。

9. 单击 **Add user**。

验证

- 用户名和所选组在**集群管理用户**列表中可见。

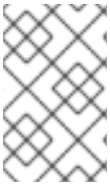
其他资源

- OpenShift Dedicated 集群管理
- ROSA 集群管理

2.4. 订阅 RED HAT OPENSHIFT AI CLOUD SERVICE

您可以使用以下方法订阅 Red Hat OpenShift AI 管理的云服务：

- 如果您使用 Amazon Web Services (AWS)或 Google Cloud Platform (GCP)上的客户云订阅(CCS)部署 Red Hat OpenShift Dedicated 集群。
- 如果您有 Red Hat OpenShift Service on AWS (ROSA Classic)集群，请通过 AWS Marketplace 订阅。



注意

您还可以将 Red Hat OpenShift AI 作为自我管理的软件购买。要购买新的订阅，请联系您的红帽客户经理。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。

2.4.1. 订阅 AWS 或 GCP 上的 OpenShift AI 管理的云服务

对于在 AWS 或 GCP 上部署的 Red Hat OpenShift Dedicated 集群，请联系您的红帽客户管理器购买新的订阅。如果您还没有帐户管理器，请在 https://cloud.redhat.com/products/dedicated/contact/ 上填写表单以请求一个。

前提条件

- 您已与红帽销售部合作支持私有提供的 OpenShift AI，请按照以下步骤接受您的提供和部署解决方案。

流程

1. 请使用红帽销售代表提供的 URL 链接访问您的私有产品。
2. 点 **Accept** terms 订阅 **AWS Marketplace 中名为 OpenShift AI 的** AMI 私有。
3. 接受优惠条款后，单击 **Continue to Configuration**。

2.4.2. 在 Red Hat OpenShift Service on AWS (ROSA)上订阅 OpenShift AI 管理的云服务

对于 ROSA Classic 集群，您可以通过 Amazon Web Services (AWS) Marketplace 订阅 OpenShift AI 管理的云服务。

先决条件

- 访问 ROSA Classic 集群，包括查看和安装附加组件的权限。
- 具有 AWS 市场查看和订阅产品权限的 AWS 帐户。

流程

1. 在 AWS 控制台中，进入 AWS Marketplace。例如：
   a. 点帮助图标，然后选择 Getting Started Resource Center。
   b. 选择 AWS Marketplace > Browse AWS Marketplace。
2. 在 top **Search** 字段中，键入： **Red Hat OpenShift AI**。
3. 根据 AWS 帐户的账单地址的地理位置，选择两个选项之一（请注意，此位置可能与集群的地理位置不同）：
   - 欧洲、中东和非州(EMEA 地区)
   - EMEA 之外的北美和地区
4. 点 **Continue to Subscribe**。

5. 点 **Continue to Configuration**，然后选择适当的 fulfillment 选项。请注意，有些选择器可能只有一个选项。
6. 点 **Continue to Launch**。
7. 将 AWS 帐户与您的红帽帐户链接，以完成您的注册：
   a. 在 AWS Marketplace 控制台中进入 **Manage Subscriptions** 页面。
   b. 在 **Red Hat OpenShift AI** 标题中，点 **Set up product**。
   c. 在顶部横幅上，单击 **Set up account**。
   此链接带您到 Red Hat Hybrid 控制台。
   d. 如果您还没有登录，请登录。
   e. 检查并接受条款和协议。
   f. 单击 **Connect accounts**。

验证

Data Science 产品页面将打开。

2.5. 在 OPENSHIFT 集群上安装 OPENSHIFT AI

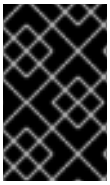
您可以使用 Red Hat OpenShift Cluster Manager 将 Red Hat OpenShift AI 作为 Red Hat OpenShift 集群的附加组件安装。

先决条件

- 订阅 Red Hat OpenShift AI 附加组件，如 订阅 AWS 或 GCP 上的 OpenShift AI 管理云服务 中所述。
- 如果您使用 AWS Marketplace 为 ROSA Classic 购买 Red Hat OpenShift AI 附加组件，您已将 AWS 帐户与您的红帽帐户相关联，如 订阅 Red Hat OpenShift Service on AWS (ROSA) 所述。
- Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)的凭证。
- 管理员对 OpenShift 集群的访问权限。
- 要支持 KServe 组件，您必须安装依赖的 Operator，包括 Red Hat OpenShift Serverless 和 Red Hat OpenShift Service Mesh Operator。如需更多信息，请参阅 Serving 大模型。

注意

有关与 Red Hat OpenShift AI 相关的生命周期的详情，请参考 Red Hat OpenShift AI 生命周期。

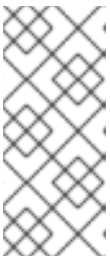


重要

Data Science pipelines 2.0 包含 Argo 工作流的安装。OpenShift AI 不支持直接客户使用此 Argo 工作流安装。要安装带有数据科学管道 2.0 的 OpenShift AI，请确保集群中没有单独的安装 Argo 工作流。

流程

1. 登录到 OpenShift Cluster Manager (https://console.redhat.com/openshift/) 。
2. 点 **Clusters**。
   Clusters 页面将打开。
3. 点您要在其上安装 OpenShift AI 的集群名称。
   此时会打开集群的 **Details** 页面。
4. 点 **Add-ons** 选项卡，找到 **Red Hat OpenShift AI** 标题。

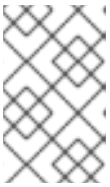


注意

如果有 **先决条件没有满足警告信息**，点 **先决条件** 标签页。请注意错误消息。如果错误消息显示您需要新的机器池，或者需要更多资源，执行适当的操作来解决这个问题。您可能需要向集群添加更多资源，或者增加默认机器池的大小。要提高集群的资源，请联系您的基础架构管理员。有关增大机器池大小的更多信息，请参阅为 OpenShift AI 用户分配其他资源。

5. 选择 **Subscription 类型** ：
   如果您通过您的红帽帐户管理器获取 Red Hat OpenShift AI 订阅，请选择 **Standard**，然后跳至第 7 步。

   如果您直接从 AWS Marketplace 获取了 Red Hat OpenShift AI 订阅，请选择 **Marketplace**，然后继续第 6 步。
6. 对于 Marketplace 订阅，请从列表中选择 AWS 帐户号。



注意

如果您的 AWS 帐号不在列表中，您可能需要链接您的红帽和 AWS 帐户，如 在 Red Hat OpenShift Service on AWS (ROSA)上订阅 OpenShift AI 管理云服务 中所述。

7. 点 **Install**。此时会出现 **Configure Red Hat OpenShift AI** 窗格。
8. 在 **通知电子邮件** 字段中，输入您要接收有关 Red Hat OpenShift AI 状态的重要警报的电子邮件地址，如中断警报。
9. 点 **Install**。

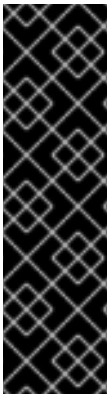
验证

- 在 OpenShift Cluster Manager 中，在集群的 **Add-ons** 选项卡中，确认 OpenShift AI 标题显示以下状态之一：
  - **Installing** - 安装正在进行中；等待此更改为 **Installed**。这大约需要 30 分钟。
  - **Installed** - 安装已完成；坚持是否出现 **View in console** 按钮。
- 在 OpenShift AI 中，点击 **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**
- **rhods-notebooks**

2.6. 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 Web 控制台在集群中安装 Red Hat OpenShift AI 的特定组件。



重要

当您将 Red Hat OpenShift AI 作为 OpenShift 集群的附加组件安装时，安装过程会自动创建默认的 **DataScienceCluster** 对象。以下流程描述了如何配置 **DataScienceCluster** 对象以 *作为新* 安装的一部分安装 Red Hat OpenShift AI 组件。

如果您从 OpenShift AI 版本 1（以前为 OpenShift Data Science）升级，升级过程还会自动创建默认的 **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 作为 Red Hat OpenShift 集群的附加组件安装。
- 有 OpenShift 集群的集群管理员特权。

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. 通过执行以下操作配置 **DataScienceCluster** 对象来安装 OpenShift AI 组件：
   a. 点 **Data Science Cluster** 选项卡。
   b. 点 **default-dsc** 对象。
   c. 选择 **YAML** 选项卡。
   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。

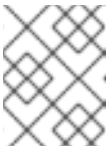
重要

- 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。
- 要了解如何安装分布式工作负载组件，请参阅安装分布式工作负载组件。

4. 点击 **Save**。

验证

- 确认每个组件都有一个正在运行的 pod：
  1. 在 OpenShift Web 控制台中，点击 **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 Web 控制台中，点 **Operators → Installed Operators**。
  2. 点 Red Hat OpenShift AI Operator。
  3. 单击 **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。

2.7. 安装分布式工作负载组件

要在 OpenShift AI 中使用分布式工作负载功能，您必须安装几个组件。

先决条件

- 已使用 **cluster-admin** 角色登录到 OpenShift，您可以访问数据科学项目。
- 已安装 Red Hat OpenShift AI。
- 您有足够的资源。除了 安装和部署 OpenShift AI 中描述的最低 OpenShift AI 资源外，还需要 1.6 vCPU 和 2 GiB 内存来部署分布式工作负载基础架构。
- 您已删除了 CodeFlare Operator 的任何以前安装实例，如 知识库解决方案 如何从数据科学集群中的单独安装的 CodeFlare Operator 迁移。
- 如果要使用图形处理单元(GPU)，在 OpenShift AI 中启用了 GPU 支持。请参阅启用 NVIDIA GPU。

注意

在 OpenShift AI 中，对于分布式工作负载，红帽只支持 NVIDIA GPU 加速器。红帽支持在同一集群中使用加速器。红帽不支持在加速器之间远程直接内存访问 (RDMA)，或使用网络上的加速器，例如使用 NVIDIA GPUDirect 或 NVLink 等技术。

- 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 中所述。不需要额外的配置来将这些证书与分布式工作负载一起使用。集中配置的自签名证书会在以下挂载点的工作负载 pod 中自动可用：
  - 集群范围的 CA 捆绑包：

    ```
    /etc/pki/tls/certs/odh-trusted-ca-bundle.crt
    /etc/ssl/certs/odh-trusted-ca-bundle.crt
    ```

  - 自定义 CA 捆绑包：

    ```
    /etc/pki/tls/certs/odh-ca-bundle.crt
    /etc/ssl/certs/odh-ca-bundle.crt
    ```

流程

1. 在 OpenShift 控制台中，点 **Operators → Installed Operators**。
2. 搜索 **Red Hat OpenShift AI** Operator，然后点 Operator 名称以打开 Operator 详情页面。
3. 点 **Data Science Cluster** 选项卡。
4. 点默认实例名称（如 **default-dsc**）打开实例详情页面。
5. 点 **YAML** 选项卡显示实例规格。
6. 启用所需的分布式工作负载组件。在 **spec.components** 部分中，为所需组件正确设置 **managementState** 字段：
   - 如果要使用 CodeFlare 框架调优模型，请启用 **codeflare**、**kue** 和 **ray** 组件。
   - 如果要使用 Kubeflow Training Operator 调优模型，请启用 **kue** 和 **trainingoperator** 组件。
   - 所需的组件列表取决于分布式工作负载是否从管道或笔记本运行，如下表所示。

表 2.1. 分布式工作负载所需的组件

组件	仅限管道	仅限笔记本	Pipelines 和笔记本
codeflare	**受管**	**受管**	**受管**
dashboard	**受管**	**受管**	**受管**
datasciencepipelines	**受管**	**删除**	**受管**
kueue	**受管**	**受管**	**受管**

组件	仅限管道	仅限笔记本	Pipelines 和笔记本
ray	**受管**	**受管**	**受管**
trainingoperator	**受管**	**受管**	**受管**
workbenches	**删除**	**受管**	**受管**

7. 点击 **Save**。片刻后，处于 **Managed** 状态的组件已就绪。

验证

检查 **codeflare-operator-manager**、**kuberay-operator** 和 **kueue-controller-manager** pod 的状态，如下所示：

1. 在 OpenShift 控制台中，从 **Project** 列表中选择 **redhat-ods-applications**。
2. 点 **Workloads** → **Deployments**。
3. 搜索 **codeflare-operator-manager**,**kuberay-operator**, 和 **kueue-controller-manager** 部署。在每个情形中，按如下所示检查状态：
   a. 单击部署名称以打开部署详情页面。
   b. 点 **Pods** 选项卡。
   c. 检查 pod 状态。
   当 **codeflare-operator-manager-** *<pod-id&* gt;、**kuberay-operator-** *<pod-id>*、和 **kue-controller-manager-***<pod-id* > pod 的状态为 **Running** 时，pod 就可以使用。
   d. 要查看每个 pod 的更多信息，请点 pod 名称以打开 pod 详情页面，然后点 **Logs** 选项卡。

下一步

配置分布式工作负载功能，如 管理分布式工作负载 中所述。

第 3 章 安装单模式服务平台

3.1. 关于单模式服务平台

为了部署大型模型，如大型语言模型(LLMs)，OpenShift AI 包含一个基于 KServe 组件的单模型服务平台。要安装单型号服务平台，需要以下组件：

- KServe ：一种 Kubernetes 自定义资源定义(CRD)，编配所有类型的模型的模型服务。KServe 包括实现给定模型服务器的加载的模型运行时。KServe 还处理部署对象、存储访问和网络设置的生命周期。
- Red Hat OpenShift Serverless ：一个云原生开发模型，允许无服务器部署模型。OpenShift Serverless 基于开源 Knative 项目。
- Red Hat OpenShift Service Mesh ：一个服务网格网络层，用于管理流量流并强制实施访问策略。OpenShift Service Mesh 基于开源 Istio 项目。

注意

目前，只支持 OpenShift Service Mesh v2。如需更多信息，请参阅 支持的配置。

您可以手动安装单型号服务平台，或自动安装：

自动安装

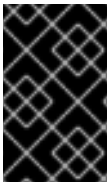
如果您还没有在 OpenShift 集群上创建 **ServiceMeshControlPlane** 或 **KNativeServing** 资源，您可以将 Red Hat OpenShift AI Operator 配置为安装 KServe 并配置其依赖项。如需更多信息，请参阅配置 KServe 的自动安装

手动安装

如果您已在 OpenShift 集群上创建了 **ServiceMeshControlPlane** 或 **KNativeServing** 资源，*则无法*配置 Red Hat OpenShift AI Operator 来安装 KServe 并配置其依赖项。在这种情况下，您必须手动安装 KServe。如需更多信息，请参阅 手动安装 KServe。

3.2. 配置 KSERVE 的自动安装

如果您还没有在 OpenShift 集群上创建 **ServiceMeshControlPlane** 或 **KNativeServing** 资源，您可以将 Red Hat OpenShift AI Operator 配置为安装 KServe 并配置其依赖项。



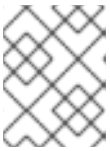
重要

如果您在集群中创建了 **ServiceMeshControlPlane** 或 **KNativeServing** 资源，Red Hat OpenShift AI Operator 无法安装 KServe 并配置其依赖项，且安装不会继续。在这种情况下，您必须按照手动安装说明来安装 KServe。

先决条件

- 有 OpenShift 集群的集群管理员特权。
- 您的集群有一个有 4 个 CPU 和 16 GB 内存的节点。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。如需更多信息，请参阅安装 OpenShift CLI (Red Hat OpenShift Dedicated)或 安装 OpenShift CLI (Red Hat OpenShift Service on AWS)。

- 已安装 Red Hat OpenShift Service Mesh Operator 和依赖的 Operator。



注意

要启用 KServe 的自动安装，*请只* 为 Red Hat OpenShift Service Mesh 安装所需的 Operator。不要执行任何其他配置或创建 **ServiceMeshControlPlane** 资源。

- 已安装 Red Hat OpenShift Serverless Operator。

注意

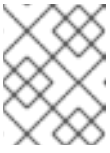
要启用 KServe 的自动安装，*请只* 安装 Red Hat OpenShift Serverless Operator。不要执行任何其他配置或创建 **KNativeServing** 资源。

- 要将 Authorino 添加为授权提供程序，以便您可以为部署的模型启用令牌授权，已安装了 **Red Hat - Authorino** Operator。请参阅 安装 Authorino Operator。

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. 按照如下所示安装 OpenShift Service Mesh：
   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，验证 **serviceMesh** 组件的 **managementState** 字段的值是否已设置为 **Managed**，如下所示：

```
spec:
 applicationsNamespace: redhat-ods-applications
 monitoring:
   managementState: Managed
   namespace: redhat-ods-monitoring
 serviceMesh:
   controlPlane:
     metricsCollection: Istio
     name: data-science-smcp
     namespace: istio-system
   managementState: Managed
```



注意

不要更改为 **serviceMesh** 组件指定的 **istio-system** 命名空间。不支持其他命名空间值。

   e. 点击 **Save**。

根据添加到 **DSCInitialization** 对象的配置，Red Hat OpenShift AI Operator 安装 OpenShift Service Mesh。

4. （仅限 Red Hat OpenShift Service on AWS）：如果您的 OpenShift 集群在 Red Hat OpenShift Service on AWS (ROSA Classic)上运行，则需要额外的设置才能使服务网格 control plane 配置正常工作。要添加此设置，请按如下方式编辑 **data-science-smcp** 服务网格 control plane 对象：

   a. 在 Web 控制台中，点 **Operators → Installed Operators**，然后点 Red Hat OpenShift Service Mesh Operator。

   b. 点 **Istio Service Mesh Control Plane**标签页。

   c. 点 **data-science-smcp** 对象。

   d. 点 **YAML** 标签。

   e. 在 **spec.security.identity** 部分中，添加名为 **type** 的字段，并将值设为 **ThirdParty**，如下所示。

```
security:
  dataPlane:
    mtls: true
  identity:
    type: ThirdParty
```

   f. 点击 **Save**。

5. 安装 KServe 和 OpenShift Serverless，如下所示：

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

   b. 点 **Data Science Cluster** 选项卡。

   c. 单击 **default-dsc** DSC 对象。

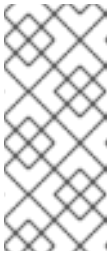
   d. 点 **YAML** 标签。

   e. 在 **spec.components** 部分中，配置 **kserve** 组件，如下所示。

```
spec:
 components:
   kserve:
     managementState: Managed
     serving:
       ingressGateway:
         certificate:
           secretName: knative-serving-cert
           type: OpenshiftDefaultIngress
       managementState: Managed
       name: knative-serving
```

   f. 点击 **Save**。
   上述配置为 OpenShift Serverless 创建一个入口网关，以接收来自 OpenShift Service Mesh 的流量。在这个配置中，观察以下详情：

- 显示的配置使用为 OpenShift 配置的默认入口证书来保护进入 OpenShift 集群的流量，并将证书存储在 **secretName** 字段中指定的 **knative-serving-cert** secret 中。
- **secretName** 字段只能在安装时设置。**secretName** 字段的默认值为 **knative-serving-cert**。必须手动对证书 secret 进行后续更改。
- 如果您在安装过程中没有使用默认的 **secretName** 值，请在 **istio-system** 命名空间中创建一个名为 **knative-serving-cert** 的新 secret，然后重启 **istiod-datascience-smcp-<suffix>** pod。
- 要提供自己的证书，请更新 **secretName** 字段的值，以指定您的 secret 名称，并将 **type** 字段的值更改为 **Provided**。



注意

如果提供自己的证书，证书必须指定 OpenShift 集群的入口控制器使用的域名。您可以运行以下命令来检查这个值：

$ oc get ingresses.config.openshift.io cluster -o jsonpath='{.spec.domain}'

- 对于 **kserve** 和 **serving** 组件，您必须将 **managementState** 字段的值设置为 **Managed**。将 **kserve.managementState** 设置为 **Managed** 触发器自动安装 KServe。将 **service.managementState** 设置为 **Managed** 会触发 OpenShift Serverless 自动安装。但是，如果 **kserve.managementState** 没有设置为 **Managed**，则 *不会触发* OpenShift Serverless 安装。

验证

- 验证 OpenShift Service Mesh 安装，如下所示：
  - 在 Web 控制台中，点击 **Workloads → Pods**。
  - 在项目列表中选择 **istio-system**。这是安装 OpenShift Service Mesh 的项目。
  - 确认存在用于服务网格 control plane、入口网关和出口网关的 pod。这些 pod 具有以下示例中显示的命名模式：

```
NAME                                  READY   STATUS    RESTARTS   AGE
istio-egressgateway-7c46668687-fzsqj     1/1     Running   0          22h
istio-ingressgateway-77f94d8f85-fhsp9    1/1     Running   0          22h
istiod-data-science-smcp-cc8cfd9b8-2rkg4   1/1     Running   0          22h
```

- 验证 OpenShift Serverless 的安装，如下所示：
  - 在 Web 控制台中，点击 **Workloads → Pods**。
  - 从项目列表中，选择 **knative-serving**。这是安装 OpenShift Serverless 的项目。
  - 确认 **knative-serving** 项目中有许多正在运行的 Pod，包括 activator、autoscaler、controller 和域映射 Pod，以及用于 Knative Istio 控制器（控制 OpenShift Serverless 和 OpenShift Service Mesh 集成）的 Pod。显示了一个示例。

```
NAME                                READY   STATUS    RESTARTS   AGE
activator-7586f6f744-nvdlb          2/2     Running   0          22h
activator-7586f6f744-sd77w          2/2     Running   0          22h
```

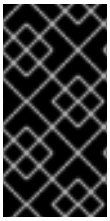
```
autoscaler-764fdf5d45-p2v98              2/2     Running   0          22h
autoscaler-764fdf5d45-x7dc6              2/2     Running   0          22h
autoscaler-hpa-7c7c4cd96d-2lkzg          1/1     Running   0          22h
autoscaler-hpa-7c7c4cd96d-gks9j          1/1     Running   0          22h
controller-5fdfc9567c-6cj9d              1/1     Running   0          22h
controller-5fdfc9567c-bf5x7              1/1     Running   0          22h
domain-mapping-56ccd85968-2hjvp          1/1     Running   0          22h
domain-mapping-56ccd85968-lg6mw          1/1     Running   0          22h
domainmapping-webhook-769b88695c-gp2hk   1/1     Running   0          22h
domainmapping-webhook-769b88695c-npn8g   1/1     Running   0          22h
net-istio-controller-7dfc6f668c-jb4xk    1/1     Running   0          22h
net-istio-controller-7dfc6f668c-jxs5p    1/1     Running   0          22h
net-istio-webhook-66d8f75d6f-bgd5r       1/1     Running   0          22h
net-istio-webhook-66d8f75d6f-hld75       1/1     Running   0          22h
webhook-7d49878bc4-8xjbr                 1/1     Running   0          22h
webhook-7d49878bc4-s4xx4                 1/1     Running   0          22h
```

- 验证 KServe 安装，如下所示：
  - 在 Web 控制台中，点击 **Workloads → Pods**。
  - 从项目列表中，选择 **redhat-ods-applications**。这是安装 OpenShift AI 组件的项目，包括 KServe。
  - 确认项目包含 KServe 控制器管理器正在运行的 pod，如下例所示：

```
NAME                                        READY   STATUS    RESTARTS   AGE
kserve-controller-manager-7fbb7bccd4-t4c5g  1/1     Running   0          22h
odh-model-controller-6c4759cc9b-cftmk       1/1     Running   0          129m
odh-model-controller-6c4759cc9b-ngj8b       1/1     Running   0          129m
odh-model-controller-6c4759cc9b-vnhq5       1/1     Running   0          129m
```

3.3. 手动安装 KSERVE

如果您已经安装了 Red Hat OpenShift Service Mesh Operator 并创建了 **ServiceMeshControlPlane** 资源，或者已安装 Red Hat OpenShift Serverless Operator 并创建了 **KNativeServing** 资源，Red Hat OpenShift AI Operator 无法安装 KServe 并配置其依赖项。在这种情况下，您必须手动安装 KServe。

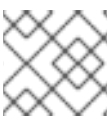


重要

本节中的步骤演示了如何执行 KServe 及其依赖项的新安装，并作为完整的安装和配置参考。*如果您已经安装并配置了 OpenShift Service Mesh 或 OpenShift Serverless，您可能不需要遵循所有步骤。如果您不确定要将哪些更新应用到现有配置以使用 KServe，请联系红帽支持。*

3.3.1. 安装 KServe 依赖项

在安装 KServe 之前，您必须安装并配置一些依赖项。具体来说，您必须创建 Red Hat OpenShift Service Mesh 和 Knative Serving 实例，然后为 Knative Serving 配置安全网关。



注意

目前，只支持 OpenShift Service Mesh v2。如需更多信息，请参阅 支持的配置。

3.3.2. 创建 OpenShift Service Mesh 实例

以下流程演示了如何创建 Red Hat OpenShift Service Mesh 实例。

先决条件

- 有 OpenShift 集群的集群管理员特权。
- 您的集群有一个有 4 个 CPU 和 16 GB 内存的节点。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅安装 OpenShift CLI (Red Hat OpenShift Dedicated) 或 安装 OpenShift CLI (Red Hat OpenShift Service on AWS)。
- 已安装 Red Hat OpenShift Service Mesh Operator 和依赖的 Operator。

流程

1. 在一个终端窗口中，如果您还没有以集群管理员登录到 OpenShift 集群，请登录 OpenShift CLI，如下例所示：

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

2. 为 Red Hat OpenShift Service Mesh 创建所需的命名空间。

```
$ oc create ns istio-system
```

   您会看到以下输出：

```
namespace/istio-system created
```

3. 在名为 **smcp.yaml** 的 YAML 文件中定义一个 **ServiceMeshControlPlane** 对象，其内容如下：

```
apiVersion: maistra.io/v2
kind: ServiceMeshControlPlane
metadata:
  name: minimal
  namespace: istio-system
spec:
  tracing:
    type: None
  addons:
    grafana:
      enabled: false
    kiali:
      name: kiali
      enabled: false
    prometheus:
      enabled: false
    jaeger:
      name: jaeger
  security:
    dataPlane:
      mtls: true
    identity:
      type: ThirdParty
```

```
techPreview:
  meshConfig:
    defaultConfig:
      terminationDrainDuration: 35s
gateways:
  ingress:
    service:
      metadata:
        labels:
          knative: ingressgateway
proxy:
  networking:
    trafficControl:
      inbound:
        excludedPorts:
          - 8444
          - 8022
```

如需有关 YAML 文件中的值的更多信息，请参阅 Service Mesh control plane 配置参考。

4. 创建服务网格 control plane。

```
$ oc apply -f smcp.yaml
```

验证

- 验证服务网格实例的创建，如下所示：
  - 在 OpenShift CLI 中输入以下命令：

    ```
    $ oc get pods -n istio-system
    ```

    前面的命令列出了 **istio-system** 项目中运行的所有 pod。这是安装 OpenShift Service Mesh 的项目。

  - 确认存在用于服务网格 control plane、入口网关和出口网关的 pod。这些 pod 具有以下命名模式：

    ```
    NAME                                  READY   STATUS    RESTARTS   AGE
    istio-egressgateway-7c46668687-fzsqj  1/1     Running   0          22h
    istio-ingressgateway-77f94d8f85-fhsp9 1/1     Running   0          22h
    istiod-data-science-smcp-cc8cfd9b8-2rkg4  1/1   Running   0        22h
    ```

3.3.3. 创建 Knative Serving 实例

以下流程演示了如何安装 Knative Serving，然后创建实例。

先决条件

- 有 OpenShift 集群的集群管理员特权。
- 您的集群有一个有 4 个 CPU 和 16 GB 内存的节点。

- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅安装 OpenShift CLI (Red Hat OpenShift Dedicated) 或 安装 OpenShift CLI (Red Hat OpenShift Service on AWS)。
- 您已创建了 Red Hat OpenShift Service Mesh 实例。
- 已安装 Red Hat OpenShift Serverless Operator。

流程

1. 在一个终端窗口中，如果您还没有以集群管理员登录到 OpenShift 集群，请登录 OpenShift CLI，如下例所示：

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

2. 检查 Knative Serving 所需的项目（即 命名空间）是否已存在。

```
$ oc get ns knative-serving
```

如果项目存在，您会看到类似以下示例的输出：

```
NAME              STATUS   AGE
knative-serving   Active   4d20h
```

3. 如果 **knative-serving** 项目不存在，请创建它。

```
$ oc create ns knative-serving
```

您会看到以下输出：

```
namespace/knative-serving created
```

4. 在名为 **default-smm.yaml** 的 YAML 文件中定义 **ServiceMeshMember** 对象，其内容如下：

```
apiVersion: maistra.io/v1
kind: ServiceMeshMember
metadata:
  name: default
  namespace: knative-serving
spec:
  controlPlaneRef:
    namespace: istio-system
    name: minimal
```

5. 在 **istio-system** 命名空间中创建 **ServiceMeshMember** 对象。

```
$ oc apply -f default-smm.yaml
```

您会看到以下输出：

```
servicemeshmember.maistra.io/default created
```

6. 在名为 **knativeserving-istio.yaml** 的 YAML 文件中定义 **KnativeServing** 对象，其内容如下：

```
apiVersion: operator.knative.dev/v1beta1
kind: KnativeServing
metadata:
  name: knative-serving
  namespace: knative-serving
  annotations:
    serverless.openshift.io/default-enable-http2: "true"
spec:
  workloads:
    - name: net-istio-controller
      env:
        - container: controller
          envVars:
            - name: ENABLE_SECRET_INFORMER_FILTERING_BY_CERT_UID
              value: 'true'
    - annotations:
        sidecar.istio.io/inject: "true" (1)
        sidecar.istio.io/rewriteAppHTTPProbers: "true" (2)
      name: activator
    - annotations:
        sidecar.istio.io/inject: "true"
        sidecar.istio.io/rewriteAppHTTPProbers: "true"
      name: autoscaler
  ingress:
    istio:
      enabled: true
  config:
    features:
      kubernetes.podspec-affinity: enabled
      kubernetes.podspec-nodeselector: enabled
      kubernetes.podspec-tolerations: enabled
```

*前面的文件为 **KnativeServing** 对象定义自定义资源(CR)。CR 还为每个激活器和自动扩展 pod 添加以下操作：*

1. *将 Istio sidecar 注入 pod。这使得 pod 成为服务网格的一部分。*
2. *启用 Istio sidecar 为 pod 重写 HTTP 存活度和就绪度探测。*

注意

*如果为 Knative 服务配置自定义域，您可以使用 TLS 证书来保护映射的服务。要做到这一点，您必须创建一个 TLS secret，然后更新 **DomainMapping** CR 以使用您创建的 TLS secret。如需更多信息，请参阅 Red Hat OpenShift Serverless 文档中的使用 TLS 证书保护映射的服务。*

7. *在指定的 **knative-serving** 命名空间中创建 **KnativeServing** 对象。*

```
$ oc apply -f kniveserving-istio.yaml
```

您会看到以下输出：

```
knativeserving.operator.knative.dev/knative-serving created
```

验证

- *查看 **istio-system** 命名空间中的默认 **ServiceMeshMemberRoll** 对象。*

```
$ oc describe smmr default -n istio-system
```

*在 **ServiceMeshMemberRoll** 对象的描述中，找到 **Status.Members** 字段，并确认它包含 **knative-serving** 命名空间。*

- *验证 Knative Serving 实例的创建，如下所示：*
  - *在 OpenShift CLI 中输入以下命令：*

```
$ oc get pods -n knative-serving
```

*前面的命令列出了 **knative-serving** 项目中运行的所有 Pod。这是在其中创建 Knative Serving 实例的项目。*

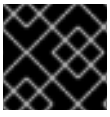
  - *确认 **knative-serving** 项目中有许多正在运行的 Pod，包括 activator、autoscaler、controller 和域映射 Pod，以及用于 Knative Istio 控制器的 Pod，用于控制 OpenShift Serverless 和 OpenShift Service Mesh 集成。显示了一个示例。*

```
NAME                                    READY   STATUS    RESTARTS   AGE
activator-7586f6f744-nvdlb              2/2     Running   0          22h
activator-7586f6f744-sd77w              2/2     Running   0          22h
autoscaler-764fdf5d45-p2v98             2/2     Running   0          22h
autoscaler-764fdf5d45-x7dc6             2/2     Running   0          22h
autoscaler-hpa-7c7c4cd96d-2lkzg         1/1     Running   0          22h
autoscaler-hpa-7c7c4cd96d-gks9j         1/1     Running   0          22h
controller-5fdfc9567c-6cj9d             1/1     Running   0          22h
controller-5fdfc9567c-bf5x7             1/1     Running   0          22h
domain-mapping-56ccd85968-2hjvp         1/1     Running   0          22h
domain-mapping-56ccd85968-lg6mw         1/1     Running   0          22h
domainmapping-webhook-769b88695c-gp2hk  1/1     Running   0          22h
domainmapping-webhook-769b88695c-npn8g  1/1     Running   0          22h
net-istio-controller-7dfc6f668c-jb4xk   1/1     Running   0          22h
net-istio-controller-7dfc6f668c-jxs5p   1/1     Running   0          22h
net-istio-webhook-66d8f75d6f-bgd5r      1/1     Running   0          22h
net-istio-webhook-66d8f75d6f-hld75      1/1     Running   0          22h
webhook-7d49878bc4-8xjbr                1/1     Running   0          22h
webhook-7d49878bc4-s4xx4                1/1     Running   0          22h
```

3.3.4. 为 Knative Serving 创建安全网关

要保护 Knative Serving 实例和服务网格之间的流量，您必须为 Knative Serving 实例创建安全网关。

以下流程描述了如何使用 OpenSSL 生成通配符证书和密钥，然后使用它们为 Knative Serving 创建本地和入口网关。



重要

如果您有自己的通配符证书和密钥在配置网关时指定，您可以跳过到此流程的第 11 步。

先决条件

- 有 OpenShift 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 (Red Hat OpenShift Dedicated) 或 安装 OpenShift CLI (Red Hat OpenShift Service on AWS)。
- 您已创建了 Red Hat OpenShift Service Mesh 实例。
- 您已创建了 Knative Serving 实例。
- 如果要生成通配符证书和密钥，您已 下载并安装 OpenSSL。

流程

1. 在一个终端窗口中，如果您还没有以集群管理员登录到 OpenShift 集群，请登录 OpenShift CLI，如下例所示：

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

重要

如果您有自己的通配符证书和密钥在配置网关时指定，请跳至此步骤的第 11 步。

2. 设置环境变量，以定义为网关生成通配符证书和密钥的基础目录。

```
$ export BASE_DIR=/tmp/kserve
$ export BASE_CERT_DIR=${BASE_DIR}/certs
```

3. 设置环境变量以定义 OpenShift 集群的入口控制器使用的通用名称。

```
$ export COMMON_NAME=$(oc get ingresses.config.openshift.io cluster -o
jsonpath='{.spec.domain}' | awk -F'.' '{print $(NF-1)"."$NF}')
```

4. 设置环境变量以定义 OpenShift 集群的入口控制器使用的域名。

```
$ export DOMAIN_NAME=$(oc get ingresses.config.openshift.io cluster -o
jsonpath='{.spec.domain}')
```

5. 根据之前设置的环境变量，为证书生成创建所需的基础目录。

```
$ mkdir ${BASE_DIR}
$ mkdir ${BASE_CERT_DIR}
```

6. 为生成通配符证书创建 OpenSSL 配置。

```
$ cat <<EOF> ${BASE_DIR}/openssl-san.config
[ req ]
distinguished_name = req
[ san ]
subjectAltName = DNS:*.${DOMAIN_NAME}
EOF
```

7. 生成 root 证书。

```
$ openssl req -x509 -sha256 -nodes -days 3650 -newkey rsa:2048 \
-subj "/O=Example Inc./CN=${COMMON_NAME}" \
-keyout $BASE_DIR/root.key \
-out $BASE_DIR/root.crt
```

8. 生成由 root 证书签名的通配符证书。

```
$ openssl req -x509 -newkey rsa:2048 \
-sha256 -days 3560 -nodes \
-subj "/CN=${COMMON_NAME}/O=Example Inc." \
-extensions san -config ${BASE_DIR}/openssl-san.config \
-CA $BASE_DIR/root.crt \
-CAkey $BASE_DIR/root.key \
-keyout $BASE_DIR/wildcard.key  \
-out $BASE_DIR/wildcard.crt

$ openssl x509 -in ${BASE_DIR}/wildcard.crt -text
```

9. 验证通配符证书。

```
$ openssl verify -CAfile ${BASE_DIR}/root.crt ${BASE_DIR}/wildcard.crt
```

10. 将脚本创建的通配符密钥和证书导出到新的环境变量。

```
$ export TARGET_CUSTOM_CERT=${BASE_CERT_DIR}/wildcard.crt
$ export TARGET_CUSTOM_KEY=${BASE_CERT_DIR}/wildcard.key
```

11. 可选：要将 您自己的 通配符密钥和证书导出到新环境变量，请输入以下命令：

```
$ export TARGET_CUSTOM_CERT=<path_to_certificate>
$ export TARGET_CUSTOM_KEY=<path_to_key>
```

注意

在您提供的证书中，您必须指定 OpenShift 集群的入口控制器使用的域名。您可以运行以下命令来检查这个值：

$ oc get ingresses.config.openshift.io cluster -o jsonpath='{.spec.domain}'

12. 使用您为通配符证书和密钥设置的环境变量，在 **istio-system** 命名空间中创建 TLS secret。

```
$ oc create secret tls wildcard-certs --cert=${TARGET_CUSTOM_CERT} --
key=${TARGET_CUSTOM_KEY} -n istio-system
```

13. 使用以下内容创建 **gateways.yaml** YAML 文件：

```
apiVersion: v1
kind: Service (1)
metadata:
  labels:
    experimental.istio.io/disable-gateway-port-translation: "true"
  name: knative-local-gateway
```

```
  namespace: istio-system
spec:
  ports:
    - name: http2
      port: 80
      protocol: TCP
      targetPort: 8081
  selector:
    knative: ingressgateway
  type: ClusterIP
---
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
  name: knative-ingress-gateway 2
  namespace: knative-serving
spec:
  selector:
    knative: ingressgateway
  servers:
    - hosts:
        - '*'
      port:
        name: https
        number: 443
        protocol: HTTPS
      tls:
        credentialName: wildcard-certs
        mode: SIMPLE
---
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
 name: knative-local-gateway 3
 namespace: knative-serving
spec:
 selector:
   knative: ingressgateway
 servers:
   - port:
       number: 8081
       name: https
       protocol: HTTPS
     tls:
       mode: ISTIO_MUTUAL
     hosts:
       - "*"
```

1 在 **istio-system** 命名空间中为 Knative 本地网关定义服务。

2 在 **knative-serving 命名空间中** 定义一个入口网关。网关使用您在此流程前面创建的 TLS secret。入口网关处理 Knative 的外部流量。

3 在 **knative-serving** 命名空间中为 Knative 定义本地网关。

14. 应用 **gateways.yaml** 文件来创建定义的资源。

```
$ oc apply -f gateways.yaml
```

您会看到以下输出：

```
service/knative-local-gateway created
gateway.networking.istio.io/knative-ingress-gateway created
gateway.networking.istio.io/knative-local-gateway created
```

验证

- 查看您创建的网关。

```
$ oc get gateway --all-namespaces
```

确认您看到在 **knative-serving** 命名空间中创建的本地和入口网关，如下例所示：

```
NAMESPACE         NAME                     AGE
knative-serving   knative-ingress-gateway   69s
knative-serving   knative-local-gateway     2m
```

3.3.5. 安装 KServe

要完成 KServe 的手动安装，您必须安装 Red Hat OpenShift AI 附加组件，它将安装 Red Hat OpenShift AI Operator。然后，您可以使用 Operator 来安装 KServe。

先决条件

- 有 OpenShift 集群的集群管理员特权。
- 您的集群有一个有 4 个 CPU 和 16 GB 内存的节点。
- 您已下载并安装 OpenShift 命令行界面 (CLI)。请参阅安装 OpenShift CLI (Red Hat OpenShift Dedicated) 或 安装 OpenShift CLI (Red Hat OpenShift Service on AWS)。
- 您已创建了 Red Hat OpenShift Service Mesh 实例。
- 您已创建了 Knative Serving 实例。
- 您已为 Knative Serving 创建安全网关。
- 您已在 OpenShift 集群中安装了 Red Hat OpenShift AI 附加组件。这会安装 Red Hat OpenShift AI Operator，并创建一个默认的 **DataScienceCluster** 对象。

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. 要安装 KServe，请按如下所示配置 OpenShift Service Mesh 组件：

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，添加和配置 **serviceMesh** 组件，如下所示：

```
spec:
 serviceMesh:
   managementState: Unmanaged
```

e. 点击 **Save**。

4. 要安装 KServe，请配置 KServe 和 OpenShift Serverless 组件，如下所示：

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

b. 点 **Data Science Cluster** 选项卡。

c. 单击 **default-dsc** DSC 对象。

d. 点 **YAML** 标签。

e. 在 **spec.components** 部分中，配置 **kserve** 组件，如下所示：

```
spec:
 components:
   kserve:
     managementState: Managed
```

f. 在 **kserve** 组件中，添加 **服务组件** 并配置它，如下所示：

```
spec:
 components:
   kserve:
     managementState: Managed
     serving:
       managementState: Unmanaged
```

g. 点击 **Save**。

3.3.6. 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**。

3.4. 为 SINGLE-MODEL 服务平台添加授权供应商

您可以将 Authorino 添加为 single-model 服务平台的授权供应商。通过添加授权供应商，您可以为平台上部署的模型启用令牌授权，这样可确保只有授权方能够对模型发出推测请求。

用于将 Authorino 添加为授权提供程序的方法取决于您如何安装单model服务平台。该平台的安装选项如下所述：

自动安装

如果您还没有在 OpenShift 集群上创建 **ServiceMeshControlPlane** 或 **KNativeServing** 资源，您可以将 Red Hat OpenShift AI Operator 配置为安装 KServe 及其依赖项。您可以包含 Authorino 作为自动安装过程的一部分。
有关自动安装的更多信息，包括 Authorino，请参阅配置 KServe 的自动安装。

手动安装

如果您已在 OpenShift 集群上创建了 **ServiceMeshControlPlane** 或 **KNativeServing** 资源，则无法配置 Red Hat OpenShift AI Operator 来安装 KServe 及其依赖项。在这种情况下，您必须手动安装 KServe。您还必须手动配置 Authorino。
有关手动安装的更多信息，包括 Authorino，请参阅 手动安装 KServe。

3.4.1. 手动添加授权供应商

您可以将 Authorino 添加为 single-model 服务平台的授权供应商。通过添加授权供应商，您可以为平台上部署的模型启用令牌授权，这样可确保只有授权方能够对模型发出推测请求。

要手动将 Authorino 添加为授权提供程序，您必须安装 **Red Hat - Authorino** Operator，创建 Authorino 实例，然后将 OpenShift Service Mesh 和 KServe 组件配置为使用该实例。

重要

要手动添加授权供应商，您必须对 OpenShift Service Mesh 实例进行配置更新。要确保 OpenShift Service Mesh 实例处于支持状态，请只进行本节中显示的更新。

先决条件

- 您已查看了将 Authorino 作为授权提供程序添加的选项，并将手动安装作为适当的选项。请参阅添加授权提供程序。
- 您已手动安装 KServe 及其依赖项，包括 OpenShift Service Mesh。请参阅 手动安装 KServe。
- 手动安装 KServe 时，您可以将 **serviceMesh** 组件的 **managementState** 字段的值设置为 **Unmanaged**。手动添加 Authorino 需要此设置。请参阅 安装 KServe。

3.4.2. 安装 Red Hat Authorino Operator

在将 Autorino 添加为授权提供程序前，您必须在 OpenShift 集群上安装 **Red Hat - Authorino** Operator。

先决条件

- 有 OpenShift 集群的集群管理员特权。

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 → OperatorHub**。
3. 在 **OperatorHub** 页面上，在 **Filter by keyword** 字段中键入 **Red Hat - Authorino**。
4. 点 **Red Hat - Authorino** Operator。
5. 在 **Red Hat - Authorino** Operator 页面中，查看 Operator 信息，然后点 **Install**。
6. 在 **Install Operator** 页面中，保留 **Update channel**,**Version**,**Installation mode**,**Installed Namespace** 和 **Update Approval** 的默认值。
7. 点 **Install**。

验证

- 在 OpenShift Web 控制台中，点 **Operators → Installed Operators**，并确认 **Red Hat - Authorino** 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。

3.4.3. 创建 Authorino 实例

在 OpenShift 集群上安装 **Red Hat - Authorino** Operator 时，您必须创建一个 Authorino 实例。

先决条件

- 已安装 **Red Hat - Authorino** Operator。
- 您有将资源添加到创建 OpenShift Service Mesh 实例的项目中的权限。请参阅创建 OpenShift Service Mesh 实例。
  如需有关 OpenShift 权限的更多信息，请参阅使用 RBAC 定义和应用权限 (Red Hat OpenShift Dedicated)或使用 RBAC 定义并应用权限 (Red Hat OpenShift Service on AWS)。

流程

1. 打开一个新的终端窗口。
2. 登录到 OpenShift 命令行界面(CLI)，如下所示：

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

3. 创建一个命名空间来安装 Authorino 实例。

```
$ oc new-project <namespace_for_authorino_instance>
```

注意

自动安装过程为 Authorino 实例创建一个名为 **redhat-ods-applications-auth-provider** 的命名空间。考虑在手动安装时使用相同的命名空间名称。

4. 要在现有 OpenShift Service Mesh 实例中为 Authorino 实例注册新命名空间，请创建一个包含以下内容的新 YAML 文件：

```
apiVersion: maistra.io/v1
kind: ServiceMeshMember
metadata:
  name: default
  namespace: <namespace_for_authorino_instance>
spec:
  controlPlaneRef:
    namespace: <namespace_for_service_mesh_instance>
    name: <name_of_service_mesh_instance>
```

5. 保存 YAML 文件。
6. 在集群中创建 **ServiceMeshMember** 资源。

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

7. 要配置 Authorino 实例，请创建一个新的 YAML 文件，如下例所示：

```
apiVersion: operator.authorino.kuadrant.io/v1beta1
kind: Authorino
metadata:
  name: authorino
```

```
    namespace: <namespace_for_authorino_instance>
  spec:
    authConfigLabelSelectors: security.opendatahub.io/authorization-group=default
    clusterWide: true
    listener:
      tls:
        enabled: false
    oidcServer:
      tls:
        enabled: false
```

8. 保存 YAML 文件。
9. 在集群中创建 **Authorino** 资源。

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

10. 对 Authorino 部署进行补丁以注入 Istio sidecar，这会使 OpenShift Service Mesh 实例的 Authorino 实例的一部分。

```
$ oc patch deployment <name_of_authorino_instance> -n
<namespace_for_authorino_instance> -p '{"spec": {"template":{"metadata":{"labels":
{"sidecar.istio.io/inject":"true"}}}} }'
```

验证

- 确认 Authorino 实例正在运行，如下所示：
  1. 检查您为 Authorino 实例创建的命名空间中运行的 pod （和容器），如下例所示：

```
$ oc get pods -n redhat-ods-applications-auth-provider -o="custom-
columns=NAME:.metadata.name,STATUS:.status.phase,CONTAINERS:.spec.containers[*
].name"
```

  2. 确认输出类似以下示例：

```
NAME                        STATUS    CONTAINERS
authorino-6bc64bd667-kn28z  Running   authorino,istio-proxy
```

如示例所示，有一个用于 Authorino 实例的 pod。pod 为 Authorino 和您注入的 Istio sidecar 容器。

3.4.4. 将 OpenShift Service Mesh 实例配置为使用 Authorino

创建 Authorino 实例时，您必须将 OpenShift Service Mesh 实例配置为使用 Authorino 作为授权供应商。

重要

为确保 OpenShift Service Mesh 实例处于支持状态，请只 进行以下流程中显示的配置更新。

先决条件

- 您已创建了 Authorino 实例，并在 OpenShift Service Mesh 实例中注册 Authorino 实例的命名空间。
- 有修改 OpenShift Service Mesh 实例的权限。请参阅创建 OpenShift Service Mesh 实例。

流程

1. 在一个终端窗口中，如果您还没有以具有更新 OpenShift Service Mesh 实例的用户身份登录 OpenShift 集群，请登录 OpenShift CLI，如下例所示：

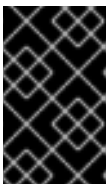
```
$ oc login <openshift_cluster_url> -u <username> -p <password>
```

2. 创建包含以下内容的新 YAML 文件：

```
spec:
 techPreview:
  meshConfig:
    extensionProviders:
    - name: redhat-ods-applications-auth-provider
      envoyExtAuthzGrpc:
        service: <name_of_authorino_instance>-authorino-
authorization.<namespace_for_authorino_instance>.svc.cluster.local
        port: 50051
```

3. 保存 YAML 文件。
4. 使用 **oc patch** 命令将 YAML 文件应用到 OpenShift Service Mesh 实例。

```
$ oc patch smcp <name_of_service_mesh_instance> --type merge -n
<namespace_for_service_mesh_instance> --patch-file <file_name>.yaml
```



重要

只有在 OpenShift Service Mesh 实例中尚未指定其他扩展供应商时，才能应用显示的配置作为补丁。如果您已经指定了其他扩展供应商，则必须手动编辑 **ServiceMeshControlPlane** 资源来添加配置。

验证

- 验证您的 Authorino 实例是否已作为扩展供应商添加到 OpenShift Service Mesh 配置中，如下所示：
  1. 检查 OpenShift Service Mesh 实例的 **ConfigMap** 对象：

```
$ oc get configmap istio-<name_of_service_mesh_instance> -n
<namespace_for_service_mesh_instance> --output=jsonpath={.data.mesh}
```

  2. 确认您看到与以下示例类似的输出，这表明 Authorino 实例已成功添加为扩展提供程序。

```
defaultConfig:
  discoveryAddress: istiod-data-science-smcp.istio-system.svc:15012
  proxyMetadata:
    ISTIO_META_DNS_AUTO_ALLOCATE: "true"
    ISTIO_META_DNS_CAPTURE: "true"
```

```
    PROXY_XDS_VIA_AGENT: "true"
  terminationDrainDuration: 35s
  tracing: {}
dnsRefreshRate: 300s
enablePrometheusMerge: true
extensionProviders:
- envoyExtAuthzGrpc:
    port: 50051
    service: authorino-authorino-authorization.opendatahub-auth-provider.svc.cluster.local
  name: opendatahub-auth-provider
ingressControllerMode: "OFF"
rootNamespace: istio-system
trustDomain: null%
```

3.4.5. 为 KServe 配置授权

要将 single-model 服务平台配置为使用 Authorino，您必须创建一个全局 **AuthorizationPolicy** 资源，该资源应用到部署模型时创建的 KServe predictor pod。另外，要考虑在对模型发出 inference 请求时发生的多个网络跃点，您必须创建一个 **EnvoyFilter** 资源，以持续将 HTTP 主机标头重置为最初包含在ference 请求中的标头。

先决条件

- 您已创建了 Authorino 实例，并将 OpenShift Service Mesh 配置为使用它。
- 有更新集群中的 KServe 部署的权限。
- 您有将资源添加到创建 OpenShift Service Mesh 实例的项目中的权限。请参阅创建 OpenShift Service Mesh 实例。

流程

1. 在一个终端窗口中，如果您还没有以具有更新 KServe 部署权限的用户身份登录 OpenShift 集群，请登录 OpenShift CLI，如下例所示：

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

2. 创建包含以下内容的新 YAML 文件：

```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: kserve-predictor
spec:
  action: CUSTOM
  provider:
    name: redhat-ods-applications-auth-provider (1)
  rules:
    - to:
        - operation:
            notPaths:
              - /healthz
              - /debug/pprof/
              - /metrics
```

```
            - /wait-for-drain
  selector:
    matchLabels:
      component: predictor
```

1 *您指定的名称必须与添加到 OpenShift Service Mesh 实例的扩展供应商名称匹配。*

3. *保存 YAML 文件。*

4. *在命名空间中为 OpenShift Service Mesh 实例创建* ***AuthorizationPolicy*** *资源。*

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

5. *创建包含以下内容的另一个新 YAML 文件：*

```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: activator-host-header
spec:
  priority: 20
  workloadSelector:
    labels:
      component: predictor
  configPatches:
  - applyTo: HTTP_FILTER
    match:
      listener:
        filterChain:
          filter:
            name: envoy.filters.network.http_connection_manager
    patch:
      operation: INSERT_BEFORE
      value:
        name: envoy.filters.http.lua
        typed_config:
          '@type': type.googleapis.com/envoy.extensions.filters.http.lua.v3.Lua
          inlineCode: |
           function envoy_on_request(request_handle)
             local headers = request_handle:headers()
             if not headers then
               return
             end
             local original_host = headers:get("k-original-host")
             if original_host then
               port_seperator = string.find(original_host, ":", 7)
               if port_seperator then
                 original_host = string.sub(original_host, 0, port_seperator-1)
               end
               headers:replace('host', original_host)
             end
           end
```

显示的 ***EnvoyFilter*** *资源持续将 HTTP 主机标头重置为最初包含在任何推销请求中的标头。*

6. *在命名空间中为您的 OpenShift Service Mesh 实例创建 **EnvoyFilter** 资源。*

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

验证

- *检查 **AuthorizationPolicy** 资源是否已成功创建。*

```
$ oc get authorizationpolicies -n <namespace_for_service_mesh_instance>
```

确认输出类似以下示例：

```
NAME              AGE
kserve-predictor  28h
```

- *检查 **EnvoyFilter** 资源是否已成功创建。*

```
$ oc get envoyfilter -n <namespace_for_service_mesh_instance>
```

确认输出类似以下示例：

```
NAME                              AGE
activator-host-header             28h
```

第 4 章 安装多模式服务平台

为了部署小型和中型模型，OpenShift AI 包含一个基于 ModelMesh 组件的 多模式服务平台。在多型号服务平台上，可以从同一模型服务器部署多个模型并共享服务器资源。

要安装多模型服务平台或 ModelMesh，请按照 使用 Web 控制台安装 Red Hat OpenShift AI 组件 中描述的步骤操作。

第 5 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 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，如 将用户添加到 OpenShift AI 用户组 中所述。

流程

1. 登录 OpenShift Web 控制台。
2. 点应用程序启动程序()。
3. 右键单击 **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。

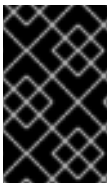
其他资源

- 登录到 OpenShift AI

将用户添加到 OpenShift AI 用户组.

第 6 章 启用 NVIDIA GPU

在 OpenShift AI 中使用 NVIDIA GPU 之前，您必须安装 NVIDIA GPU Operator。



重要

NVIDIA GPU 附加组件不再被支持。反之，通过安装 NVIDIA GPU Operator 来启用 GPU。如果您的部署有一个之前安装的 NVIDIA GPU 附加组件，在安装 NVIDIA GPU Operator 前，请使用 Red Hat OpenShift Cluster Manager 从集群中卸载 NVIDIA GPU 附加组件。

先决条件

- 已登陆到您的 OpenShift 集群。
- 在 OpenShift 集群中具有 **cluster-admin** 角色。

流程

1. 要在 OpenShift 集群上启用 GPU 支持，请按照 NVIDIA 文档中的 Red Hat OpenShift Container Platform 上的 NVIDIA GPU Operator 的说明进行操作。
2. 删除 **migration-gpu-status** ConfigMap。
   a. 在 OpenShift 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. 重启仪表板 replicaset。
   a. 在 OpenShift 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 Web 控制台的 Operators → Installed Operators 页面中。**

- reset 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。

注意

在 OpenShift AI 中，红帽支持在同一集群中使用加速器。红帽不支持在加速器之间远程直接内存访问(RDMA)，或使用网络上的加速器，例如使用 NVIDIA GPUDirect 或 NVLink 等技术。

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 使用证书

证书供 OpenShift 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。

7.1. 了解 OPENSHIFT AI 中的证书

对于依赖自签名证书的 OpenShift 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。

7.1.1. 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>
```

7.1.2. 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。
- **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>
```

7.2. 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 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，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 集群中的 **DSCInitialization** 资源。
- 如 *CLI 入门* 所述，安装了 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 OpenShift。
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：

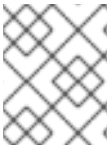
```
$ oc get configmaps --all-namespaces -l app.kubernetes.io/part-of=opendatahub-operator | grep odh-trusted-ca-bundle
```

- 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
```

7.3. 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

只有当您将 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

- 有 OpenShift 集群的集群管理员特权。
- 如 CLI 入门所述，安装了 OpenShift 命令行界面(oc)。

流程

1. 在 OpenShift 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。

7.4. 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

- 有 OpenShift 集群的集群管理员特权。
- 如 CLI 入门 所述，安装了 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。

7.5. 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

- 有 OpenShift 集群的集群管理员特权。

流程

1. 在 OpenShift 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
    customCABundle: example123
```

2. 点击 Save。
3. 点 Workloads → ConfigMaps。
4. 从项目列表中选择一个项目。
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。

7.6. 在 OPENSHIFT 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。

7.6.1. 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 中所述。

在数据科学项目管道中使用这些证书不需要额外的配置。

7.6.1.1. 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。
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 OpenShift 控制台。
2. 进入与 data Science 项目对应的 OpenShift 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：

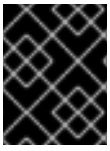
```
pod=$(oc get pod -n ${dsProject} -l app=ds-pipeline-${dspa} --no-headers | awk '{print $1}')
```

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。

7.6.2. 使用带有工作台的证书



重要

默认情况下，自签名证书应用到您在集中配置证书后创建的工作台，如使用 证书 中所述。要将集中配置的证书应用到现有的工作台，请停止并重启工作台。

自签名证书存储在 /etc/pki/tls/custom-certs/ca-bundle.crt 中。工作台使用环境变量预设，该变量将软件包指向此路径，并涵盖许多流行的 HTTP 客户端软件包。对于默认未包含的软件包，您可以提供此证书路径。例如，对于 kfp 软件包连接到数据科学管道服务器：

```
from kfp.client import Client

with open(sa_token_file_path, 'r') as token_file:
    bearer_token = token_file.read()

    client = Client(
        host='https://<GO_TO_ROUTER_OF_DS_PROJECT>/',
        existing_token=bearer_token,
        ssl_ca_cert='/etc/pki/tls/custom-certs/ca-bundle.crt'
    )
    print(client.list_experiments())
```

7.6.2.1. 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 8 章 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 集群中的 DSCInitialization 资源。
- 如 CLI 入门 所述，安装了 OpenShift 命令行界面(oc)。

流程

1. 以集群管理员身份登录 OpenShift。
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。

8.1. 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 -I {} 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 9 章 常见安装问题的故障排除

如果您在安装 Red Hat OpenShift AI Add-on 时遇到问题，请阅读本节以了解导致问题的原因以及如何解决这个问题。

如果此处或发行注记中没有包括该问题，请联系红帽支持团队。在提交问题单时，包含集群的调试信息会很有帮助。您可以使用 must-gather 工具来收集此信息，如 Must-Gather for Red Hat OpenShift AI 和收集集群数据 中所述。

您还可以调整 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。如需更多信息，请参阅配置 OpenShift AI Operator 日志记录器。

9.1. RED HAT OPENSHIFT AI OPERATOR 无法从镜像 REGISTRY 中检索

问题

当尝试从镜像 registry 检索 Red Hat OpenShift AI Operator 时，会出现 **Failure to pull from quay** 错误信息。在以下情况下，Red Hat OpenShift AI Operator 可能无法检索：

- 镜像 registry 不可用。
- 您的网络连接存在问题。
- 集群无法正常工作，因此无法检索镜像 registry。

诊断

检查 OpenShift 中的 Events 部分中的日志，以了解有关 **Failure to pull from quay** 错误消息的更多信息。

解决方案

- 联系红帽支持。

9.2. 由于集群资源不足而无法安装 OPENSHIFT AI

问题

尝试安装 OpenShift AI 时，会出现一条错误消息，指出尚未满足安装先决条件。

诊断

1. 登录到 Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)。
2. 点 Clusters。
   Clusters 页面将打开。
3. 点您要在其上安装 OpenShift AI 的集群名称。
   此时会打开集群的 Details 页面。
4. 点 Add-ons 选项卡，找到 Red Hat OpenShift AI 标题。
5. 点 Install。此时会出现 Configure Red Hat OpenShift AI 窗格。
6. 如果安装失败，点先决条件标签页。

7. 请注意错误消息。如果错误消息显示您需要新的机器池，或者需要更多资源，执行适当的操作来解决这个问题。

解决方案

- 您可能需要向集群添加更多资源，或者增加机器池的大小。要提高集群的资源，请联系您的基础架构管理员。有关增大机器池大小的更多信息，请参阅 节点 并将 其他资源分配给 OpenShift AI 用户。

9.3. OPENSHIFT AI 在不支持的基础架构上无法安装

问题

您在 Red Hat OpenShift AI Operator 不支持的环境中部署。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将 Project 设置为 All Projects 或 redhat-ods-operator。
4. 点 rhods-operator-<random string> pod。
   Pod 详情页面会显示。
5. 点 Logs。
6. 从下拉列表中选择 rhods-operator。
7. 检查日志中的 ERROR: Deploying on $infrastructure, not supported。Failing Installation 错误信息。

解决方案

- 在继续新安装前，请确保您有完全受支持的环境来安装 OpenShift AI。如需更多信息，请参阅 Red Hat OpenShift AI: 支持的配置。

9.4. 创建 OPENSHIFT AI 自定义资源(CR)失败

问题

在安装过程中，OpenShift AI 自定义资源(CR)不会被创建。这个问题在未知情况下发生。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将 Project 设置为 All Projects 或 redhat-ods-operator。
4. 点 rhods-operator-<random string> pod。
   Pod 详情页面会显示。

5. 点 Logs。
6. 从下拉列表中选择 rhods-operator。
7. 检查 ERROR: Attempt 的日志以创建 ODH CR failed. 错误消息。

解决方案

- 联系红帽支持。

9.5. OPENSHIFT AI NOTEBOOKS 自定义资源(CR)的创建失败

问题

在安装过程中，OpenShift AI Notebooks 自定义资源(CR)不会被创建。这个问题在未知情况下发生。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将 Project 设置为 All Projects 或 redhat-ods-operator。
4. 点 rhods-operator-<random string> pod。
   Pod 详情页面会显示。
5. 点 Logs。
6. 从下拉列表中选择 rhods-operator。
7. 检查 ERROR: Attempt logs to create the RHODS Notebooks CR failed. 错误信息。

解决方案

- 联系红帽支持。

9.6. OPENSHIFT AI 仪表板无法访问

问题

安装 OpenShift AI 后，redhat-ods-applications、redhat-ods-monitoring 和 redhat-ods-operator 项目命名空间是 Active，但由于 pod 中的错误，您无法访问仪表板。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将项目设置为 All Projects。
4. 单击 Filter，然后选中 Running 和 Completed 以外的每个状态的复选框。
   页面中显示出错的 pod。

解决方案

- *要查看 Pod 的更多信息和故障排除步骤，请在 Pods 页面中点击 Pod 的 Status 列中的链接。*
- *如果 Status 列没有显示链接，请单击 Pod 名称以打开 Pod 详情页面，然后单击 Logs 选项卡。*

9.7. 无法创建基于 DEDICATED-ADMINS 基于角色的访问控制(RBAC)策略

问题

无法为目标项目中的 dedicated-admins 组提供基于角色的访问控制(RBAC)策略。这个问题在未知情况下发生。

诊断

1. *在 OpenShift Web 控制台中，切换到 Administrator 视角。*
2. *点击 Workloads → Pods。*
3. *将 Project 设置为 All Projects 或 redhat-ods-operator。*
4. *点 rhods-operator-<random string> pod。*
   Pod 详情页面会显示。
5. *点 Logs。*
6. *从下拉列表中选择 rhods-operator。*
7. *检查 ERROR: Attempt 的日志，以在 $target_project failed. 错误消息中为 dedicated admins 组创建 RBAC 策略。*

解决方案

- *联系红帽支持。*

9.8. DEAD MAN 的 SNITCH OPERATOR 的 SECRET 没有被创建

问题

Managed Tenants SRE 自动化进程存在问题，从而导致 Dead Man 的 Snitch Operator 的 secret 不会被创建。

诊断

1. *在 OpenShift Web 控制台中，切换到 Administrator 视角。*
2. *点击 Workloads → Pods。*
3. *将 Project 设置为 All Projects 或 redhat-ods-operator。*
4. *点 rhods-operator-<random string> pod。*
   Pod 详情页面会显示。
5. *点 Logs。*
6. *从下拉列表中选择 rhods-operator。*

7. 检查 **ERROR: Dead Man Snitch secret does not exist** 的日志。 错误消息。

解决方案

- 联系红帽支持。

9.9. PAGERDUTY SECRET 不会被创建

问题

Managed Tenants SRE 自动化进程存在问题，导致 PagerDuty 的 secret 无法被创建。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将 Project 设置为 All Projects 或 redhat-ods-operator。
4. 点 **rhods-operator-<random string>** pod。
   Pod 详情页面会显示。
5. 点 Logs。
6. 从下拉列表中选择 rhods-operator。
7. 检查 **ERROR: Pagerduty secret does not exist** 错误信息的日志。

解决方案

- 联系红帽支持。

9.10. SMTP SECRET 不存在

问题

Managed Tenants SRE 自动化进程存在问题，会导致 SMTP secret 无法创建。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将 Project 设置为 All Projects 或 redhat-ods-operator。
4. 点 **rhods-operator-<random string>** pod。
   Pod 详情页面会显示。
5. 点 Logs。
6. 从下拉列表中选择 rhods-operator。
7. 检查日志中的 **ERROR: SMTP secret does not exist** 错误消息。

解决方案

- 联系红帽支持。

9.11. ODH 参数 SECRET 不会被创建

问题

OpenShift AI 附加组件流的问题可能会导致无法创建 ODH 参数。

诊断

1. 在 OpenShift Web 控制台中，切换到 Administrator 视角。
2. 点击 Workloads → Pods。
3. 将 Project 设置为 All Projects 或 redhat-ods-operator。
4. 点 rhods-operator-<random string> pod。
   Pod 详情页面会显示。
5. 点 Logs。
6. 从下拉列表中选择 rhods-operator。
7. 检查 **ERROR: Addon managed odh parameter secret does not exist** 的日志。错误消息。

解决方案

- 联系红帽支持。

9.12. 安装 OPENSHIFT AI 2.9 或更高版本后，因为现有的 ARGO WORKFLOWS 资源，不会启用数据科学管道

问题

在安装了未由 OpenShift AI 安装的 Argo 工作流安装 OpenShift AI 2.9 或更高版本后，虽然 Datasciencepipelines 组件在 **DataScienceCluster** 对象中启用了 **datasciencepipelines** 组件，则不会启用数据科学管道。

诊断

安装 OpenShift AI 2.9 或更高版本后，Data Science Pipelines 选项卡在 OpenShift AI 仪表板导航菜单中不可见。

解决方案

- 删除集群中 Argo 工作流的独立安装。在从集群中删除任何不是由 OpenShift AI 创建的 Argo Workflows 资源后，会自动启用数据科学管道。

第 10 章 卸载 OPENSHIFT AI

使用 Red Hat OpenShift Cluster Manager 从 OpenShift 集群卸载 Red Hat OpenShift AI。

10.1. 了解卸载过程

为 OpenShift AI 的不同组件安装 Red Hat OpenShift AI 在 OpenShift 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 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
- 订阅、**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 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 集群中，但它们无法正常工作。卸载后，红帽建议您查看 OpenShift 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

10.2. 从 AMAZON EBS 备份存储数据

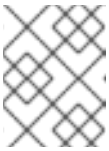
红帽建议定期备份持久性卷声明(PVC)中的数据。在删除用户和卸载 OpenShift AI 之前备份您的数据非常重要，因为在卸载 OpenShift AI 时，所有 PVC 都会被删除。

先决条件

- *您有 Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)的凭证。*
- *具有 OpenShift Dedicated 集群的管理员访问权限。*
- *您有部署 OpenShift Dedicated 集群的 Amazon Web Services (AWS) 帐户的凭证。*

流程

1. *确定您要备份的持久性卷 (PV) 的 ID。*
   a. *在 OpenShift Dedicated Web 控制台中，更改到 Administrator 视角。*
   b. *点 Home → Projects。*
   c. *点 rhods-notebooks 项目。*
   将打开项目的 Details 页面。
   d. *点 Inventory 部分中的 PersistentVolumeClaims。*
   PersistentVolumeClaims 页面将打开。
   e. *记录您要备份的持久性卷 (PV) 的 ID。*



注意

您需要注意的持久性卷 (PV)，才能识别正确的 EBS 卷以便在 AWS 实例中备份。

2. *找到包含您要备份的 PV 的 EBS 卷。*
   如需更多信息，请参阅 Amazon Web Services 文档：创建 Amazon EBS 快照。
   a. *登录到 AWS (https://aws.amazon.com)，并确保您查看部署了 OpenShift Dedicated 集群的区域。*
   b. *点 Services。*
   c. *点 Compute → EC2。*
   d. *在侧边导航中点 Elastic Block Storage → Volumes。*
   此时会打开 Volumes 页面。
   e. *在搜索栏中，输入您之前记下的持久性卷 (PV) 的 ID。*
   卷页面会重新加载以显示搜索结果。
   f. *单击显示的卷，并验证任何 kubernetes.io/created-for/pvc/namespace 标签是否包含值 rhods-notebooks，以及任何 kubernetes.io/ created-for/pvc/name 标签是否与 EC2 卷用于的持久性卷的名称匹配，如 jupyter-nb-user1-pvc。*
3. *备份包含持久性卷 (PV) 的 EBS 卷。*
   a. *右键点击您要备份的卷，然后从列表中选择 Create Snapshot。*
   Create Snapshot 页面将打开。
   b. *为卷输入 Description。*

c. 点 Create Snapshot。
创建卷的快照。

d. 点 Close。

验证

- 您创建的快照在 AWS 中的 Snapshots 页面中可见。

其他资源

- Amazon Web Services 文档：创建 Amazon EBS 快照

10.3. 从 GOOGLE PERSISTENT DISK 备份存储数据

红帽建议定期备份持久性卷声明(PVC)中的数据。在删除用户和卸载 OpenShift AI 之前备份您的数据非常重要，因为卸载 OpenShift AI 时所有 PVC 都会被删除。

先决条件

- 您有 Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)的凭证。
- 具有 OpenShift Dedicated 集群的管理员访问权限。
- 您有部署 OpenShift Dedicated 集群的 Google Cloud Platform (GCP)帐户的凭证。

流程

1. 确定您要备份的持久性卷 (PV) 的 ID。
   a. 在 OpenShift Dedicated Web 控制台中，更改到 Administrator 视角。
   b. 点 Home → Projects。
   c. 点 rhods-notebooks 项目。
   将打开项目的 Details 页面。
   d. 点 Inventory 部分中的 PersistentVolumeClaims。
   PersistentVolumeClaims 页面将打开。
   e. 记录您要备份的持久性卷 (PV) 的 ID。
   需要持久性卷(PV) ID 来识别要在 GCP 实例中备份的正确持久性磁盘。
2. 找到包含您要备份的 PV 的持久性磁盘。
   a. 登录到 Google Cloud 控制台(https://console.cloud.google.com)，并确保您查看部署了 OpenShift Dedicated 集群的区域。
   b. 单击导航菜单(categories)，然后单击 Compute Engine。
   c. 在侧边导航中，在 Storage 下点 Disks。
   Disks 页面将打开。
   d. 在 Filter 查询框中，输入您之前记下的持久性卷(PV)的 ID。
   Disks 页面会重新加载以显示搜索结果。

e. 单击显示的磁盘，并验证任何 kubernetes.io/created-for/pvc/namespace 标签是否包含值 rhods-notebooks，以及任何 kubernetes.io/ created-for/pvc/name 标签是否与持久性卷的名称匹配，例如 jupyterhub-nb-user1-pvc。

3. 备份包含持久性卷(PV)的持久磁盘。

a. 从顶部导航中选择 CREATE SNAPSHOT。
Create a snapshot 页面将打开。

b. 输入快照的唯一 Name。

c. 在 Source disk 下，验证您要备份的持久磁盘是否显示。

d. 根据需要更改任何可选设置。

e. 点 CREATE。
创建持久磁盘的快照。

验证

- 您创建的快照在 GCP 的 Snapshots 页面中可见。

其他资源

- Google Cloud 文档：创建和管理磁盘快照

10.4. 卸载 OPENSHIFT AI

您可以使用 Red Hat OpenShift Cluster Manager 从 OpenShift 集群安全地卸载 Red Hat OpenShift AI。

先决条件

- Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)的凭证。
- 管理员对 OpenShift 集群的访问权限。
- 对于 AWS 集群，您备份了包含持久性卷声明(PVC)的 EBS 卷。如需更多信息，请参阅 Amazon Web Services 文档：创建 Amazon EBS 快照。
- 对于 GCP 集群，您备份了包含持久性卷声明(PVC)的持久性卷。如需更多信息，请参阅 Google Cloud 文档：创建和管理磁盘快照。

流程

1. 登录到 Red Hat OpenShift Cluster Manager (https://console.redhat.com/openshift/)。
2. 点 Clusters。
Clusters 页面将打开。
3. 点托管要卸载的 OpenShift AI 实例的集群名称。
此时会打开集群的 Details 页面。
4. 点 Add-ons 选项卡，找到 Red Hat OpenShift AI 标题。
5. 点 Uninstall。

完成此过程需要大约 30 分钟时间。卸载 OpenShift AI 时不要手动删除任何资源，因为这可能会干扰卸载过程。

OpenShift AI 被卸载，并删除与 OpenShift AI 实例关联的任何持久性卷声明(PVC)。但是，您之前创建的 OpenShift AI 任何用户组都会在集群中。

验证

- 在 Red Hat OpenShift Cluster Manager 中，在集群的 Add-ons 选项卡中，确认 OpenShift AI 标题没有显示 **Installed** 状态。
- 在 OpenShift 集群中，点 Home → Projects 并确认以下项目命名空间不可见：
  - **redhat-ods-applications**
  - **redhat-ods-monitoring**
  - **redhat-ods-operator**

其他资源

- Amazon Web Services 文档：创建 Amazon EBS 快照
- Google Cloud 文档：创建和管理有关删除用户及其资源的磁盘分区

10.5. 其他资源

关于删除用户及其资源