



Red Hat OpenShift AI Self-Managed 2.10

升级 OpenShift AI Self-Managed

在 OpenShift Container Platform 上升级 OpenShift AI

Red Hat OpenShift AI Self-Managed 2.10 升级 OpenShift AI Self-Managed

在 OpenShift Container Platform 上升级 OpenShift AI

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 OpenShift Container Platform 上升级 OpenShift AI。

目录

前言	3
第 1 章 升级 OPENSIFT AI SELF-MANAGED 概述	4
第 2 章 为 OPENSIFT AI 配置升级策略	5
第 3 章 升级 OPENSIFT AI 的要求	6
第 4 章 升级后添加 CA 捆绑包	7

前言

作为集群管理员，您可以配置 OpenShift AI Operator 自动或手动升级。



注意

升级到 OpenShift AI 2.10 后，使用 Data Science Pipelines (DSP) 1.0 创建的管道继续运行，但无法从 OpenShift AI 仪表盘访问。如果您是当前的 DSP 用户，请不要升级到带有 DSP 2.0 的 OpenShift AI，直到您准备好迁移到新的管道解决方案。

第 1 章 升级 OPENSIFT AI SELF-MANAGED 概述

作为集群管理员，您可以为 Red Hat OpenShift AI Operator 配置自动或手动升级。



注意

有关在断开连接的环境中将 OpenShift AI 作为 OpenShift 集群上的自我管理软件的详情，请参考 [在断开连接的环境中升级 OpenShift AI Self-Managed](#)。



重要

Data Science Pipelines (DSP) 2.0 包含 Argo 工作流的安装。OpenShift AI 不支持将这个 Argo 工作流安装的用户使用。要使用 DSP 升级到 OpenShift AI 2.9，请确保集群中没有单独安装 Argo 工作流。



注意

升级到 OpenShift AI 2.9 或更高版本后，使用 DSP 1.0 创建的管道会继续运行，但无法从 OpenShift AI 仪表板访问。如果您是当前的 DSP 用户，请不要升级到带有 DSP 2.0 的 OpenShift AI，直到您准备好迁移到新的管道解决方案。

- 如果配置自动升级，当有新版本的 Red Hat OpenShift AI Operator 可用时，Operator Lifecycle Manager (OLM) 将自动升级 Operator 的运行实例，而无需人为干预。
- 如果配置手动升级，则当有新版本的 Red Hat OpenShift AI Operator 可用时，OLM 会创建一个更新请求。
集群管理员必须手动批准更新请求，才能将 Operator 更新至新版本。如需有关 [批准待处理的 Operator 升级](#) 的更多信息，请参阅 [手动批准待处理的 Operator 升级](#)。
- 默认情况下，Red Hat OpenShift AI Operator 遵循一个后续更新过程。这意味着，如果当前版本和您要升级到的版本之间存在几个次版本，Operator Lifecycle Manager (OLM) 会在将 Operator 升级到最终的目标版本前，将 Operator 升级到每个次版本。如果配置自动升级，OLM 会自动将 Operator 升级到最新的可用版本，而无需人为干预。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。
红帽支持当前版本和以前 OpenShift AI Self-Managed 的次版本。如需更多信息，请参阅 [Red Hat OpenShift AI Self-Managed Life Cycle](#) 知识库文章。
- 升级 OpenShift AI 时，您应该 [完成升级 OpenShift AI 的要求](#)。
- 在 OpenShift AI 中使用加速器前，您的实例必须有关联的加速器配置集。如果您的 OpenShift Container Platform 实例具有加速器，则升级后会保留其加速器配置集。有关加速器的更多信息，请参阅 [使用加速器](#)。
- 笔记本镜像会在升级过程中集成到镜像流中，然后出现在 OpenShift AI 仪表板中。



注意

笔记本镜像由外部构建；它们是预先构建的镜像，执行每季度更改，并且不会随着每个 OpenShift AI 升级而改变。

其他资源

- [Operator Lifecycle Manager 工作流](#)

第 2 章 为 OPENSIFT AI 配置升级策略

作为集群管理员，您可以为 Red Hat OpenShift AI Operator 配置自动或手动升级策略。



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，如果当前版本和您要升级到的版本之间存在多个版本，Operator Lifecycle Manager (OLM) 会在将 Operator 升级到最终的目标版本前，将 Operator 升级到每个中间版本。如果配置自动升级，OLM 会自动将 Operator 升级到 *最新的可用版本*，而无需人为干预。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](#)。

先决条件

- 具有 OpenShift Container Platform 集群的集群管理员特权。
- 安装了 Red Hat OpenShift AI Operator。

流程

1. 以集群管理员身份登录 OpenShift Container Platform 集群 Web 控制台。
2. 在 **Administrator** 视角中，在左侧菜单中选择 **Operators** → **Installed Operators**。
3. 点 **Red Hat OpenShift AI Operator**。
4. 点 **Subscription** 标签页。
5. 在 **Update approval** 下，点铅笔图标并选择以下更新策略之一：
 - **自动**：在有新更新可用时即可安装新的更新。
 - **Manual**：集群管理员必须在安装开始前批准任何新的更新。
6. 点击 **Save**。

其他资源

- 有关 Red Hat OpenShift AI Operator 版本 2 中提供的订阅频道的更多信息，请参阅安装 [Red Hat OpenShift AI Operator](#)。
- 有关升级使用 OLM 安装的 Operator 的更多信息，请参阅 OpenShift Container Platform 文档中的 [更新已安装的 Operator](#)。

第 3 章 升级 OPENSIFT AI 的要求

本节介绍升级 OpenShift AI 时应该完成的任务。

检查 DataScienceCluster 对象中的组件

升级 Red Hat OpenShift AI 时，升级过程会自动使用上一个 **DataScienceCluster** 对象中的值。

升级后，您应该检查 **DataScienceCluster** 对象，并选择性地更新任何组件的状态，如 [使用 Web 控制台更新 Red Hat OpenShift AI 组件的安装状态](#) 中所述。

重新创建现有管道运行

当您升级到较新版本时，在以前的版本中创建的任何现有管道运行将继续引用上一版本的镜像（如预期）。

您必须删除管道运行（而不是管道）并创建新管道运行。您在较新版本中创建的管道运行可以正确地引用较新版本的镜像。

如需有关管道运行的更多信息，[请参阅管理管道运行](#)。

升级到 Data Science Pipelines (DSP) 2.0

在以前的版本中，OpenShift AI 中的数据科学管道基于 KubeFlow Pipelines v1。不再可以从 OpenShift AI 2.10 中的仪表板部署、查看或编辑基于 Data Science Pipelines (DSP) 1.0 的管道详情。如果您已经使用数据科学管道，红帽建议您继续使用 OpenShift AI 2.8，直到 DSP 2.0 中的完整功能奇偶校验已在稳定的 OpenShift AI 版本中提供，您可以迁移到新的管道解决方案。

DSP 2.0 包含 Argo 工作流的安装。OpenShift AI 不支持将这个 Argo 工作流安装的用户使用。要使用 DSP 2.0 安装或升级到 OpenShift AI 2.9，请确保在集群中没有 Argo 工作流安装。

如果要在升级到 OpenShift AI 2.10 后将现有管道和工作台与 DSP 2.0 搭配使用，您必须更新工作台以使用 2024.1 笔记本镜像版本，然后手动将您的管道从 DSP 1.0 迁移到 2.0。如需更多信息，[请参阅升级到 DSP 2.0](#)。

地址 KServe 要求

对于由 single-model 服务平台用来为大型模型提供服务的 KServe 组件，您必须满足以下要求：

- 要完全安装和使用 KServe，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，[请参阅为单模式服务平台添加授权供应商](#)。
- 如果您还没有启用 KServe 组件（即，在 **DataScienceCluster** 对象中将 **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](#)。

第 4 章 升级后添加 CA 捆绑包

Red Hat OpenShift AI 2.10 支持使用自签名证书。如果您已经从 OpenShift AI 2.7 或更早版本升级，您可以将自签名证书添加到集群中的 OpenShift AI 部署和 Data Science Projects 中。

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。
- 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**)。

如需更多信息，[请参阅使用证书](#)。

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**)。
- 从 2.7 或更早版本升级了 Red Hat OpenShift AI。如果您在 Red Hat OpenShift AI 的新安装中工作，请参阅 [添加 CA 捆绑包](#)。

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. 将以下内容添加到 **spec** 部分，将 **managementState** 字段设置为 **Managed**：

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

7. 如果要使用添加到集群范围的 CA 捆绑包中的自签名证书，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。
8. 如果要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，请按照以下步骤操作：
 - a. 将自定义证书添加到 **default-dsci** 对象的 **customCABundle** 字段中，如下例所示：

```
spec:
  trustedCABundle:
    managementState: Managed
    customCABundle: |
```

```
-----BEGIN CERTIFICATE-----  
examplebundle123  
-----END CERTIFICATE-----
```

b. 点击 **Save**。

Red Hat OpenShift AI Operator 创建一个 **odh-trusted-ca-bundle** ConfigMap，其中包含所有新的和现有非reserved 命名空间中的证书。

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
```