Red Hat Trusted Application Pipeline 1.0

安装 Red Hat Trusted Application Pipeline

了解如何在集群中安装 Red Hat Trusted Application Pipeline。

Last Updated: 2024-07-02

Red Hat Trusted Application Pipeline 1.0 安装 Red Hat Trusted Application Pipeline

了解如何在集群中安装 Red Hat Trusted Application Pipeline。

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

本文档提供有关如何在集群中安装 Red Hat Trusted Application Pipeline 的信息。

目录

前言

红帽受信任的应用程序 Pipeline (RHTAP)不是单个产品。相反，它是一个一系列产品，组合组成一个高度自动化、可定制和安全的平台用于构建应用程序。

RHTAP 包括以下产品：

- Red Hat Developer Hub ：面向开发人员的自助服务门户。
- OpenShift GitOps: 管理 Kubernetes 部署及其基础架构。
- OpenShift Pipelines: 启用自动化并为软件的持续集成和持续交付(CI/CD)提供可见性。
- trusted Artifact Signer: 为 RHTAP 生成的工件签名和验证。
- 受信任的配置文件分析器 ：提供有关安全状况的可操作信息。

它还依赖于以下产品：

- quay.io ：一个容器 registry，其中 RHTAP 存储您的工件。
- 高级集群安全性(ACS) ：RHTAP 用来扫描工件的安全工具。

注意

要准确查看这些产品的 RHTAP 支持哪个版本，请参考我们 发行注记 中的兼容性和支持列表。

由于完全正常运行的 RHTAP 实例涉及上面列出的所有产品，因此安装 RHTAP 需要时间和精力。但是，我们已尽可能自动化这个过程，并在此处提供说明非常有帮助且简洁。

另外，请注意 RHTAP 安装程序不是管理器：它不支持升级。安装程序生成您的第一个 RHTAP 部署。安装后，您可以单独管理 RHTAP 中的每个产品。

在开始安装前，您必须满足六个先决条件。然后您必须完成 7 个步骤。

先决条件

- ClusterAdmin 通过 CLI 和 Web 控制台访问 OpenShift Container Platform (OCP)集群
- 一个 Red Hat Advanced Cluster Security 实例，以及该实例中的以下值：
  - ACS API 令牌。您可以按照 此处 创建 API 令牌的先决条件的说明进行操作。
  - ACS 中央端点 URL。您可以按照以下说明 配置 端点。
- 要让 ACS 访问镜像 registry 中的私有存储库，需要为您的特定 registry 配置 ACS
  - 对于 Quay.io，在 Integrations→Image Integrations 下选择 Quay.io 卡
  - 添加 OAUTH 令牌以访问您的特定 Quay.io 实例
  - 通过测试按钮验证访问权限。这将确保 RHTAP 被要求扫描私有镜像，ACS 将具有访问权限
- Quay.io 帐户

- Helm CLI 工具
- GitHub 帐户

流程

1. 为 RHTAP 创建 GitHub 应用程序
2. 分叉模板目录
3. 创建 GitOps git 令牌
4. 创建 Docker 配置值
5. 创建 private-values.yaml 文件
6. 在集群中安装 RHTAP
7. 最终调整 GitHub 应用程序

以下文档页面详细介绍了每个程序。如果您有先决条件，您可以通过创建 GitHub 应用程序来启动安装过程。

第 1 章 为 RHTAP 创建 GITHUB 应用程序

通过为 RHTAP 创建 GitHub 应用程序，开发人员可以对 Red Hat Developer Hub 进行身份验证，这是他们可以使用 RHTAP 的用户界面(UI)。此 GitHub 应用还允许 RHTAP 访问托管在 GitHub 上的开发人员的源代码。

请记住，您必须在您拥有的 GitHub 组织中创建并安装新应用程序，并希望用于 Red Hat Trusted Application Pipeline 实例。RHTAP 随后可以在该机构中创建新的存储库，以用作其构建的应用的源代码。

先决条件

- GitHub 机构的所有权

步骤

1. 登录 GitHub 并进入您的机构(**Settings & gt; Organizations**)。
2. 点击您拥有的组织，并希望用于此 RHTAP 实例。或者您可以选择 **New organization** 来创建新机构。
3. 在机构上下文中，进入 GitHub Apps 页面 (**Settings > Developer settings > GitHub Apps**)。
4. 在页面右侧，在顶部横幅旁边，选择 **New GitHub App**。
5. 如有提示，请根据需要进行身份验证。
6. 在 **GitHub App name** 字段中，输入唯一名称。
7. 在 **Homepage URL** 字段中，输入占位符值，例如 https://www.placeholder.com。
8. 在 **Callback URL** 字段中，输入占位符值。您可以使用相同的占位符值，例如 https://www.placeholder.com。
9. 在 **Webhook URL** 字段中，输入占位符值。您可以使用相同的占位符值，例如 https://www.placeholder.com。另外，确保选中了 **Active** 复选框(GitHub 默认应执行此操作)。
10. 在本地系统上创建新文件，在其中保存安装过程中后续步骤所需的多个值。在此文件中输入值时，请确保标记它们，以便稍后可以记住每个值是什么。

    ```
    $ touch ~/install_values.txt
    ```

11. 在 CLI 中，生成 secret，然后标记 并将其保存在 **~/install_values.txt** 中。
    a. 如果您没有 OpenSSL，您可以按照 下载说明进行操作。

    ```
    $ openssl rand -hex 20 >> ~/install_values.txt
    ```

    重要

    务必保存这个命令的输出！

12. 在 GitHub 中，在 **Webook secret** 字段中输入最后一个命令的输出。
13. 在 **Repository permissions** 下设置以下权限：

a. **管理： 读取和写入**

b. **check: 读取和写入**

c. **内容：读取和写入**

d. **问题：读取和写入**

e. **元数据：只读** （应该已正确设置，但验证其值）

f. **Pull requests: 读取和写入**

14. 在 **Organization permissions** 下设置以下权限：

a. **Members: Read-only**

b. **Plan: Read-only**

15. 在 **Subscribe to events** 下，选择以下订阅：

a. **Check run**

b. **检查套件**

c. **Commit comment**

d. **Issue comment**

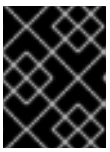
e. **Pull request**

f. **push**

16. 在什么地方 **可以安装此 GitHub 应用程序？** 选择 **任何帐户**。

17. 点 **Create GitHub App**。然后，您应该会看到 Developer Settings 页面。

18. 检索客户端 ID 和应用 ID。标签并将它们保存在 **~install_values.txt** 中。



重要

接下来的两个步骤解释了如何收集客户端机密和私钥。您必须保存客户端机密和私钥，并保持访问，才能完成 RHTAP 的安装过程！

19. 在新应用程序的页面中，在 **Client secrets** 旁边，选择 **Generate a new client secret**。在 **~/install_values.txt** 中标记并保存客户端 secret。

20. 在 GitHub 的同一页面上，在 **Private** key 下选择 **Generate a private key** 按钮。您的系统下载 **包含私钥的** 私钥文件。标签并将私钥文件的内容保存在 **~/install_values.txt** 中。私钥应以 **----- BEGIN RSA PRIVATE KEY-----** 开头，并以 **-----END RSA PRIVATE KEY-----** 结尾。

21. 仍然在 GitHub 中的同一页面上，从左侧的选项卡中选择 **Install App**。

22. 使用您的机构名称旁边的绿色 **Install** 按钮。

23. 出现提示时，选择 **All repositories**，因此 RHTAP 可以在您的机构中创建新存储库。单击绿色 **安装** 按钮。

其他资源

- 本文档中的步骤基于 Pipelines 作为代码文档来创建 GitHub 应用程序。

第 2 章 分叉 RHTAP 目录存储库

开发人员开始使用 RHTAP 实例后，您可能需要自定义您的实例，以更好地满足其需求。您可以自定义的 RHTAP 的一个方面是它所提供的软件模板集合。这些模板可帮助开发人员快速构建应用程序。

通过派生我们的目录存储库，其中包含默认软件模板集合，允许您为实例自定义模板。

先决条件

- GitHub 帐户

流程

1. 在 Web 浏览器中，导航到 RHTAP 软件目录存储库的 Releases 页面。
2. 选择与您正在使用的 RHTAP 版本对应的发行版本。
    a. 例如，如果您使用 RHTAP 的版本 1.0.0，您应该使用 此版本。
3. 在该发行版本的页面横幅下，选择 **Fork** 来分叉存储库。

注意

请务必从时间更新您的分叉，因此上游存储库的更新可能会使您的 RHTAP 实例受益。

第 3 章 创建 GITOPS GIT 令牌

在此过程中，您将在 **~/install_values.txt** 文件中创建两个需要的值来完成安装。

先决条件

- GitHub 帐户

流程

1. 在 Web 浏览器中，进入 GitHub 中的 Developer Settings 页面。
2. 在左侧面板中，在 **个人访问令牌** 下，选择 **Tokens （经典）**
3. 在页面横幅下的 **Generate new token** 下拉菜单中选择 **Generate new token (classic)**。您可能需要进行身份验证才能继续。
4. 输入名称，选择过期日期，在 **Select scopes** 中选择 **repo** （应该自动包括 **repo: status** 到 **security_events**）的所有范围。
5. 选择 **Generate token**。GitHub 将您重定向到一个新页面，其中您的令牌可见。确保将这个令牌标记为 **~/install_values.txt**。

第 4 章 创建 DOCKER 配置值

在此过程中，您将在 **~/install_values.txt** 文件中创建您需要的最终值。

先决条件

- Quay.io 帐户

流程

1. 在 Web 浏览器中，登录 Quay.io。在横幅的右侧，选择您的用户名并从下拉菜单中选择 **Account Settings**。
2. 在用户设置页面中，在 **Docker CLI Password** 下选择 **Generate Encrypted Password**。在弹出窗口中输入要进行身份验证的密码。
3. 接下来，仍然在弹出窗口中，选择 **Docker Configuration > View [username]-auth.json**。复制字符串，但不包括引号，具体为 **"auth":**。
4. 在 **~/install_values.txt** 文件中，使用您的用户名和身份验证令牌进行标记并创建 Docker 配置值（在合适的位置）：{"quay.io/[username]": {"auth": "[auth token]","email": ""}}}

注意

如果您计划在 Quay 上使用私有存储库来托管 RHTAP 构建的镜像，还必须将该私有存储库注册到 Advanced Cluster Security，如 本文档 所述。

第 5 章 创建 PRIVATE-VALUES.YAML 文件

RHTAP 依赖于 Helm 自动执行大部分安装过程。但是，Helm 需要特定信息才能正确安装 RHTAP。您必须在可在 install 命令中引用的文件中提供该信息。该文件名为 **private-values.yaml**。

此文件很复杂，可能容易将其错误准备。但是，这个过程解释了如何使准备 **private-values.yaml** 的过程变得更加简单。它指导您从 GitHub 克隆 RHTAP 安装程序存储库，并在该存储库中使用 shell 脚本，以更轻松地生成 **private-values.yaml**。

先决条件

- Git CLI 工具
- yq CLI 工具
- 包含所有必要值的 **~/install_values.txt** 文件。您在第一个、第三和第四个过程中创建此文件。
- Advanced Cluster Security 实例的 API 令牌和中央端点。

流程

1. 在 Web 浏览器中，导航到 GitHub 上的 RHTAP 安装程序存储库。
2. 选择绿色 &lt ;> **CODE** 按钮。在 Local 选项卡下，选择您首选的连接类型(HTTPS、SSH 或 GitHub CLI)进行克隆，并复制给定的命令。
3. 运行您复制的命令。例如，对于 SSH，在 CLI 中运行以下命令：

```
$ git clone git@github.com:redhat-appstudio/rhtap-installer.git
```

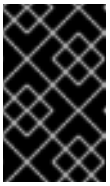
4. 在 CLI 中，导航到 RHTAP 安装程序存储库的本地克隆。

```
$ cd rhtap-installer
```

5. 运行 **bin/make.sh** 脚本。

```
$ bin/make.sh values
```

6. 该脚本提示您为每个以下字段输入值。按照说明确定您应该输入的值。如果您需要随时停止脚本，您可以这样做，只需重新运行 **bin/make.sh values** 命令来恢复进度：
   a. **RHTAP_ENABLE_GITHUB** ：如果要使用 GitHub 作为应用程序的 git 存储库，请输入 **y**
   b. **RHTAP_ENABLE_GITLAB** ：如果要将 GitLab 用作应用程序的 git 存储库，请输入 **y**



重要

在发布时，我们的文档不说明如何将 GitLab 配置为 RHTAP 的 git 主机，但可能这样做。我们正努力记录该流程。同时，如果您想要使用 GitLab，请参考此流程末尾提供的文档。

   c. **RHTAP_ENABLE_DEVELOPER_HUB**: 输入 **y**

d. **RHTAP_ENABLE_TAS** ：如果要使用红帽受信任的工件签名程序来增强软件供应链的安全性，请输入 **y**。

e. **RHTAP_ENABLE_TAS_FULCIO_OIDC_DEFAULT_VALUES :** 如果将前面的值设为 y，则输入 **y**。

f. **RHTAP_ENABLE_TPA** ：如果要使用红帽受信任的配置文件分析器来增强软件供应链的安全性，请输入 **y**。

g. **ACS__API_TOKEN** ：为您的 ACS 实例输入 API 令牌。您可以按照 此处 创建 API 令牌的先决条件的说明进行操作。

h. **ACS_CENTRAL_ENDPOINT** ：输入 ACS 实例的端点。您可以按照以下说明 配置 端点。

i. **DEVELOPER_HUB__CATALOG_URL**: 在您在上一步中创建的分叉中输入 **all.yaml** 文件的地址。

> https://github.com/[username]/tssc-sample-templates/blob/main/all.yaml

j. **GITHUB__APP__ID** ：这个值应该位于 **~/install_values.txt** 文件中。您在第一个安装过程中保存它。

k. **GITHUB__APP__CLIENT__ID** ：此值应位于 **~/install_values.txt** 文件中。您在第一个安装过程中保存它。

l. **GITHUB__APP__CLIENT__SECRET** ：此值应位于 **~/install_values.txt** 文件中。您在第一次过程中创建并保存它。

m. **GITHUB__APP__PRIVATE_KEY** ：这个值应该位于 **~/install_values.txt** 文件中。您在第一次过程中创建并保存它。

n. **GITHUB__APP__WEBHOOK_SECRET** ：这个值应该位于 **~/install_values.txt** 文件中。您在第一次过程中创建并保存它。

o. **GITOPS__GIT_TOKEN** ：这个值应该位于 **~/install_values.txt** 文件中。您创建并保存在第二个过程中。

p. **QUAY__API_TOKEN** ：如果您的镜像存储库是公共的，请使用 'null'。否则，使用读取访问权限 创建 API 令牌 并粘贴其值。

q. **QUAY__DOCKERCONFIGJSON** ：这个值应该位于 **~/install_values.txt** 文件中。您会在最后一个过程中创建并保存它。

r. **TAS__SECURESIGN__FULCIO_ORG_EMAIL**: 输入您拥有此新版 RHTAP 的人员或团队的电子邮件。

s. **TAS__SECURESIGN__FULCIO__ORG_NAME** ：输入 GitHub 组织的名称。

注意

剩余的值是您必须生成的密码和 secret。您不必在其他位置保存这些值，因为您当前使用的脚本正在创建存储您输入的所有值的文件。

t. **TPA__GUAC__PASSWORD** ：输入一个强大的密码，您和团队成员可用于验证 TPA 的 GUAC。您可以使用之前用来创建 Webhook secret 的同一 OpenSSL 命令。

```
$ openssl rand -hex 20
```

u. **TPA__KEYCLOAK__ADMIN_PASSWORD** ：输入另一个强大的密码。

v. **TPA__MINIO__ROOT_PASSWORD** ：输入另一个强大的密码。

w. **TPA__OIDC__TESTING_MANAGER_CLIENT_SECRET**: 输入另一个值，用作安全 secret。您还可以使用 OpenSSL 命令生成这个值。

x. **TPA__OIDC__TESTING_USER_CLIENT_SECRET**: 输入另一个值，用作安全 secret。

y. **TPA__OIDC__WALKER_CLIENT_SECRET** ：输入另一个值，用作安全 secret。

z. **TPA__POSTGRES__POSTGRES_PASSWORD** ：输入另一个强密码。

aa. **TPA__POSTGRES__TPA_PASSWORD** ：输入另一个强大的密码。

7. （可选）在运行 **bin/make.sh** 后，您可以更改哪个命名空间 RHTAP 用于部署应用程序。在运行 **bin/make.sh** 的同一上下文中，查找新生成的 **private-values.yaml** 文件。打开该文件，并在命名空间下编辑或添加 **其他命名空间：**，其中当前您应该看到 **rhtap-app**。

其他资源

如果要在该安装路径的文档完成前使用 GitLab 作为 git 主机，请参考以下文档：

- 将 GitLab 配置为 OAuth 2.0 身份验证身份提供程序
- 在 GitLab 中使用 Pipelines as Code

第 6 章 在集群中安装 RHTAP

创建 GitHub 应用程序和私有 values.yaml 文件后，就可以安装 RHTAP。实际安装过程非常简单。

先决条件

- ClusterAdmin 通过 CLI 访问 OCP 集群
- **private-values.yaml** 文件（在前面的过程中生成）

步骤

1. 在 CLI 中，以 ClusterAdmin 身份访问您的 OCP 集群。

```
$  oc login [cluster address] -u kubeadmin -p [password]
```

2. 将安装程序 Helm 仓库添加到本地系统中。

```
$ helm repo add openshift-helm-charts https://charts.openshift.io/
```

   a. 如果您之前添加了此 Helm 仓库，请更新 Helm 仓库。

```
$ helm repo update
```

3. 在运行 **bin/make.sh** 的目录中，运行 install 命令。安装可能需要十分钟或更长时间来完成。

```
$ helm upgrade installer openshift-helm-charts/redhat-trusted-application-pipeline --install --create-namespace --namespace rhtap --timeout 20m --values private-values.yaml
```

注意

在命名空间 后，您可以指定任何需要的命名空间；您不必使用 **rhtap**。

4. 安装完成后，安装程序会提供输出。标签并保存安装程序在 **~/install_values.txt** 中生成的 PipelineRun 文件。
5. 将整个第一个 PipelineRun 从 **cat << EOF | kubectl create** 复制并粘贴到您的命令行中。按 enter 键执行此 PipelineRun。
6. PipelineRun 完成后，它提供包括链接的输出。如果您还没有登录 OpenShift Web 控制台，请先登录。然后，打开 PipelineRun 输出中的链接。

故障排除

- 如果 Helm 无法出于某种原因成功安装 RHTAP，请尝试删除集群中的 **rhatp** 项目，然后再次运行 install 命令。

第 7 章 最终调整 GITHUB 应用程序

安装 RHTAP 后，您必须将之前在 GitHub 应用中输入的占位符值替换为特定于集群的值。这允许安装 GitHub 应用程序向 Red Hat Developer Hub 进行身份验证的用户，并使用 Red Hat Trusted Application Pipeline。

先决条件

- ClusterAdmin 通过 Web 控制台访问 OpenShift 集群

流程

1. 通过打开上一次步骤末尾生成的链接，您应该使用 **Administrator** 视图在 OpenShift 控制台中。如果没有，请导航到集群的 OpenShift Console 中的查看。
   a. 使用左侧导航条进入 **Pipelines > Pipelines**。
   b. 在 **Project** 字段中，在页面横幅下方，选择 **rhtap**。
   c. 选择 **PipelineRun** 选项卡。选择 PipelineRun，其名称以 **rhtap-pe-info-** 开头。
   d. 导航到 **Logs** 选项卡。
2. 在单独的浏览器选项卡中，返回到 GitHub Apps 页面(**Settings > Developer settings > GitHub Apps**)。
3. 在新的自定义应用程序旁边，点 **Edit**。
4. 将以下字段的占位符值替换为您在 OpenShift 控制台中执行的 PipelineRun 日志中找到的新值：
   a. **主页 URL**
   b. **回调 URL**
   c. **Webhook URL**
5. 滚动到页面底部，然后单击 **Save**。
6. 在单独的选项卡中，导航到您作为 GitHub 应用程序的新主页 URL 输入的地址。
7. 单击 **SIGN IN** 按钮，选择 GitHub 作为登录方法。
8. 在弹出窗口中，根据需要授权您的自定义 GitHub 应用程序。

您应该立即重定向到 Red Hat Developer Hub (RHDH)。下载 GitHub 应用程序的任何开发人员也可以使用该应用程序进行身份验证，并通过运行第三个过程中生成的第二个 PipelineRun。在 RHDH 中，开发人员可以利用红帽受信任的应用程序管道的自动化、可自定义和安全的 CI/CD 功能。

更新于 2024-07-02