



builds for Red Hat OpenShift 1.0

关于构建

构建简介

构建简介

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

本文档概述构建功能。它还包括发行注记以及如何获得支持的详细信息。

目录

第 1 章 发行注记	3
1.1. 兼容性和支持列表	3
1.2. 使开源包含更多	3
1.3. 构建正式发行 1.0 发行注记	4
第 2 章 构建概述	5
2.1. 构建资源	5
2.2. BUILDSTRATEGY 和 CLUSTERBUILDSTRATEGY 资源	5
2.3. BUILDRUN 资源	6
2.4. 构建控制器	6
2.5. 其他资源	7
第 3 章 构建策略	8
3.1. BUILDAH	8
3.2. SOURCE-TO-IMAGE	9
3.3. 其他资源	9

第 1 章 发行注记

发行注记包含有关新的和已弃用的功能、破坏更改以及已知问题的信息。以下发行注记适用于 OpenShift Container Platform 的最新构建版本。

构建是基于 [Shipwright 项目](#) 的可扩展构建框架，可用于在 OpenShift Container Platform 集群上构建容器镜像。您可以使用镜像构建工具（如 Source-to-Image (S2I) 和 Buildah）从源代码和 Dockerfile 构建容器镜像。您可以创建并应用构建资源，查看构建运行的日志，并管理 OpenShift Container Platform 命名空间中的构建。

构建包括以下功能：

- 用于从源代码和 Dockerfile 构建容器镜像的标准 Kubernetes 原生 API
- 支持 Source-to-Image (S2I) 和 Buildah 构建策略
- 您自己的自定义构建策略的可扩展性
- 从本地目录中的源代码执行构建
- 用于创建和查看日志的 Shipwright CLI，以及管理集群上的构建
- 使用 OpenShift Container Platform Web 控制台的 **Developer** 视角集成用户体验

有关构建的更多信息，请参阅 [构建概述](#)。

1.1. 兼容性和支持列表

在表中，组件被标记为以下状态：

TP	技术预览
GA	公开发布

[技术预览功能](#) 是实验性功能，不适用于生产环境。

表 1.1. 兼容性和支持列表

构建版本	组件版本	兼容 Openshift Pipelines 版本	OpenShift 版本
Operator	Builds (Shipwright)	CLI	
1.0	0.12.0 (GA)	0.12.0 (GA)	1.12 及更新的版本 4.14

1.2. 使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中存在问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

1.3. 构建正式发行 1.0 发行注记

OpenShift Container Platform 4.14 现在提供了构建正式发行(GA) 1.0。

1.3.1. 新功能

以下小节重点介绍了 Build 1.0 中的新内容。

1.3.1.1. Builds

- 此发行版本支持 **buildah** 和 **source-to-image** 构建策略。Red Hat OpenShift Operator 的构建会自动安装这些策略以供使用。



注意

buildpacks 构建策略目前处于 [开发者预览状态](#)。红帽计划在以后的发行版本中使用此策略正式发布。

- 在这个版本中，您可以在 **BuildStrategy** 资源中定义卷挂载。在定义策略时，您可以声明可在构建步骤和构建运行间共享的卷。您还可以为卷声明挂载点，该卷可在所有构建步骤中修复。卷可以具有固定卷源，也可以是您可以在 **Build** 或 **BuildRun** 资源中设置的可覆盖的卷源。
- 在这个版本中，您可以使用 **spec.build.spec** 字段将完整的构建规格嵌入到 **BuildRun** 资源中。通过嵌入规格，您可以在不创建和维护专用 **构建** 资源的情况下构建镜像。
- 此发行版本支持自动清理在特定时间或达到特定数量构建运行时运行的已完成构建。通过使用保留参数，您可以指定已完成的构建运行可以存在的持续时间，以及可以存在的成功或失败构建数量。
- 在这个版本中，您可以通过在 **Build** 资源中定义源、构建策略、参数值、构建程序或 docker 文件、输出、保留参数和卷来配置构建。
- 在这个版本中，您可以通过在 **BuildStrategy** 或 **ClusterBuildStrategy** 资源中定义策略参数、系统参数、步骤资源定义、注解和卷来配置构建策略。
- 在这个版本中，您可以通过定义构建引用、构建规格、参数值、服务帐户、输出、保留参数和卷来配置构建运行。
- 在这个版本中，您可以使用构建控制器指标监控构建资源。
- 在这个版本中，您可以在构建 secret 中添加注解 **build.shipwright.io/referenced.secret: "true"**。基于此注解，构建控制器在事件（如为构建 secret 创建、更新或删除触发器）时会采取协调操作。

1.3.1.2. CLI

- 在这个版本中，您可以创建、删除、列出或运行 **Build** 资源。
- 在这个版本中，您可以创建或列出 **BuildRun** 资源，或查看其日志。

第 2 章 构建概述

构建是基于 [Shipwright 项目](#) 的可扩展构建框架，可用于在 OpenShift Container Platform 集群上构建容器镜像。您可以使用镜像构建工具（如 Source-to-Image (S2I) 和 Buildah）从源代码和 Dockerfile 构建容器镜像。您可以创建并应用构建资源，查看构建运行的日志，并管理 OpenShift Container Platform 命名空间中的构建。

构建包括以下功能：

- 用于从源代码和 Dockerfile 构建容器镜像的标准 Kubernetes 原生 API
- 支持 Source-to-Image (S2I) 和 Buildah 构建策略
- 您自己的自定义构建策略的可扩展性
- 从本地目录中的源代码执行构建
- 用于创建和查看日志的 Shipwright CLI，以及管理集群上的构建
- 使用 OpenShift Container Platform Web 控制台的 **Developer** 视角集成用户体验

构建由以下自定义资源(CR)组成：

- **Build**
- **BuildStrategy** 和 **ClusterBuildStrategy**
- **BuildRun**

2.1. 构建资源

Build 资源定义应用程序的源代码以及要推送应用程序镜像的位置。以下示例显示了一个简单的构建，它由 Git 源、构建策略和输出镜像组成：

```
apiVersion: shipwright.io/v1beta1
kind: Build
metadata:
  name: buildah-golang-build
spec:
  source:
    git:
      url: https://github.com/username/taxi
  strategy:
    name: buildah
    kind: ClusterBuildStrategy
  output:
    image: registry.mycompany.com/my-org/taxi-app:latest
```

您还可以扩展 **Build** 资源，以将镜像推送到私有 registry 或使用 Dockerfile。

2.2. BUILDSTRATEGY 和 CLUSTERBUILDSTRATEGY 资源

BuildStrategy 和 **ClusterBuildStrategy** 资源定义一系列步骤来编译应用程序。您可以使用命名空间中的 **BuildStrategy** 资源和集群中的 **ClusterBuildStrategy** 资源。

BuildStrategy 或 **ClusterBuildStrategy** 资源的规格由 **steps** 对象组成。以下示例显示了 **buildah** 集群构建策略的规格：

```
apiVersion: shipwright.io/v1beta1
kind: ClusterBuildStrategy
metadata:
  name: buildah
spec:
  steps:
    - name: build-and-push
      image: quay.io/containers/buildah:v1.31.0
      workingDir: $(params.shp-source-root)
      command:
        - /bin/bash
        # ...
    # ...
```

2.3. BUILD RUN 资源

BuildRun 资源在集群中调用构建，类似于任何集群作业或 Tekton 任务运行。**BuildRun** 资源代表集群中的工作负载，这会导致运行 pod。**BuildRun** 是构建的运行实例。它使用集群上的特定参数实例化构建来执行。

BuildRun 资源可帮助您定义以下元素：

- 用于监控构建状态的唯一 **BuildRun** 名称
- 构建期间要使用的引用的 **Build** 实例
- 托管构建的所有 secret 的服务帐户

每个 **BuildRun** 资源都位于一个命名空间中。

2.4. 构建控制器

构建控制器监控 **Build** 资源中的任何更新，并执行以下任务：

- 验证 **Build** 资源中是否存在引用的 **Strategy** 对象。
- 验证 **Build** CR 中指定的参数是否存在于引用的构建策略中。它还验证参数名称与任何保留名称冲突。
- 验证容器 registry 输出 secret 是否存在于 **Build** 资源中。
- 验证 **Build** 资源中是否存在引用的 **spec.source.git.url** 端点 URL。

构建运行控制器监控 **Build** 或 **TaskRun** 资源中的任何更新，并执行以下任务：

- 搜索任何现有 **TaskRun** 资源并更新其父 **BuildRun** 资源状态。
- 检索指定的服务帐户，并将它与 **Build** 资源中的输出 secret 一起设置。
- 如果 **TaskRun** 资源不存在，控制器会生成新的 Tekton **TaskRun** 资源，并设置对 **TaskRun** 资源的引用。

- 对于 **TaskRun** 资源中的任何后续更新，控制器会更新父 **BuildRun** 资源。

2.4.1. 构建验证

为了避免因为不正确的或缺少依赖项或配置设置而触发 **BuildRun** 资源，构建控制器会提前验证它们。如果所有验证都成功，您可以查看名为 **Succeeded** 的 **status.reason** 字段。但是，如果有任何验证失败，您必须检查 **status.reason** 和 **status.message** 字段以了解根本原因。

表 2.1. 通过构建控制器验证构建

status.reason field	描述
BuildStrategyNotFound	命名空间级别上引用的策略不存在。
ClusterBuildStrategyNotFound	集群级别上引用的策略不存在。
SetOwnerReferenceFailed	在 Build 和 BuildRun 资源之间设置所有者引用会失败。当您在构建中将 spec.retention.atBuildDeletion 字段设置为 true 时，会触发此状态。
SpecSourceSecretRefNotFound	用于向 Git 进行身份验证的 secret 不存在。
SpecOutputSecretRefNotFound	用于向容器 registry 进行身份验证的 secret 不存在。
SpecBuilderSecretRefNotFound	用于向容器 registry 进行身份验证的 secret 不存在。
MultipleSecretRefNotFound	缺少用于身份验证的多个 secret。
RestrictedParametersInUse	一个或多个定义的 params 与任何保留参数共存。
UndefinedParameter	这些参数没有在引用的策略中定义。您必须在策略的 spec.parameters 规格中定义这些参数。
RemoteRepositoryUnreachable	未找到定义的 spec.source.git.url 规格。此验证只用于 HTTP 和 HTTPS 协议。
BuildNameInvalid	metadata.name 字段中的构建名称无效。您必须将有效的标签值用于构建名称。
SpecEnvNameCanNotBeBlank	表示用户提供的环境变量的名称为空。
SpecEnvValueCanNotBeBlank	表示用户提供的环境变量的值为空。

2.5. 其他资源

- [使用控制台安装构建](#)
- [使用 CLI 安装构建](#)

第 3 章 构建策略

您可以在 OpenShift Container Platform 集群上使用一组策展的构建策略或集群构建策略。Red Hat OpenShift Operator 的构建会自动安装这些策略以供使用。这种自动化策略安装可帮助您快速开始使用构建。

构建支持以下集群构建策略：

- **Buildah**：在所有平台上都被支持
- **Source-to-image**：在 linux/amd64 平台上支持



注意

buildpacks 构建策略目前处于 [开发者预览状态](#)。如需更多信息，请参阅 [buildpacks 示例](#)。

3.1. BUILDDAH

buildah 集群构建策略使用 Dockerfile 来构建容器镜像并将其推送到目标 registry。您必须在 **Build CR** 的 **spec.paramValues** 字段中指定 Dockerfile。

您可以在集群中的不同命名空间间共享 **buildah** 策略，因为 Red Hat OpenShift Operator 的构建会在集群级别安装 **buildah** 策略。

您可以为 **buildah** 策略配置以下参数：

表 3.1. buildah 的配置参数

Name	类型	描述	Default (默认)
build-args	数组	构建期间使用的 Dockerfile 所需的参数的键值对	[]
registries-block	数组	必须被阻断的 registry 列表	[]
registries-insecure	数组	带有其完全限定域名 (FQDN) 的不安全 registry 列表	[]
registries-search	数组	用于搜索短名称镜像的 registry 列表	["registry.redhat.io", "quay.io"]
dockerfile	string	构建期间使用的 Dockerfile 的路径	"Dockerfile"
storage-driver	string	buildah 使用的存储驱动程序，如 overlay 或 vfs	"vfs"



注意

如需更多信息，请参阅附加资源部分中的 [配置构建策略](#)。

3.2. SOURCE-TO-IMAGE

此构建策略由 **source-to-image** 和 **buildah** 组成。您可以使用此策略来生成容器文件，并准备应用以使用构建器镜像构建。您必须在 **Build CR** 的 **spec.paramValues** 字段中指定构建器镜像。

您可以在集群中的不同命名空间间共享 **source-to-image** 策略，因为 Red Hat OpenShift Operator 的构建会在集群级别安装 **Source-to-image** 策略。

您可以为 **source-to-image** 策略配置以下参数：

表 3.2. **source-to-image** 的配置参数

Name	类型	描述	Default (默认)
registries-block	数组	必须被阻断的 registry 列表	[]
registries-insecure	数组	带有 FQDN 的不安全 registry 列表	[]
registries-search	数组	用于搜索短名称镜像的 registry 列表	["registry.redhat.io", "quay.io"]
builder-image	string	构建期间使用的构建器镜像的位置	不适用
storage-driver	string	Source-to-image 使用的存储驱动程序，如 overlay 或 vfs	"vfs"

3.3. 其他资源

- [使用 Web 控制台安装构建](#)
- [使用 CLI 安装构建](#)
- [配置构建策略](#)
- [使用 Web 控制台创建 ShipwrightBuild 资源](#)