



OpenShift Container Platform 4.1

发行注记

OpenShift Container Platform 4.1 发行版本的重点介绍及变化

OpenShift Container Platform 4.1 发行注记

OpenShift Container Platform 4.1 发行版本的重点介绍及变化

法律通告

Copyright © 2019 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

此发行注记介绍了 OpenShift Container Platform 4.1 的新功能、功能增强、重要的技术变化、以及对以前版本中的错误作出的主要修正。另外，还包括在此版本正式发行（GA）时存在的已知问题的信息。

目录

第 1 章 OPENSIFT CONTAINER PLATFORM 4.1 发行注记	3
1.1. 关于此版本	3
1.1.1. 致谢	3
1.1.2. 已过时的功能	3
1.2. 新功能及功能增强	5
1.2.1. Operators	5
1.2.1.1. Operator Lifecycle Manager (OLM)	5
1.2.2. 安装和升级	5
1.2.2.1. OperatorHub	6
1.2.3. 存储	6
1.2.4. 扩展	6
1.2.4.1. 集群的限制	6
1.2.4.2. Node Tuning Operator	7
1.2.5. Cluster Monitoring	7
1.2.5.1. 基于定制的 metrics API 对 pod 进行横向的自动扩展 (技术预览)	7
1.2.5.2. 新的提示用户界面	7
1.2.5.3. Telemeter	7
1.2.5.4. 基于资源 metrics API 对 pod 进行横向的自动扩展	8
1.2.6. 开发者体验	8
1.2.6.1. 代码就绪容器	8
1.2.6.2. 全面支持 Multi-stage Dockerfile 镜像构建功能	8
1.2.7. Registry	8
1.2.7.1. registry 现在由一个 Operator 管理	8
1.2.8. 网络	8
1.2.8.1. Cluster Network Operator (CNO)	8
1.2.8.2. OpenShift SDN	8
1.2.8.3. Multus	8
1.2.9. Web 控制台	8
1.2.9.1. 开发者目录	8
1.2.9.2. 新的管理界面	8
1.2.10. 安全性	9
1.3. 主要的技术变化	9
由 buildah 进行镜像构建	9
SecurityContextConstraints	9
服务 CA bundle 的变化	9
OpenShift Service Broker 和 service Catalog 已过时	9
Service Catalog 不再被默认安装	9
Template Service Broker 不会被默认安装	9
OpenShift Ansible Service Broker 不再会被默认安装	9
多个 oc adm 命令现已过时	9
imagepolicyadmission 的可配置性已不存在	10
1.4. 技术预览功能	10
1.5. 已知问题	13
1.6. 异步勘误更新	14
1.6.1. RHBA-2019:0758 - OpenShift Container Platform 4.1 镜像发行公告	14
1.6.1.1. 镜像	15
第 2 章 OPENSIFT CONTAINER PLATFORM 版本政策	18

第 1 章 OPENSIFT CONTAINER PLATFORM 4.1 发行注记

Red Hat OpenShift Container Platform 为软件开发人员和 IT 机构提供了一个混合云应用平台。使用这个平台可以在配置和管理成本最小化的情况下，利用安全、可扩展的资源部署新的或已有的应用程序。OpenShift Container Platform 支持大量编程语言和开发平台，如 Java、JavaScript、Python、Ruby 和 PHP。

OpenShift Container Platform 基于 Red Hat Enterprise Linux 和 Kubernetes，为当今的企业级应用程序提供了一个更加安全、可扩展的多租户操作系统，同时提供了集成的应用程序运行时及程序库。OpenShift Container Platform 可以满足用户对安全性、隐私、合规性及监管的要求。

1.1. 关于此版本

Red Hat OpenShift Container Platform (RHBA-2019:0758) 现已正式发行。这个版本使用 Kubernetes 1.13。OpenShift Container Platform 4.1 的新功能、改变以及已知的问题包括在此文档中。

红帽没有公开发布 OpenShift Container Platform 4.0，在版本 3.11 后的第一个正式发行版本即为 OpenShift Container Platform 4.1。

OpenShift Container Platform 4.1 集群包括在 <https://cloud.openshift.com>。您可以通过 Unified Hybrid Cloud (UHC) application for OpenShift Container Platform 在内部环境或云环境中部署 OpenShift 集群。

OpenShift Container Platform 4.1 需要运行在 Red Hat Enterprise Linux 7.6 及更新的版本，或 Red Hat Enterprise Linux CoreOS 4.1 上。

您必须使用 Red Hat Enterprise Linux CoreOS (RHCOS) 作为 control plane (或 master) 的系统，而 compute (或 worker) 系统可以使用 RHCOS 或 Red Hat Enterprise Linux 7.6。



重要

因为当前只支持使用 Red Hat Enterprise Linux 7.6 作为 compute 系统，所以不能把使用 Red Hat Enterprise Linux 的 compute 系统升级到版本 8。

您可以使用安装程序置备的基础架构 (installer-provisioned infrastructure) 在 Amazon Web Services (AWS) 上安装 OpenShift Container Platform 4.1，或使用用户提供的基础架构 (user-provided infrastructure) 在 AWS、裸机或 VMware vSphere 主机上安装 OpenShift Container Platform 4.1。如果使用安装程序置备的基础架构安装，集群将会为您置备和管理所有的集群基础架构。

OpenShift Container Platform 需要包括用来运行安装进程在内的所有机器都可以直接连接到互联网以用来为平台容器获取镜像 (pull image)，并向红帽提供 telemetry 数据。这些操作不能通过代理服务器进行。

1.1.1. 致谢

红帽全球支持服务团队借此感谢 Rushil Sharma、JooHo Lee 和 Suresh Gaikwad 在评估和测试 OpenShift Container Platform 4.1 过程中所做出的杰出贡献。

1.1.2. 已过时的功能

因为 OpenShift Container Platform 4.1 的底层架构以及安装的过程有很大改变，所以 OpenShift Container Platform 3.x 的许多功能现在已过时。

表 1.1. 在版本 4.1 中已过时的功能

功能	原因
Hawkular	被 cluster monitoring (集群监控) 替代。
Cassandra	被 cluster monitoring (集群监控) 替代。
Heapster	由 Prometheus adapter 替代。
Atomic Host	由 Red Hat Enterprise Linux CoreOS 替代。
系统容器	由 Red Hat Enterprise Linux CoreOS 替代。
projectatomic/docker-1.13 additional search registries	CRI-O is the default container runtime for OpenShift Container Platform 4.x on RHCOS and Red Hat Enterprise Linux.
oc adm diagnostics	基于 Operator 的诊断工具。
oc adm registry	Replaced by the Image Registry Operator.
使用 Docker 进行自定义 (custom) 策略构建镜像	如果希望继续使用自定义构建, 则需要把 Docker 替换为 Podman 或 Buildah。虽然自定义构建策略没有从 OpenShift Container Platform 4.1 中删除, 但是它的功能有很大变化。
Cockpit	已经过改进的 OpenShift Container Platform 4.1 web 控制台。
Stand-alone registry installations	Quay is Red Hat's enterprise container image registry.
DNSmasq	CoreDNS is the default.
外部的 etcd 节点	etcd is always on the cluster in OpenShift Container Platform 4.1.
CloudForms OpenShift Provider 和 Podified CloudForms	被内置的管理工具所取代。
通过安装程序 (installer) 进行卷置备	被动态卷替代。如果需要 NFS, 则使用 NFS provisioner.
蓝绿安装方法	Ease of upgrade is a core value of OpenShift Container Platform 4.1.

功能	原因
OpenShift Service Broker 和 Service Catalog	Service Catalog 和 OpenShift service brokers 将会在以后发行的几个 OpenShift 4 版本中被逐渐替代。使用 Operator Framework 和 Operator Lifecycle Manager (OLM) 来继续向 OpenShift 4 集群提供应用程序。这些新功能为完整管理应用程序的生命周期提供了多个好处。
oc adm ca	Certificates are managed by Operators internally.
oc adm create-api-client-config	Functions are managed by Operators internally.
oc adm create-bootstrap-policy-file	
Web 控制台	OpenShift Container Platform 3.11 的 web 控制台已被 OpenShift Container Platform 4.1 提供的新的 web 控制台所替代。

1.2. 新功能及功能增强

此版本对以下方面进行了改进

1.2.1. Operators

Operators 是一组软件，可以用来减低运行其他软件的复杂程度。它可以被看作是软件厂商的工程团队的扩展，可以在 Kubernetes 环境中（如 OpenShift Container Platform）监控软件的运行情况，并根据软件的当前状态实时做出决定。Advanced Operators 被设计为用来无缝地处理升级过程，并对出现的错误自动进行响应，而且不会采取“捷径”（如跳过软件备份过程来节省时间）。

1.2.1.1. Operator Lifecycle Manager (OLM)

OpenShift Container Platform 4.1 当前还没有完全支持这个功能。

OLM 可以帮助集群管理员进行安装、升级并对在集群中运行的 Operators 设置访问权限：

- 包括一个经过策划的 Operators 目录，可以把其他 Operators 加载到集群中
- 对所有 Operator 进行滚动更新来升级到新版本
- 支持使用基于角色的访问控制（RBAC）机制来控制特定的团队使用特定的 Operator

请参阅 [Understanding the Operator Lifecycle Manager \(OLM\)](#) 来获得更详细的信息。

1.2.2. 安装和升级

Red Hat OpenShift Container Platform 4.1 带有一个安装程序置备的基础架构（installer-provisioned infrastructure），这样安装程序就可以控制安装过程的所有方面。另外，针对 AWS 实例，安装程序置备的基础架构还会提供一个基于最佳实践方案的 OpenShift Container Platform 4.1 部署。它会提供一个精简的默认安装，用户可以通过 OperatorHub 在以后逐渐添加新的功能。

另外，在 AWS、裸机或 vSphere 主机上，还可以安装用户提供的基础架构（user-provided infrastructure）。如果使用安装程序置备的基础架构，集群将会为您置备并管理所有的集群基础架构。

当前，还不支持从 3.x 直接升级到 4.1。用户需要进行一个全新的 OpenShift Container Platform 4.1 安装。

简单的、OTA 模式的 OpenShift Container Platform 4.x z-stream 发行版本的升级过程。集群管理员可以通过 web 控制台中的 **Cluster Settings** 标签页进行升级。详细信息请参阅 [Updating a cluster](#)。

1.2.2.1. OperatorHub

OperatorHub 可以帮助系统管理员方便地发现并安装所有可选的组件和应用程序。这包括红帽自己的产品，以及红帽的合作伙伴及社区的产品。

表 1.2. 基本安装和 OperatorHub 提供的功能

功能	新的安装程序 (installer)	OperatorHub
控制台和身份验证	* [x]	-
Prometheus cluster monitoring	* [x]	-
OTA (Over-the-air) 升级	* [x]	-
机器管理	* [x]	-
可选的服务代理	-	* [x]
可选的 OpenShift Container Platform 组件	-	* [x]
Red Hat product Operators	-	* [x]
Red Hat partner Operators	-	* [x]
Community Operators	-	* [x]

请参阅 [Understanding the OperatorHub](#) 来获得更详细的信息。

1.2.3. 存储

OpenShift Container Platform 4.1 提供了和 OpenShift Container Platform 3.11 相同的存储支持。另外，它还以“技术预览”的形式支持以下存储：EFS (通过 Amazon 处理的 CSI Driver)、Manila provisioner/operator 以及 Snapshot。

1.2.4. 扩展

1.2.4.1. 集群的限制

[Cluster Limits](#) 包括了与 OpenShift Container Platform 4.1 相关的内容。

使用 [OpenShift Container Platform Limit Calculator](#) 可以估算出您的环境的集群限制。

1.2.4.2. Node Tuning Operator

从版本 4.1 开始，[Node Tuning Operator](#) 会作为标准 OpenShift Container Platform 安装的一部分。

Node Tuning Operator 可以帮助您通过 tuned 守护进程来管理节点一级的性能优化。大多数高性能应用程序都需要一定程度的内核级性能优化。Node Tuning Operator 为用户提供了一个统一的、节点一级的 sysctls 管理接口，并可以根据具体用户的需要灵活地添加定制的性能优化设置（当前还是一个技术预览功能）。Node Tuning Operator 把为 OpenShift Container Platform 容器化的 tuned 守护进程作为一个 Kubernetes DaemonSet 进行管理。它保证了定制的优化设置以可以被守护进程支持的格式传递到在集群中运行的所有容器化的 tuned 守护进程中。相应的守护进程会在集群的所有节点上运行，每个节点上运行一个。

1.2.5. Cluster Monitoring

1.2.5.1. 基于定制的 metrics API 对 pod 进行横向的自动扩展 (技术预览)

此功能（目前为技术预览）允许您根据自定义的 metrics API 对 pod 进行横向的自动扩展（horizontal pod autoscaling，简称 HPA）。作为这种技术预览的一部分，现在可以部署一个 Prometheus Adapter 组件来为定制的 metrics API 提供应用程序的指标数据（metrics）。

限制：

- 该适配器只连接到一个 Prometheus 实例（或一组使用 Kubernetes 服务的，实现负载均衡功能的一组实例）。
- 手动部署适配器并把它配置为使用 Prometheus。
- Prometheus Adapter 配置的语法规则可能会在以后有所变化。
- 如果 OpenShift Container Platform 包括了一个现成的自定义 metrics adapter，用来向自定义 metrics adapter 实例发送 Kubernetes API 聚合的 **APIService** 配置可能会在以后发行的版本中改变。

1.2.5.2. 新的提示用户界面

一个提示信息 UI 现已集成到 OpenShift Container Platform web 控制台中。现在，您可以从一个统一的地方查看集群一级的提示信息及提示规则。

1.2.5.3. Telemeter

Telemeter 可以以匿名的方式收集与集群相关的 metrics 数据来帮助用户预防 OpenShift Container Platform 集群的潜在问题。它包括：

- 收集 OpenShift Container Platform 环境关键的健康状况指标数据。
- 启用多个 OpenShift Container Platform 升级的可行反馈回路。
- 收集每个集群的节点数量及大小（CPU内核和RAM）。
- 收集 etcd 的大小。
- 收集在 OpenShift 集群中安装的 OpenShift 框架组件的健康状况和状态。

1.2.5.4. 基于资源 metrics API 对 pod 进行横向的自动扩展

默认情况下，OpenShift Cluster Monitoring 会通过 Kubernetes 资源 metrics API 提供 CPU 和内存利用率的数据。因此，不再需要装单独的 metrics 服务器。

1.2.6. 开发者体验

1.2.6.1. 代码就绪容器

oc cluster 命令、Minishift 以及 CDK 现在由一个本地的 OpenShift Container Platform 4.1 桌面实例所替代。OpenShift Container Platform 4.1 的设计宗旨是简化访问过程及提供原生的体验。它在 macOS 和 Microsoft Windows 系统上带有原生的安装程序，并提供原生的 hypervisor 支持，而且可以集成到桌面工具栏图标中。

1.2.6.2. 全面支持 Multi-stage Dockerfile 镜像构建功能

现在，所有 **Docker** 策略构建都支持 Multi-stage Dockerfile。

1.2.7. Registry

1.2.7.1. registry 现在由一个 Operator 管理

现在，registry 通过一个 Operator 进行管理，而不再需要使用 **oc adm registry**。

1.2.8. 网络

1.2.8.1. Cluster Network Operator (CNO)

集群网络现在由一个 Operator 进行配置和管理。Operator 对集群网络进行升级和监控。

1.2.8.2. OpenShift SDN

现在默认的格式是 **NetworkPolicy**。

1.2.8.3. Multus

Multus 是 Kubernetes Container Network Interface (CNI) 的元数据插件，用户可以使用它来在每个 pod 中创建多个网络接口。

1.2.9. Web 控制台

1.2.9.1. 开发者目录

OpenShift Container Platform 4.1 带有一个重新设计的开发者目录 (Developer Catalog)，它把所有新的 Operators 和已存在的代理服务 (broker services) 集成在一起，使用新的方法来发现、排序并理解如何使用这些服务。开发者目录是开发人员访问所有可用服务的一个起点。它把 Operators、Service Catalog、broker 和 Source-to-Image (S2I) 的功能集成在一起。

1.2.9.2. 新的管理界面

OpenShift Container Platform 4.1 所提供的新的管理界面支持自动化操作。例如管理机器的设置、taint、tolerations 以及集群设置。

1.2.10. 安全性

在 OpenShift Container Platform 4.1 中，可以使用一个 Operator 来安装、配置和管理证书签发服务器。证书作为 secret 进行管理，并以加密的形式保存在 etcd 中。

1.3. 主要的技术变化

OpenShift Container Platform 4.1 引入了以下显著的技术变化。

由 buildah 进行镜像构建

使用 Source 和 Docker 策略进行构建将通过 buildah 执行，而不再使用 docker 守护进程。

SecurityContextConstraints

SecurityContextConstraints 现在包括在 **security.openshift.io** 组中。

服务 CA bundle 的变化

pod 可以信任由集群生成的证书，这些证书只为内部的 DNS 名称签发。它使用一个 CA bundle 来自动注入到任何使用 **service.beta.openshift.io/inject-cabundle=true** 注解的 configMap 中。CA bundle 作为 **service-ca.crt** 下的使用 PEM 编码的数据存在。这个注解将会删除 configMap 中现存的内容。

当前使用来自于 **/var/run/secrets/kubernetes.io/serviceaccount/service-ca.crt** 下的 service-serving CA bundle 的 pod 需要被迁移为使用来自带有 **service.beta.openshift.io/inject-cabundle=true** 注解的 configMap 中的 CA bundle。

/var/run/secrets/kubernetes.io/serviceaccount/service-ca.crt 文件现已过时，并将从以后的发行版本中删除。

OpenShift Service Broker 和 service Catalog 已过时

Service Catalog 和 OpenShift service brokers 将会在以后发行的几个 OpenShift 4 版本中被逐渐替代。当重要的依赖内容被移植到新的使用 Operator 的解决方案后，红帽就将会不再使用 Template Service Broker 和 OpenShift Ansible Broker。我们建议用户关注 Operator Framework and Operator Lifecycle Manager (OLM) 来继续向 OpenShift 4 集群提供应用程序。这些新功能为完整管理应用程序的生命周期提供了多个好处。

Service Catalog 不再被默认安装

Service Catalog 在 OpenShift Container Platform 4.1 中不会被默认安装，如果需要使用 OpenShift Ansible broker 或 template service broker，则要手工安装它。在 OpenShift Container Platform 4.1 中，Service Catalog API 服务器会被安装到 **openshift-service-catalog-apiserver** 命名空间中，Service Catalog controller manager 会被安装到 **openshift-service-catalog-controller-manager** 命名空间中。在 OpenShift Container Platform 3.11 中，这两个组件都被安装到 **kube-service-catalog** 命名空间中。

Template Service Broker 不会被默认安装

在 OpenShift Container Platform 4.1 中，Template Service Broker 不会被默认安装。如果用户需要从 web 控制台访问模板应用程序，集群的系统管理员则需要手工安装 Template Service Broker。

OpenShift Ansible Service Broker 不再会被默认安装

在 OpenShift Container Platform 4.1 中，OpenShift Ansible Service Broker 不再会被默认安装。

多个 oc adm 命令现已过时

过时的 **oc adm** 命令包括：

- **oc adm create-master-certs** - 创建密钥和证书
- **oc adm create-key-pair** - 创建一个 RSA 密钥对。
- **oc adm create-server-cert** - 创建一个密钥和服务器证书。

- **oc adm create-signer-cert** - 创建一个自签发的 CA。

imagepolicyadmission 的可配置性已不存在

imagepolicyadmission 插件的可配置性在 OpenShift Container Platform 4.1 中已不存在。这个插件可以运行，但只能使用默认配置。如果需要对它进行配置，则使用其他不被支持的覆盖机制。

1.4. 技术预览功能

这个版本中的一些功能当前还处于技术预览状态。它们并不适用于在生产环境中使用。请参阅红帽门户网站中关于对技术预览功能支持范围的信息：

[技术预览功能支持范围](#)

在下表中，标记为 **TP** 的功能代表 *技术预览*，标记为 **GA** 的功能代表 *正式发布 (General Availability)*。

表 1.3. 技术预览

功能	OCP 3.11	OCP 4.1
Prometheus Cluster Monitoring	GA	GA
本地存储持久卷	TP	TP
CRI-O for runtime pods	GA* [a]	GA
Tenant Driven Snapshotting	TP	TP
oc CLI Plug-ins	TP	TP
Service Catalog	GA	GA
Template Service Broker	GA	GA
OpenShift Ansible Service Broker	GA	GA
Network Policy	GA	GA
Multus	-	GA
Service Catalog Initial Experience	GA	GA
New Add Project Flow	GA	GA
Search Catalog	GA	GA
Cron Jobs	GA	GA
Kubernetes Deployments	GA	GA

功能	OCP 3.11	OCP 4.1
StatefulSets	GA	GA
Explicit Quota	GA	GA
Mount Options	GA	GA
System Containers for Docker, CRI-O	-	-
Hawkular Agent	-	-
Pod PreSets	-	-
experimental-qos-reserved	TP	TP
Pod sysctls	GA	GA。请参阅 已知问题 以了解当前的限制。
中央审计	GA	GA
外部项目网络通讯的静态 IP	GA	GA
模板完整性检测	GA	GA
replicaSet	GA	GA
Fluentd Mux	TP	TP
集群的 MongoDB 模板	-	-
集群的 MySQL 模板	-	-
Kubernetes 资源的镜像流 (image stream)	GA	GA
设备管理器	GA	GA
重新调整持久性卷的大小	GA	GA
大内存页	GA	GA
CPU 固定 (CPU pinning)	GA	GA
Admission Webhooks	TP	GA

功能	OCP 3.11	OCP 4.1
AWS EFS 的外部置备系统	TP	TP
Pod Unidler	TP	TP
节点问题检测程序	TP	TP
为临时存储设置 Limit/Requests	TP	TP
Descheduler	TP	TP
CephFS	TP	TP
Podman	TP	TP
Kuryr CNI 插件	TP	TP
Istio	TP	TP
PID 命名空间的共享控制	TP	TP
Manila Provisioner	TP	TP
Cluster Administrator 控制台	GA	GA
Cluster Autoscaling (只适用于 AWS)	GA	GA
Container Storage Interface (CSI)	TP	TP
Operator Lifecycle Manager	TP	GA
Red Hat OpenShift Service Mesh	TP	TP
"Fully Automatic" Egress IPs	GA	GA
Pod Priority and Preemption	GA	GA
Dockerfiles 的 Multi-stage build	TP	GA
OVN		TP
基于 Prometheus 的 HPA 定制 metrics adapter		TP
机器健康状态检查		TP

功能	OCP 3.11	OCP 4.1
[a] 带有 * 标记的功能代表通过 z-stream 更新提供。		

1.5. 已知问题

- 不安全的 `sysctl` 不能在 OpenShift Container Platform 4.1 中使用。 ([BZ#1690754](#))
- 如果一个实例被从云供应商中删除（用户删除了它，或因为云供应商的某些事件删除了它），`machine-controller` 可能会发现这个实例已不存在并试图创建这个实例。这个行为没有包括在产品的文档中，用户在其工作流程中不应该依赖这个行为。这个行为可能会影响到当前或以后的组件，如 `node-health-checker`。 ([BZ#1712068](#))
- 使用 `shell` 替换功能来生成环境变量的构建可能会失败。 ([BZ#1712245](#))
- 当删除一个机器对象时（直接删除，或因为要缩减 `machine-set`），如果相关联的节点因为某些原因已被删除（可能被一个集群管理员删除），`machine-controller` 可能无法成功删除其支持的云实例，`machine-object` 将会停留在 `deleting` 状态。 ([BZ#1713061](#))
- 因为提供给客户端的证书为空，所以在 **JBoss EAP** 镜像中查询 **Jolokia** 会出错。**Jolokia SSL** 客户端用户验证将会失败，并需要提供用户名和密码（如果已启用这个功能）。 ([BZ#1708640](#))
- **TokenRequest** API 在 OpenShift Container Platform 4.1 不可用。请求一个 **ServiceAccountTokenVolumeProjection** 卷在 OpenShift Container Platform 4.1 中不被支持。如果使用了一个 **ServiceAccountTokenVolumeProjection**，`kubelet` 将会报错。。 ([BZ#1695196](#))
- 当部署了 3 个节点时，ElasticSearch CRD 实例中的 `es nodeCount` 将无法被扩展。扩展功能可以在部署了 1 个、2 个、4 个、5 个或 6 个 ElasticSearch CRD 实例时正常工作。 ([BZ#1712955](#))
- 从 OperatorHub 创建的 ElasticSearch 实例在部署时会会有一个 CPU 的限制，即使没有设置相应的限制。 ([BZ#1710657](#))
- 在一个 AWS 安装后，`openshiftClustID` tag 不存在。 ([BZ#1685089](#))
- 无法使用 `oc patch` 和 `oc edit` 命令升级 `scc(CRD)` 资源。其结果是，策略性合并也将失败。 ([BZ#1707679](#))
- OpenShift Container Platform 4.1 registry 服务使用端口 5000 而不是端口 443。 ([BZ#1701422](#))
- 如果所选的 Availability Zone 没有请求的资源，则在 AWS 环境中进行 `machineset scaling` 操作可能会失败。 ([BZ#1713157](#))
- 在通过自定义的 PKI 配置 `ingress wildcard` 证书后使用 **OAuth** 端点会导致登陆失败。。 ([BZ#1712525](#))
- Source-to-Image (S2I) 镜像构建在 OpenShift Container Platform 4.1 中可能会需要更长的时间完成。这是因为，OpenShift Container Platform 4.1 不会象以前的版本那样在构建镜像时利用一个共享的镜像缓存。 ([BZ#1685352](#))
- 因为内存的限制，`cloud-credential-operator` 可能会在带有大量项目的集群中崩溃。 ([BZ#1711402](#))

- 在一个集群升级后，Marketplace 无法检测到 **opsrc**。其结果是，**csc** 软件包为空，并无法下载 **packagemanifests**。Marketplace 会在同步后自动解决这个问题（大概需要1个小时的时间）。([BZ#1695550](#))
- 在 OpenShift Container Platform 4.1 中，**oc version** 输出中的 **oc** 和 **openshift-install** 版本的 **GitTreeState**: 可能是 “dirty”。([BZ#1715001](#))
- 在 AWS 环境中，如果一个 master 节点已停止，**kubeapiserver** 会因为一个 pod 处于 **Pending** 状态而无法被部署。([BZ#1713292](#))
- 使用 Broadwell CPU model 79 (type m4)，AWS 中的所有 m4 实例都无法验证 ([CVE-2019-1109](#))，因为 **microcode_ctl** 不会被更新。([BZ#1710981](#))
- 如果另外一个 ElasticSearch 部署处于正在删除的状态，则新的 ElasticSearch 可能无法被创建。([BZ#1711044](#))
- 在 OpenShift Container Platform 4.1 web 控制台中没有 *Open Java Console* 链接。([BZ#1713656](#))
- 在运行了几天后，**openshift-cluster-node-tuning-operator** 可能会产生大量的 secret。([BZ#1714484](#))
- Autoscaling for Memory Utilization 功能可能无法象预期的情况运行。为基于内存的自动扩展 (autoscaling) 创建 HPA 会在查找资源时失败。([BZ#1707785](#))
- 在成功进行更新后，**oc clusterversion** 可能会错误地报告更新没有被应用。([BZ#1711964](#))
- 在出现一个 FATAL 事件时，installer 可能会有一个 **0** 返回代码。([BZ#1712409](#))

1.6. 异步勘误更新

OpenShift Container Platform 4.1 的安全更新、程序错误修正、功能增强更新将会通过红帽网络以异步勘误的形式发布。所有的 OpenShift Container Platform 4.1 勘误都可以通过[红帽客户门户网站](#)获得。OpenShift Container Platform 生命周期包括了详细的与异步勘误相关的内容。

红帽客户门户网站的用户可以在红帽订阅管理 (RHSM) 帐户设置中启用勘误通知功能。当勘误通知被启用后，用户会在有与其注册的系统相关的勘误发行时接收到电子邮件通知。



注意

用户的红帽客户门户网站账户需要有注册的系统，以及使用 OpenShift Container Platform 的权限才可以接收到 OpenShift Container Platform 的勘误通知。

本节的内容将会持续更新，以提供以后发行的与 OpenShift Container Platform 4.1 相关的异步勘误信息。异步子版本（例如，OpenShift Container Platform 4.1.z）的具体信息会包括在相应的子章节中。此外，在发行公告中因为空间限制没有包括在其中的勘误内容也会包括在这里的相应的子章节中。



重要

对于任何 OpenShift Container Platform 发行版本，请仔细参阅 [updating your cluster](#) 中的内容。

1.6.1. RHBA-2019:0758 - OpenShift Container Platform 4.1 镜像发行公告

发布日期：2019年6月4日

OpenShift Container Platform release 4.1 现已正式发布。其软件包列表包括在 [RHBA-2019:1173](#) 公告中。此更新包括的容器镜像及程序错误修正由 [RHBA-2019:0758](#) 公告提供。

请参阅以下部分来了解这个发行版本的容器镜像的注记信息。

1.6.1.1. 镜像

此版本更新了 Red Hat Container Registry (registry.access.redhat.com)，包括以下镜像：

- openshift4/apb-base
- openshift4/apb-tools
- openshift4/mariadb-apb
- openshift4/mediawiki-apb
- openshift4/mediawiki
- openshift4/mysql-apb
- openshift4/ose-ansible-operator
- openshift4/ose-ansible-service-broker
- openshift4/ose-ansible-service-broker-operator
- openshift4/ose-aws-machine-controllers
- openshift4/ose-cli-artifacts
- openshift4/ose-cli
- openshift4/ose-cloud-credential-operator
- openshift4/ose-cluster-authentication-operator
- openshift4/ose-cluster-autoscaler
- openshift4/ose-cluster-autoscaler-operator
- openshift4/ose-cluster-bootstrap
- openshift4/ose-cluster-capacity
- openshift4/ose-cluster-config-operator
- openshift4/ose-cluster-dns-operator
- openshift4/ose-cluster-image-registry-operator
- openshift4/ose-cluster-ingress-operator
- openshift4/ose-cluster-kube-apiserver-operator
- openshift4/ose-cluster-kube-controller-manager-operator
- openshift4/ose-cluster-kube-scheduler-operator
- openshift4/ose-cluster-logging-operator
- openshift4/ose-cluster-machine-approver
- openshift4/ose-cluster-monitoring-operator
- openshift4/ose-cluster-network-operator
- openshift4/ose-cluster-node-tuned
- openshift4/ose-cluster-node-tuning-operator
- openshift4/ose-cluster-openshift-apiserver-operator
- openshift4/ose-cluster-openshift-controller-manager-operator
- openshift4/ose-cluster-samples-operator
- openshift4/ose-cluster-storage-operator
- openshift4/ose-cluster-svcat-apiserver-operator
- openshift4/ose-cluster-svcat-controller-manager-operator
- openshift4/ose-cluster-version-operator
- openshift4/ose-configmap-reloader
- openshift4/ose-console
- openshift4/ose-console-operator
- openshift4/ose-container-networking-plugins-supported
- openshift4/ose-container-networking-plugins-unsupported
- openshift4/ose-coredns
- openshift4/ose-deployer
- openshift4/ose-descheduler
- openshift4/ose-descheduler-operator

openshift4/ose-docker-builder
openshift4/ose-docker-registry
openshift4/ose-egress-dns-proxy
openshift4/ose-egress-http-proxy
openshift4/ose-egress-router
openshift4/ose-elasticsearch-operator
openshift4/ose-etcd
openshift4/ose-grafana
openshift4/ose-haproxy-router
openshift4/ose-hyperkube
openshift4/ose-hypershift
openshift4/ose-installer-artifacts
openshift4/ose-installer
openshift4/ose-jenkins-agent-base
openshift4/ose-jenkins-agent-maven
openshift4/ose-jenkins-agent-nodejs
openshift4/ose-jenkins
openshift4/ose-k8s-prometheus-adapter
openshift4/ose-keepalived-ipfailover
openshift4/ose-kube-client-agent
openshift4/ose-kube-etcd-signer-server
openshift4/ose-kube-proxy
openshift4/ose-kube-rbac-proxy
openshift4/ose-kube-state-metrics
openshift4/ose-libvirt-machine-controllers
openshift4/ose-logging-curator5
openshift4/ose-logging-elasticsearch5
openshift4/ose-logging-eventrouter
openshift4/ose-logging-fluentd
openshift4/ose-logging-kibana5
openshift4/ose-machine-api-operator
openshift4/ose-machine-config-controller
openshift4/ose-machine-config-daemon
openshift4/ose-machine-config-operator
openshift4/ose-machine-config-server
openshift4/ose-multus-cni
openshift4/ose-must-gather
openshift4/ose-node
openshift4/ose-oauth-proxy
openshift4/ose-operator-lifecycle-manager
openshift4/ose-operator-marketplace
openshift4/ose-operator-registry
openshift4/ose-ovn-kubernetes
openshift4/ose-pod
openshift4/ose-prometheus-alertmanager
openshift4/ose-prometheus-config-reloader
openshift4/ose-prometheus
openshift4/ose-prometheus-node-exporter
openshift4/ose-prometheus-operator
openshift4/ose-prom-label-proxy
openshift4/ose-recycler
openshift4/ose-service-ca-operator
openshift4/ose-service-catalog
openshift4/ose-setup-etcd-environment
openshift4/ose-sriov-cni
openshift4/ose-sriov-network-device-plugin

- openshift4/ose-telemeter
- openshift4/ose-template-service-broker-operator
- openshift4/ose-template-service-broker
- openshift4/ose-tests
- openshift4/postgresql-apb
- openshift4/ose-haproxy-router-base

第 2 章 OPENSIFT CONTAINER PLATFORM 版本政策

OpenShift Container Platform 对所有支持的 API 提供了严格的后向兼容保证，但这不包括 alpha API（这些 API 可能会在不通知的情况下被改变），以及 beta API（这些 API 偶尔可能会被改变且不保证后向兼容）。

红帽没有公开发布 OpenShift Container Platform 4.0，在版本 3.11 后的第一个正式发行版本即为 OpenShift Container Platform 4.1。

master 主机和节点（node）主机使用的 OpenShift Container Platform 版本必须相互匹配（在集群升级过程中出现的临时不匹配除外）。例如，在一个 4.1 集群中，所有的 master 必需是 4.1，所有节点也必需是 4.1。如果使用的 **oc** 版本早于 4.1，则无法使用 OpenShift Container Platform 4.1 中的所有命令。您需要下载并安装新版本的 **oc**。

因为非安全的原因改变 API 将会最少涉及到 2 个次发行版本（例如，3.4 到 3.5 到 3.6）来更新旧的 **oc**，使用新功能可能会需要使用新版本的 **oc**。一个 3.2 版本的服务器可能会带有版本 3.1 的 **oc** 不能使用的功能，而一个版本为 3.2 的 **oc** 也可能会有不被版本 3.1 服务器支持的功能。

表 2.1. 兼容性列表

	X.Y (oc Client)	X.Y+N ^[a] (oc Client)
X.Y (Server)	1	3
X.Y+N ^[a] (Server)	2	1

[a] 其中N 是一个大于 1 的数字。

- 1** 完全兼容。
- 2** **oc** 可能无法访问服务器的功能。
- 3** **oc** 可能会包括与要访问的服务器不兼任的选项和功能。