



Red Hat OpenShift Container Storage 4.8

使用 Amazon Web Services 部署 OpenShift Container Storage

如何在 OpenShift Container Platform AWS 集群上安装并设置 OpenShift Container Storage

Red Hat OpenShift Container Storage 4.8 使用 Amazon Web Services 部署 OpenShift Container Storage

如何在 OpenShift Container Platform AWS 集群上安装并设置 OpenShift Container Storage

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律通告

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Deploying_OpenShift_Container_Storage_using_Amazon_Web_Services.ent file |.

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

有关使用 Amazon Web Services 进行云存储安装 Red Hat OpenShift Container Storage 4.8 的说明，请参阅本文档。

目录

让开源更具包容性	3
对红帽文档提供反馈	4
前言	5
第 1 章 准备部署 OPENSIFT CONTAINER STORAGE	6
1.1. 为基于 RED HAT ENTERPRISE LINUX 的节点上的容器启用文件系统访问	6
1.2. 在 VAULT 中启用键值后端路径和策略	7
第 2 章 使用动态存储设备进行部署	8
2.1. 安装 RED HAT OPENSIFT CONTAINER STORAGE OPERATOR	8
2.2. 以内部模式创建 OPENSIFT CONTAINER STORAGE CLUSTER SERVICE	9
第 3 章 验证 OPENSIFT CONTAINER STORAGE 部署	12
3.1. 验证 POD 的状态	12
3.2. 验证 OPENSIFT CONTAINER STORAGE 集群是否正常运行	13
3.3. 验证 MULTICLOUD 对象网关是否健康	13
3.4. 验证 OPENSIFT CONTAINER STORAGE 特定的存储类是否存在	14
第 4 章 卸载 OPENSIFT CONTAINER STORAGE	15
4.1. 在内部模式中卸载 OPENSIFT CONTAINER STORAGE	15
4.2. 从 OPENSIFT CONTAINER STORAGE 中删除监控堆栈	21
4.3. 从 OPENSIFT CONTAINER STORAGE 中删除 OPENSIFT CONTAINER PLATFORM REGISTRY	24
4.4. 从 OPENSIFT CONTAINER STORAGE 中删除集群日志记录 OPERATOR	25

让开源更具包容性

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。如需了解更多详细信息，请参阅 [CTO Chris Wright 信息](#)。

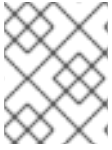
对红帽文档提供反馈

我们感谢您对文档提供反馈信息。请告诉我们如何让它更好。提供反馈：

- 关于特定内容的简单评论：
 1. 请确定您使用 *Multi-page HTML* 格式查看文档。另外，确定 **Feedback** 按钮出现在文档页的右上方。
 2. 用鼠标指针高亮显示您想评论的文本部分。
 3. 点在高亮文本上弹出的 **Add Feedback**。
 4. 按照显示的步骤操作。
- 要提交更复杂的反馈，请创建一个 Bugzilla ticket：
 1. 进入 [Bugzilla](#) 网站。
 2. 在 **Component** 部分中，选择 **文档**。
 3. 在 **Description** 中输入您要提供的信息。包括文档相关部分的链接。
 4. 点 **Submit Bug**。

前言

Red Hat OpenShift Container Storage 4.8 支持在连接或断开连接的环境中的现有 Red Hat OpenShift Container Platform (RHOCP) AWS 集群上部署，以及代理环境的开箱即用支持。



注意

AWS 仅支持内部 OpenShift Container Storage 集群。如需有关部署要求的更多信息，请参阅[规划部署并准备部署 OpenShift Container Storage](#)。

要部署 OpenShift Container Storage，请从 [准备部署 OpenShift Container Storage](#) 章节的要求开始，然后按照环境的部署过程操作：

- [使用动态存储设备进行部署](#)

第 1 章 准备部署 OPENSIFT CONTAINER STORAGE

使用动态存储设备在 OpenShift Container Platform 上部署 OpenShift Container Storage 可让您选择创建内部集群资源。这将会在内部置备基础服务，这有助于为应用提供额外的存储类。

在开始部署 Red Hat OpenShift Container Storage 前，请按照以下步骤执行：

1. 对于用于 worker 节点的基于 Red Hat Enterprise Linux 的主机，[为基于 Red Hat Enterprise Linux 的节点上的容器启用文件系统访问](#)。



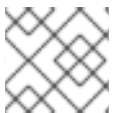
注意

对于 Red Hat Enterprise Linux CoreOS (RHCOS) 可以跳过此步骤。

2. 可选：如果要使用外部密钥管理系统(KMS)启用集群范围加密：
 - 确保存在具有令牌的策略，并且启用了 Vault 中的键值后端路径。请参阅[在 Vault 中启用键值后端路径和策略](#)。
 - 确保您在 Vault 服务器上使用签名的证书。
3. 最低的启动节点要求 [\[技术预览\]](#)
当不符合标准部署资源要求时，OpenShift Container Storage 集群将以最小配置进行部署。请参阅规划指南中的[资源要求](#)部分。

1.1. 为基于 RED HAT ENTERPRISE LINUX 的节点上的容器启用文件系统访问

在用户置备的基础架构 (UPI) 的 Red Hat Enterprise Linux 基础上使用 worker 节点部署 OpenShift Container Storage 不会自动提供对底层 Ceph 文件系统的容器访问。



注意

对于基于 Red Hat Enterprise Linux CoreOS (RHCOS) 的主机，跳过这一步。

步骤

1. 登录基于 Red Hat Enterprise Linux 的节点并打开一个终端。
2. 对于集群中的每个节点：
 - a. 验证节点有权访问 rhel-7-server-extras-rpms 存储库。

```
# subscription-manager repos --list-enabled | grep rhel-7-server
```

如果您没有在输出中看到 **rhel-7-server-rpms** 和 **rhel-7-server-extras-rpms**，或者没有输出，请运行以下命令来启用相关的存储库：

```
# subscription-manager repos --enable=rhel-7-server-rpms
# subscription-manager repos --enable=rhel-7-server-extras-rpms
```

- b. 安装所需的软件包。

```
# yum install -y policycoreutils container-selinux
```

c. 在 SELinux 中永久启用 Ceph 文件系统的容器使用。

```
# setsebool -P container_use_cephfs on
```

1.2. 在 VAULT 中启用键值后端路径和策略

先决条件

- 管理员对 Vault 的访问权限。
- 仔细选择唯一路径名称作为遵循命名惯例的后端路径，因为它无法在以后更改。

步骤

1. 在 Vault 中启用 Key/Value(KV)后端路径。
对于 Vault KV secret 引擎 API，版本 1：

```
$ vault secrets enable -path=ocs kv
```

对于 Vault KV secret 引擎 API，版本 2：

```
$ vault secrets enable -path=ocs kv-v2
```

2. 使用以下命令，创建一个策略来限制用户对 secret 执行写入或删除操作：

```
echo '
path "ocs/*" {
  capabilities = ["create", "read", "update", "delete", "list"]
}
path "sys/mounts" {
  capabilities = ["read"]
}' | vault policy write ocs -
```

3. 创建与上述策略匹配的令牌：

```
$ vault token create -policy=ocs -format json
```

第 2 章 使用动态存储设备进行部署

使用 AWS EBS 提供的动态存储设备 (type: gp2) 在 OpenShift Container Platform 上部署 OpenShift Container Storage 为您提供了创建内部集群资源的选项。这将会在内部置备基础服务，这有助于为应用提供额外的存储类。



注意

AWS 仅支持内部 OpenShift Container Storage 集群。如需有关部署要求的更多信息，请参阅[规划部署](#)。

另外，请确保已满足 [准备部署 OpenShift Container Storage](#) 章节的要求，然后按照以下步骤使用动态存储设备进行部署：

1. [安装 Red Hat OpenShift Container Storage Operator](#)。
2. [创建 OpenShift Container Storage Cluster Service](#)。

2.1. 安装 RED HAT OPENSIFT CONTAINER STORAGE OPERATOR

您可以使用 Red Hat OpenShift Container Platform Operator Hub 安装 Red Hat OpenShift Container Storage Operator。

先决条件

- 使用具有 cluster-admin 和 operator 安装权限的账户访问 OpenShift Container Platform 集群。
- 在 Red Hat OpenShift Container Platform 集群中至少有三个 worker 节点。
- 您满足了任何额外的要求。如需更多信息，请参阅 [规划部署](#)。



注意

- 当您需要覆盖 OpenShift Container Storage 的集群范围默认节点选择器时，您可以使用以下命令为 **openshift-storage** 命名空间指定空白节点选择器（在这种情况下创建 openshift-storage 命名空间）：

```
$ oc annotate namespace openshift-storage openshift.io/node-selector=
```

- 将节点作为 **infra** 污点，以确保只在该节点上调度 Red Hat OpenShift Container Storage 资源。这有助于您节省订阅成本。如需更多信息，请参阅[管理和分配存储资源指南中的如何将专用 worker 节点用于 Red Hat OpenShift Container Storage](#) 一章。

步骤

1. 登录 OpenShift Web 控制台。
2. 点 **Operators** → **OperatorHub**。
3. 从操作器列表中搜索 **OpenShift Container Storage** 并点它。
4. 点击 **Install**。

5. 在 **Install Operator** 页面中设置以下选项：

- a. 频道为 **stable-4.8**。
 - b. 安装模式是 **A specific namespace on the cluster**。
 - c. Installed Namespace 为 **Operator recommended namespace openshift-storage**。如果 Namespace **openshift-storage** 不存在，它会在 Operator 安装过程中创建。
 - d. **批准策略**为 **Automatic** 或 **Manual**。
 - e. 点击 **Install**。
- 如果选择 **Automatic** 更新，Operator Lifecycle Manager(OLM)将自动升级 Operator 的运行实例，而无需任何干预。

如果选择 **手动** 更新，则 OLM 会创建一个更新请求。作为集群管理员，您必须手动批准该更新请求，才可将 Operator 更新至新版本。

验证步骤

- 验证 **OpenShift Container Storage Operator** 是否显示绿色勾号，指示安装成功。

2.2. 以内部模式创建 OPENSIFT CONTAINER STORAGE CLUSTER SERVICE

安装 OpenShift Container Storage operator 后，请使用此流程创建 OpenShift Container Storage Cluster Service。

先决条件

- OpenShift Container Storage Operator 必须从 Operator Hub 安装。如需更多信息，请参阅[使用 Operator Hub 安装 OpenShift Container Storage Operator](#)。

步骤

1. 登录 OpenShift Web 控制台。
2. 点 **Operators** → **Installed Operators** 查看所有已安装的 Operator。确保所选项目为 **openshift-storage**。
3. 点 **OpenShift Container Storage** > **Create Instance Cluster** 链接。
4. 选择 **Mode** 被默认设置为 **Internal**。
5. 选择 **Capacity** 和节点
 - a. 选择 **Storage Class**。默认情况下，它被设置为 **gp2**。
 - b. 从下拉列表中选择 **Requested Capacity**。默认设置为 **2 TiB**。您可以使用下拉菜单修改容量值。



注意

选择初始存储容量后，集群扩展将使用所选的可用容量（原始存储的 3 倍）执行。

- c. 在 **Select Nodes** 部分中，选择至少三个可用节点。
对于具有多个可用区的云平台，请确保节点分布在不同的位置/可用性区域。

如果选择的节点与一个聚合的 30 个 CPU 和 72 GiB RAM 的要求不匹配，则会部署一个最小的集群。如需最低起始节点要求，请参阅规划指南中的[资源要求](#)部分。

- d. 点击 **Next**。
6. (可选) 设置**安全性和网络配置**
 - a. 选中 **启用加密** 复选框，以加密块和文件存储。
 - b. 选择任意一个或两个 **加密级别**：
 - **Cluster-wide encryption** 来加密整个集群（块存储和文件存储）。
 - **Storage class encryption** 以使用加密启用的存储类创建加密的持久性卷（仅限块）。
 - c. 选择 **Connect to an external key management service**复选框。这是集群范围加密的可选选项。
 - i. 默认情况下，**Key Management Service Provider**设置为 **Vault**。
 - ii. 输入 **Vault Service Name**、Vault 服务器的主机**地址**（'https://<hostname 或 ip>'）、**端口号**和 **Token**。
 - iii. 展开 **Advanced Settings** 根据您的 Vault 配置输入额外的设置和证书详情：
 - A. 在 **后端路径**中输入为 OpenShift Container Storage 专用且唯一的 Key Value secret 路径。
 - B. (可选) 输入 **TLS 服务器名称**和 **Vault Enterprise 命名空间**。
 - C. 通过上传相应的 PEM 编码证书文件提供 **CA 证书**、**客户端证书**和**客户端私钥**。
 - D. 点击 **Save**。
 7. 如果使用一个单一网络，选择 **Default (SDN)**；如果使用多个网络借口，选择 **Custom (Multus) Network**。
 - a. 从下拉菜单中选择**公共网络接口**。
 - b. 从下拉菜单中选择 **Cluster Network Interface**。



注意

如果只使用一个额外网络接口，请为公共网络接口选择单个 NetworkAttachmentDefinition（如 ocs-public-cluster），并将 Cluster Network Interface 留空。

8. 点击 **Next**。
9. 检查配置详情。若要修改任何配置设置，请单击 **Back** 以返回到上一配置页面。
10. 点 **Create**。

11. 如果 Vault Key/Value(KV)secret 引擎 API, 则编辑 configmap, 版本 2 则用于使用密钥管理系统 (KMS)进行集群范围的加密。
 - a. 在 OpenShift Web 控制台中, 导航到 **Workloads → ConfigMaps**。
 - b. 要查看 KMS 连接详情, 点 **ocs-kms-connection-details**。
 - c. 编辑 configmap。
 - i. 点 **Action menu (⋮) → Edit ConfigMap**。
 - ii. 将 **VAULT_BACKEND** 参数设置为 **v2**。

```

kind: ConfigMap
apiVersion: v1
metadata:
  name: ocs-kms-connection-details
[...]
data:
  KMS_PROVIDER: vault
  KMS_SERVICE_NAME: vault
[...]
  VAULT_BACKEND: v2
[...]

```

- iii. 点击 **Save**。

验证步骤

1. 在存储集群详情页面中, 存储集群名称旁边显示一个绿色勾号, 表示集群创建成功。
2. 验证已安装存储集群的最后一个 **Status** 显示为 **Phase: Ready**, 并带有绿色勾号标记。
 - 点 **Operators → Installed Operators → Storage Cluster**链接来查看存储集群安装状态。
 - 另外, 当使用 Operator **Details** 选项卡时, 您可以点击 **Storage Cluster** 选项卡查看状态。
3. 要验证 OpenShift Container Storage 的所有组件是否已成功安装, 请参阅[验证 OpenShift Container Storage 安装](#)。

第 3 章 验证 OPENSIFT CONTAINER STORAGE 部署

使用本节验证 OpenShift Container Storage 是否已正确部署。

3.1. 验证 POD 的状态

要验证 OpenShift Container Storage 的 pod 是否正在运行，请按照以下步骤操作：

步骤

1. 登录 OpenShift Web 控制台。
2. 从 OpenShift Web 控制台左侧窗格中，点击 **Workloads → Pods**。
3. 从 **Project** 下拉列表中，选择 **openshift-storage**。
有关每个组件预期的 pod 数量及其变化取决于节点数量的更多信息，请参阅 [表 3.1“对应 OpenShift Container 存储集群的 Pod”](#)。
4. 点击 **Running** 和 **Completed** 选项卡以验证 pod 正在运行并处于 completed 状态：

表 3.1. 对应 OpenShift Container 存储集群的 Pod

组件	对应的 pod
OpenShift Container Storage Operator	<ul style="list-style-type: none"> ● OCS-operator-* (在任何 worker 节点上有 1 个 pod) ● ocs-metrics-exporter-*
Rook-ceph Operator	rook-ceph-operator-* (任何 worker 节点上有 1 个 pod)
多云对象网关	<ul style="list-style-type: none"> ● noobaa-operator-* (任何 worker 节点上 1 个 pod) ● noobaa-core-* (任何存储节点上 1 个 pod) ● noobaa-db-pg-* (任何存储节点上 1 个 pod) ● noobaa-endpoint-* (任何存储节点上 1 个 pod)
MON	rook-ceph-mon-* (在存储节点间分布 3 个 pod)
MGR	rook-ceph-mgr-* (任何存储节点上的 1 个 pod)

组件	对应的 pod
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* (2 个 pod 在存储节点间分布)
CSI	<ul style="list-style-type: none"> ● cephfs <ul style="list-style-type: none"> ○ csi-cephfsplugin-* (每个 worker 节点上 1 个 pod) ○ csi-cephfsplugin-provisioner-* (2 个 pod 在不同的 worker 节点上分布) ● rbd <ul style="list-style-type: none"> ○ csi-rbdplugin-* (每个 worker 节点上 1 个 pod) ○ csi-rbdplugin-provisioner-* (2 个 pod 在不同的 worker 节点上分步)
rook-ceph-crashcollector	rook-ceph-crashcollector-* (每个存储节点上 1 个 pod)
OSD	<ul style="list-style-type: none"> ● rook-ceph-osd-* (每个设备 1 个 pod) ● rook-ceph-osd-prepare-ocs-deviceset-* (每个设备 1 个 pod)

3.2. 验证 OPENSIFT CONTAINER STORAGE 集群是否正常运行

要验证 OpenShift Container Storage 的集群是否正常运行，请按照以下步骤执行。

步骤

1. 点 **Storage** → **Overview**，点 **Block and File** 选项卡。
2. 在 **Status** 卡中，验证 *Storage Cluster* 和 *Data Resiliency* 具有绿色勾号标记。
3. 在 **Details** 卡中，验证是否显示集群信息。

如需有关使用 Block and File 仪表板的 OpenShift Container Storage 集群健康状况的更多信息，请参阅 [监控 OpenShift Container Storage](#)。

3.3. 验证 MULTICLOUD 对象网关是否健康

要验证 OpenShift Container Storage Multicloud 对象网关是否健康，请按照以下步骤执行。

步骤

1. 在 OpenShift Web 控制台中点 **Storage → Overview**，然后单击 **Object** 选项卡。
2. 在 **Status** 卡中，验证 *Object Service* 和 *Data Resiliency* 都处于 **Ready** 状态（绿色钩号）。
3. 在 **Details** 卡中，验证是否显示了 Multicloud 对象网关信息。

如需有关使用对象服务仪表板的 OpenShift Container Storage 集群健康状况的更多信息，请参阅[监控 OpenShift Container Storage](#)。

3.4. 验证 OPENSIFT CONTAINER STORAGE 特定的存储类是否存在

要验证集群中的存储类是否存在，请按照以下步骤执行。

步骤

1. 从 OpenShift Web 控制台点 **Storage → Storage Classes**
2. 验证是否在创建 OpenShift Container Storage 集群时创建了以下存储类：
 - **ocs-storagecluster-ceph-rbd**
 - **ocs-storagecluster-cephfs**
 - **openshift-storage.noobaa.io**

第 4 章 卸载 OPENSIFT CONTAINER STORAGE

4.1. 在内部模式中卸载 OPENSIFT CONTAINER STORAGE

使用本节中的步骤卸载 OpenShift Container Storage。

卸载注解

Storage Cluster 上的注解用于更改卸载过程的行为。要定义卸载行为，在存储集群中引入了以下两个注解：

- `uninstall.ocs.openshift.io/cleanup-policy: delete`
- `uninstall.ocs.openshift.io/mode: graceful`

下表提供了有关可用于这些注解的不同值的信息：

表 4.1. `uninstall.ocs.openshift.io` 卸载注解描述

注解	订阅价值	Default (默认)	行为
<code>cleanup-policy</code>	删除	是	Rook 清理物理驱动器和 DataDirHostPath
<code>cleanup-policy</code>	<code>retain</code>	否	Rook 不会 清理物理驱动器和 DataDirHostPath
模式	<code>graceful</code>	是	Rook 和 NooBaa 暂停 卸载过程，直到管理员/用户移除 PVC 和 OBC
模式	<code>forced</code>	否	Rook 和 NooBaa 即使使用 Rook 和 NooBaa 置备的 PVC/OBC 分别存在，也会继续卸载。

您可以通过使用以下命令编辑注解值来更改清理策略或卸载模式：

```
$ oc annotate storagecluster -n openshift-storage ocs-storagecluster
uninstall.ocs.openshift.io/cleanup-policy="retain" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

```
$ oc annotate storagecluster -n openshift-storage ocs-storagecluster
uninstall.ocs.openshift.io/mode="forced" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

先决条件

- 确保 OpenShift Container Storage 集群处于健康状态。当因为资源或节点不足而导致部分 pod 无法成功终止时，卸载过程可能会失败。如果集群处于不健康状态，请在卸载 OpenShift Container Storage 前联络红帽客户支持。

- 使用 OpenShift Container Storage 提供的存储类，确保应用程序不使用持久性卷声明 (PVC) 或对象存储桶声明 (OBC)。
- 如果管理员创建了任何自定义资源（如自定义存储类、cephblockpools），则管理员必须在移除消耗这些资源后将它们删除。

步骤

1. 删除使用 OpenShift Container Storage 的卷快照。

- a. 列出来自所有命名空间的卷快照。

```
$ oc get volumesnapshot --all-namespaces
```

- b. 在上一命令的输出中，识别和删除使用 OpenShift Container Storage 的卷快照。

```
$ oc delete volumesnapshot <VOLUME-SNAPSHOT-NAME> -n <NAMESPACE>
```

2. 删除使用 OpenShift Container Storage 的 PVC 和 OBC。

在默认的卸载模式 (graceful) 中，卸载程序会等待所有使用 OpenShift Container Storage 的 PVC 和 OBC 被删除。

如果要事先删除 PVC 来删除存储集群，您可以将卸载模式注解设置为**强制**并跳过此步骤。这样做会导致系统处于孤立的 PVC 和 OBC。

- a. 使用 OpenShift Container Storage 删除 OpenShift Container Platform 监控堆栈 PVC。如需更多信息，请参阅 [第 4.2 节“从 OpenShift Container Storage 中删除监控堆栈”](#)。
- b. 使用 OpenShift Container Storage 删除 OpenShift Container Platform Registry PVC。更多信息请参阅 [第 4.3 节“从 OpenShift Container Storage 中删除 OpenShift Container Platform registry”](#)。
- c. 使用 OpenShift Container Storage 删除 OpenShift Container Platform 日志 PVC。如需更多信息，请参阅 [第 4.4 节“从 OpenShift Container Storage 中删除集群日志记录 Operator”](#)。
- d. 删除使用 OpenShift Container Storage 置备的其他 PVC 和 OBC。

- 以下示例是示例脚本，用于识别使用 OpenShift Container Storage 置备的 PVC 和 OBC。该脚本忽略 OpenShift Container Storage 内部使用的 PVC。

```
#!/bin/bash

RBD_PROVISIONER="openshift-storage.rbd.csi.ceph.com"
CEPHFS_PROVISIONER="openshift-storage.cephfs.csi.ceph.com"
NOOBAA_PROVISIONER="openshift-storage.noobaa.io/obc"
RGW_PROVISIONER="openshift-storage.ceph.rook.io/bucket"

NOOBAA_DB_PVC="noobaa-db"
NOOBAA_BACKINGSTORE_PVC="noobaa-default-backing-store-noobaa-pvc"

# Find all the OCS StorageClasses
OCS_STORAGECLASSES=$(oc get storageclasses | grep -e
"$RBD_PROVISIONER" -e "$CEPHFS_PROVISIONER" -e
"$NOOBAA_PROVISIONER" -e "$RGW_PROVISIONER" | awk '{print $1}')
```

```
# List PVCs in each of the StorageClasses
for SC in $OCS_STORAGECLASSES
do
    echo
    "=====
=="
    echo "$SC StorageClass PVCs and OBCs"
    echo
    "=====
=="
    oc get pvc --all-namespaces --no-headers 2>/dev/null | grep $SC | grep -v -e
"$NOOBAA_DB_PVC" -e "$NOOBAA_BACKINGSTORE_PVC"
    oc get obc --all-namespaces --no-headers 2>/dev/null | grep $SC
    echo
done
```



注意

云平台省略 **RGW_PROVISIONER**。

- 删除 OBC。

```
$ oc delete obc <obc name> -n <project name>
```

- 删除 PVC。

```
$ oc delete pvc <pvc name> -n <project-name>
```



注意

确保您已删除了集群中创建的任何自定义后备存储、存储桶类等。

3. 删除 Storage Cluster 对象并等待相关资源被删除。

```
$ oc delete -n openshift-storage storagecluster --all --wait=true
```

4. 检查 **uninstall.ocs.openshift.io/cleanup-policy** 是否已设置为 **delete**（默认），并确保其状态为 **Completed**。

```
$ oc get pods -n openshift-storage | grep -i cleanup
NAME                READY STATUS  RESTARTS AGE
cluster-cleanup-job-<xx> 0/1   Completed 0      8m35s
cluster-cleanup-job-<yy> 0/1   Completed 0      8m35s
cluster-cleanup-job-<zz> 0/1   Completed 0      8m35s
```

5. 确认目录 **/var/lib/rook** 现在为空。只有 **uninstall.ocs.openshift.io/cleanup-policy** 注解设置为 **delete**（默认）时，此目录才为空。

```
$ for i in $(oc get node -l cluster.ocs.openshift.io/openshift-storage= -o jsonpath='{
.items[*].metadata.name }'); do oc debug node/${i} -- chroot /host ls -l /var/lib/rook; done
```

6. 如果在安装时启用了加密，在所有 OpenShift Container Storage 节点上的 OSD 设备中删除 **dm-crypt** 管理的 **device-mapper** 映射。

- a. 创建 **debug** pod 和 **chroot** 到存储节点上的主机。

```
$ oc debug node/<node name>
$ chroot /host
```

- b. 获取设备名称并记录 OpenShift Container Storage 设备。

```
$ dmsetup ls
ocs-deviceset-0-data-0-57snx-block-dmccrypt (253:1)
```

- c. 删除映射的设备。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-0-data-0-57snx-block-dmccrypt
```



注意

如果上述命令因为权限不足而卡住，请运行以下命令：

- 按 **CTRL+Z** 退出上述命令。
- 查找阻塞的进程的 PID。

```
$ ps -ef | grep crypt
```

- 使用 **kill** 命令终止进程。

```
$ kill -9 <PID>
```

- 验证设备名称是否已移除。

```
$ dmsetup ls
```

7. 删除命名空间并等待删除完成。如果 **openshift-storage** 是活跃的项目，则需要切换到另一个项目。

例如：

```
$ oc project default
$ oc delete project openshift-storage --wait=true --timeout=5m
```

如果以下命令返回 **NotFound** 错误，则项目被删除。

```
$ oc get project openshift-storage
```



注意

卸载 OpenShift Container Storage 时，如果没有完全删除 **命名空间** 并处于 **Terminating** 状态，请执行 [故障排除和删除 Uninstall 过程中剩余的资源](#) 的步骤，以识别阻塞命名空间的对象。

8. 取消标记存储节点。

```
$ oc label nodes --all cluster.ocs.openshift.io/openshift-storage-
$ oc label nodes --all topology.rook.io/rack-
```

9. 如果节点有污点，则删除 OpenShift Container Storage 污点。

```
$ oc adm taint nodes --all node.ocs.openshift.io/storage-
```

10. 确认已删除使用 OpenShift Container Storage 置备的所有 PV。如果有任何 PV 处于 **Released** 状态，请将其删除。

```
$ oc get pv
$ oc delete pv <pv name>
```

11. 删除 Multicloud 对象网关存储类。

```
$ oc delete storageclass openshift-storage.noobaa.io --wait=true --timeout=5m
```

12. 删除 **CustomResourceDefinitions**。

```
$ oc delete crd backingstores.noobaa.io bucketclasses.noobaa.io
cephblockpools.ceph.rook.io cephclusters.ceph.rook.io cephfilesystems.ceph.rook.io
cephnfses.ceph.rook.io cephobjectstores.ceph.rook.io cephobjectstoreusers.ceph.rook.io
noobaas.noobaa.io ocsinitializations.ocs.openshift.io storageclusters.ocs.openshift.io
cephclients.ceph.rook.io cephobjectrealms.ceph.rook.io cephobjectzonegroups.ceph.rook.io
cephobjectzones.ceph.rook.io cephrbdmirrors.ceph.rook.io --wait=true --timeout=5m
```

13. 可选：要确保永久删除 vault 密钥，您需要手动删除与 vault 密钥关联的元数据。



注意

只有在 Vault Key/Value(KV)机密引擎 API 时才执行此步骤，使用密钥管理系统 (KMS) 版本 2 进行集群范围加密，因为 vault 密钥被标记为删除并在 OpenShift Container Storage 卸载过程中永久删除。如果需要，您可以在以后恢复它。

- a. 列出密码库中的密钥。

```
$ vault kv list <backend_path>
```

<backend_path>

是存储加密密钥的密码库中的路径。

例如：

```
$ vault kv list kv-v2
```

输出示例：

```
Keys
-----
NOOBAA_ROOT_SECRET_PATH/
```

```
rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
rook-ceph-osd-encryption-key-ocs-deviceset-thin-1-data-0sq227
rook-ceph-osd-encryption-key-ocs-deviceset-thin-2-data-0xzszb
```

- b. 列出与 vault 密钥关联的元数据。

```
$ vault kv get kv-v2/<key>
```

对于 Multicloud Object Gateway(MCG)密钥：

```
$ vault kv get kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

是加密密钥。

例如：

```
$ vault kv get kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-data-0m27q8
```

输出示例：

```
==== Metadata =====
Key          Value
---          -
created_time 2021-06-23T10:06:30.650103555Z
deletion_time 2021-06-23T11:46:35.045328495Z
destroyed    false
version      1
```

- c. 删除元数据。

```
$ vault kv metadata delete kv-v2/<key>
```

对于 MCG 密钥：

```
$ vault kv metadata delete kv-v2/NOOBAA_ROOT_SECRET_PATH/<key>
```

<key>

是加密密钥。

例如：

```
$ vault kv metadata delete kv-v2/rook-ceph-osd-encryption-key-ocs-deviceset-thin-0-
data-0m27q8
```

输出示例：

```
Success! Data deleted (if it existed) at: kv-v2/metadata/rook-ceph-osd-encryption-key-
ocs-deviceset-thin-0-data-0m27q8
```

- d. 重复这些步骤，以删除与所有 vault 密钥关联的元数据。

14. 在 OpenShift Container Platform Web 控制台中，确保完全卸载 OpenShift Container Storage，
 - a. 点 **Storage**。
 - b. 验证 **Overview** 不再显示在 Storage 下。

4.2. 从 OPENSIFT CONTAINER STORAGE 中删除监控堆栈

使用本节清理 OpenShift Container Storage 中的监控堆栈。

在配置监控堆栈时创建的 PVC 位于 **openshift-monitoring** 命名空间中。

先决条件

- PVC 被配置为使用 OpenShift Container Platform 监控堆栈。
如需更多信息，请参阅 [配置监控堆栈](#)。

步骤

1. 列出当前在 **openshift-monitoring** 命名空间中运行的 pod 和 PVC。

```
$ oc get pod,pvc -n openshift-monitoring
```

NAME	READY	STATUS	RESTARTS	AGE
pod/alertmanager-main-0	3/3	Running	0	8d
pod/alertmanager-main-1	3/3	Running	0	8d
pod/alertmanager-main-2	3/3	Running	0	8d
pod/cluster-monitoring-operator-84457656d-pkrxm	1/1	Running	0	8d
pod/grafana-79ccf6689f-2ll28	2/2	Running	0	8d
pod/kube-state-metrics-7d86fb966-rvd9w	3/3	Running	0	8d
pod/node-exporter-25894	2/2	Running	0	8d
pod/node-exporter-4dsd7	2/2	Running	0	8d
pod/node-exporter-6p4zc	2/2	Running	0	8d
pod/node-exporter-jbjvg	2/2	Running	0	8d
pod/node-exporter-jj4t5	2/2	Running	0	6d18h
pod/node-exporter-k856s	2/2	Running	0	6d18h
pod/node-exporter-rf8gn	2/2	Running	0	8d
pod/node-exporter-rmb5m	2/2	Running	0	6d18h
pod/node-exporter-zj7kx	2/2	Running	0	8d
pod/openshift-state-metrics-59dbd4f654-4clng	3/3	Running	0	8d
pod/prometheus-adapter-5df5865596-k8dzn	1/1	Running	0	7d23h
pod/prometheus-adapter-5df5865596-n2gj9	1/1	Running	0	7d23h
pod/prometheus-k8s-0	6/6	Running	1	8d
pod/prometheus-k8s-1	6/6	Running	1	8d
pod/prometheus-operator-55cfb858c9-c4zd9	1/1	Running	0	6d21h
pod/telemeter-client-78fc8fc97d-2rgfp	3/3	Running	0	8d

NAME	STATUS	VOLUME	
CAPACITY	ACCESS MODES	STORAGECLASS	AGE

```
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-0 Bound pvc-0d519c4f-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-1 Bound pvc-0d5a9825-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-alertmanager-claim-alertmanager-main-2 Bound pvc-0d6413dc-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-0 Bound pvc-0b7c19b0-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
persistentvolumeclaim/my-prometheus-claim-prometheus-k8s-1 Bound pvc-0b8aed3f-15a5-11ea-baa0-026d231574aa 40Gi RWO ocs-storagecluster-ceph-rbd 8d
```

2. 编辑监控 **configmap**。

```
$ oc -n openshift-monitoring edit configmap cluster-monitoring-config
```

3. 删除引用 OpenShift Container Storage 存储类的所有 **config** 部分，如下例所示并保存。
编辑前

```
.  
. .  
apiVersion: v1  
data:  
  config.yaml: |  
    alertmanagerMain:  
      volumeClaimTemplate:  
        metadata:  
          name: my-alertmanager-claim  
        spec:  
          resources:  
            requests:  
              storage: 40Gi  
          storageClassName: ocs-storagecluster-ceph-rbd  
  prometheusK8s:  
    volumeClaimTemplate:  
      metadata:  
        name: my-prometheus-claim  
      spec:  
        resources:  
          requests:  
            storage: 40Gi  
        storageClassName: ocs-storagecluster-ceph-rbd  
kind: ConfigMap  
metadata:  
  creationTimestamp: "2019-12-02T07:47:29Z"  
  name: cluster-monitoring-config  
  namespace: openshift-monitoring  
  resourceVersion: "22110"  
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config  
  uid: fd6d988b-14d7-11ea-84ff-066035b9efa8  
. . .
```

编辑后

```

.
.
.
apiVersion: v1
data:
  config.yaml: |
kind: ConfigMap
metadata:
  creationTimestamp: "2019-11-21T13:07:05Z"
  name: cluster-monitoring-config
  namespace: openshift-monitoring
  resourceVersion: "404352"
  selfLink: /api/v1/namespaces/openshift-monitoring/configmaps/cluster-monitoring-config
  uid: d12c796a-0c5f-11ea-9832-063cd735b81c
.
.
.

```

在本例中，**alertmanagerMain** 和 **prometheusK8s** 监控组件使用 OpenShift Container Storage PVC。

4. 删除相关的 PVC。请确定删除所有消耗存储类的 PVC。

```
$ oc delete -n openshift-monitoring pvc <pvc-name> --wait=true --timeout=5m
```

4.3. 从 OPENSIFT CONTAINER STORAGE 中删除 OPENSIFT CONTAINER PLATFORM REGISTRY

要从 OpenShift Container Storage 清理 OpenShift Container Platform registry，请按照以下步骤执行。

如果要配置其他存储，请参阅 [镜像 registry](#)

作为配置 OpenShift Container Platform registry 的一部分创建的 PVC 位于 **openshift-image-registry** 命名空间中。

先决条件

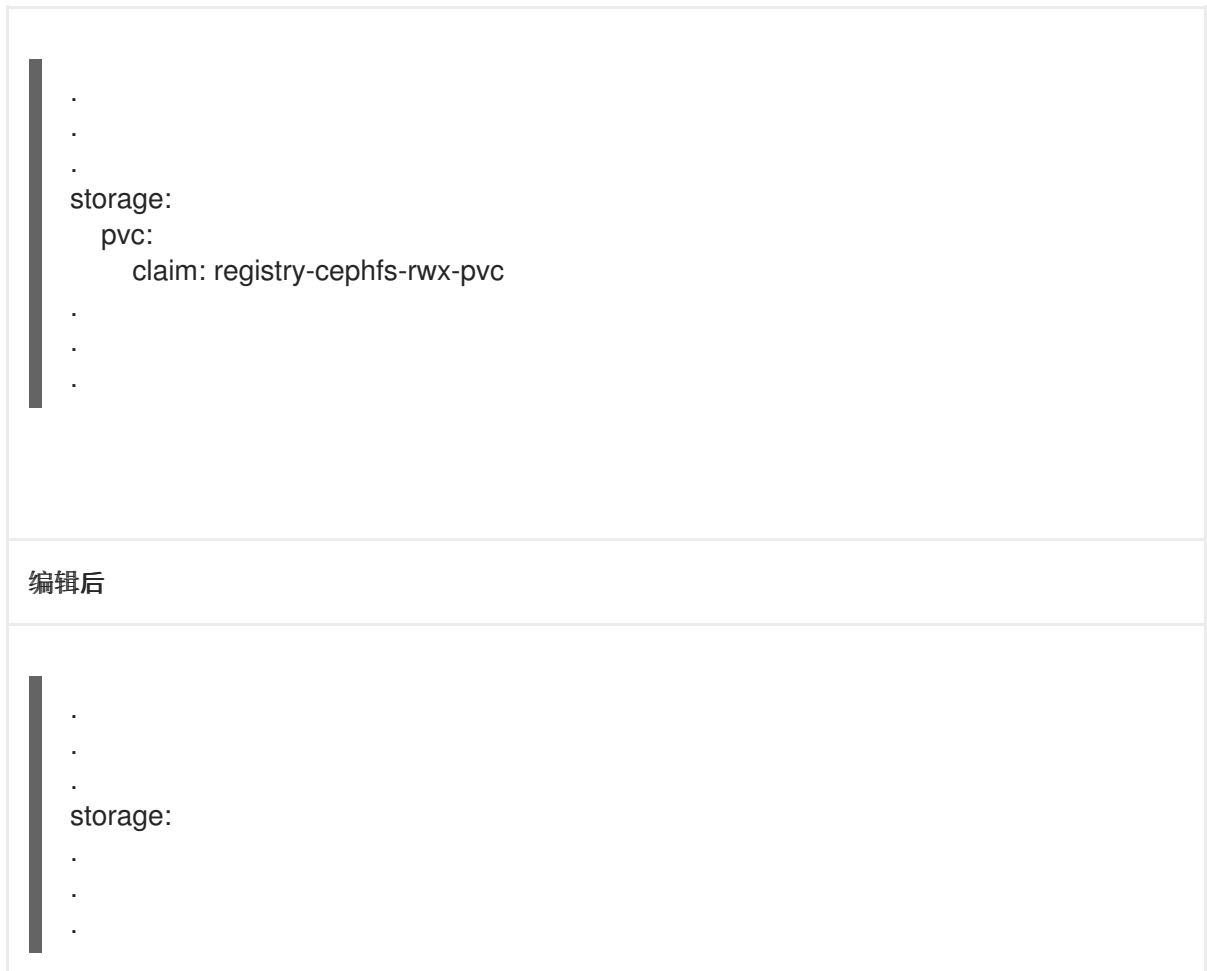
- 镜像 registry 必须配置为使用 OpenShift Container Storage PVC。

步骤

1. 编辑 **configs.imageregistry.operator.openshift.io** 对象，并删除 **storage** 部分中的内容。

```
$ oc edit configs.imageregistry.operator.openshift.io
```

编辑前



在本例中，PVC 称为 **registry-cephfs-rwx-pvc**，现在可以安全地删除。

2. 删除 PVC。

```
$ oc delete pvc <pvc-name> -n openshift-image-registry --wait=true --timeout=5m
```

4.4. 从 OPENSIFT CONTAINER STORAGE 中删除集群日志记录 OPERATOR

要从 OpenShift Container Storage 清理集群日志记录 Operator，请按照以下步骤执行。

作为配置集群日志记录 Operator 的一部分而创建的 PVC 位于 **openshift-logging** 命名空间中。

先决条件

- 集群日志记录实例必须配置为使用 OpenShift Container Storage PVC。

步骤

1. 删除命名空间中的 **ClusterLogging** 实例。

```
$ oc delete clusterlogging instance -n openshift-logging --wait=true --timeout=5m
```

openshift-logging 命名空间中的 PVC 现在可以安全地删除。

2. 删除 PVC。

```
$ oc delete pvc <pvc-name> -n openshift-logging --wait=true --timeout=5m
```