



# Red Hat OpenShift Container Storage 4.8

## 使用 IBM Z 基础架构部署 OpenShift Container Storage

如何安装和设置 IBM Z 环境



# Red Hat OpenShift Container Storage 4.8 使用 IBM Z 基础架构部署 OpenShift Container Storage

---

如何安装和设置 IBM Z 环境

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\_IBM\_Z\_infrastructure.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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

有关安装 Red Hat OpenShift Container Storage 4.8 以便在 IBM Z 基础架构中使用本地存储的说明，请参阅本文档。 While this document refers only to IBM Z, all information in it also applies to LinuxONE.

---

# 目录

使开源包含更多 .....	3
对红帽文档提供反馈 .....	4
前言 .....	5
<b>第 1 章 准备部署 OPENSIFT CONTAINER STORAGE .....</b>	<b>6</b>
1.1. 使用本地存储设备安装 OPENSIFT CONTAINER STORAGE 的要求	6
<b>第 2 章 使用本地存储设备部署 OPENSIFT CONTAINER STORAGE .....</b>	<b>7</b>
2.1. 安装 RED HAT OPENSIFT CONTAINER STORAGE OPERATOR	7
2.2. 安装 LOCAL STORAGE OPERATOR	8
2.3. 查找可用的存储设备（可选）	8
2.4. 在 IBM Z 上创建 OPENSIFT CONTAINER STORAGE 集群	9
<b>第 3 章 为内部附加设备模式验证 OPENSIFT CONTAINER STORAGE 部署 .....</b>	<b>13</b>
3.1. 验证 POD 的状态	13
3.2. 验证 OPENSIFT CONTAINER STORAGE 集群是否正常运行	14
3.3. 验证 OPENSIFT CONTAINER STORAGE 特定的存储类是否存在	14
<b>第 4 章 卸载 OPENSIFT CONTAINER STORAGE .....</b>	<b>16</b>
4.1. 以内部附加设备模式卸载 OPENSIFT CONTAINER STORAGE	16
4.1.1. 删除本地存储 Operator 配置	19
4.2. 从 OPENSIFT CONTAINER STORAGE 中删除监控堆栈	23
4.3. 从 OPENSIFT CONTAINER STORAGE 中删除 OPENSIFT CONTAINER PLATFORM REGISTRY	26
4.4. 从 OPENSIFT CONTAINER STORAGE 中删除集群日志记录 OPERATOR	27



## 使开源包含更多

红帽承诺替换我们的代码、文档和网页属性中存在问题的语言。我们从这四个术语开始：master、slave、blacklist 和 whitelist。这些更改将在即将发行的几个发行本中逐渐实施。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

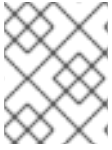
我们感谢您对文档提供反馈信息。请告诉我们如何让它更好。提供反馈：

- 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 中选择 **Documentation**。
  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)IBM Z 集群上部署，以及代理环境的开箱即用支持。



### 注意

IBM Z 上只支持内部 OpenShift Container Storage 集群。如需更多与部署相关的信息，请[参阅规划 部署并准备部署 OpenShift Container Storage](#)。

要部署 OpenShift Container Storage，请遵循适合您的环境的部署过程：

- 内部附加设备模式
  - [使用本地存储设备部署](#)

# 第 1 章 准备部署 OPENSIFT CONTAINER STORAGE

使用本地存储设备在 OpenShift Container Platform 上部署 OpenShift Container Storage 时，您可以创建内部集群资源。此方法在内部调配基础服务。然后，所有应用程序都可以访问其他存储类。

在使用本地存储开始部署 Red Hat OpenShift Container Storage 前，请确保满足您的资源要求。请参阅 [使用本地存储设备安装 OpenShift Container Storage 的要求](#)。

在解决了以上问题后，按照给出的顺序执行这些步骤：

1. [安装 Red Hat OpenShift Container Storage Operator](#)。
2. [安装 Local Storage Operator](#)。
3. [查找可用的存储设备](#)。
4. [在 IBM Z 上创建 OpenShift Container Storage 集群服务](#)。

## 1.1. 使用本地存储设备安装 OPENSIFT CONTAINER STORAGE 的要求

### 节点要求

集群必须至少包含三个 OpenShift Container Platform worker 节点，每个节点都有本地附加存储设备。

- 三个所选节点的每个节点必须至少有一个原始块设备可供 OpenShift Container Storage 使用。
- 您使用的设备必须为空；磁盘不得包含物理卷 (PV)，卷组 (VG) 或逻辑卷 (LV) 不能保留在磁盘上。

请参阅规划指南中的 [资源要求](#) 部分。

- 对于存储设备，需要 FCP 存储设备。
- 不支持多云对象网关。

### 最低起始节点要求 [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。

## 第 2 章 使用本地存储设备部署 OPENSIFT CONTAINER STORAGE

使用本地存储设备在 OpenShift Container Platform 上部署 OpenShift Container Storage 可让您选择创建内部集群资源。按照这个部署方法，使用本地存储来支持 OpenShift Container Platform 应用程序的持久性卷。

使用本节在已安装 OpenShift Container Platform 的 IBM Z 基础架构上部署 OpenShift Container Storage。

### 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。
- 如需额外的资源要求，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，您可以在命令行界面中使用以下命令为 **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。

#### 流程

1. 在 Web 控制台中导航至点 **Operators → OperatorHub**。
2. 在 Filter by keyword 框中滚动或输入关键字以搜索 **OpenShift Container Storage Operator**。
3. 在 OpenShift Container Storage operator 页中点 **Install**。
4. 在 **Install Operator** 页面中，默认选择以下所需选项：
  - a. 将 Channel 更新为 **stable-4.8**。
  - b. 安装模式 **作为集群中的一个特定命名空间**。
  - 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. 将 Approval Strategy 选为 **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。

如果选择了 **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。

## 后续步骤

- 创建 OpenShift Container Storage 集群。

如需更多信息，请参阅在 [IBM Z 上创建 OpenShift Container Storage 集群](#)。

## 2.2. 安装 LOCAL STORAGE OPERATOR

在本地存储设备上创建 OpenShift Container Storage 集群前，使用此流程从 Operator Hub 安装 Local Storage Operator。

### 流程

1. 登录 OpenShift Web 控制台。
2. 点 **Operators → OperatorHub**。
3. 在 **Filter by keyword...** 框中键入 **local storage** 以从操作器列表中搜索 **Local Storage** operator 并单击它。
4. 点击 **Install**。
5. 在 **Install Operator** 页面中设置以下选项：
  - a. 将 Channel 更新为 **stable-4.8**。
  - b. 安装模式 作为集群中的一个特定命名空间。
  - c. Installed Namespace 为 **Operator recommended namespace openshift-local-storage**。
  - d. 批准策略为 **Automatic**。
6. 点击 **Install**。
7. 验证 Local Storage Operator 是否显示 **Status** 为 **Succeeded**。

## 2.3. 查找可用的存储设备（可选）

此步骤是额外的信息，可以在存储集群创建过程中自动发现磁盘时跳过。在为 IBM Z 创建持久性卷(PV)前，使用 OpenShift Container Storage 标签 **cluster.ocs.openshift.io/openshift-storage="** 标记的三个或更多 worker 节点的设备名称。

### 流程

1. 使用 OpenShift Container Storage 标签列出并验证 worker 节点的名称。

```
$ oc get nodes -l=cluster.ocs.openshift.io/openshift-storage=
```

输出示例：

```
NAME          STATUS  ROLES  AGE   VERSION
bmworker01   Ready  worker 6h45m v1.16.2
bmworker02   Ready  worker 6h45m v1.16.2
bmworker03   Ready  worker 6h45m v1.16.2
```

2. 登录到用于 OpenShift Container Storage 资源的每个 worker 节点，并为每个可用原始块设备查找唯一的 **by-id** 设备名称。

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

输出示例：

```
$ oc debug node/bmworker01
Starting pod/bmworker01-debug ...
To use host binaries, run `chroot /host`
Pod IP: 10.0.135.71
If you don't see a command prompt, try pressing enter.
sh-4.2# chroot /host
sh-4.4# lsblk
NAME                MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop0                7:0  0  500G  0 loop
sda                  8:0  0  120G  0 disk
|-sda1                8:1  0   384M  0 part /boot
`-sda4                8:4  0  119.6G  0 part
`-coreos-luks-root-nocrypt 253:0  0  119.6G  0 dm  /sysroot
sdb                  8:16  0  500G  0 disk
```

在本例中，对于 **bmworker01**，可用的本地设备是 **sdb**。

3. 确定在第 2 步中选择的每个设备的唯一 ID。

```
sh-4.4#ls -l /dev/disk/by-id/ | grep sdb
lrwxrwxrwx. 1 root root 9 Feb  3 16:49 scsi-360050763808104bc2800000000000259 ->
../sdb
lrwxrwxrwx. 1 root root 9 Feb  3 16:49 scsi-SIBM_2145_00e020412f0aXX00 -> ../sdb
lrwxrwxrwx. 1 root root 9 Feb  3 16:49 scsi-0x60050763808104bc2800000000000259 ->
../sdb
```

在上例中，本地设备 **sdb** 的 ID

```
scsi-0x60050763808104bc2800000000000259
```

4. 重复上述步骤，识别所有由 OpenShift Container Storage 使用存储设备其他节点的设备 ID。详情请查看本[知识库文章](#)。

## 2.4. 在 IBM Z 上创建 OPENSIFT CONTAINER STORAGE 集群

使用这个流程在 IBM Z 上创建存储集群。

## 先决条件

- 确保满足 [使用本地存储设备安装 OpenShift Container Storage 的要求](#) 部分中的所有要求。
- 您必须有三个存储类型和大小相同的 worker 节点（例如 200 GB），才能使用 IBM Z 或 LinuxONE 上的本地存储设备。

## 流程

1. 登录 OpenShift Web 控制台。
2. 点 **Operators** → **Installed Operators** 查看所有已安装的 Operator。确保所选的 **Project** 为 **openshift-storage**。
3. 点 **OpenShift Container Storage**。
4. 单击 Storage Cluster 的 **Create Instance** 链接。
5. 为 **Select Mode** 选择 **Internal-Attached** 设备。默认选择 **Internal**。
6. 使用向导创建存储集群，其中包括磁盘发现、存储类创建和存储集群创建。如果还没有安装，系统会提示您安装 Local Storage Operator。点 **Install** 并安装 Operator，如 [Installing Local Storage Operator](#) 所述。

### 发现磁盘

您可以在所选节点上发现一个潜在的可用磁盘列表。发现不使用且可用于置备持久性卷 (PV) 的磁盘和分区。

a. 选择以下任意一项：

- **可发现所有节点中磁盘的所有节点。**
- **选择节点以从列出节点的子集发现磁盘。**  
要查找集群中的特定 worker 节点，您可以根据 Name 或 Label 过滤节点。Name 允许您按节点名称搜索，而 Label 则允许您选择预定义的标签进行搜索。

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。



### 注意

如果要选择的节点有污点且没有在向导中发现，请按照 [红帽知识库解决方案](#) 中提供的步骤作为为 Local Storage Operator 资源添加容忍度的临时解决方案。

b. 点 **Next**。

### 创建存储类

您可以通过过滤一组存储卷来创建专用的存储类来消耗存储。

- a. 输入 **Volume Set Name**。
- b. 输入 **Storage Class Name**。默认情况下，存储类名称会出现卷集名称。
- c. 在上一步中为磁盘发现选择的节点会在 **Filter Disks** 部分显示。选择以下任意一项：

- 选择 **All nodes** 可以选择您发现设备的所有节点。
- 选择 **Select nodes** 可以选择一组节点来发现设备。  
要查找集群中的特定 worker 节点，您可以根据 Name 或 Label 过滤节点。Name 允许您按节点名称搜索，而 Label 则允许您选择预定义的标签进行搜索。

建议 worker 节点分散到三个不同的物理节点、机架或故障域中以实现高可用性。



### 注意

确保 OpenShift Container Storage 机架标签与数据中心中的物理机架一致，以防止在故障域级别出现双节点故障。

- d. 选择所需的 **Disk Type**。可用的选项如下：

All	选择节点上存在的所有磁盘类型。默认情况下会选择这个选项。
SSD/NVME E	仅选择 SSD NVME 类型磁盘。
HDD	仅选择 HDD 类型磁盘。

- e. 在 **Advanced** 部分，您可以设置以下内容：

卷模式	默认会选择块。
磁盘大小	需要包含的设备的最小和最大可用大小。   <b>注意</b> 您必须为该设备设置最小 100GB。
最大磁盘 限制	这表示节点上可以创建的 PV 数量上限。如果此字段留空，则为匹配节点上的所有可用磁盘创建 PV。

- f. （可选）您可以使用 *Select Capacity Chart* 查看所选节点上的磁盘所选容量。  
此图表可能需要几分钟时间来反映上一步中发现的磁盘。

您可以单击图表中的 **Nodes** 和 **Disks** 链接，以调出节点和磁盘列表以查看更多详细信息。

- g. 点 **Next**。
- h. 在消息警报中单击 **Yes** 以确认创建存储类。  
本地卷集和存储类被创建后，无法返回该步骤。

## 创建存储集群

- a. 选择所需的存储类。  
您可能需要等待一两分钟，以便与所选存储类对应的存储节点被填充。与存储类对应的节点会根据从下拉列表中选择存储类显示。

- b. 点 **Next**。
- c. （可选）在 Encryption 部分中，将切换设置为 **Enabled** 以在集群中启用数据加密。安全配置：
- d. 点 **Save**。
- e. 点 **Next** 查看您的存储集群。
- f. 点击 **Create**。  
只有在最少选择了三个节点时，才会启用 **Create** 按钮。创建一个新存储集群，卷数等于附加到每个节点的设备总数。默认配置使用 1 的复制因数。

若要扩展初始集群的容量，请参阅 [扩展存储](#) 指南。

## 验证步骤

请参阅 [验证 OpenShift Container Storage 安装](#)。



## 第 3 章 为内部附加设备模式验证 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

要确定 OpenShift Container 存储是否已成功部署，您可以验证 pod 是否处于 **Running** 状态。

#### 流程

1. 从 OpenShift Web 控制台左侧窗格中，点击 **Workloads → Pods**。
2. 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”](#)。
3. 点 **Running** 和 **Completed** 标签页验证以下 pod 是否处于运行状态并完成状态：

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

组件	对应的 pod
MDS	<b>rook-ceph-mds-ocs-storagecluster-cephfilesystem-*</b>  (2 个 pod 在存储节点间分布)
RGW	<b>rook-ceph-rgw-ocs-storagecluster-cephobjectstore-*</b> (任何存储节点上的 1 个 pod)
CSI	<ul style="list-style-type: none"> <li>● <b>cephfs</b> <ul style="list-style-type: none"> <li>○ <b>csi-cephfsplugin-*</b> (每个 worker 节点上 1 个 pod)</li> <li>○ <b>csi-cephfsplugin-provisioner-*</b> (2 个 pod 在存储节点间分布)</li> </ul> </li> <li>● <b>rbd</b> <ul style="list-style-type: none"> <li>○ <b>csi-rbdplugin-*</b> (每个 worker 节点上 1 个 pod)</li> <li>○ <b>csi-rbdplugin-provisioner-*</b> (2 个 pod 在存储节点间分布)</li> </ul> </li> </ul>
rook-ceph-crashcollector	<b>rook-ceph-crashcollector-*</b>  (每个存储节点上 1 个 pod)
OSD	<ul style="list-style-type: none"> <li>● <b>rook-ceph-osd-*</b> (每个设备 1 个 pod)</li> <li>● <b>rook-ceph-osd-prepare-ocs-device-*</b> (每个设备 1 个 pod)</li> </ul>

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

1. 在 OpenShift Web 控制台左侧窗格中点击 **Storage → Overview**, 然后点击 **Block and File** 选项卡。
2. 在 **Status** 卡中, 验证 *Storage Cluster and 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 OPENSIFT CONTAINER STORAGE 特定的存储类是否存在

验证集群中是否存在存储类：

1. 从 OpenShift Web 控制台左侧窗格中, 点击 **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**
- **ocs-storagecluster-ceph-rgw**

## 第 4 章 卸载 OPENSIFT CONTAINER STORAGE

### 4.1. 以内部附加设备模式卸载 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	值	默认	行为
<code>cleanup-policy</code>	<code>delete</code>	是	Rook 清理物理驱动器和 <b>DataDirHostPath</b>
<code>cleanup-policy</code>	<code>retain</code>	否	Rook <b>不会</b> 清理物理驱动器和 <b>DataDirHostPath</b>
<code>mode</code>	<code>graceful</code>	是	Rook 和 NooBaa <b>暂停</b> 卸载过程，直到管理员/用户移除 PVC 和 OBC
<code>mode</code>	<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 -n openshift-storage annotate storagecluster ocs-storagecluster
uninstall.ocs.openshift.io/cleanup-policy="retain" --overwrite
storagecluster.ocs.openshift.io/ocs-storagecluster annotated
```

```
$ oc -n openshift-storage annotate storagecluster 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。

## 流程

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。这样做会导致系统中出现孤立 PVC 和 OBC。

- a. 使用 OpenShift Container Storage 删除 OpenShift Container Platform 监控堆栈 PVC。  
请参阅 [从 OpenShift Container Storage 中删除监控堆栈](#)
- b. 使用 OpenShift Container Storage 删除 OpenShift Container Platform Registry PVC。  
[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。  
[从 OpenShift Container Storage 中删除集群日志记录 Operator](#)

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
```

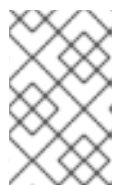
7. 删除命名空间并等待删除完成。如果 **openshift-storage** 是活跃的项目，则需要切换到另一个项目。

例如：

```
$ oc project default
$ oc delete project openshift-storage
```

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. 如果您使用本地存储设备部署了 OpenShift Container Storage，请删除本地存储 Operator 配置。请参阅 [删除本地存储 Operator 配置](#)。

- 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-
```

- 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-
```

- 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>
```

- 删除 Multicloud 对象网关存储类。

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

- 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 cephrobdmirrors.ceph.rook.io --wait=true --timeout=5m
```

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

#### 4.1.1. 删除本地存储 Operator 配置

只有在使用本地存储设备部署了 OpenShift Container Storage 时，才使用本节中的说明。



#### 注意

对于仅使用 **localvolume** 资源的 OpenShift Container Storage 部署，请直接转至第 8 步。

#### 流程

- 标识 **LocalVolumeSet** 以及 OpenShift Container Storage 使用的对应 **StorageClassName**。

```
$ oc get localvolumesets.local.storage.openshift.io -n openshift-local-storage
```

- 将变量 SC 设置为提供 **LocalVolumeSet** 的 **StorageClass**。

```
$ export SC="<StorageClassName>"
```

- 列出并记下稍后要清理的设备。要列出磁盘的设备 ID，请按照这里所述的步骤进行操作，请参阅[查找可用的存储设备](#)。

输出示例：

```
/dev/disk/by-id/scsi-360050763808104bc2800000000000eb
/dev/disk/by-id/scsi-360050763808104bc2800000000000ef
/dev/disk/by-id/scsi-360050763808104bc2800000000000f3
```

- 删除 **LocalVolumeSet**。

```
$ oc delete localvolumesets.local.storage.openshift.io <name-of-volumeset> -n openshift-
local-storage
```

- 删除给定 **StorageClassName** 的本地存储 PV。

```
$ oc get pv | grep $SC | awk '{print $1}' | xargs oc delete pv
```

- 删除 **StorageClassName**。

```
$ oc delete sc $SC
```

- 删除 **LocalVolumeSet** 创建的符号链接。

```
[[ ! -z $SC ]] && 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 rm -rfv
/mnt/local-storage/${SC}; done
```

- 删除 **LocalVolumeDiscovery**。

```
$ oc delete localvolumediscovery.local.storage.openshift.io/auto-discover-devices -n
openshift-local-storage
```

- 删除 **LocalVolume** 资源（如果有）。

使用以下步骤删除在当前或以前的 OpenShift Container Storage 版本中置备 PV 的 **LocalVolume** 资源。此外，确保这些资源不提供给集群上的其他租户使用。

对于每个本地卷，请执行以下操作：

- 标识 **LocalVolume** 以及 OpenShift Container Storage 使用的对应 **StorageClassName**。

```
$ oc get localvolume.local.storage.openshift.io -n openshift-local-storage
```

- 将变量 LV 设置为 LocalVolume 的名称，变量 SC 设置为 StorageClass 的名称  
例如：

```
$ LV=local-block
$ SC=localblock
```

- 列出并记下稍后要清理的设备。

```
$ oc get localvolume -n openshift-local-storage $LV -o jsonpath='{
.spec.storageClassDevices[].devicePaths[]}'
```



输出示例：

```
/dev/sdb
/dev/sdc
/dev/sdd
/dev/sde
```

d. 删除本地卷资源。

```
$ oc delete localvolume -n openshift-local-storage --wait=true $LV
```

e. 删除剩余的 PV 和 StorageClasses（如果存在）。

```
$ oc delete pv -l storage.openshift.com/local-volume-owner-name=${LV} --wait --
timeout=5m
$ oc delete storageclass $SC --wait --timeout=5m
```

f. 从该资源的存储节点中清理工件。

```
$ [[ ! -z $SC ]] && 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 rm -rfv
/mnt/local-storage/${SC}/; done
```

输出示例：

```
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-yyy-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
Starting pod/node-zzz-debug ...
To use host binaries, run `chroot /host`
removed '/mnt/local-storage/localblock/nvme2n1'
removed directory '/mnt/local-storage/localblock'
```

```
Removing debug pod ...
```

10. 分别擦除第 1 和第 8 步中列出的每个本地卷组或本地卷的磁盘，以便可以重复使用它们。

a. 列出存储节点。

```
oc get nodes -l cluster.ocs.openshift.io/openshift-storage=
```

输出示例：

```
NAME          STATUS  ROLES  AGE  VERSION
```

```
node-xxx Ready worker 4h45m v1.18.3+6c42de8
node-yyy Ready worker 4h46m v1.18.3+6c42de8
node-zzz Ready worker 4h45m v1.18.3+6c42de8
```

- b. 获取节点控制台并在出现提示时执行 **chroot /host** 命令。

```
$ oc debug node/node-xxx
Starting pod/node-xxx-debug ...
To use host binaries, run `chroot /host`
Pod IP: w.x.y.z
If you don't see a command prompt, try pressing enter.
sh-4.2# chroot /host
```

- c. 将磁盘路径存储在 **DISKS** 变量中的引号内。有关磁盘路径列表，请查看第 3 步和第 8.c 步以了解本地 volumeset 和本地卷。

输出示例：

```
sh-4.4# DISKS="/dev/disk/by-id/scsi-360050763808104bc2800000000000eb
/dev/disk/by-id/scsi-360050763808104bc2800000000000ef /dev/disk/by-id/scsi-
360050763808104bc2800000000000f3 "
or
sh-4.2# DISKS="/dev/sdb /dev/sdc /dev/sdd /dev/sde ".
```

- d. 在所有 **磁盘上运行 `sgdisk --zap-all`**。

```
sh-4.4# for disk in $DISKS; do sgdisk --zap-all $disk;done
```

输出示例：

```
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
Creating new GPT entries.
GPT data structures destroyed! You may now partition the disk using fdisk or
other utilities.
```

- e. 退出 shell，再对其他节点重复此操作。

```
sh-4.4# exit
exit
sh-4.2# exit
exit
Removing debug pod ...
```

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

例如：

-

```
$ oc project default
$ oc delete project openshift-local-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-local-storage
```

## 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](#)。

### 流程

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

使用这个部分从 OpenShift Container Storage 清理 OpenShift Container Platform registry。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

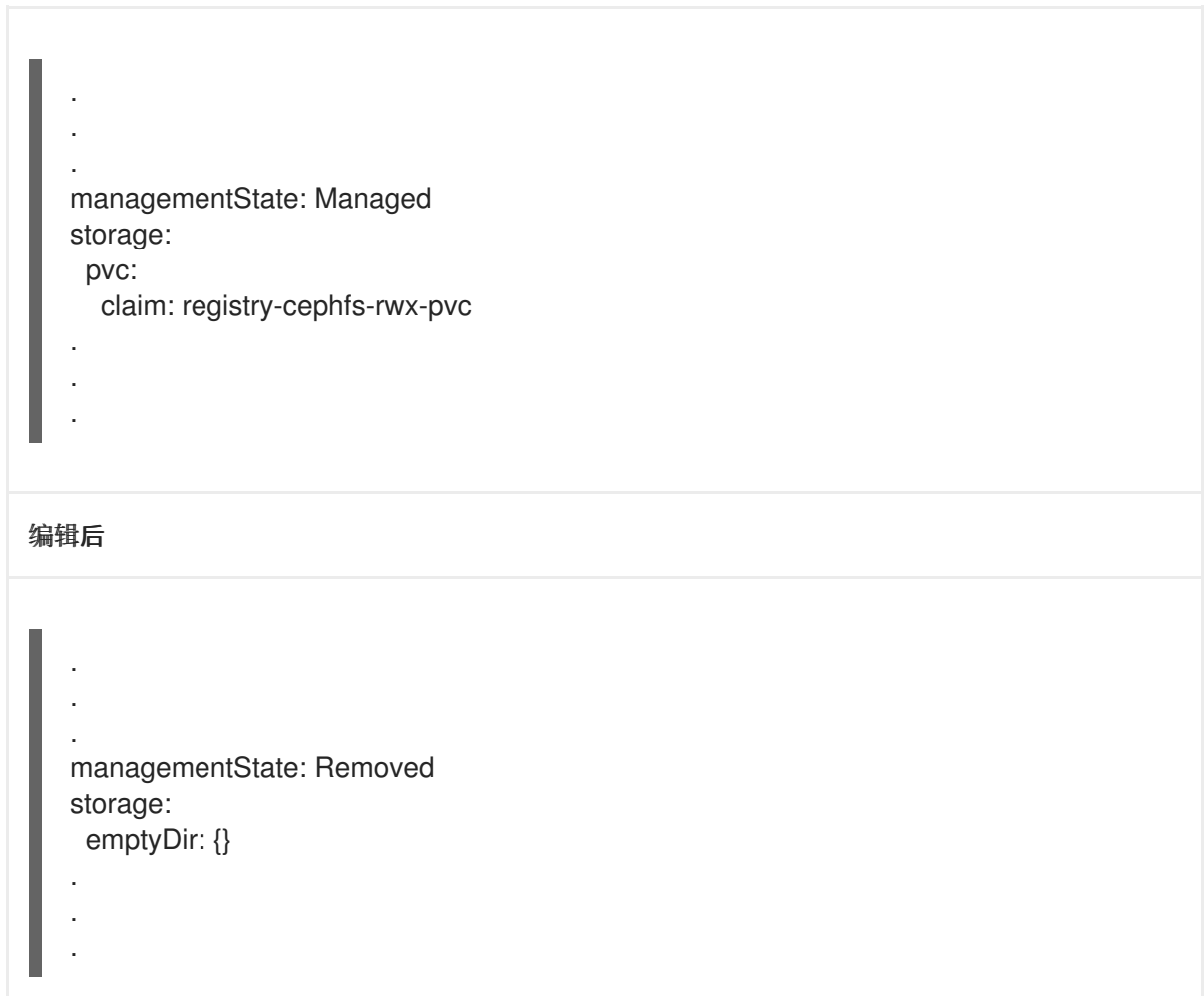
- 镜像 registry 应配置为使用 OpenShift Container Storage PVC。

#### 流程

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

使用本节从 OpenShift Container Storage 清理集群日志记录 Operator。

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

### 先决条件

- 集群日志记录实例应该已配置为使用 OpenShift Container Storage PVC。

### 流程

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
```