



# Red Hat OpenShift Container Storage 4.8

## 使用 Red Hat Virtualization 平台部署 OpenShift Container Storage

如何安装



# Red Hat OpenShift Container Storage 4.8 使用 Red Hat Virtualization 平台部署 OpenShift Container Storage

---

如何安装

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\_Red\_Hat\_Virtualization\_platform.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

有关使用 Red Hat Virtualization 平台安装 Red Hat OpenShift Container Storage 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
<b>第 1 章 准备使用 RED HAT VIRTUALIZATION 平台部署 OPENSIFT CONTAINER STORAGE .....</b>	<b>6</b>
1.1. 在 VAULT 中启用键值后端路径和策略 .....	6
1.2. 使用本地存储设备安装 OPENSIFT CONTAINER STORAGE 的要求 .....	7
<b>第 2 章 使用动态存储设备进行部署 .....</b>	<b>8</b>
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
<b>第 3 章 使用本地存储设备部署 .....</b>	<b>12</b>
3.1. 安装 LOCAL STORAGE OPERATOR .....	12
3.2. 安装 RED HAT OPENSIFT CONTAINER STORAGE OPERATOR .....	12
3.3. 在 RED HAT VIRTUALIZATION 平台上创建 OPENSIFT CONTAINER STORAGE 集群 .....	13
<b>第 4 章 验证 OPENSIFT CONTAINER STORAGE 部署 .....</b>	<b>18</b>
4.1. 验证 POD 的状态 .....	18
4.2. 验证 OPENSIFT CONTAINER STORAGE 集群是否正常运行 .....	19
4.3. 验证 MULTICLOUD 对象网关是否健康 .....	19
4.4. 验证 OPENSIFT CONTAINER STORAGE 特定的存储类是否存在 .....	20
<b>第 5 章 卸载 OPENSIFT CONTAINER STORAGE .....</b>	<b>21</b>
5.1. 在内部模式中卸载 OPENSIFT CONTAINER STORAGE .....	21
5.1.1. 删除本地存储 Operator 配置 .....	27
5.2. 从 OPENSIFT CONTAINER STORAGE 中删除监控堆栈 .....	28
5.3. 从 OPENSIFT CONTAINER STORAGE 中删除 OPENSIFT CONTAINER PLATFORM REGISTRY .....	31
5.4. 从 OPENSIFT CONTAINER STORAGE 中删除集群日志记录 OPERATOR .....	32



## 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 支持在现有 Red Hat OpenShift Container Platform(RHOCP)Red Hat Virtualization Platform 集群上部署。

使用 Red Hat Virtualization 安装程序置备的基础架构 (IPI) 提供的共享存储设备在 OpenShift Container Platform 上部署 OpenShift Container Storage 可让您创建内部集群资源。



### 注意

只有 Red Hat Virtualization 平台支持内部 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](#)。

根据您的要求，执行以下部署方法之一：

- [使用动态存储设备进行部署](#)，以使用动态存储设备部署 OpenShift Container Storage。
- [使用本地存储设备](#) 部署使用本地存储设备部署 OpenShift Container Storage。

# 第 1 章 准备使用 RED HAT VIRTUALIZATION 平台部署 OPENSIFT CONTAINER STORAGE

在使用动态或本地存储开始部署 Red Hat OpenShift Container Storage 前，请确保满足您的资源要求。请参阅 [规划部署](#)。

1. 可选：如果要使用外部密钥管理系统(KMS)启用集群范围加密：
  - 确保存在具有令牌策略的策略，并且启用了 Vault 中的键值后端路径。请参阅在 [vault 中启用键值后端和策略](#)。
  - 确保您在 Vault 服务器上使用签名的证书。
2. 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。
3. 确保 [满足使用本地存储设备安装 OpenShift Container Storage](#) 的要求。

## 1.1. 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
```

## 1.2. 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。

如需更多信息，请参阅规划指南中的[资源要求](#)部分。

### 仲裁程序扩展集群要求 [技术预览]

在这种情况下，单个集群将扩展到两个区域，并有第三个区域作为仲裁者的位置。这是一个技术预览功能，目前用于在 OpenShift Container Platform 内部部署。

有关详细要求和说明，请参阅[为 Metro-DR 扩展集群配置 OpenShift Container Storage](#)。



#### 注意

因为有冲突的扩展逻辑，您无法同时启用灵活的扩展和仲裁程序。通过灵活的扩展，您可以一次向 OpenShift Container Storage 集群添加一个节点。但是，在仲裁集群中，您需要在每个 2 个数据区中添加一个节点。

### 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。

如需更多信息，请参阅规划指南中的[资源要求](#)部分。

## 第 2 章 使用动态存储设备进行部署

使用 Red Hat Virtualization 提供的动态存储设备在 OpenShift Container Platform 上部署 OpenShift Container Storage 可让您选择创建内部集群资源。这会导致在内部置备基础服务，这有助于为应用提供额外的存储类。

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. 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。

- b. 在 **Select Nodes** 部分中，选择至少三个可用节点。
  - c. 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。

## 第 3 章 使用本地存储设备部署

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。这会导致在内部置备基础服务，这有助于为应用提供额外的存储类。

使用本节在已安装 OpenShift Container Platform 的 Red Hat Virtualization 上部署 OpenShift Container Storage。

另外，在执行后续步骤前，请确保您已满足 [准备部署 OpenShift Container Storage](#) 章节的要求。

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

### 3.1. 安装 LOCAL STORAGE OPERATOR

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

#### 前提条件

- 使用具有 cluster-admin 和 Operator 安装权限的账户访问 OpenShift Container Platform 集群。

#### 步骤

1. 登录 OpenShift Web 控制台。
2. 点 **Operators** → **OperatorHub**。
3. 在 **Filter by keyword...** 框中键入 **local storage**，从操作器列表中搜索 **Local Storage operator** 并单击它。
4. 单击 **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-local-storage**。
  - d. 批准策略为 **Automatic**。
6. 单击 **Install**。

#### 验证步骤

- 验证 Local Storage Operator 是否显示 **Status** 为 **Succeeded**。

### 3.2. 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。

## 3.3. 在 RED HAT VIRTUALIZATION 平台上创建 OPENSIFT CONTAINER STORAGE 集群

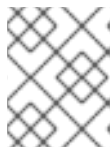
安装 OpenShift Container Storage Operator 后，使用此流程使用本地存储设备创建 OpenShift Container Storage Cluster Service。

## 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](#)。
- 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。

## 步骤

1. 登录 OpenShift Web 控制台。
2. 点 **Operators** → **Installed Operators** 查看所有已安装的 Operator。确保所选的 **Project** 为 **openshift-storage**。
3. 点 Storage Cluster 的 **OpenShift Container Storage** → **Create Instance** 链接。
4. 选择**选择模式** 作为**内部附加设备**。



### 注意

如果还没有安装，系统会提示您安装 Local Storage Operator。点 **Install** 并按照以下步骤进行操作，如 [Installing Local Storage Operator](#) 所述。

5. 发现磁盘
  - 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 以从可用节点的子集发现磁盘。



### 重要

对于使用 arbiter 模式，不要选择 **All nodes** 选项。反之，使用 **Select nodes** 选项从两个数据中心区域中选择附加存储设备的带标记的节点。



### 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。

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。

- b. 点击 **Next**。
6. 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 **本地卷集名称**。

- b. 输入 **Storage Class Name**。默认情况下，存储类名称会出现卷集名称。您还可以更改名称。
- c. 在上一步中为磁盘发现选择的节点显示在 **Filter Disks By** 部分。选择以下任意一项：
  - **Disks on all nodes** 以选择所有节点来发现设备。
  - **Disks on selected nodes** 以选择所有节点中的一部分节点来发现设备。将 worker 节点分布到三个不同的物理节点、机架或故障域以实现高可用性。



### 重要

在创建分布在少于最低要求的 3 个可用区中的带有 3 个或更多节点的存储集群中时，灵活扩展功能会被启用。此功能仅适用于 OpenShift Container Storage 4.7 集群的新部署，且不支持升级的集群。有关灵活扩展的信息，请参阅 [扩展存储指南](#)。

- d. 从可用列表中选择 **SSD/NVME Disk Type**。
- e. 展开 **Advanced** 部分并设置以下选项：

卷模式	默认会选择块。
设备类型	从下拉列表中选择一个或多个磁盘类型。
磁盘大小	为设备设置最小 100GB 大小，以及需要包含的设备的最大可用大小。
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. 点击 **Next**。此时会显示一个用于确认创建新存储类的弹出窗口。
  - g. 单击 **Yes** 以继续。
7. 设置 **容量和节点**
    - a. 选择 **Storage Class**。默认情况下，会选择上一步中创建的新存储类。
    - b. **所选节点** 显示上一步中选择的节点。此列表需要几分钟时间来反映上一步中发现的磁盘。
    - c. 点击 **Next**。
  8. (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. 选择以下 **加密级别** 之一：
      - **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. 选中**连接到外部密钥管理服务**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**。
9. 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**。
10. 点击 **Next**。
11. 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。
12. 点 **Create**。
13. 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

- 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。
- 要验证是否在存储集群中启用了灵活的扩展，请执行以下步骤（对于仲裁模式，请禁用灵活的扩展）：
  1. 在 **Storage Cluster** 标签页中点 **ocs-storagecluster**。
  2. 在 YAML 选项卡中，在 **spec** 部分搜索键 **flexibleScaling**，在 **status** 部分搜索 **failureDomain**。如果 **灵活扩展** 且将 **failureDomain** 设置为 **host**，则启用灵活的扩展功能。

```
spec:  
flexibleScaling: true  
[...]  
status:  
failureDomain: host
```

- 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](#)。

#### 其他资源

- 若要扩展初始集群的容量，请参阅[扩展存储](#)。

## 第 4 章 验证 OPENSIFT CONTAINER STORAGE 部署

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

### 4.1. 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 4.1 “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 4.1. 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></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>noobaa-db-pg-*</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 在不同的 worker 节点上分布)</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 在不同的 worker 节点上分步)</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-deviceset-*</b> (每个设备 1 个 pod)</li> </ul>

## 4.2. 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](#)。

## 4.3. 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](#)。

## 4.4. 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**
  - **ocs-storagecluster-ceph-rgw**



## 第 5 章 卸载 OPENSIFT CONTAINER STORAGE

### 5.1. 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 5.1. `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 <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>
模式	<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
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。更多信息请参阅 [第 5.2 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 5.3 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。更多信息请参阅 [第 5.4 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. 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](#)。

9. 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-
```

10. 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-
```

11. 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>
```

12. 删除 Multicloud 对象网关存储类。

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

13. 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
```

14. 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。
15. 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。

### 5.1.1. 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

对于只使用 **localvolume** 资源的 OpenShift Container Storage 部署，请参阅第 8 步。

#### 步骤

1. 标识 **LocalVolumeSet** 以及 OpenShift Container Storage 使用的对应 **StorageClassName**。
2. 将变量 SC 设置为提供 **LocalVolumeSet** 的 **StorageClass**。

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

3. 删除 **LocalVolumeSet**。

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

4. 删除给定 **StorageClassName** 的本地存储 PV。

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

5. 删除 **StorageClassName**。

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

6. 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
```

7. 删除 **LocalVolumeDiscovery**。

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

8. 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。另外，也要确保这些资源没有被集群中的其他租户使用。

对于每个本地卷，请执行以下操作：

- a. 标识 **LocalVolume** 以及 OpenShift Container Storage 使用的对应 **StorageClassName**。
- b. 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
```

- c. 删除本地卷资源。

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

- d. 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
```

- e. 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 ...
```

## 5.2. 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	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
```

- 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
```

### 5.3. 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

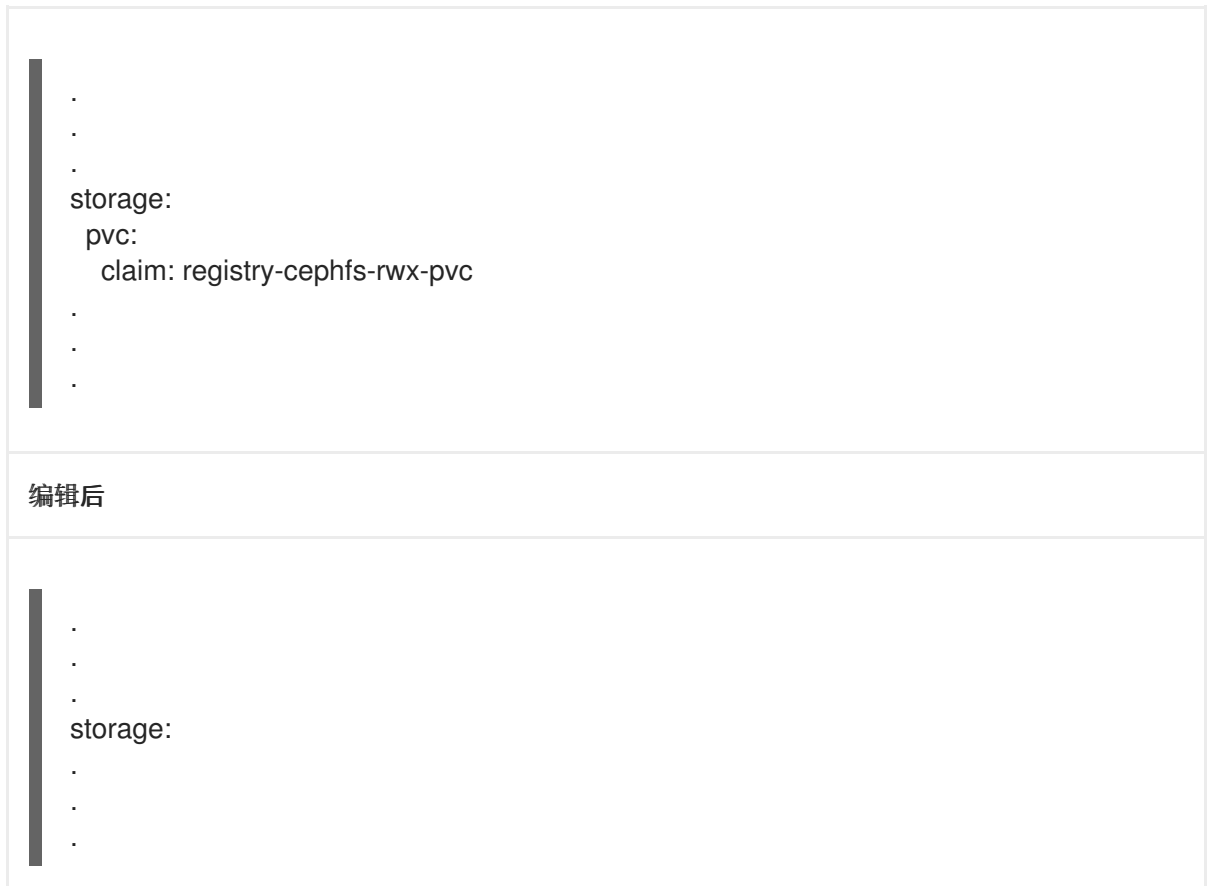
- 镜像 registry 应配置为使用 OpenShift Container Storage PVC。

#### 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
```

## 5.4. 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
```

