



Red Hat OpenShift Container Storage 4.8

使用 IBM Power 系统部署 OpenShift Container Storage

如何安装和设置您的 IBM Power 系统环境

Red Hat OpenShift Container Storage 4.8 使用 IBM Power 系统部署 OpenShift Container Storage

如何安装和设置您的 IBM Power 系统环境

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_Power_Systems.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

有关安装 Red Hat OpenShift Container Storage 4.8 在 IBM Power Systems 上使用本地存储的说明，请参阅本文档。

目录

使开源包含更多	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. 使用本地存储设备安装 OPENSIFT CONTAINER STORAGE 的要求	6
1.2. 在 VAULT 中启用键值后端路径和策略	6
第 2 章 使用本地存储设备部署 OPENSIFT CONTAINER STORAGE	8
2.1. 安装 LOCAL STORAGE OPERATOR	8
2.2. 安装 RED HAT OPENSIFT CONTAINER STORAGE OPERATOR	8
2.3. 查找可用的存储设备	9
2.4. 在 IBM POWER 系统中创建 OPENSIFT CONTAINER STORAGE 集群	11
第 3 章 为内部模式验证 OPENSIFT CONTAINER STORAGE 部署	15
3.1. 验证 POD 的状态	15
3.2. 验证 OPENSIFT CONTAINER STORAGE 集群是否正常运行	16
3.3. 验证 MULTICLOUD 对象网关是否健康	16
3.4. 验证 OPENSIFT CONTAINER STORAGE 特定的存储类是否存在	17
第 4 章 卸载 OPENSIFT CONTAINER STORAGE	18
4.1. 在内部模式中卸载 OPENSIFT CONTAINER STORAGE	18
4.1.1. 删除本地存储 Operator 配置	22
4.2. 从 OPENSIFT CONTAINER STORAGE 中删除监控堆栈	24
4.3. 从 OPENSIFT CONTAINER STORAGE 中删除 OPENSIFT CONTAINER PLATFORM REGISTRY	27
4.4. 从 OPENSIFT CONTAINER STORAGE 中删除集群日志记录 OPERATOR	28

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](#)。

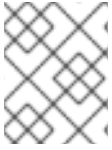
对红帽文档提供反馈

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 Power 集群上部署，以及代理环境的开箱即用支持。



注意

IBM Power 系统上只支持内部 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，请遵循适当的部署过程：

- 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 IBM Power Systems 提供的本地存储设备在 OpenShift Container Platform 上部署 OpenShift Container Storage 可让您创建内部集群资源。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额外的存储类。



注意

IBM Power 系统上只支持内部 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](#)。

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](#)。

- 在外部密钥管理系统(KMS)上，
 - 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。

解决了以上问题后，请按照给出的顺序执行以下步骤：

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

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

节点要求

- 集群必须包含集群中的至少三个 OpenShift Container Platform worker 节点，每个节点都带有本地附加的存储设备。
 - 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。
 - 要使用的设备必须是空的，也就是说，磁盘上应当没有剩余的持久性卷(PV)、卷组(VG)或本地卷(LV)。
- 您必须至少有三个标记的节点。
 - 具有 OpenShift Container Storage 使用的本地存储设备的每个节点都必须具有特定的标签才能部署 OpenShift Container Storage Pod。要标记节点，请使用以下命令：

```
$ oc label nodes <NodeNames> cluster.ocs.openshift.io/openshift-storage="
```

请参阅规划指南中的 [资源要求](#) 部分。

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。

流程

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 使用本地存储设备部署 OPENSIFT CONTAINER STORAGE

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

按照给出的顺序按照以下步骤执行操作：

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

2.1. 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. 单击 **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. 单击 **Install**。
7. 验证 Local Storage Operator 是否显示 **Status** 为 **Succeeded**。

2.2. 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。

有关硬件和软件要求的详情，请参阅 [规划您的部署](#)。

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。
- RHOCP 集群中必须至少有三个工作程序节点。



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 OpenShift Web 控制台左侧窗格中导航，以点击 **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**。
如果选择了 **Automatic** 更新，Operator Lifecycle Manager(OLM)将自动升级 Operator 的正在运行的实例，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。

2.3. 查找可用的存储设备

在为 IBM Power 系统创建 PV 前，使用此流程为您标记了 OpenShift Container Storage 标签 **cluster.ocs.openshift.io/openshift-storage=** 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
worker-0  Ready  worker 2d11h v1.21.1+f36aa36
worker-1  Ready  worker 2d11h v1.21.1+f36aa36
worker-2  Ready  worker 2d11h v1.21.1+f36aa36
```

2. 登录到用于 OpenShift Container Storage 资源的每个 worker 节点，并查找您在部署 OpenShift Container Platform 时附加的额外磁盘的名称。

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

输出示例：

```
$ oc debug node/worker-0
Starting pod/worker-0-debug ...
To use host binaries, run `chroot /host`
Pod IP: 192.168.0.63
If you don't see a command prompt, try pressing enter.
sh-4.4#
sh-4.4# chroot /host
sh-4.4# lsblk
NAME MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
loop1 7:1 0 500G 0 loop
sda 8:0 0 500G 0 disk
sdb 8:16 0 120G 0 disk
|-sdb1 8:17 0 4M 0 part
|-sdb3 8:19 0 384M 0 part
`sdb4 8:20 0 119.6G 0 part
sdc 8:32 0 500G 0 disk
sdd 8:48 0 120G 0 disk
|-sdd1 8:49 0 4M 0 part
|-sdd3 8:51 0 384M 0 part
`sdd4 8:52 0 119.6G 0 part
sde 8:64 0 500G 0 disk
sdf 8:80 0 120G 0 disk
|-sdf1 8:81 0 4M 0 part
|-sdf3 8:83 0 384M 0 part
`sdf4 8:84 0 119.6G 0 part
sdg 8:96 0 500G 0 disk
sdh 8:112 0 120G 0 disk
|-sdh1 8:113 0 4M 0 part
|-sdh3 8:115 0 384M 0 part
`sdh4 8:116 0 119.6G 0 part
sdi 8:128 0 500G 0 disk
sdj 8:144 0 120G 0 disk
|-sdj1 8:145 0 4M 0 part
|-sdj3 8:147 0 384M 0 part
`sdj4 8:148 0 119.6G 0 part
sdk 8:160 0 500G 0 disk
sdl 8:176 0 120G 0 disk
|-sdl1 8:177 0 4M 0 part
|-sdl3 8:179 0 384M 0 part
```

```

`-sdl4 8:180 0 119.6G 0 part /sysroot
sdm 8:192 0 500G 0 disk
sdn 8:208 0 120G 0 disk
|-sdn1 8:209 0 4M 0 part
|-sdn3 8:211 0 384M 0 part /boot
`-sdn4 8:212 0 119.6G 0 part
sdo 8:224 0 500G 0 disk
sdp 8:240 0 120G 0 disk
|-sdp1 8:241 0 4M 0 part
|-sdp3 8:243 0 384M 0 part
`sdp4 8:244 0 119.6G 0 part

```

在本例中，对于 worker-0，可用的 500G 本地设备为 500G、**sdc**、**sde**、**sdg**、**sdi**、**sdk**、**sdm**、**sdo**。

3. 为所有其他 worker 节点重复上述步骤，它们具有要由 OpenShift Container Storage 使用的存储设备。详情请查看本[知识库文章](#)。

2.4. 在 IBM POWER 系统中创建 OPENSIFT CONTAINER STORAGE 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。
- 您必须至少有三个 worker 节点，每个节点的存储类型和大小都相同（例如：200 GB SSD），才能在 IBM Power 系统上使用本地存储设备。
- 验证 OpenShift Container Platform worker 节点是否已标记为 OpenShift Container Storage：

```

oc get nodes -l cluster.ocs.openshift.io/openshift-storage -o jsonpath='{range .items[*]}
{.metadata.name}{"\n"}'

```

要识别每个节点中的存储设备，请参阅 [查找可用的存储设备](#)。

流程

1. 登录 OpenShift Web 控制台。
2. 在 **openshift-local-storage** 命名空间中点击 **Operators → Installed Operators** 来查看安装的 Operator。
3. 单击 **Local Storage installed** 操作器。
4. 在 **Operator Details** 页面中，点 **Local Volume** 链接。
5. 单击 **Create Local Volume**。
6. 点 **YAML 视图** 来配置本地卷。
7. 使用以下 YAML 为块 PV 定义 **LocalVolume** 自定义资源。

```

apiVersion: local.storage.openshift.io/v1
kind: LocalVolume
metadata:
  name: localblock

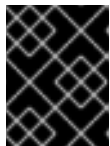
```

```

namespace: openshift-local-storage
spec:
  logLevel: Normal
  managementState: Managed
  nodeSelector:
    nodeSelectorTerms:
      - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
              - worker-0
              - worker-1
              - worker-2
  storageClassDevices:
    - devicePaths:
        - /dev/sda
      storageClassName: localblock
      volumeMode: Block

```

以上定义从 **worker-0**、**worker-1** 和 **worker-2** 节点中选择 **sda** 本地设备。**本地块存储** 类被创建，持久性卷从 **sda** 置备。



重要

根据您的环境，指定 `nodeSelector` 的适当值。在所有 worker 节点上，设备名称都应该相同。您还可以指定多个 `devicePath`。

8. 点击 **Create**。
9. 确认是否已创建 **diskmaker-manager** pod 和持久卷。
 - a. 对于 **Pod**
 - i. 从 OpenShift Web 控制台左侧窗格中，点击 **Workloads → Pods**。
 - ii. 从 **Project** 下拉列表中，选择 **openshift-local-storage**。
 - iii. 检查您在创建 LocalVolume CR 时使用的每个 worker 节点是否有 **diskmaker-manager** pod。
 - b. 对于 **持久性卷**
 - i. 从 OpenShift Web 控制台左侧窗格中，点击 **Storage → PersistentVolumes**。
 - ii. 使用名称 **local-pv-*** 检查持久性卷。持久性卷的数量将等同于 worker 节点的数量和在创建 LocalVolume CR 时置备的存储设备数量。



重要

在创建分布在 3 个或多个可用区的最低要求的存储集群上，启用灵活的扩展功能。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](#)

10. 点击 OpenShift Web 控制台左侧窗格中的 **Operators → Installed Operators** 来查看安装的 Operator。

11. 从 **Project** 下拉列表中，选择 **openshift-storage**。
12. 点 **OpenShift Container Storage installed operator**。
13. 在 **Operator Details** 页面中，点 **Storage Cluster** 链接。
14. 点 **Create Storage Cluster**。
 - a. 为 **Select Mode** 选择 **Internal-Attached** 设备。
 - b. 单击 **Storage** 和 **Nodes**。
 - c. 选择所需的存储类。
 - d. 与存储类对应的节点会根据从下拉菜单中选择的存储类显示。
 - e. 点 **Next**。
 - f. (可选) **设置安全性和网络配置**
 - i. 选中 **启用加密** 复选框，以加密块和文件存储。
 - ii. 选择任意一个或两个 **加密级别**：
 - **用于加密整个集群（块和文件）的集群范围加密**。
 - **存储类加密**，以使用加密启用的存储类创建加密的持久性卷（仅限块）。
 - iii. 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。
 - A. 默认情况下，**Key Management Service Provider** 设置为 **Vault**。
 - B. 输入 **Vault Service Name**、**Vault 服务器的主机地址**（'https://<hostname 或 ip>'）、**端口号**和 **Token**。
 - iv. 展开 **Advanced Settings** 以输入附加设置和证书详情：
 - 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. 输入 **TLS Server Name**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 **Save**。
 - g. 点 **Next**。
 - h. 查看配置详情。若要修改任何配置设置，请单击 **Back** 以返回到上一配置页面。
 - i. 单击 **Create**。

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，您可以单击 **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：
 1. 点 **ocs-storagecluster** in **Storage Cluster** 标签页。
 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  
  [...]br/>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](#) 指南。

第 3 章 为内部模式验证 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"> ● 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 节点上的 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-* (任何存储节点上的 1 个 pod) ● noobaa-endpoint-* (任何存储节点上 1 个 pod)
MON	rook-ceph-mon-* (每个存储节点上的 3 个 pod)
MGR	rook-ceph-mgr-* (任何存储节点上的 1 个 pod)
MDS	rook-ceph-mds-ocs-storagecluster-cephfilesystem-* (2 个 pod 在存储节点上分发)

组件	对应的 pod
RGW	rook-ceph-rgw-ocs-storagecluster-cephobjectstore-* (任何存储节点上的 1 个 pod)
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 在存储节点间分布) ● rbd <ul style="list-style-type: none"> ○ csi-rbdplugin-* (每个 worker 节点上 1 个 pod) ○ csi-rbdplugin-provisioner-* (2 个 pod 在存储节点间分布)
rook-ceph-crashcollector	rook-ceph-crashcollector-* (每个存储节点上的 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

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 MULTICLOUD 对象网关是否健康

1. 点击 OpenShift Web 控制台左侧窗格中的 **Storage → Overview**, 然后点击 **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

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. 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	值	默认	行为
<code>cleanup-policy</code>	<code>delete</code>	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
<code>mode</code>	<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
<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。如需更多信息，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。如需更多信息，请参阅 [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 [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。

- 下面是一个示例脚本，用于标识使用 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-localblock-0-data-1x4r7g-block-dmccrypt (253:1)
```

- c. 删除映射的设备。

```
$ cryptsetup luksClose --debug --verbose ocs-deviceset-localblock-0-data-1x4r7g-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 cephrbdmirrors.ceph.rook.io --wait=true --timeout=5m
```

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。

4.1.1. 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

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

```
$ LV=localblock
$ SC=localblock
```

- 删除本地卷资源。

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

- 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
```

- 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/sda'
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/sda'
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/sda'
removed directory '/mnt/local-storage/localblock'

Removing debug pod ...
```

- 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

- 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-jbjpg       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

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

```

.
.
.
storage:
  pvc:
    claim: registry-cephfs-rwx-pvc
.
.
.

```

编辑后

```

.
.
.
storage:
  emptyDir: {}
.
.
.

```

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
```