



# Red Hat OpenShift Data Foundation 4.12

## 替换节点

有关如何安全替换 OpenShift Data Foundation 集群中的节点的说明。



## Red Hat OpenShift Data Foundation 4.12 替换节点

---

有关如何安全替换 OpenShift Data Foundation 集群中的节点的说明。

## 法律通告

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<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

本文档介绍如何安全地替换 Red Hat OpenShift Data Foundation 集群中的节点。

---

## 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 OPENSIFT DATA FOUNDATION 使用动态设备部署 .....</b>	<b>6</b>
1.1. 在 AWS 上部署的 OPENSIFT DATA FOUNDATION .....	6
1.2. 在 VMWARE 上部署的 OPENSIFT DATA FOUNDATION .....	12
1.3. 在 RED HAT VIRTUALIZATION 上部署的 OPENSIFT DATA FOUNDATION .....	20
1.4. 在 MICROSOFT AZURE 上部署的 OPENSIFT DATA FOUNDATION .....	23
<b>第 2 章 使用本地存储设备部署的 OPENSIFT DATA FOUNDATION .....</b>	<b>27</b>
2.1. 在裸机基础架构中替换存储节点 .....	27
2.2. 替换 IBM Z 或 LINUXONE 基础架构中的存储节点 .....	38
2.3. 替换 IBM POWER 基础架构上的存储节点 .....	45
2.4. 替换 VMWARE 基础架构上的存储节点 .....	50
2.5. 在 RED HAT VIRTUALIZATION 基础架构中替换存储节点 .....	73



---

## 使开源包含更多

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。详情请查看 [CTO Chris Wright 的信息](#)。

## 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。

要提供反馈，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

对于 OpenShift Data Foundation，可以为操作节点主动执行节点替换，并为以下部署针对故障节点主动执行节点：

- 对于 Amazon Web Services(AWS)
  - 用户置备的基础架构
  - 安装程序置备的基础架构
- 对于 VMware
  - 用户置备的基础架构
  - 安装程序置备的基础架构
- Red Hat Virtualization
  - 安装程序置备的基础架构
- 对于 Microsoft Azure
  - 安装程序置备的基础架构
- 用于本地存储设备
  - 裸机
  - VMware
  - Red Hat Virtualization
  - IBM Power
- 有关以外部模式替换存储节点，请参阅 [Red Hat Ceph Storage 文档](#)。

# 第 1 章 OPENSIFT DATA FOUNDATION 使用动态设备部署

## 1.1. 在 AWS 上部署的 OPENSIFT DATA FOUNDATION

- 要替换正常运行的节点，请参阅：
  - [第 1.1.1 节 “在用户置备的基础架构中替换可运行的 AWS 节点”](#)。
  - [第 1.1.2 节 “在安装程序置备的基础架构中替换可正常工作的 AWS 节点”](#)。
- 要替换失败的节点，请参阅：
  - [第 1.1.3 节 “在用户置备的基础架构中替换失败的 AWS 节点”](#)。
  - [第 1.1.4 节 “在安装程序置备的基础架构中替换失败的 AWS 节点”](#)。

### 1.1.1. 在用户置备的基础架构中替换可运行的 AWS 节点

#### 先决条件

- 确保替换节点配置类似的基础架构和资源，以用于被替换的节点。
- 您必须登录到 OpenShift Container Platform 集群。

#### 流程

1. 确定需要替换的节点。
2. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

**<node\_name>**

指定替换的节点的名称。

3. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

4. 删除节点：

```
$ oc delete nodes <node_name>
```

5. 使用所需的基础架构创建一个新的 Amazon Web Service(AWS)机器实例。请参阅 [平台要求](#)。
6. 使用新的 AWS 机器集创建新的 OpenShift Container Platform 节点。
7. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

- 批准新节点所需的所有 OpenShift Container Platform CSR :

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

- 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
- 使用以下之一将 OpenShift Data Foundation 标签应用到新节点 :

#### 从用户界面

- 对于新节点, 点击 **Action Menu( : )** → **Edit Labels**。
- 添加 **cluster.ocs.openshift.io/openshift-storage**, 然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点 :

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

**<new\_node\_name>**

指定新节点的名称。

#### 验证步骤

- 验证输出中是否存在新节点 :

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

- 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态 :

- csi-cephfsplugin-\***
- csi-rbdplugin-\***

- 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
- 验证新的对象存储设备(OSD)pod 是否在替换节点上运行 :

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选 : 如果在集群中启用了集群范围的加密, 请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点, 请执行以下操作 :

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境 :

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.1.2. 在安装程序置备的基础架构中替换可正常工作的 AWS 节点

### 流程

1. 登录 OpenShift Web 控制台，再点击 **Compute** → **Nodes**。
2. 确定需要替换的节点。记录其 **Machine Name**。
3. 将节点标记为不可调度：

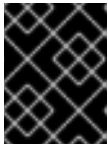
```
$ oc adm cordon <node_name>
```

**<node\_name>**

指定替换的节点的名称。

4. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

5. 点 **Compute** → **Machines**。搜索所需的机器。
6. 除了所需机器外，点击 **Action 菜单(⋮)** → **Delete Machine**。
7. 点 **Delete** 以确认机器已被删除。会自动创建新机器。
8. 等待新计算机启动并过渡到 **Running** 状态。



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。

9. 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
10. 将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮)** → **Edit Labels**。

- b. 添加 `cluster.ocs.openshift.io/openshift-storage`，然后点 **Save**。

### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

### 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d ' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。

4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-device** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

### 1.1.3. 在用户置备的基础架构中替换失败的 AWS 节点

#### 先决条件

- 确保替换节点配置类似的基础架构和资源，以用于被替换的节点。

- 您必须登录到 OpenShift Container Platform 集群。

## 流程

1. 识别您要替换的节点的 Amazon Web Service(AWS)机器实例。
2. 登录 AWS，再终止您确定的 AWS 机器实例。
3. 使用所需基础架构创建新的 AWS 机器集。请参阅 [平台要求](#)。
4. 使用新的 AWS 机器集创建新的 OpenShift Container Platform 节点。
5. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

6. 批准新节点所需的所有 OpenShift Container Platform CSR：

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

7. 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
8. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

### 从用户界面

- a. 对于新节点，点击 **Action Menu**(**:**) → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

### 使用命令行界面

- 执行以下命令，将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：
  - **csi-cephfsplugin-\***
  - **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

#### 1.1.4. 在安装程序置备的基础架构中替换失败的 AWS 节点

##### 流程

1. 登录 OpenShift Web 控制台，再点击 **Compute → Nodes**。
2. 确定有故障的节点，再单击其 **Machine Name**。
3. 点击 **Actions → Edit Annotations**，然后点 **Add More**。
4. 添加 **machine.openshift.io/exclude-node-draining** 并点 **Save**。
5. 点 **Actions → Delete Machine**，然后点 **Delete**。
6. 新机器会自动创建，等待新机器启动。



##### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

7. 点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
8. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

##### 从用户界面

- a. 对于新节点，点击 **Action Menu (⋮) → Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

## 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

9. 可选：如果没有自动删除失败的 Amazon Web Service(AWS)实例，请从 AWS 控制台终止实例。

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。

4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-device** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.2. 在 VMWARE 上部署的 OPENSIFT DATA FOUNDATION

- 要替换正常运行的节点，请参阅：
  - [第 1.2.1 节 “在用户置备的基础架构中替换可运行的 VMware 节点”](#)。
  - [第 1.2.2 节 “在安装程序置备的基础架构中替换可运行的 VMware 节点”](#)。



- 要替换失败的节点，请参阅：
  - [第 1.2.3 节 “在用户置备的基础架构中替换失败的 VMware 节点”](#)。
  - [第 1.2.4 节 “在安装程序置备的基础架构中替换失败的 VMware 节点”](#)。

### 1.2.1. 在用户置备的基础架构中替换可运行的 VMware 节点

#### 先决条件

- 确保替换节点配置类似的基础架构和资源，以用于被替换的节点。
- 您必须登录到 OpenShift Container Platform 集群。

#### 流程

1. 识别需要替换的节点及其虚拟机(VM)。
2. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

**<node\_name>**

指定替换的节点的名称。

3. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



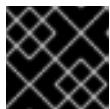
#### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

4. 删除节点：

```
$ oc delete nodes <node_name>
```

5. 登录到 VMware vSphere，终止您识别的虚拟机：



#### 重要

仅从清单中删除虚拟机，而不从磁盘中删除。

6. 在带有所需基础架构的 VMware vSphere 上创建新虚拟机。请参阅 [平台要求](#)。
7. 使用新虚拟机创建新的 OpenShift Container Platform worker 节点。
8. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

- 批准新节点所需的所有 OpenShift Container Platform CSR :

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

- 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。

- 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点 :

#### 从用户界面

- 对于新节点，点击 **Action Menu( ! )** → **Edit Labels**。
- 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点 :

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

**<new\_node\_name>**

指定新节点的名称。

#### 验证步骤

- 验证输出中是否存在新节点 :

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d ' ' -f1
```

- 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态 :

- csi-cephfsplugin-\***
- csi-rbdplugin-\***

- 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。

- 验证新的对象存储设备(OSD)pod 是否在替换节点上运行 :

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选 : 如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。

对于上一步中标识的每个新节点，请执行以下操作 :

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境 :

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.2.2. 在安装程序置备的基础架构中替换可运行的 VMware 节点

### 流程

1. 登录 OpenShift Web 控制台，再点击 **Compute → Nodes**。
2. 确定需要替换的节点。记录其 **Machine Name**。
3. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

**<node\_name>**

指定替换的节点的名称。

4. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

5. 点 **Compute → Machines**。搜索所需的机器。
6. 除了所需机器外，点击 **Action 菜单( ! ) → Delete Machine**。
7. 点 **Delete** 以确认机器已被删除。会自动创建新机器。
8. 等待新计算机启动并过渡到 **Running** 状态。



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。

9. 点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
10. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu( ! ) → Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

## 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。

4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.2.3. 在用户置备的基础架构中替换失败的 VMware 节点

### 先决条件

- 确保替换节点配置类似的基础架构和资源，以用于被替换的节点。
- 您必须登录到 OpenShift Container Platform 集群。

## 流程

1. 找出您要替换的节点及其虚拟机(VM)。
2. 删除节点：

```
$ oc delete nodes <node_name>
```

**<node\_name>**

指定替换的节点的名称。

3. 登录 VMware vSphere 并终止您确定的虚拟机。



### 重要

仅从清单中删除虚拟机，而不从磁盘中删除。

4. 在带有所需基础架构的 VMware vSphere 上创建新虚拟机。请参阅 [平台要求](#)。
5. 使用新虚拟机创建新的 OpenShift Container Platform worker 节点。
6. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

7. 批准新节点所需的所有 OpenShift Container Platform CSR：

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

8. 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
9. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

### 从用户界面

- a. 对于新节点，点击 **Action Menu( ! )** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。

4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.2.4. 在安装程序置备的基础架构中替换失败的 VMware 节点

### 流程

1. 登录 OpenShift Web 控制台，再点击 **Compute** → **Nodes**。
2. 确定有故障的节点，再单击其 **Machine Name**。
3. 点击 **Actions** → **Edit Annotations**，然后点 **Add More**。
4. 添加 **machine.openshift.io/exclude-node-draining** 并点 **Save**。
5. 点 **Actions** → **Delete Machine**，然后点 **Delete**。
6. 会自动创建新机器。等待启动新机器。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

7. 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
8. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮)** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

9. 可选：如果没有自动删除失败的虚拟机(VM)，请从 VMware vSphere 中终止虚拟机。

#### 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.3. 在 RED HAT VIRTUALIZATION 上部署的 OPENSIFT DATA FOUNDATION

### 1.3.1. 在安装程序置备的基础架构中替换可运行的 Red Hat Virtualization 节点

#### 流程

1. 登录 **OpenShift Web 控制台**，再点击 **Compute → Nodes**。
2. 确定需要替换的节点。记录其 **Machine Name**。
3. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

**<node\_name>**

指定替换的节点的名称。

4. 排空节点：

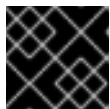
```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

5. 点 **Compute → Machines**。搜索所需的机器。
6. 除了所需机器外，点击 **Action 菜单(⋮) → Delete Machine**。
7. 点 **Delete** 以确认机器已被删除。会自动创建新机器。等待新计算机启动并过渡到 **Running** 状态。



#### 重要

此活动可能至少需要 5 到 10 分钟或更多。

8. 点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
9. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮) → Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面



- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

### 1.3.2. 在安装程序置备的基础架构中替换失败的 Red Hat Virtualization 节点

#### 流程

1. 登录 **OpenShift Web 控制台**，再点击 **Compute** → **Nodes**。
2. 确定出现故障的节点。记录其 **Machine Name**。
3. 删除 Virtual Machine(VM)实例时，确保磁盘不会被删除。

- 登录 **Red Hat Virtualization 管理门户**，并从失败的虚拟机中删除与监控 pod 和对象存储设备(OSD)关联的虚拟磁盘。



### 重要

当您删除一个或多个磁盘时，不要选择 **Remove Perently** 选项。

4. 在 **OpenShift Web 控制台**中，点 **Compute → Machines**。搜索所需的机器。
5. 点 **Actions → Edit Annotations**，然后点 **Add More**。
6. 添加 **machine.openshift.io/exclude-node-draining** 并点 **Save**。
7. 点 **Actions → Delete Machine**，然后点 **Delete**。  
会自动创建新机器。等待新机器启动。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

8. 点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
9. 使用以下任一方法之一将 **OpenShift Data Foundation** 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮) → Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 **OpenShift Data Foundation** 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

10. 可选：如果没有自动删除失败的虚拟机，请从 **Red Hat Virtualization 管理门户**中删除虚拟机。

#### 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads → Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

- 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
- 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-device** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.4. 在 MICROSOFT AZURE 上部署的 OPENSIFT DATA FOUNDATION

### 1.4.1. 替换 Azure 安装程序置备的基础架构上的操作节点

#### 流程

- 登录 OpenShift Web 控制台，再点击 **Compute → Nodes**。
- 确定需要替换的节点。记录其 **Machine Name**。
- 将节点标记为不可调度：

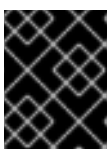
```
$ oc adm cordon <node_name>
```

**<node\_name>**

指定替换的节点的名称。

- 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

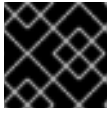


#### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

- 点 **Compute → Machines**。搜索所需的机器。

6. 除了所需的机器外，点击 **Action 菜单(⋮) → Delete Machine**。
7. 点 **Delete** 以确认机器已被删除。会自动创建新机器。
8. 等待新计算机启动并过渡到 **Running** 状态。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。

9. 点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
10. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮) → Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 执行以下命令，将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

#### 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点 **Workloads → Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-devicset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 1.4.2. 在 Azure 安装程序置备的基础架构中替换失败的节点

### 流程

1. 登录 OpenShift Web 控制台，再点击 **Compute → Nodes**。
2. 确定有故障的节点，再单击其 **Machine Name**。
3. 点击 **Actions → Edit Annotations**，然后点 **Add More**。
4. 添加 **machine.openshift.io/exclude-node-draining** 并点 **Save**。
5. 点 **Actions → Delete Machine**，然后点 **Delete**。
6. 会自动创建新机器。等待新机器启动。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

7. 点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
8. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu (⋮) → Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

9. 可选：如果没有自动删除失败的 Azure 实例，从 Azure 控制台终止实例。

### 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d ' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
4. 验证对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 第 2 章 使用本地存储设备部署的 OPENSIFT DATA FOUNDATION

### 2.1. 在裸机基础架构中替换存储节点

- 要替换操作节点，请参阅 [第 2.1.1 节 “在裸机用户置备的基础架构中替换操作节点”](#)。
- 要替换出现故障的节点，请参阅 [第 2.1.2 节 “在裸机用户置备的基础架构中替换失败的节点”](#)。

#### 2.1.1. 在裸机用户置备的基础架构中替换操作节点

##### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

##### 流程

1. 找出该节点，并获取您要替换的节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

2. 识别要替换的节点中运行的监控器 pod（若有）和 OSD：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 缩减上一步中标识的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>  
--replicas=0 -n openshift-storage
```

4. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

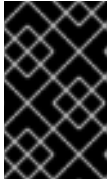
5. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

6. 删除节点：

```
$ oc delete node <node_name>
```

- 获取具有所需基础架构的新裸机计算机。请参阅[在裸机上安装](#)。



### 重要

有关如何在三节点 OpenShift 紧凑的裸机集群中安装 OpenShift Data Foundation 时如何替换 master 节点的详情，请参阅 OpenShift Container Platform 文档中的[备份和恢复指南](#)。

- 使用新的裸机机器创建新的 OpenShift Container Platform 节点。
- 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

- 批准新节点所需的所有 OpenShift Container Platform CSR：

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

- 在 OpenShift Web 控制台中，点 **Compute → Nodes**。确认新节点处于 **Ready** 状态。
- 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- 对于新节点，点击 **Action Menu(⋮) → Edit Labels**。
- 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

- 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```



输出示例：

```
openshift-local-storage
```

14. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，删除了 **server3.example.com**，**newnode.example.com** 是新节点。

- b. 确定要编辑的 **localVolumeSet**：

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
```

```
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

15. 验证新的 **localblock** 持久性卷 (PV) 是否可用：

```
$oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

16. 进入 **openshift-storage** 项目：

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

17. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

18. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

19. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



## 重要

如果 **ocs-osd-removal-job** 失败，且 pod 处于预期的 **Completed** 状态，请检查 pod 日志以进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

20. 识别与持久性卷声明 (PVC) 关联的持久性卷 (PV)：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

21. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

22. 删除 **ocs-osd-removal-job**：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

- 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

- 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的 incremental **mon**，且处于 **Running** 状态：

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

```
rook-ceph-mon-a-cd575c89b-b6k66    2/2    Running
0      38m
rook-ceph-mon-b-6776bc469b-tzzt8    2/2    Running
0      38m
rook-ceph-mon-d-5ff5d488b5-7v8xh    2/2    Running
0      4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

- 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.1.2. 在裸机用户置备的基础架构中替换失败的节点

### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

## 流程

1. 找出该节点，并获取您要替换的节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

2. 识别要替换的节点中运行的监控器 pod（若有）和 OSD：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 缩减上一步中标识的 pod 的部署：

例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

5. 删除处于 **Terminating** 状态的 pod：

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

6. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. 删除节点：

```
$ oc delete node <node_name>
```

8. 获取具有所需基础架构的新裸机计算机。请参阅[在裸机上安装](#)。



### 重要

有关如何在三节点 OpenShift 紧凑的裸机集群中安装 OpenShift Data Foundation 时如何替换 master 节点的详情，请参阅 OpenShift Container Platform 文档中的[备份和恢复](#)指南。

9. 使用新的裸机机器创建新的 OpenShift Container Platform 节点。

10. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

11. 批准新节点所需的所有 OpenShift Container Platform CSR :

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

12. 在 OpenShift Web 控制台中，点 **Compute** → **Nodes**，确认新节点处于 **Ready** 状态。
13. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu**(**:**) → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

14. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

15. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
```

```
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，删除了 **server3.example.com**，**newnode.example.com** 是新节点。

- b. 确定要编辑的 **localVolumeSet** :

```
# oc get -n $local_storage_project localvolumeset
```

输出示例 :

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点，并删除故障节点 :

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例 :

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

16. 验证新的 **localblock** 持久性卷 (PV) 是否可用 :

```
$oc get pv | grep localblock | grep Available
```

输出示例 :

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. 进入 **openshift-storage** 项目：

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

18. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

#### <failed\_osd\_id>

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如 **FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

19. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

#### 重要

如果 **ocs-osd-removal-job** 失败，且 pod 处于预期的 **Completed** 状态，请检查 pod 日志以进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

21. 识别与持久性卷声明 (PVC) 关联的持久性卷 (PV)：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：



```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

## 22. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

## 23. 删除 **ocs-osd-removal-job**：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

### 1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

### 2. 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

### 3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的 incremental **mon**，且处于 **Running** 状态：

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

■

```
rook-ceph-mon-a-cd575c89b-b6k66    2/2    Running
0      38m
rook-ceph-mon-b-6776bc469b-tzzt8    2/2    Running
0      38m
rook-ceph-mon-d-5ff5d488b5-7v8xh    2/2    Running
0      4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

- 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceSet** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.2. 替换 IBM Z 或 LINUXONE 基础架构中的存储节点

您可以选择以下步骤之一来替换存储节点：

- [第 2.2.1 节 “替换 IBM Z 或 LinuxONE 基础架构上的操作节点”](#)。
- [第 2.2.2 节 “在 IBM Z 或 LinuxONE 基础架构中替换失败的节点”](#)。

### 2.2.1. 替换 IBM Z 或 LinuxONE 基础架构上的操作节点

使用这个流程替换 IBM zSystem 或 LinuxONE 基础架构上的操作节点。

#### 流程

- 识别节点，并在要替换的节点上获取标签。记录机架标签。

```
$ oc get nodes --show-labels | grep <node_name>
```

- 识别要替换的节点中运行的 mon（如果有）和对象存储设备(OSD)pod。

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- 缩减上一步中确定的容器集部署。

例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

- 将节点标记为不可调度。

```
$ oc adm cordon <node_name>
```

- 删除处于 Terminating 状态的 pod。

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -
n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

- 排空节点。

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

- 删除节点。

```
$ oc delete node <node_name>
```

- 获取一个新的 IBM zSystem 存储节点作为替换。

- 检查与处于 **Pending** 状态的 OpenShift Data Foundation 相关的证书签名请求 (CSR)：

```
$ oc get csr
```

- 批准新节点所需的所有 OpenShift Data Foundation CSR：

```
$ oc adm certificate approve <Certificate_Name>
```

- 在 OpenShift Web 控制台中点 **Compute** → **Nodes**，确认新节点是否处于 **Ready** 状态。

- 使用以下任一方法之一将 **openshift-storage** 标签应用到新节点：

#### 从用户界面

- 对于新节点，点击 **Action Menu( ! )** → **Edit Labels**
- 添加 **cluster.ocs.openshift.io/openshift-storage** 并点 **Save**。

#### 使用命令行界面

- 执行以下命令，将 OpenShift Data Foundation 标签应用到新节点：

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

- 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点并删除失败的节点。

```
# oc edit -n local-storage-project localvolumediscovery auto-discover-devices
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在上例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

- b. 决定要编辑的 **localVolumeSet**。

在以下命令中，将 *local-storage-project* 替换为您的本地存储项目的名称。默认项目名称是 OpenShift Data Foundation 4.6 及之后的版本中的 **openshift-local-storage**。之前的版本默认使用 **local-storage**。

```
# oc get -n local-storage-project localvolumeset
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点并删除失败的节点。

```
# oc edit -n local-storage-project localvolumeset localblock
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在上例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

14. 验证新的 **localblock** PV 是否可用。

```
$ oc get pv | grep localblock
          CAPA- ACCESS RECLAIM          STORAGE
NAME     CITY MODES POLICY STATUS  CLAIM          CLASS  AGE
local-pv- 931Gi RWO  Delete Bound   openshift-storage/ localblock 25h
```

```

3e8964d3                ocs-deviceset-2-0
                        -79j94
local-pv- 931Gi RWO Delete Bound   openshift-storage/ localblock 25h
414755e0                ocs-deviceset-1-0
                        -959rp
local-pv- 931Gi RWO Delete Available localblock 3m24s b481410
local-pv- 931Gi RWO Delete Bound   openshift-storage/ localblock 25h
d9c5cbd6                ocs-deviceset-0-0
                        -nvs68

```

15. 更改到 **openshift-storage** 项目。

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

16. 从集群移除出现故障的 OSD。如果需要，您可以指定多个故障 OSD。

- a. 根据需要识别 PVC，我们需要删除与该特定 PVC 关联的 PV。

```

$ osd_id_to_remove=1
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd-${osd_id_to_remove} |
grep ceph.rook.io/pvc

```

其中，**osd\_id\_to\_remove** 是 pod 名称中紧接在 **rook-ceph-osd-prefix** 后面的整数。在本例中，部署名称为 **rook-ceph-osd-1**。

输出示例：

```

ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc

```

在本例中，PVC 名称为 **ocs-deviceset-localblock-0-data-0-g2mmc**。

- b. 从集群移除出现故障的 OSD。

```

$ oc process -n openshift-storage ocs-osd-removal -p
FAILED_OSD_IDS=${osd_id_to_remove} |oc create -f -

```

您可以通过在命令中添加逗号分隔的 OSD ID 来移除多个 OSD。（例如：**FAILED\_OSD\_IDS=0,1,2**）

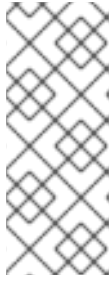


### 警告

这一步会导致 OSD 完全从集群中移除。确保提供了 **osd\_id\_to\_remove** 的正确值。

17. 通过检查 **ocs-osd-removal** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-osd_id_to_remove -n openshift-storage
```



## 注意

如果 **ocs-osd-removal** 失败且 pod 不处于预期的 **Completed** 状态，请检查 pod 日志以进一步调试。例如：

```
# oc logs -l job-name=ocs-osd-removal-osd_id_to_remove -n openshift-storage --tail=-1
```

可能需要手动清理移除的 OSD，如下所示：

```
ceph osd crush remove osd.osd_id_to_remove
ceph osd rm osd_id_to_remove
ceph auth del osd.osd_id_to_remove
ceph osd crush rm osd_id_to_remove
```

### 18. 删除与故障节点关联的 PV。

#### a. 标识与 PVC 关联的 PV。

PVC 名称必须与从集群中删除故障 OSD 时获取的名称相同。

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-
localblock-0-data-0-g2mmc localblock 24h worker-0
```

#### b. 如果有一个 PV 处于 **Released** 状态，请将其删除。

```
# oc delete pv <persistent-volume>
```

例如：

```
# oc delete pv local-pv-5c9b8982
persistentvolume "local-pv-5c9b8982" deleted
```

### 19. 确定 **crashcollector** pod 部署。

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=
<failed_node_name> -n openshift-storage
```

如果已存在 **crashcollector** pod 部署，请将其删除。

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=
<failed_node_name> -n openshift-storage
```

### 20. 删除 **ocs-osd-removal** 任务。

```
# oc delete job ocs-osd-removal-${osd_id_to_remove}
```

输出示例：

```
job.batch "ocs-osd-removal-0" deleted
```

流程

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。

4. 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage| egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了数据加密，请验证新 OSD 设备是否已加密。

对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.2.2. 在 IBM Z 或 LinuxONE 基础架构中替换失败的节点

### 流程

1. 登录 OpenShift Web 控制台，再点击 **Compute** → **Nodes**。
2. 确定有故障的节点，再单击其 **Machine Name**。
3. 点击 **Actions** → **Edit Annotations**，然后点 **Add More**。
4. 添加 **machine.openshift.io/exclude-node-draining** 并点 **Save**。
5. 点 **Actions** → **Delete Machine**，然后点 **Delete**。
6. 会自动创建新机器。等待新机器启动。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。Ceph 周期内生成的 Ceph 错误是临时的，当您标记新节点并且可以正常工作时，会自动解决。

- 点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
- 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- 对于新节点，点击 **Action Menu(⋮)** → **Edit Labels**。
- 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

#### 验证步骤

- 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

- 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- csi-cephfsplugin-\***
- csi-rbdplugin-\***

- 验证所有其他所需的 OpenShift Data Foundation pod 是否处于 **Running** 状态。
- 验证新的对象存储设备(OSD)pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了数据加密，请验证新 OSD 设备是否已加密。  
对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。



## 2.3. 替换 IBM POWER 基础架构上的存储节点

对于 OpenShift Data Foundation，您可以主动对一个操作节点执行节点替换，并主动进行与 IBM Power 相关的部署。

### 2.3.1. 替换 IBM Power 上的操作或失败的存储节点

#### 先决条件

- 确保替换节点配置类似的基础架构和资源，以用于被替换的节点。
- 您必须登录到 OpenShift Container Platform 集群。

#### 流程

1. 找出该节点，并获取您要替换的节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

2. 识别在需要替换的节点中运行的 **mon**（若有）和 Object Storage Device(OSD)pod：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 缩减上一步中标识的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-a --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-1 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

5. 删除处于 **Terminating** 状态的 pod：

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n "$1 " delete pods "$2 " --grace-period=0 " " --force ")}'
```

6. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. 删除节点：

```
$ oc delete node <node_name>
```

8. 使用所需基础架构获取新的 IBM Power 机器。请参阅 [在 IBM Power 上安装集群](#)。
9. 使用新的 IBM Power 机器创建新的 OpenShift Container Platform 节点。
10. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

11. 批准新节点所需的所有 OpenShift Container Platform CSR：

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

12. 在 OpenShift Web 控制台中，点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
13. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu( ! )** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- a. 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

14. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

15. 将新添加的 worker 节点添加到 **localVolume** 中。

- a. 确定您需要编辑的 **localVolume** :

```
# oc get -n $local_storage_project localvolume
```

输出示例 :

```
NAME      AGE
localblock 25h
```

- b. 更新 **localVolume** 定义, 使其包含新节点, 并删除故障节点 :

```
# oc edit -n $local_storage_project localvolume localblock
```

输出示例 :

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
  operator: In
  values:
  #- worker-0
  - worker-1
  - worker-2
  - worker-3
[...]
```

记住在退出编辑器之前进行保存。

在本例中, **worker-0** 被删除, **worker-3** 是新节点。

16. 验证新的 **localblock** 持久性卷 (PV) 是否可用 :

```
$ oc get pv | grep localblock
```

输出示例 :

```
NAME          CAPACITY  ACCESSMODES  RECLAIMPOLICY  STATUS  CLAIM
STORAGECLASS  AGE
local-pv-3e8964d3  500Gi  RWO          Delete         Bound   ocs-deviceset-localblock-2-
data-0-mdbg9  localblock  25h
local-pv-414755e0  500Gi  RWO          Delete         Bound   ocs-deviceset-localblock-1-
data-0-4cslf  localblock  25h
local-pv-b481410  500Gi  RWO          Delete         Available
localblock     3m24s
local-pv-5c9b8982  500Gi  RWO          Delete         Bound   ocs-deviceset-localblock-0-
data-0-g2mmc  localblock  25h
```

17. 进入 **openshift-storage** 项目 :

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

18. 从集群移除出现故障的 OSD。如果需要，您可以指定多个故障 OSD。

a. 识别 PVC：

```
$ osd_id_to_remove=1
```

```
$ oc get -n openshift-storage -o yaml deployment rook-ceph-osd- $\{<osd\_id\_to\_remove>\}$  | grep ceph.rook.io/pvc
```

其中，**<osd\_id\_to\_remove>** 是 **rook-ceph-osd** 前缀后的 pod 名字中的整数。

在本例中，部署名称为 **rook-ceph-osd-1**。

输出示例：

```
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
ceph.rook.io/pvc: ocs-deviceset-localblock-0-data-0-g2mmc
```

b. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS= $<failed\_osd\_id>$  -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如 **FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。



### 警告

此步骤的结果是从集群中完全移除 OSD。确保提供了 **osd\_id\_to\_remove** 的正确值。

19. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态 **Completed** 确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



## 重要

如果 **ocs-osd-removal-job** 失败，并且 pod 不在预期的 **Completed** 状态，请检查 pod 日志以进一步调试。

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

### 21. 删除与故障节点关联的 PV。

#### a. 识别与 PVC 关联的 PV：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-5c9b8982 500Gi RWO Delete Released openshift-storage/ocs-deviceset-
localblock-0-data-0-g2mmc localblock 24h worker-0
```

PVC 名称必须与从集群中删除故障 OSD 时获取的名称相同。

#### b. 如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-5c9b8982
```

输出示例：

```
persistentvolume "local-pv-5c9b8982" deleted
```

### 22. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

### 23. 删除 **ocs-osd-removal-job**：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点击 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的 incremental **mon**，且处于 **Running** 状态：

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

```
rook-ceph-mon-b-74f6dc9dd6-4llzq          1/1   Running   0        6h14m
rook-ceph-mon-c-74948755c-h7wtx          1/1   Running   0        4h24m
rook-ceph-mon-d-598f69869b-4bv49        1/1   Running   0        162m
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

4. 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-device** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.4. 替换 VMWARE 基础架构上的存储节点

- 要替换正常运行的节点，请参阅：
  - [第 2.4.1 节“在 VMware 用户置备的基础架构上替换操作节点”](#)。
  - [第 2.4.2 节“在 VMware 安装程序置备的基础架构中替换操作节点”](#)。

- 要替换失败的节点，请参阅：
  - [第 2.4.3 节“在 VMware 用户置备的基础架构中替换失败的节点”](#)。
  - [第 2.4.4 节“在 VMware 安装程序置备的基础架构中替换失败的节点”](#)。

### 2.4.1. 在 VMware 用户置备的基础架构上替换操作节点

#### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

#### 流程

1. 找出该节点，并获取您要替换的节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

2. 识别要替换的节点中运行的监控器 pod（若有）和 OSD：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 缩减上一步中标识的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

5. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

6. 删除节点：

```
$ oc delete node <node_name>
```

7. 登录到 VMware vSphere 并终止您已识别的虚拟机(VM)。
8. 在带有所需基础架构的 VMware vSphere 上创建新虚拟机。请参阅[基础架构要求](#)。

9. 使用新虚拟机创建新的 OpenShift Container Platform worker 节点。
10. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

11. 批准新节点所需的所有 OpenShift Container Platform CSR：

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

12. 在 OpenShift Web 控制台中，点 **Compute** → **Nodes**，确认新节点处于 **Ready** 状态。
13. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(!)** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

14. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

15. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。
  - a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```



输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，删除了 **server3.example.com**，**newnode.example.com** 是新节点。

- b. 确定要编辑的 **localVolumeSet**：

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

16. 验证新的 **localblock** 持久性卷 (PV) 是否可用：

```
$oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. 进入 **openshift-storage** 项目：

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

18. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

19. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

### 重要

如果 **ocs-osd-removal-job** 失败，且 pod 处于预期的 **Completed** 状态，请检查 pod 日志以进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

21. 识别与持久性卷声明 (PVC) 关联的持久性卷 (PV)：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

## 22. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

## 23. 删除 **ocs-osd-removal-job**：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

### 1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

### 2. 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

### 3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的 incremental **mon**，且处于 **Running** 状态：

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

■

```
rook-ceph-mon-a-cd575c89b-b6k66    2/2    Running
0      38m
rook-ceph-mon-b-6776bc469b-tzzt8    2/2    Running
0      38m
rook-ceph-mon-d-5ff5d488b5-7v8xh    2/2    Running
0      4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

- 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-device** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.4.2. 在 VMware 安装程序置备的基础架构中替换操作节点

### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

### 流程

- 登录 OpenShift Web 控制台，再点 **Compute** → **Nodes**。
- 确定需要替换的节点。记录其 **Machine Name**。
- 获取节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

- 识别在节点上运行的 **mon**（若有）和对象存储设备 (OSD) pod：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. 缩减您在上一步中确定的 pod 的部署：

例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name>
--replicas=0 -n openshift-storage
```

6. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

7. 排空节点：

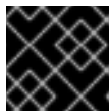
```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

8. 点 **Compute** → **Machines**。搜索所需的机器。

9. 除了所需机器外，点击 **Action 菜单( ! )** → **Delete Machine**。

10. 单击 **Delete** 以确认删除机器。会自动创建新机器。

11. 等待新计算机启动并过渡到 **Running** 状态。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。

12. 在 OpenShift Web 控制台中，点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
13. 向节点物理地添加新设备。
14. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu( ! )** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

15. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

16. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点并删除失败的节点。

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，删除了 **server3.example.com**，**newnode.example.com** 是新节点。

- b. 确定您需要编辑的 **localVolumeSet**：

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点并删除失败的节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，删除了 **server3.example.com**，**newnode.example.com** 是新节点。

17. 验证新的 **localblock** 持久性卷 (PV) 是否可用：

```
$ oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

18. 进入 **openshift-storage** 项目：

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

19. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

20. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

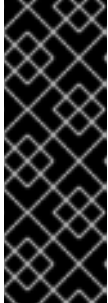
```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



### 重要

如果 **ocs-osd-removal-job** 失败并且 pod 不在预期的 **Completed** 状态，请检查 pod 日志来进一步调试。

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

22. 识别与持久性卷声明(PVC)关联的 PV：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

23. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

24. 删除 **ocs-osd-removal-job**：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```



输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

### 验证步骤

1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的增量 **mon**，并处于 **Running** 状态。

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

```
rook-ceph-mon-a-cd575c89b-b6k66      2/2   Running
0          38m
rook-ceph-mon-b-6776bc469b-tzxt8     2/2   Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh     2/2   Running
0          4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

4. 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceSet** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

### 2.4.3. 在 VMware 用户置备的基础架构中替换失败的节点

#### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

#### 流程

1. 找出该节点，并获取您要替换的节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

2. 识别要替换的节点中运行的监控器 pod（若有）和 OSD：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

3. 缩减上一步中标识的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

4. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

5. 删除处于 **Terminating** 状态的 pod：

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

6. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

7. 删除节点：

```
$ oc delete node <node_name>
```

8. 登录到 VMware vSphere 并终止您已识别的虚拟机(VM)。
9. 在带有所需基础架构的 VMware vSphere 上创建新虚拟机。请参阅[基础架构要求](#)。

10. 使用新虚拟机创建新的 OpenShift Container Platform worker 节点。
11. 检查与处于 **Pending** 状态的 OpenShift Container Platform 相关的证书签名请求(CSR)：

```
$ oc get csr
```

12. 批准新节点所需的所有 OpenShift Container Platform CSR：

```
$ oc adm certificate approve <certificate_name>
```

**<certificate\_name>**

指定 CSR 的名称。

13. 在 OpenShift Web 控制台中，点 **Compute** → **Nodes**，确认新节点处于 **Ready** 状态。
14. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(!)** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

15. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

16. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。
  - a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，删除了 **server3.example.com**，**newnode.example.com** 是新节点。

- b. 确定要编辑的 **localVolumeSet**：

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

17. 验证新的 **localblock** 持久性卷 (PV) 是否可用：

```
$oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

18. 进入 **openshift-storage** 项目：

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

19. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

20. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

### 重要

如果 **ocs-osd-removal-job** 失败，且 pod 处于预期的 **Completed** 状态，请检查 pod 日志以进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

22. 识别与持久性卷声明 (PVC) 关联的持久性卷 (PV)：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

### 23. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

### 24. 删除 **ocs-osd-removal-job**：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

#### 1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

#### 2. 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

#### 3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的 incremental **mon**，且处于 **Running** 状态：

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

■

```
rook-ceph-mon-a-cd575c89b-b6k66    2/2    Running
0      38m
rook-ceph-mon-b-6776bc469b-tzzt8    2/2    Running
0      38m
rook-ceph-mon-d-5ff5d488b5-7v8xh    2/2    Running
0      4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

- 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-device** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。

#### 2.4.4. 在 VMware 安装程序置备的基础架构中替换失败的节点

##### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

##### 流程

- 登录 OpenShift Web 控制台，再点 **Compute** → **Nodes**。
- 确定需要替换的节点。记录其 **Machine Name**。
- 获取节点上的标签：

```
$ oc get nodes --show-labels | grep _<node_name>_
```

**<node\_name>**

指定替换的节点的名称。

- 识别在节点上运行的 **mon**（若有）和对象存储设备 (OSD) pod：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>_
```

5. 缩减上一步中标识的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

6. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

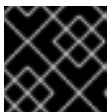
7. 删除处于 **Terminating** 状态的 pod：

```
$ oc get pods -A -o wide | grep -i <node_name>_ | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

8. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

9. 点 **Compute** → **Machines**。搜索所需的机器。
10. 除了所需机器外，点击 **Action 菜单(⋮)** → **Delete Machine**。
11. 点 **Delete** 以确认机器已被删除。会自动创建新机器。
12. 等待新计算机启动并过渡到 **Running** 状态。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。

13. 在 OpenShift Web 控制台中，点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
14. 向节点物理地添加新设备。
15. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮)** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

■



```
$ oc label node _<new_node_name>_ cluster.ocs.openshift.io/openshift-storage=""
```

**<new\_node\_name>**

指定新节点的名称。

16. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

17. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- **newnode.example.com**
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

- b. 确定您需要编辑的 **localVolumeSet**。

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点并删除失败的节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- **newnode.example.com**
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

18. 验证新的 **localblock** PV 是否可用：

```
$ oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock      26s
```

19. 进入 **openshift-storage** 项目：

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

20. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

21. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态为 **Completed**，确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

22. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



### 重要

如果 **ocs-osd-removal-job** 失败并且 pod 不在预期的 **Completed** 状态，请检查 pod 日志来进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

23. 识别与持久性卷声明(PVC)关联的 PV：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-d6bf175b 1490Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h compute-1
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv _<persistent_volume>_
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d9c5cbd6" deleted
```

24. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=_<failed_node_name>_ -n openshift-storage
```

如果有现有的 **crashcollector** pod 部署，请删除它：

■

```
$ oc delete deployment --selector=app=rook-ceph-
crashcollector,node_name=<failed_node_name> -n openshift-storage
```

## 25. 删除 **ocs-osd-removal-job** :

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例 :

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

### 1. 验证输出中是否存在新节点 :

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

### 2. 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态 :

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

### 3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的 incremental **mon**，且处于 **Running** 状态 :

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例 :

```
rook-ceph-mon-a-cd575c89b-b6k66      2/2   Running
0          38m
rook-ceph-mon-b-6776bc469b-tzxt8     2/2   Running
0          38m
rook-ceph-mon-d-5ff5d488b5-7v8xh     2/2   Running
0          4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

### 4. 验证新 OSD pod 是否在替换节点上运行 :

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

### 5. 可选 : 如果在集群中启用了集群范围的加密, 请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点, 请执行以下操作 :

#### a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境 :

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

```
$ chroot /host
```

- b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.5. 在 RED HAT VIRTUALIZATION 基础架构中替换存储节点

- 要替换操作节点，请参阅 [第 2.5.1 节“在 Red Hat Virtualization 安装程序置备的基础架构中替换操作节点”](#)。
- 要替换出现故障的节点，请参阅 [第 2.5.2 节“在 Red Hat Virtualization 安装程序置备的基础架构中替换失败的节点”](#)。

### 2.5.1. 在 Red Hat Virtualization 安装程序置备的基础架构中替换操作节点

#### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

#### 流程

1. 登录 OpenShift Web 控制台，再点 **Compute** → **Nodes**。
2. 确定需要替换的节点。记录其 **Machine Name**。
3. 获取节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

**<node\_name>**

指定替换的节点的名称。

4. 识别在节点上运行的 **mon**（若有）和对象存储设备 (OSD) pod：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

5. 缩减您在上一步中确定的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

6. 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

7. 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

8. 点 **Compute** → **Machines**。搜索所需的机器。
9. 除了所需机器外，点击 **Action 菜单(⋮)** → **Delete Machine**。
10. 点 **Delete** 以确认机器已被删除。会自动创建新机器。等待新计算机启动并过渡到 **Running** 状态。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。

11. 在 OpenShift Web 控制台中点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
12. 物理向节点添加一个或多个新设备。
13. 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- a. 对于新节点，点击 **Action Menu(⋮)** → **Edit Labels**。
- b. 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

**<new\_node\_name>**

指定新节点的名称。

14. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

15. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

- b. 确定您需要编辑的 **localVolumeSet**：

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点并删除失败的节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

16. 验证新的 **localblock** 持久性卷 (PV) 是否可用：

```
$ oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

17. 进入 **openshift-storage** 项目：

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

18. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

19. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态 **Completed** 确认 OSD 移除作业已成功。

```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

20. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```

### 重要

如果 **ocs-osd-removal-job** 失败并且 pod 不在预期的 **Completed** 状态，请检查 pod 日志来进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage -tail=-1
```



## 21. 识别与持久性卷声明(PVC)关联的 PV :

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例 :

```
local-pv-d6bf175b 512Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h server3.example.com
```

如果有一个 PV 处于 **Released** 状态, 请删除它 :

```
# oc delete pv <persistent_volume>
```

例如 :

```
# oc delete pv local-pv-d6bf175b
```

输出示例 :

```
persistentvolume "local-pv-d6bf175b" deleted
```

22. 确定 **crashcollector** pod 部署 :

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod, 请删除它 :

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name>_ -n openshift-storage
```

23. 删除 **ocs-osd-removal** 作业 :

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例 :

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

## 1. 验证输出中是否存在新节点 :

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= |cut -d' ' -f1
```

2. 点 **Workloads** → **Pods**, 确认新节点上至少以下 pod 处于 **Running** 状态 :

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。

确保创建了新的增量 **mon**，并处于 **Running** 状态。

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

```
rook-ceph-mon-a-cd575c89b-b6k66      2/2   Running 0 38m
rook-ceph-mon-b-6776bc469b-tzzt8     2/2   Running 0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh     2/2   Running 0 4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

- 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

- 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

- 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

- 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

- 如果验证步骤失败，请联系[红帽支持团队](#)。

## 2.5.2. 在 Red Hat Virtualization 安装程序置备的基础架构中替换失败的节点

### 先决条件

- 确保将替换节点配置为与您替换的节点类似基础架构、资源和磁盘。
- 您必须登录到 OpenShift Container Platform 集群。

### 流程

- 登录 OpenShift Web 控制台，再点 **Compute** → **Nodes**。
- 确定需要替换的节点。记录其 **Machine Name**。
- 获取节点上的标签：

```
$ oc get nodes --show-labels | grep <node_name>
```

```
<node_name>
```

指定替换的节点的名称。

- 识别在节点上运行的 **mon**（若有）和对象存储设备 (OSD) pod：

```
$ oc get pods -n openshift-storage -o wide | grep -i <node_name>
```

- 缩减您在上一步中确定的 pod 的部署：  
例如：

```
$ oc scale deployment rook-ceph-mon-c --replicas=0 -n openshift-storage
```

```
$ oc scale deployment rook-ceph-osd-0 --replicas=0 -n openshift-storage
```

```
$ oc scale deployment --selector=app=rook-ceph-crashcollector,node_name=<node_name> --replicas=0 -n openshift-storage
```

- 将节点标记为不可调度：

```
$ oc adm cordon <node_name>
```

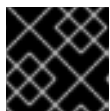
- 删除处于 **Terminating** 状态的 pod：

```
$ oc get pods -A -o wide | grep -i <node_name> | awk '{if ($4 == "Terminating") system ("oc -n " $1 " delete pods " $2 " --grace-period=0 " " --force ")}'
```

- 排空节点：

```
$ oc adm drain <node_name> --force --delete-emptydir-data=true --ignore-daemonsets
```

- 点 **Compute** → **Machines**。搜索所需的机器。
- 除了所需的机器外，点击 **Action 菜单(⋮)** → **Delete Machine**。
- 点 **Delete** 以确认机器已被删除。会自动创建新机器。等待新计算机启动并过渡到 **Running** 状态。



### 重要

此活动可能至少需要 5 到 10 分钟或更多。

- 在 OpenShift Web 控制台中点 **Compute** → **Nodes**。确认新节点处于 **Ready** 状态。
- 物理向节点添加一个或多个新设备。
- 使用以下任一方法之一将 OpenShift Data Foundation 标签应用到新节点：

#### 从用户界面

- 对于新节点，点击 **Action Menu(⋮)** → **Edit Labels**。
- 添加 **cluster.ocs.openshift.io/openshift-storage**，然后点 **Save**。

#### 使用命令行界面

- 将 OpenShift Data Foundation 标签应用到新节点：

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

15. 识别安装 OpenShift 本地存储 Operator 的命名空间，并将其分配给 **local\_storage\_project** 变量：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

例如：

```
$ local_storage_project=$(oc get csv --all-namespaces | awk '{print $1}' | grep local)
```

```
echo $local_storage_project
```

输出示例：

```
openshift-local-storage
```

16. 将新的 worker 节点添加到 **localVolumeDiscovery** 和 **localVolumeSet**。

- a. 更新 **localVolumeDiscovery** 定义，使其包含新节点，并删除故障节点：

```
# oc edit -n $local_storage_project localvolumediscovery auto-discover-devices
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

- b. 确定您需要编辑的 **localVolumeSet**：

```
# oc get -n $local_storage_project localvolumeset
```

输出示例：

```
NAME      AGE
localblock 25h
```

- c. 更新 **localVolumeSet** 定义，使其包含新节点并删除失败的节点：

```
# oc edit -n $local_storage_project localvolumeset localblock
```

输出示例：

```
[...]
nodeSelector:
nodeSelectorTerms:
- matchExpressions:
- key: kubernetes.io/hostname
operator: In
values:
- server1.example.com
- server2.example.com
#- server3.example.com
- newnode.example.com
[...]
```

记住在退出编辑器之前进行保存。

在本例中，**server3.example.com** 已被删除，**newnode.example.com** 是新节点。

17. 验证新的 **localblock** 持久性卷 (PV) 是否可用：

```
$oc get pv | grep localblock | grep Available
```

输出示例：

```
local-pv-551d950 512Gi RWO Delete Available
localblock 26s
```

18. 进入 **openshift-storage** 项目：

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

19. 从集群移除出现故障的 OSD。如果需要，您可以指定多个失败的 OSD：

```
$ oc process -n openshift-storage ocs-osd-removal \
-p FAILED_OSD_IDS=<failed_osd_id> -p FORCE_OSD_REMOVAL=true | oc create -f -
```

**<failed\_osd\_id>**

是 pod 名称中紧接在 **rook-ceph-osd** 前缀后面的整数。

您可以通过在命令中添加逗号分隔的 OSD ID，以移除多个 OSD，例如

**FAILED\_OSD\_IDS=0,1,2**。

**FORCE\_OSD\_REMOVAL** 值必须在有三个 OSD 的集群中更改为 **true**，或者有足够空间的集群在移除 OSD 后恢复所有这三个数据副本。

20. 通过检查 **ocs-osd-removal-job** pod 的状态，验证 OSD 是否已成功移除。状态 **Completed** 确认 OSD 移除作业已成功。

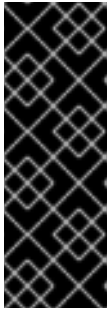
```
# oc get pod -l job-name=ocs-osd-removal-job -n openshift-storage
```

21. 确保 OSD 移除已完成。

```
$ oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1 | egrep -i 'completed removal'
```

输出示例：

```
2022-05-10 06:50:04.501511 I | cephosd: completed removal of OSD 0
```



### 重要

如果 **ocs-osd-removal-job** 失败并且 pod 不在预期的 **Completed** 状态，请检查 pod 日志来进一步调试：

例如：

```
# oc logs -l job-name=ocs-osd-removal-job -n openshift-storage --tail=-1
```

22. 识别与持久性卷声明(PVC)关联的 PV：

```
# oc get pv -L kubernetes.io/hostname | grep localblock | grep Released
```

输出示例：

```
local-pv-d6bf175b 512Gi RWO Delete Released openshift-storage/ocs-deviceset-0-data-0-6c5pw localblock 2d22h server3.example.com
```

如果有一个 PV 处于 **Released** 状态，请删除它：

```
# oc delete pv <persistent_volume>
```

例如：

```
# oc delete pv local-pv-d6bf175b
```

输出示例：

```
persistentvolume "local-pv-d6bf175b" deleted
```

23. 确定 **crashcollector** pod 部署：

```
$ oc get deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name> -n openshift-storage
```

如果有现有的 **crashcollector** pod，请删除它：

```
$ oc delete deployment --selector=app=rook-ceph-crashcollector,node_name=<failed_node_name>_ -n openshift-storage
```

24. 删除 **ocs-osd-removal** 作业：

```
# oc delete -n openshift-storage job ocs-osd-removal-job
```

输出示例：

```
job.batch "ocs-osd-removal-job" deleted
```

## 验证步骤

## 1. 验证输出中是否存在新节点：

```
$ oc get nodes --show-labels | grep cluster.ocs.openshift.io/openshift-storage= | cut -d' ' -f1
```

2. 点 **Workloads** → **Pods**。确认新节点上至少以下 pod 处于 **Running** 状态：

- **csi-cephfsplugin-\***
- **csi-rbdplugin-\***

3. 验证所有其他必需的 OpenShift 数据基础容器集是否都处于 **Running** 状态。确保创建了新的增量 **mon**，并处于 **Running** 状态。

```
$ oc get pod -n openshift-storage | grep mon
```

输出示例：

```
rook-ceph-mon-a-cd575c89b-b6k66      2/2   Running 0 38m
rook-ceph-mon-b-6776bc469b-tzxt8    2/2   Running 0 38m
rook-ceph-mon-d-5ff5d488b5-7v8xh    2/2   Running 0 4m8s
```

OSD 和监控 pod 可能需要几分钟才能进入 **Running** 状态。

## 4. 验证新 OSD pod 是否在替换节点上运行：

```
$ oc get pods -o wide -n openshift-storage | egrep -i <new_node_name> | egrep osd
```

## 5. 可选：如果在集群中启用了集群范围的加密，请验证新 OSD 设备是否已加密。对于上一步中标识的每个新节点，请执行以下操作：

## a. 创建 debug pod 并为一个或多个所选主机打开 chroot 环境：

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

```
$ chroot /host
```

## b. 显示可用块设备列表：

```
$ lsblk
```

检查一个或多个 **ocs-deviceset** 名称旁的 **crypt** 关键字。

6. 如果验证步骤失败，请联系[红帽支持团队](#)。