OpenShift Container Platform 4.16

备份和恢复

备份和恢复 OpenShift Container Platform 集群

Last Updated: 2024-07-03

OpenShift Container Platform 4.16 备份和恢复

备份和恢复 OpenShift Container Platform 集群

法律通告

摘要

本文档提供了备份集群数据以及从各种灾难场景中恢复的步骤。

目录

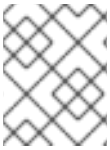
第 1 章 备份和恢复

1.1. CONTROL PLANE 备份和恢复操作

作为集群管理员，您可能需要在一段时间内停止 OpenShift Container Platform 集群，并在以后重启集群。重启集群的一些原因是您需要对集群执行维护或希望降低资源成本。在 OpenShift Container Platform 中，您可以对集群执行安全关闭，以便在以后轻松重启集群。

您必须在关闭集群前 备份 etcd 数据 ；etcd 是 OpenShift Container Platform 的键值存储，它会保留所有资源对象的状态。etcd 备份在灾难恢复中扮演着关键角色。在 OpenShift Container Platform 中，您还可以替换不健康的 etcd 成员。

当您希望集群再次运行时，请安全地重启集群。



注意

集群的证书在安装日期后一年后过期。您可以关闭集群，并在证书仍有效时安全地重启集群。虽然集群自动检索过期的 control plane 证书，但您仍需要批准证书签名请求(CSR)。

您可能会遇到 OpenShift Container Platform 无法按预期工作的一些情况，例如：

- 您有一个在重启后无法正常工作的集群，因为意外状况（如节点故障或网络连接问题）无法正常工作。
- 您已错误地删除了集群中的某些关键内容。
- 您丢失了大多数 control plane 主机，从而导致 etcd 仲裁丢失。

通过使用保存的 etcd 快照，始终可以通过将集群恢复到之前的状态来从灾难中恢复。

其他资源

- 使用机器生命周期 hook 进行仲裁保护

1.2. 应用程序备份和恢复操作

作为集群管理员，您可以使用 OpenShift API 进行数据保护(OADP)来备份和恢复在 OpenShift Container Platform 上运行的应用程序。

根据下载 Velero CLI 工具中的表，按照命名空间粒度来备份和恢复 Kubernetes 资源和内部镜像。OADP 使用快照或 Restic 来备份和恢复持久性卷(PV)。详情请查看 OADP 功能。

1.2.1. OADP 要求

OADP 有以下要求：

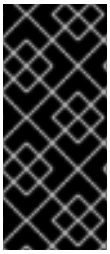
- 您必须以具有 **cluster-admin** 角色的用户身份登录。
- 您必须具有用于存储备份的对象存储，比如以下存储类型之一：
  - OpenShift Data Foundation
  - Amazon Web Services

- Microsoft Azure
- Google Cloud Platform
- S3 兼容对象存储
- IBM Cloud® Object Storage S3

注意

如果要在 OCP 4.11 及之后的版本中使用 CSI 备份，请安装 OADP 1.1.*x*。

OADP 1.0.*x* 不支持 OCP 4.11 及更高版本上的 CSI 备份。OADP 1.0. *x* 包括 Velero 1.7.*x*，并需要 API 组 **snapshot.storage.k8s.io/v1beta1**，这在 OCP 4.11 及更高版本中不存在。

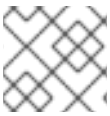


重要

S3 存储的 **CloudStorage** API 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

- 要使用快照备份 PV，您必须有具有原生快照 API 的云存储，或者支持 Container Storage Interface(CSI)快照，如以下供应商：
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - 支持 CSI 快照的云存储，如 Ceph RBD 或 Ceph FS



注意

如果您不想使用快照备份 PV，可以使用 Restic，这由 OADP Operator 安装。

1.2.2. 备份和恢复应用程序

您可以通过创建一个 **Backup** 自定义资源 (CR) 来备份应用程序。请参阅 创建备份 CR。您可以配置以下备份选项：

- 创建备份 hook，以便在备份操作之前或之后运行命令
- 调度备份
- 使用文件系统备份对应用程序进行备份：Kopia 或 Restic
- 您可以通过创建一个 **Restore** (CR) 来恢复应用程序备份。请参阅 创建 Restore CR。
- 您可以配置 restore hook，以便在 init 容器或应用程序容器中运行命令。

第 2 章 安全地关闭集群

本文档描述了安全关闭集群的过程。出于维护或者节约资源成本的原因，您可能需要临时关闭集群。

2.1. 先决条件

- 在关闭集群前进行 etcd 备份。

重要

执行此流程前务必要进行 etcd 备份，以便在重启集群遇到任何问题时可以恢复集群。

例如，以下条件可能会导致重启的集群失败：

- 关机过程中的 etcd 数据崩溃
- 因硬件原因造成节点故障
- 网络连接问题

如果集群无法恢复，请按照以下步骤恢复到以前的集群状态。

2.2. 关闭集群

您可以以安全的方式关闭集群，以便稍后重启集群。

注意

您可以在安装日期起的一年内关闭集群，并期望它可以正常重启。安装日期起一年后，集群证书会过期。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 已进行 etcd 备份。

流程

1. 如果您要长时间关闭集群，请确定证书过期的日期，并运行以下命令：

```
$ oc -n openshift-kube-apiserver-operator get secret kube-apiserver-to-kubelet-signer -o jsonpath='{.metadata.annotations.auth\.openshift\.io/certificate-not-after}'
```

输出示例

```
2022-08-05T14:37:50Zuser@user:~ $ 1
```

1 为确保集群可以正常重启，请计划在指定的日期或之前重启集群。当集群重启时，可能需要您手动批准待处理的证书签名请求 (CSR) 来恢复 kubelet 证书。

2. 将集群中的所有节点标记为不可调度。您可以从云供应商的 web 控制台或运行以下循环来完成此操作：

```
$ for node in $(oc get nodes -o jsonpath='{.items[*].metadata.name}'); do echo ${node} ; oc adm cordon ${node} ; done
```

输出示例

```
ci-ln-mgdnf4b-72292-n547t-master-0
node/ci-ln-mgdnf4b-72292-n547t-master-0 cordoned
ci-ln-mgdnf4b-72292-n547t-master-1
node/ci-ln-mgdnf4b-72292-n547t-master-1 cordoned
ci-ln-mgdnf4b-72292-n547t-master-2
node/ci-ln-mgdnf4b-72292-n547t-master-2 cordoned
ci-ln-mgdnf4b-72292-n547t-worker-a-s7ntl
node/ci-ln-mgdnf4b-72292-n547t-worker-a-s7ntl cordoned
ci-ln-mgdnf4b-72292-n547t-worker-b-cmc9k
node/ci-ln-mgdnf4b-72292-n547t-worker-b-cmc9k cordoned
ci-ln-mgdnf4b-72292-n547t-worker-c-vcmtn
node/ci-ln-mgdnf4b-72292-n547t-worker-c-vcmtn cordoned
```

3. 使用以下方法撤离 pod：

```
$ for node in $(oc get nodes -l node-role.kubernetes.io/worker -o jsonpath='{.items[*].metadata.name}'); do echo ${node} ; oc adm drain ${node} --delete-emptydir-data --ignore-daemonsets=true --timeout=15s --force ; done
```

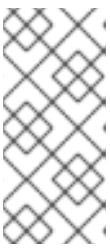
4. 关闭集群中的所有节点。您可以从云供应商的 web 控制台或运行以下循环来完成此操作：

```
$ for node in $(oc get nodes -o jsonpath='{.items[*].metadata.name}'); do oc debug node/${node} -- chroot /host shutdown -h 1; done
```

输出示例

```
Starting pod/ip-10-0-130-169us-east-2computeinternal-debug ...
To use host binaries, run `chroot /host`
Shutdown scheduled for Mon 2021-09-13 09:36:17 UTC, use 'shutdown -c' to cancel.
Removing debug pod ...
Starting pod/ip-10-0-150-116us-east-2computeinternal-debug ...
To use host binaries, run `chroot /host`
Shutdown scheduled for Mon 2021-09-13 09:36:29 UTC, use 'shutdown -c' to cancel.
```

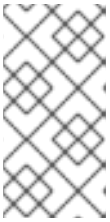
使用以下方法关闭节点可让 pod 安全终止，从而减少数据崩溃的可能性。



注意

为大规模集群调整关闭时间：

```
$ for node in $(oc get nodes -o jsonpath='{.items[*].metadata.name}'); do oc debug node/${node} -- chroot /host shutdown -h 10; done
```



注意

在关闭前，不需要排空 OpenShift Container Platform 中附带的标准 pod 的 control plane 节点。集群管理员负责确保在集群重启后，彻底重启自己的工作负载。如果因为自定义工作负载的原因已在关闭前排空 control plane 节点，您必须在重启后将 control plane 节点标记为可调度，然后集群才可以重新正常工作。

5. 关闭不再需要的集群依赖项，如外部存储或 LDAP 服务器。在进行操作前请务必查阅您的厂商文档。

重要

如果您在云供应商平台上部署了集群，请不要关闭、挂起或删除关联的云资源。如果您删除挂起的虚拟机的云资源，OpenShift Container Platform 可能无法成功恢复。

2.3. 其他资源

- 正常重启集群

第 3 章 正常重启集群

本文档论述了在安全关闭后重启集群的过程。

尽管在重启后集群应该可以正常工作，但可能会因为意外状况集群可能无法恢复，例如：

- 关机过程中的 etcd 数据崩溃
- 因硬件原因造成节点故障
- 网络连接问题

如果集群无法恢复，请按照以下步骤恢复到以前的集群状态。

3.1. 先决条件

- 已安全关闭集群。

3.2. 重启集群

您可以在集群被安全关闭后重启它。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 此流程假设您安全关闭集群。

流程

1. 启动所有依赖设备，如外部存储或 LDAP 服务器。
2. 启动所有集群机器。
   使用适合您的云环境的方法启动机器，例如从云供应商的 Web 控制台启动机器。

   等待大约 10 分钟，然后继续检查 control plane 节点的状态。
3. 验证所有 control plane 节点都已就绪。

   ```
   $ oc get nodes -l node-role.kubernetes.io/master
   ```

   如果状态为 **Ready**，如以下输出中所示，则代表 control plane 节点已就绪：

   ```
   NAME                          STATUS   ROLES    AGE   VERSION
   ip-10-0-168-251.ec2.internal   Ready    master   75m   v1.29.4
   ip-10-0-170-223.ec2.internal   Ready    master   75m   v1.29.4
   ip-10-0-211-16.ec2.internal    Ready    master   75m   v1.29.4
   ```
4. 如果 control plane 节点*没有*就绪，请检查是否有待批准的证书签名请求 (CSR)。
   a. 获取当前 CSR 列表：

      ```
      $ oc get csr
      ```

b. 查看一个 CSR 的详细信息以验证其是否有效：

```
$ oc describe csr <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

c. 批准每个有效的 CSR：

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

5. 在 control plane 节点就绪后，验证所有 worker 节点是否已就绪。

```
$ oc get nodes -l node-role.kubernetes.io/worker
```

如果状态为 **Ready**，如下所示，则代表 worker 节点已就绪：

```
NAME                         STATUS   ROLES    AGE   VERSION
ip-10-0-179-95.ec2.internal  Ready    worker   64m   v1.29.4
ip-10-0-182-134.ec2.internal Ready    worker   64m   v1.29.4
ip-10-0-250-100.ec2.internal Ready    worker   64m   v1.29.4
```

6. 如果 worker 节点 *未* 就绪，请检查是否有待批准的证书签名请求(CSR)。

a. 获取当前 CSR 列表：

```
$ oc get csr
```

b. 查看一个 CSR 的详细信息以验证其是否有效：

```
$ oc describe csr <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

c. 批准每个有效的 CSR：

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

7. 验证集群是否已正确启动。

a. 检查是否有降级的集群 Operator。

```
$ oc get clusteroperators
```

确定没有 **DEGRADED** 条件为 **True** 的集群 Operator。

```
NAME                                       VERSION   AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.16.0    True        False         False     59m
cloud-credential                           4.16.0    True        False         False     85m
cluster-autoscaler                         4.16.0    True        False         False     73m
config-operator                            4.16.0    True        False         False     73m
```

```
console                                    4.16.0    True        False        False      62m
csi-snapshot-controller                    4.16.0    True        False        False      66m
dns                                        4.16.0    True        False        False      76m
etcd                                       4.16.0    True        False        False      76m
...
```

b. 检查所有节点是否处于 **Ready** 状态：

```
$ oc get nodes
```

检查所有节点的状态是否为 **Ready**。

```
NAME                           STATUS   ROLES    AGE   VERSION
ip-10-0-168-251.ec2.internal   Ready    master   82m   v1.29.4
ip-10-0-170-223.ec2.internal   Ready    master   82m   v1.29.4
ip-10-0-179-95.ec2.internal    Ready    worker   70m   v1.29.4
ip-10-0-182-134.ec2.internal   Ready    worker   70m   v1.29.4
ip-10-0-211-16.ec2.internal    Ready    master   82m   v1.29.4
ip-10-0-250-100.ec2.internal   Ready    worker   69m   v1.29.4
```

如果集群无法正确启动，您可能需要使用 etcd 备份来恢复集群。

8. 在 control plane 和 worker 节点就绪后，将集群中的所有节点标记为可调度。运行以下命令:

```
for node in $(oc get nodes -o jsonpath='{.items[*].metadata.name}'); do echo ${node} ; oc adm uncordon ${node} ; done
```

其他资源

- 如果集群重启后无法恢复，请参阅恢复到以前的集群状态。

第 4 章 OADP 应用程序备份和恢复

4.1. OPENSHIFT API FOR DATA PROTECTION 简介

OpenShift API for Data Protection (OADP) 产品保护 OpenShift Container Platform 上的客户应用程序。它提供全面的灾难恢复保护，涵盖 OpenShift Container Platform 应用程序、应用程序相关的集群资源、持久性卷和内部镜像。OADP 还能够备份容器化应用程序和虚拟机 (VM)。

但是，OADP 不会充当 etcd 或 OpenShift Operator 的灾难恢复解决方案。

4.1.1. OpenShift API for Data Protection API

OpenShift API for Data Protection (OADP) 提供了 API，它允许多种方法自定义备份，并防止包含不必要的或不当的资源。

OADP 提供以下 API：

- Backup
- 恢复
- 调度
- BackupStorageLocation
- VolumeSnapshotLocation

其他资源

- 备份 etcd

4.2. OADP 发行注记

4.2.1. OADP 1.3 发行注记

OpenShift API for Data Protection (OADP) 的发行注记介绍了新的功能和增强功能、已弃用的功能、产品建议、已知问题和解决问题。

4.2.1.1. OADP 1.3.2 发行注记

OpenShift API for Data Protection (OADP) 1.3.2 发行注记列出了已解决的问题和已知问题。

4.2.1.1.1. 已解决的问题

如果使用了有效的自定义 secret 进行 BSL，则 DPA 无法协调

如果将有效的自定义 secret 用于 Backup Storage Location (BSL)，但缺少默认 secret，则 DPA 无法协调。解决办法是首先创建所需的默认 **cloud-credentials**。当重新创建了自定义 secret，可以使用它并检查其是否存在。

OADP-3193

CVE-2023-45290: oadp-velero-container: Golang net/http: Memory exhaustion in Request.ParseMultipartForm

net/http Golang 标准库软件包中发现了一个安全漏洞，这会影响到之前 OADP 版本。在以前的版本中，当解析 **multipart** 表单时，可以明确使用 **Request.ParseMultipartForm**，或使用 **Request.FormValue**, **Request.PostFormValue**, 或 **Request.FormFile** 方法隐式应用，解析表单的总大小限制不适用于在读单一表单行时消耗的内存。这可能会允许在恶意设计的输入中包含非常长的行，从而导致分配大量内存，这可能会导致内存耗尽。这个安全漏洞已在 OADP 1.3.2 中解决。

如需了解更多详细信息，请参阅 CVE-2023-45290。

CVE-2023-45289: oadp-velero-container: Golang net/http/cookiejar: Incorrect forwarding of sensitive headers and cookies on HTTP redirect

net/http/cookiejar Golang 标准库软件包中发现了一个安全漏洞，这会影响到之前 OADP 版本。 当 HTTP 重定向到不是子域匹配或与初始域完全匹配的域时，**http.Client** 不会转发敏感标头，如 **Authorization** 或 **Cookie**。恶意精心设计的 HTTP 重定向可能会导致敏感标头被意外转发。这个安全漏洞已在 OADP 1.3.2 中解决。

如需了解更多详细信息，请参阅 CVE-2023-45289。

CVE-2024-24783: oadp-velero-container: Golang crypto/x509: Verify panics on certificates with an unknown public key algorithm

在 **crypto/x509** Golang 标准库软件包中发现了一个安全漏洞，这会影响到之前的 OADP 版本。验证包含带有未知公钥算法的证书的证书链会导致 **Certificate.Verify** panic。这会影响将 **Config.ClientAuth** 设置为 **VerifyClientCertIfGiven** 或 **RequireAndVerifyClientCert** 的所有 **crypto/tls** 客户端和服务器。默认行为是 TLS 服务器无法验证客户端证书。这个安全漏洞已在 OADP 1.3.2 中解决。

如需了解更多详细信息，请参阅 CVE-2024-24783。

CVE-2024-24784: oadp-velero-plugin-container: Golang net/mail: Comments in display names are incorrectly handled

net/mail Golang 标准库软件包中发现了一个安全漏洞，这会影响到之前 OADP 版本。**ParseAddressList** 函数错误地处理注释、文本（括号中的文本）和显示名称。由于这与地址解析程序不匹配，因此可能会导致使用不同解析器的程序进行不同的信任决策。这个安全漏洞已在 OADP 1.3.2 中解决。

如需了解更多详细信息，请参阅 CVE-2024-24784。

CVE-2024-24785: oadp-velero-container: Golang: html/template: errors returned from MarshalJSON methods may break template escaping

在 **html/template** Golang 标准库软件包中发现了一个安全漏洞，这会影响到之前 OADP 版本。 如果来自 **MarshalJSON** 方法返回的错误包含用户控制的数据，则它们可能会用来破坏 HTML/template 软件包的上下文自动转义行为，以便后续操作将意外内容注入模板。 这个安全漏洞已在 OADP 1.3.2 中解决。

如需了解更多详细信息，请参阅 CVE-2024-24785。

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.3.2 解决的问题列表。

4.2.1.1.2. 已知问题

恢复 OADP 后，Cassandra 应用程序 pod 进入 CrashLoopBackoff 状态

在 OADP 恢复后，**CrashLoopBackoff** 状态可能会进入 Cassandra 应用程序 pod。要临时解决这个问题，在恢复 OADP 后删除返回错误或 **CrashLoopBackoff** 状态的 **StatefulSet** pod。**StatefulSet** 控制器重新创建这些 pod，它正常运行。

OADP-3767

4.2.1.2. OADP 1.3.1 发行注记

OpenShift API for Data Protection (OADP) 1.3.1 发行注记列出了新功能、解决的问题和错误以及已知的问题。

4.2.1.2.1. 新功能

OADP 1.3.0 Data Mover 现在被完全支持

在 OADP 1.3.0 中作为技术预览引进的 OADP 内置 Data Mover，现在在容器化和虚拟机工作负载中被完全支持。

4.2.1.2.2. 已解决的问题

现在支持 IBM Cloud(R) Object Storage 作为存储供应商

IBM Cloud® Object Storage 是 AWS S3 兼容备份存储供应商之一，它在之前不被支持。在这个版本中，IBM Cloud® Object Storage 作为 AWS S3 兼容备份存储供应商被支持。

OADP-3788

OADP operator 现在可以正确地报告缺少的区域错误

在以前的版本中，当在 AWS Backup Storage Location (BSL) 配置中在没有指定 **region** 的情况下指定 **profile:default** 时，OADP operator 无法报告数据保护应用程序 (DPA) 自定义资源(CR) 中 **missing region** 错误。在这个版本中，修正了 AWS 的 DPA BSL 规格的验证。因此，OADP Operator 会报告 **missing region** 错误。

OADP-3044

自定义标签不会从 openshift-adp 命名空间中删除

在以前的版本中，**openshift-adp-controller-manager** pod 会重置附加到 **openshift-adp** 命名空间的标签。这会导致需要自定义标签（如 Argo CD）的应用程序出现同步问题，从而导致功能不正确。在这个版本中，这个问题已被解决，自定义标签不会从 **openshift-adp** 命名空间中删除。

OADP-3189

OADP must-gather 镜像收集 CRD

在以前的版本中，OADP **must-gather** 镜像不会收集 OADP 提供的自定义资源定义 (CRD)。因此，您无法使用 **omg** 工具在 support shell 中提取数据。在这个版本中，**must-gather** 镜像会收集 OADP 附带的 CRD，并可使用 **omg** 工具提取数据。

OADP-3229

垃圾回收具有默认频率值的正确描述

在以前的版本中，**garbage-collection-frequency** 字段对默认频率值有一个错误的描述。在这个版本中，**garbage-collection-frequency** 具有 **gc-controller** 协调默认频率的正确值（一小时）。

OADP-3486

OperatorHub 中提供了 FIPS 模式标记

通过将 **fips-compliant** 标志设置为 **true**，现在 FIPS 模式标记添加到 OperatorHub 中的 OADP Operator 列表中。此功能在 OADP 1.3.0 中启用，但没有在启用了 FIPS 的 Red Hat Container 目录中显示。

OADP-3495

当 csiSnapshotTimeout 设置为短持续时间时，CSI 插件不会出现 nil pointer 的错误

在以前的版本中，当 **csiSnapshotTimeout** 参数设置为较短的持续时间时，CSI 插件遇到以下错误：**plugin panicked: runtime error: invalid memory address or nil pointer dereference**。

在这个版本中，备份会失败并显示以下错误： **Timed out awaiting reconciliation of volumesnapshot**。

OADP-3069

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.3.1 解决的问题列表。

4.2.1.2.3. 已知问题

IBM Power (R) 和 IBM Z (R) 平台上部署的单节点 OpenShift 集群的备份和存储限制

查看在 IBM Power® 和 IBM Z® 平台上部署的单节点 OpenShift 集群的备份和存储相关限制：

Storage

目前，只有 NFS 存储与 IBM Power® 和 IBM Z® 平台上部署的单节点 OpenShift 集群兼容。

Backup

备份和恢复操作只支持使用文件系统备份（如 **kopia** 和 **restic** ）备份应用程序。

OADP-3787

恢复 OADP 后，CrashLoopBackoff 状态会进入 CrashLoopBackoff 状态

在 OADP 恢复后，**CrashLoopBackoff** 状态可能会进入 Cassandra 应用程序 pod。要临时解决这个问题，在恢复 OADP 后删除带有任何错误或 **CrashLoopBackoff** 状态的 **StatefulSet** pod。**StatefulSet** 控制器重新创建这些 pod，它正常运行。

OADP-3767

4.2.1.3. OADP 1.3.0 发行注记

OpenShift API for Data Protection (OADP) 1.3.0 发行注记列出了新功能、解决的问题和错误以及已知的问题。

4.2.1.3.1. 新功能

Velero 内置 DataMover

OADP 1.3 包含一个内置的 Data Mover，您可以使用它将 Container Storage Interface (CSI) 卷快照移到远程对象存储。如果发生故障、意外删除或损坏，内置的 Data Mover 可让您从远程对象存储中恢复有状态的应用程序。它使用 Kopia 作为上传程序机制来读取快照数据并写入 Unified Repository。

Velero 内置 DataMover 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

使用文件系统备份对应用程序进行备份：Kopia 或 Restic

Velero 的文件系统备份(FSB)支持两个备份库：Restic 和 Kopia。

Velero 允许用户在两个之间进行选择。

对于备份，通过 **uploader-type** 标志指定在安装过程中的路径。有效值为 **restic** 或 **kopia**。如果没有指定值，则默认为 **kopia**。安装后无法更改选择。

GCP Cloud 身份验证

Google Cloud Platform (GCP) 身份验证可让您使用短生命 Google 凭证。

带有 Workload Identity Federation 的 GCP 可让您使用 Identity and Access Management (IAM) 授予外部身份 IAM 角色，包括模拟服务帐户的功能。这消除了与服务帐户密钥相关的维护和安全风险。

AWS ROSA STS 身份验证

您可以将 OpenShift API 用于 Red Hat OpenShift Service on AWS (ROSA) 集群的数据保护 (OADP) 来备份和恢复应用程序数据。

ROSA 提供与各种 AWS 计算、数据库、分析、机器学习、联网、移动和其他服务进行无缝整合，进一步加快向客户构建和交付不同体验。

您可以直接从 AWS 帐户订阅该服务。

创建集群后，您可以使用 OpenShift Web 控制台对集群进行操作。ROSA 服务还使用 OpenShift API 和命令行界面 (CLI) 工具。

4.2.1.3.2. 已解决的问题

在恢复后，ACM 应用程序会被删除并重新创建

在恢复激活后，ACM 应用程序会在受管集群中被删除并重新创建。OpenShift API for Data Protection (OADP 1.2) 备份和恢复过程比旧版本快。在恢复 ACM 资源时，OADP 性能的变化导致了这个结果。因此，有些资源会在其他资源前恢复，这会导致从受管集群中删除应用程序。OADP-2686

由于 Pod 安全标准，Restic 恢复部分失败

在互操作性测试过程中，OpenShift Container Platform 4.14 将 pod 安全模式设置为 **enforce**，这会导致 pod 被拒绝。这是因为恢复顺序造成的。pod 在安全性上下文约束(SCC)资源之前创建，因为 pod 违反了 **podSecurity** 标准，因此拒绝 Pod。当在 Velero 服务器上设置 restore 优先级字段时，恢复可以成功。OADP-2688

如果 Velero 在多个命名空间中安装，则可能会出现 pod 卷备份失败

当在几个命名空间中安装 Velero 时，Pod 卷备份 (PVB) 功能中存在安全漏洞。PVB 控制器没有正确限制为自己命名空间中的 PVB。OADP-2308

OADP Velero 插件返回 "received EOF, stop recv loop" 信息

在 OADP 中，Velero 插件作为单独的进程启动。当 Velero 操作完成后，无论是否成功，它们都会退出。因此，如果您看到 **received EOF, stopping recv loop** 消息，这并不意味着发生了错误。它代表一个插件操作已完成。OADP-2176

CVE-2023-39325 启用的多个 HTTP/2 的 Web 服务器容易受到 DDoS 攻击(Rapid Reset Attack) 的影响

在之前的 OADP 版本中，HTTP/2 协议易受拒绝服务攻击的影响，因为请求可以快速重置多个流。服务器需要在没有达到每个连接的最大活跃流数量在服务器端的限制的情况下，设置和处理流。这会导致因为服务器的资源被耗尽而出现拒绝服务的问题。

如需更多信息，请参阅 CVE-2023-39325 (Rapid Reset Attack)

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.3.0 解决的问题列表。

4.2.1.3.3. 已知问题

当 csiSnapshotTimeout 设置为较短的持续时间时，空指针上的 CSI 插件错误

当 **csiSnapshotTimeout** 设置为较短的持续时间时，空指针上的 CSI 插件错误有时，它会在短时间内成功完成快照，但通常会出现备份 **PartiallyFailed** 的 panics，并带有以下错误： **plugin panicked: runtime error: invalid memory address 或 nil pointer dereference**。

当 volumeSnapshotContent CR 出错时，备份被标记为 PartiallyFailed

如果有任何 **VolumeSnapshotContent** CR 存在一个与删除 **VolumeSnapshotBeingCreated** 注解相关的错误，它会将备份移到 **WaitingForPluginOperationsPartiallyFailed** 阶段。OADP-2871

第一次恢复 30,000 资源时的性能问题

当第一次恢复 30,000 个资源时，没有 existing-resource-policy，恢复它们所需的时间需要两倍长（与第二次第三次在 existing-resource-policy 设置为 **update** 时尝试恢复所需时间相比）。OADP-3071

在 Datadownload 操作释放相关的 PV 前，恢复后 hook 可能会开始运行

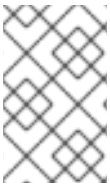
由于 Data Mover 操作的异步性质，在 Data Mover 持久性卷声明(PVC)发布相关的 pod 持久性卷(PV)前可能会尝试 post-hook。

GCP-Workload Identity Federation VSL 备份 PartiallyFailed

当在 GCP 上配置 GCP 工作负载身份时，VSL 备份 **PartiallyFailed**。

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.3.0 已知问题 列表。

4.2.1.3.4. 升级备注



注意

始终升级到下一个次版本。**不要** 跳过版本。要升级到更新的版本，请一次只升级一个频道。例如，若要从 OpenShift API for Data Protection (OADP) 1.1 升级到 1.3，首先升级到 1.2，然后再升级到 1.3。

4.2.1.3.4.1. 从 OADP 1.2 改为 1.3

Velero 服务器已从版本 1.11 更新至 1.12。

OpenShift API for Data Protection (OADP) 1.3 使用 Velero 内置 Data Mover 而不是 VolumeSnapshotMover (VSM) 或 Volsync Data Mover。

这会更改以下内容：

- **spec.features.dataMover** 字段和 VSM 插件与 OADP 1.3 不兼容，您必须从 **DataProtectionApplication** (DPA) 配置中删除配置。

- Data Mover 功能不再需要 Volsync Operator，您可以删除它。
- 自定义资源定义 **volumesnapshotbackups.datamover.oadp.openshift.io** 和 **volumesnapshotrestores.datamover.oadp.openshift.io** 不再需要，您可以删除它们。
- 不再需要用于 OADP-1.2 Data Mover 的 secret，您可以删除它们。

OADP 1.3 支持 Kopia，它是 Restic 的替代文件系统备份工具。

- 要使用 Kopia，请使用新的 **spec.configuration.nodeAgent** 字段，如下例所示：

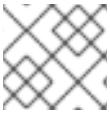
  Example

```
spec:
  configuration:
    nodeAgent:
      enable: true
      uploaderType: kopia
# ...
```

- **spec.configuration.restic** 字段在 OADP 1.3 中已弃用，并将在以后的 OADP 版本中删除。要避免出现弃用警告，请删除 **restic** 键及其值，并使用以下语法：

  Example

```
spec:
  configuration:
    nodeAgent:
      enable: true
      uploaderType: restic
# ...
```



注意

在以后的 OADP 发行版本中，计划 **kopia** 工具成为默认的 **uploaderType** 值。

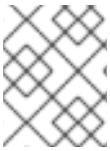
4.2.1.3.4.2. 升级步骤

4.2.1.3.4.3. 从 OADP 1.2 技术预览 Data Mover 升级

OpenShift API for Data Protection (OADP) 1.2 Data Mover 备份 **无法使用** OADP 1.3 恢复。要防止应用程序的数据保护出现差距，请在升级到 OADP 1.3 前完成以下步骤：

流程

1. 如果您的集群备份足够，且 Container Storage Interface (CSI) 存储可用，请使用 CSI 备份备份应用程序。
2. 如果您需要关闭集群备份：
   a. 使用使用 **--default-volumes-to-fs-backup=true 或 backup.spec.defaultVolumesToFsBackup** 选项的文件系统备份应用程序。
   b. 使用对象存储插件备份应用程序，如 **velero-plugin-for-aws**。



注意

Restic 文件系统备份的默认超时值为一小时。在 OADP 1.3.1 及更高版本中，Restic 和 Kopia 的默认超时值为四小时。

重要

要恢复 OADP 1.2 Data Mover 备份，您必须卸载 OADP，并安装和配置 OADP 1.2。

4.2.1.3.4.4. 备份 DPA 配置

您必须备份当前的 **DataProtectionApplication** (DPA) 配置。

流程

- 运行以下命令来保存您当前的 DPA 配置：

  Example

```
$ oc get dpa -n openshift-adp -o yaml > dpa.orig.backup
```

4.2.1.3.4.5. 升级 OADP Operator

在升级 OpenShift API for Data Protection (OADP) Operator 时，使用以下序列。

流程

1. 将 OADP Operator 的订阅频道从 **stable-1.2** 改为 **stable-1.3**。
2. 允许 Operator 和容器更新并重启的时间。

其他资源

- 更新安装的 Operator

4.2.1.3.4.6. 将 DPA 转换为新版本

如果您需要使用 Data Mover 移出集群，请重新配置 **DataProtectionApplication** (DPA) 清单，如下所示。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 部分中，点 **View more**。
3. 在 **DataProtectionApplication** 框中点 **Create instance**。
4. 点 **YAML View** 显示当前 DPA 参数。

   当前 DPA 示例

```
spec:
  configuration:
```

```
    features:
      dataMover:
      enable: true
      credentialName: dm-credentials
    velero:
      defaultPlugins:
      - vsm
      - csi
      - openshift
# ...
```

5. 更新 DPA 参数：

   - 从 DPA 中删除 **features.dataMover** 键和值。
   - 删除 VolumeSnapshotMover (VSM) 插件。
   - 添加 **nodeAgent** 键和值。

     更新的 DPA 示例

```
spec:
  configuration:
    nodeAgent:
      enable: true
      uploaderType: kopia
    velero:
      defaultPlugins:
      - csi
      - openshift
# ...
```

6. 等待 DPA 成功协调。

4.2.1.3.4.7. 验证升级

使用以下步骤验证升级。

流程

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                1/1     Running   0          94s
pod/node-agent-m4lts                                1/1     Running   0          94s
pod/node-agent-pv4kr                                1/1     Running   0          95s
pod/velero-588db7f655-n842v                         1/1     Running   0          95s

NAME                                                TYPE       CLUSTER-IP       EXTERNAL-IP
```

```
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc                   ClusterIP   172.30.10.0     <none>
8085/TCP   8h

NAME                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                              1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输**出示例**

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输**出示例**

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

在 OADP 1.3 中，您可以为每个备份启动数据移动，而不是创建 **DataProtectionApplication** (DPA)配置。

Example

```
$ velero backup create example-backup --include-namespaces mysql-persistent --snapshot-move-
data=true
```

Example

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: example-backup
  namespace: openshift-adp
```

```
spec:
  snapshotMoveData: true
  includedNamespaces:
  - mysql-persistent
  storageLocation: dpa-sample-1
  ttl: 720h0m0s
# ...
```

4.2.2. OADP 1.2 发行注记

OpenShift API for Data Protection (OADP) 1.2 的发行注记介绍了新的功能和增强功能、已弃用的功能、产品建议、已知问题和解决问题。

4.2.2.1. OADP 1.2.5 发行注记

OpenShift API for Data Protection (OADP) 1.2.5 是一个 Container Grade only (CGO) 版本，用于刷新容器的健康等级，与 OADP 1.2.4 相比，产品本身的代码并没有改变。

4.2.2.1.1. 已解决的问题

CVE-2023-2431: oadp-velero-plugin-for-microsoft-azure-container: Bypass of seccomp profile enforcement

Kubernetes 中发现了一个安全漏洞，它会影响到 OADP 的早期版本。当为 **seccomp** 配置集使用 localhost 类型但指定了一个空的配置集字段时存在一个漏洞，会导致 Kubernetes 允许本地的、经过验证的攻击者可以绕过安全限制。攻击者可以通过发送一个精心设计的请求来绕过 **seccomp** 配置集强制。这个安全漏洞已在 OADP 1.2.5 中解决。

如需了解更多详细信息，请参阅 (CVE-2023-2431)。

CSI 恢复以 'PartiallyFailed' 状态结束，PVC 不会被创建

CSI 恢复以 **PartiallyFailed** 状态结尾。PVC 不会被创建，pod 处于 **Pending** 状态。这个问题已在 OADP 1.2.5 中解决。

(OADP-1956)

在完成的 pod 卷中，PodVolumeBackup 失败

在 OADP 1.2 的早期版本中，当 Restic **podvolumebackup** 或 Velero 备份使用的命名空间中挂载了已完成的 pod 时，备份将无法成功完成。当 **defaultVolumesToFsBackup** 设置为 **true** 时会出现这种情况。这个问题已在 OADP 1.2.5 中解决。

(OADP-1870)

4.2.2.1.2. 已知问题

当凭证 secret 被更新时，数据保护应用程序 (DPA) 不会协调

目前，当您更新 **cloud-credentials** secret 时，OADP Operator 不会协调。这是因为 **cloud-credentials** secret 中没有 OADP 特定的标签或所有者引用。如果您创建带有不正确的凭证的 **cloud-credentials** secret，如空数据，Operator 会协调并创建带有空数据的 Backup Storage Location (BSL) 和 registry 部署。因此，当使用正确的凭证更新 **cloud-credentials** secret 时，OADP Operator 不会立即协调来捕获新凭证。

临时解决方案：更新到 OADP 1.3。

(OADP-3327)

4.2.2.2. OADP 1.2.4 发行注记

OpenShift API for Data Protection (OADP) 1.2.4 是一个 Container Grade only (CGO) 版本，用于刷新容器的健康等级，与 OADP 1.2.3 相比，产品本身的代码并没有改变。

4.2.2.2.1. 已解决的问题

OADP 1.2.4 中没有解决的问题。

4.2.2.2.2. 已知问题

OADP 1.2.4 有以下已知问题：

当凭证 secret 被更新时，数据保护应用程序 (DPA) 不会协调

目前，当您更新 **cloud-credentials** secret 时，OADP Operator 不会协调。这是因为 **cloud-credentials** secret 中没有 OADP 特定的标签或所有者引用。 如果您创建带有不正确的凭证的 **cloud-credentials** secret，如空数据，Operator 会协调并创建带有空数据的 Backup Storage Location (BSL) 和 registry 部署。因此，当使用正确的凭证更新 **cloud-credentials** secret 时，Operator 不会立即协调来捕获新凭证。

临时解决方案：更新到 OADP 1.3。

(OADP-3327)

4.2.2.3. OADP 1.2.3 发行注记

4.2.2.3.1. 新功能

OpenShift API for Data Protection (OADP) 1.2.3 版本没有包括新的功能。

4.2.2.3.2. 已解决的问题

以下主要问题已在 OADP 1.2.3 中解决：

启用的多个 HTTP/2 的 Web 服务器容易受到 DDoS 攻击(Rapid Reset Attack) 的影响

在之前的 OADP 1.2 版本中，HTTP/2 协议易受拒绝服务攻击的影响，因为请求可以快速重置多个流。服务器需要在没有达到每个连接的最大活跃流数量在服务器端的限制的情况下，设置和处理流。这会导致因为服务器的资源被耗尽而出现拒绝服务的问题。有关与此 CVE 关联的所有 OADP 问题列表，请查看以下 JIRA 列表。

如需更多信息，请参阅 CVE-2023-39325 (Rapid Reset Attack) 。

有关 OADP 1.2.3 发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.2.3 解决的问题列表。

4.2.2.3.3. 已知问题

OADP 1.2.3 有以下已知问题：

当凭证 secret 被更新时，数据保护应用程序 (DPA) 不会协调

目前，当您更新 **cloud-credentials** secret 时，OADP Operator 不会协调。这是因为 **cloud-credentials** secret 中没有 OADP 特定的标签或所有者引用。 如果您创建带有不正确的凭证的 **cloud-credentials**

secret，如空数据，Operator 会协调并创建带有空数据的 Backup Storage Location (BSL) 和 registry 部署。因此，当使用正确的凭证更新 **cloud-credentials** secret 时，Operator 不会立即协调来捕获新凭证。

临时解决方案：更新到 OADP 1.3。

(OADP-3327)

4.2.2.4. OADP 1.2.2 发行注记

4.2.2.4.1. 新功能

OpenShift API for Data Protection (OADP) 1.2.2 版本没有包括新的功能。

4.2.2.4.2. 已解决的问题

以下主要问题已在 OADP 1.2.2 中解决：

因为 Pod 安全标准，Restic 恢复部分失败

在之前的 OADP 1.2 版本中，OpenShift Container Platform 4.14 强制执行一个 pod 安全准入 (PSA) 策略，在 Restic 恢复过程中会阻止 pod 的就绪度。

这个问题已在 OADP 1.2.2 版本中解决，同时也在 OADP 1.1.6 中解决。因此，建议用户升级到这些版本。

如需更多信息，请参阅因为更改 PSA 策略，在 OCP 4.14 上进行 Restic 恢复部分失败。(OADP-2094)

使用内部镜像备份应用程序部分失败并显示插件 panicked 错误

在以前的 OADP 1.2 版本中，带有内部镜像的应用程序备份部分会失败，并显示插件 panicked 错误。备份部分失败，在 Velero 日志中出现这个错误：

```
time="2022-11-23T15:40:46Z" level=info msg="1 errors encountered backup up item"
backup=openshift-adp/django-persistent-67a5b83d-6b44-11ed-9cba-902e163f806c
logSource="/remote-source/velero/app/pkg/backup/backup.go:413" name=django-psql-persistent
time="2022-11-23T15:40:46Z" level=error msg="Error backing up item" backup=openshift-
adp/django-persistent-67a5b83d-6b44-11ed-9cba-902e163f8
```

这个问题已在 OADP 1.2.2 中解决。(OADP-1057)。

因为恢复顺序的原因，ACM 集群恢复无法如预期正常工作。

在以前的 OADP 1.2 版本中，因为恢复顺序，ACM 集群恢复无法正常工作。在恢复激活后，ACM 应用程序会在受管集群中被删除并重新创建。(OADP-2505)

由于卷大小不匹配，在备份和恢复时，使用 filesystemOverhead 的虚拟机会失败

在以前的 OADP 1.2 版本中，因为存储供应商实现选择，当应用程序持久性卷声明 (PVC) 存储请求和同一 PVC 的快照大小之间有区别时，使用 filesystemOverhead 的虚拟机在备份和恢复时会失败。 这个问题已在 OADP 1.2.2 的 Data Mover 中解决。(OADP-2144)

OADP 没有包含设置 VolSync 复制源修剪间隔的选项

在之前的 OADP 1.2 版本中，没有设置 VolSync 复制源 **pruneInterval** 的选项。(OADP-2052)

如果 Velero 在多个命名空间中安装，则可能会出现 pod 卷备份失败

在以前的 OADP 1.2 版本中，如果在多个命名空间中安装 Velero，则可能会出现 pod 卷备份失败。(OADP-2409)

当 VSL 使用自定义 secret 时，备份存储位置会进入不可用阶段

在之前的 OADP 1.2 版本中，当卷快照位置使用自定义 secret 时，备份存储位置将进入不可用阶段。(OADP-1737)

有关 OADP 1.2.2 发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.2.2 解决的问题列表。

4.2.2.4.3. 已知问题

在 OADP 1.2.2 发行版本中，以下问题已被明确标识为已知问题：

must-gather 命令无法删除 ClusterRoleBinding 资源

oc adm must-gather 命令无法删除 **ClusterRoleBinding** 资源，这些资源因为准入 Webhook 留在集群中。因此，删除 **ClusterRoleBinding** 资源的请求会被拒绝。(OADP-27730)

```
admission webhook "clusterrolebindings-validation.managed.openshift.io" denied the request:
Deleting ClusterRoleBinding must-gather-p7vwj is not allowed
```

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的OADP 1.2.2 已知问题 列表。

4.2.2.5. OADP 1.2.1 发行注记

4.2.2.5.1. 新功能

OpenShift API for Data Protection (OADP) 1.2.1 版本没有包括新的功能。

4.2.2.5.2. 已解决的问题

有关 OADP 1.2.1 发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.2.1 解决的问题列表。

4.2.2.5.3. 已知问题

在 OADP 1.2.1 发行版本中，以下问题已被明确标识为已知问题：

DataMover Restic retain 和 prune 策略无法按预期正常工作

VolSync 和 Restic 提供的 retention（保留）和 prune（修剪）功能无法按预期正常工作。因为没有在 VolSync 复制中设置修剪间隔的可用选项，所以您必须在 OADP 之外的 S3 存储上远程管理和修剪存储备份。如需了解更多详细信息，请参阅：

- OADP-2052
- OADP-2048
- OADP-2175
- OADP-1690

重要

OADP Data Mover 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.2.1 已知问题列表。

4.2.2.6. OADP 1.2.0 发行注记

OADP 1.2.0 发行注记包括有关新功能、错误修复和已知问题的信息。

4.2.2.6.1. 新功能

资源超时

新的 **resourceTimeout** 选项指定等待各种 Velero 资源的超时时间（以分钟为单位）。这个选项适用于资源，如 Velero CRD 可用性、**volumeSnapshot** 删除和备份存储库可用性。默认持续时间为 10 分钟。

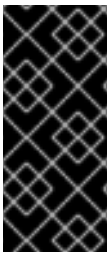
AWS S3 兼容备份存储供应商

您可以在 AWS S3 兼容供应商上备份对象和快照。

4.2.2.6.1.1. 技术预览功能

data Mover

OADP Data Mover 可让您将 Container Storage Interface (CSI) 卷快照备份到远程对象存储。如果启用了 Data Mover，当出现意外删除、集群故障或数据崩溃的情况时，可以使用从对象存储中拉取的 CSI 卷快照来恢复有状态的应用程序。



重要

OADP Data Mover 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

4.2.2.6.2. 已解决的问题

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.2.0 解决的问题列表。

4.2.2.6.3. 已知问题

在 OADP 1.2.0 发行版本中，以下问题已被明确标识为已知问题：

启用的多个 HTTP/2 的 Web 服务器容易受到 DDoS 攻击(Rapid Reset Attack) 的影响

HTTP/2 协议易受拒绝服务攻击的影响，因为请求可以快速重置多个流。服务器需要在没有达到每个连接的最大活跃流数量在服务器端的限制的情况下，设置和处理流。这会导致因为服务器的资源被耗尽而出现拒绝服务的问题。

建议升级到 OADP 1.2.3，它解决了这个问题。

如需更多信息，请参阅 CVE-2023-39325 (Rapid Reset Attack)。

在生成的路由中更改主机名时，可以创建不正确的主机名。

默认情况下，OpenShift Container Platform 集群确保打开 **openshift.io/host.generated: true** 注解，并填写生成的路由以及未生成的路由的字段。

您无法根据生成的和非生成的路由中的集群基本域名来修改 **.spec.host** 字段的值。

如果您修改了 **.spec.host** 字段的值，则无法恢复 OpenShift Container Platform 集群生成的默认值。恢复 OpenShift Container Platform 集群后，Operator 会重置字段的值。

4.2.2.6.4. 升级备注

注意

始终升级到下一个次版本。**不要** 跳过版本。要升级到更新的版本，请一次只升级一个频道。例如，若要从 OpenShift API for Data Protection (OADP) 1.1 升级到 1.3，首先升级到 1.2，然后再升级到 1.3。

4.2.2.6.4.1. 从 OADP 1.1 更改为 1.2

Velero 服务器已从 1.9 更新至 1.11。

在 OADP 1.2 中，**DataProtectionApplication** (DPA) 配置 **dpa.spec.configuration.velero.args** 有以下更改：

- **default-volumes-to-restic** 字段被重命名为 **default-volumes-to-fs-backup**。如果使用 **dpa.spec.configuration.velero.args**，则必须在升级 OADP 后将新名称重新添加到 DPA 中。
- **restic-timeout** 字段被重命名为 **fs-backup-timeout**。如果使用 **dpa.spec.configuration.velero.args**，则必须在升级 OADP 后将新名称重新添加到 DPA 中。
- **restic** 守护进程集被重命名为 **node-agent**。OADP 自动更新守护进程集的名称。
- 自定义资源定义 **resticrepositories.velero.io** 被重命名为 **backuprepositories.velero.io**。
- 自定义资源定义 **resticrepositories.velero.io** 可以从集群中移除。

4.2.2.6.5. 升级步骤

4.2.2.6.5.1. 备份 DPA 配置

您必须备份当前的 **DataProtectionApplication** (DPA) 配置。

流程

- 运行以下命令来保存您当前的 DPA 配置：

  Example

  ```
  $ oc get dpa -n openshift-adp -o yaml > dpa.orig.backup
  ```

4.2.2.6.5.2. 升级 OADP Operator

在升级 OpenShift API for Data Protection (OADP) Operator 时，使用以下序列。

流程

1. 将 OADP Operator 的订阅频道从 **stable-1.1** 改为 **stable-1.2**。
2. 允许 Operator 和容器更新并重启的时间。

其他资源

- 配置 Amazon Web Services
- 对 CSI 快照使用 Data Mover
- 更新安装的 Operator

4.2.2.6.5.3. 将 DPA 转换为新版本

如果使用 **spec.configuration.velero.args** 小节中更新的字段，您必须配置 **DataProtectionApplication** (DPA) 清单以使用新的参数名称。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 选择 **Provided APIs**，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 显示当前 DPA 参数。

   当前 DPA 示例

   ```
   spec:
     configuration:
       velero:
         args:
           default-volumes-to-fs-backup: true
           default-restic-prune-frequency: 6000
           fs-backup-timeout: 600
   # ...
   ```

4. 更新 DPA 参数：
5. 更新 DPA 参数名称而不更改它们的值：

   a. 将 **default-volumes-to-restic** 键更改为 **default-volumes-to-fs-backup**。

   b. 将 **default-restic-prune-frequency** 键更改为 **default-repo-maintain-frequency**。

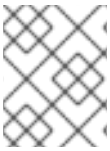
   c. 将 **restic-timeout** 键更改为 **fs-backup-timeout**。

   .Example 更新的 DPA

   ```
   spec:
     configuration:
       velero:
   ```

```
    args:
      default-volumes-to-fs-backup: true
      default-repo-maintain-frequency: 6000
      fs-backup-timeout: 600
# ...
```

6. 等待 DPA 成功协调。



注意

Restic 文件系统备份的默认超时值为一小时。在 OADP 1.3.1 及更高版本中，Restic 和 Kopia 的默认超时值为四小时。

4.2.2.6.5.4. 验证升级

使用以下步骤验证升级。

流程

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                   TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                        1/1     1            1           96s

NAME                                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                        1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME          PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1  Available   1s               3d16h   true
```

4.2.3. OADP 1.1 发行注记

OpenShift API for Data Protection (OADP) 1.1 的发行注记介绍了新的功能和增强功能、已弃用的功能、产品建议、已知问题和解决问题。

4.2.3.1. OADP 1.1.8 发行注记

OpenShift API for Data Protection (OADP) 1.1.8 发现注记列出了所有已知的问题。这个版本没有已解决的问题。

4.2.3.1.1. 已知问题

有关 OADP 1.1.8 中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.1.8 已知问题列表。

4.2.3.2. OADP 1.1.7 发行注记

OADP 1.1.7 发行注记列出了所有已解决的问题和已知问题。

4.2.3.2.1. 已解决的问题

以下主要问题已在 OADP 1.1.7 中解决：

启用的多个 HTTP/2 的 Web 服务器容易受到 DDoS 攻击(Rapid Reset Attack) 的影响

在之前的 OADP 1.1 版本中，HTTP/2 协议易受拒绝服务攻击的影响，因为请求可以快速重置多个流。服务器需要在没有达到每个连接的最大活跃流数量在服务器端的限制的情况下，设置和处理流。这会导致因为服务器的资源被耗尽而出现拒绝服务的问题。有关与此 CVE 关联的所有 OADP 问题列表，请查看以下 JIRA 列表。

如需更多信息，请参阅 CVE-2023-39325 (Rapid Reset Attack) 。

有关 OADP 1.1.7 发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.1.7 解决的问题列表。

4.2.3.2.2. 已知问题

OADP 1.1.7 发行版本中没有已知的问题。

4.2.3.3. OADP 1.1.6 发行注记

OADP 1.1.6 发行注记列出了任何新功能、解决的问题和错误以及已知的问题。

4.2.3.3.1. 已解决的问题

由于 Pod 安全标准，Restic 恢复部分失败

OCP 4.14 引入了 pod 安全标准，这意味着 **privileged** 配置集是 **enforced**。在以前的 OADP 版本中，这个配置集会导致 pod 收到 **permission denied** 错误。造成这个问题的原因是恢复顺序。pod 在安全性上下文约束 (SCC) 资源之前创建。由于此 pod 违反了 pod 安全标准，因此 pod 被拒绝，然后失败。OADP-2420

恢复作业资源部分失败

在以前的 OADP 版本中，恢复作业资源在 OCP 4.14 中部分失败。旧的 OCP 版本中没有此问题。此问题是由一个额外的标签指向作业资源造成的，这些资源在旧的 OCP 版本中不存在。OADP-2530

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.1.6 解决的问题列表。

4.2.3.3.2. 已知问题

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.1.6 已知问题列表。

4.2.3.4. OADP 1.1.5 发行注记

OADP 1.1.5 发行注记列出了任何新功能、解决的问题和错误以及已知的问题。

4.2.3.4.1. 新功能

这个版本 OADP 是一个服务发行版本。此版本不会添加新功能。

4.2.3.4.2. 已解决的问题

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.1.5 解决的问题列表。

4.2.3.4.3. 已知问题

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.1.5 已知问题列表。

4.2.3.5. OADP 1.1.4 发行注记

OADP 1.1.4 发行注记列出了任何新功能、解决的问题和错误以及已知的问题。

4.2.3.5.1. 新功能

这个版本 OADP 是一个服务发行版本。此版本不会添加新功能。

4.2.3.5.2. 已解决的问题

添加对所有 velero 部署服务器参数的支持

在之前的 OADP 版本中，OADP 无法促进所有上游 Velero 服务器参数的支持。这个问题已在 OADP 1.1.4 中解决，所有上游 Velero 服务器参数都支持。OADP-1557

当存在多个 VSR 用于恢复名称和 pvc 名称时，数据 Mover 可以从不正确的快照中恢复

在之前的 OADP 版本中，如果集群中有多个带有相同的 **restore** 名和 PersistentVolumeClaim (pvc) 名的多个 Volume Snapshot Restore (VSR) 资源，OADP Data Mover 可能会从不正确的快照进行恢复。OADP-1822

Cloud Storage API BSLs 需要 OwnerReference

在之前的 OADP 版本中，ACM 备份调度失败，因为使用 **dpa.spec.backupLocations.bucket** 创建的 Backup Storage Locations (BSLs) 中缺少 **OwnerReference**。OADP-1511

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.1.4 解决的问题列表。

4.2.3.5.3. 已知问题

这个版本有以下已知问题：

OADP 备份可能会失败，因为集群中的 UID/GID 范围可能已更改

OADP 备份可能会失败，因为在恢复应用程序的集群上可能会更改 UID/GID 范围，因此 OADP 不会备份和恢复 OpenShift Container Platform UID/GID 范围元数据。要避免这个问题，如果支持的应用程序需要特定的 UUID，请确保恢复时范围可用。一个额外的解决方法是允许 OADP 在恢复操作中创建命名空间。

如果 ArgoCD 使用了 ArgoCD 的标签，则恢复可能会失败

在处理过程中如果使用了 ArgoCD，则恢复可能会失败。这是因为 ArgoCD 使用的一个标签 **app.kubernetes.io/instance** 造成的。该标签用于标识 ArgoCD 需要管理的资源，它可能会导致与 OADP 在恢复过程中管理资源的过程有冲突。要临时解决这个问题，将 ArgoCD YAML 上的 **.spec.resourceTrackingMethod** 设置为 **annotation+label** 或 **annotation**。如果问题仍然存在，请在开始恢复前禁用 ArgoCD，并在恢复完成后再次启用它。

OADP Velero 插件返回 "received EOF, stop recv loop" 信息

Velero 插件作为单独的进程启动。当 Velero 操作完成后，无论是否成功，它们都会退出。因此，如果您看到 **received EOF, stopping recv loop** 消息，这并不意味着发生了错误。消息显示插件操作已完成。OADP-2176

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.1.4 已知问题列表。

4.2.3.6. OADP 1.1.3 发行注记

OADP 1.1.3 发行注记列出了任何新功能、解决的问题和错误以及已知的问题。

4.2.3.6.1. 新功能

这个版本 OADP 是一个服务发行版本。此版本不会添加新功能。

4.2.3.6.2. 已解决的问题

有关本发行版本中解决的所有问题的完整列表，请参阅 JIRA 中的 OADP 1.1.3 解决的问题列表。

4.2.3.6.3. 已知问题

有关本发行版本中所有已知问题的完整列表，请参阅 JIRA 中的 OADP 1.1.3 已知问题 列表。

4.2.3.7. OADP 1.1.2 发行注记

OADP 1.1.2 发行注记包括产品建议、修复的错误列表和已知问题的描述。

4.2.3.7.1. 产品建议

VolSync

要准备从 VolSync 0.5.1 升级到 VolSync **stable** 频道中的最新版本，您必须运行以下命令在 **openshift-adp** 命名空间中添加此注解：

```
$ oc annotate --overwrite namespace/openshift-adp volsync.backube/privileged-movers='true'
```

Velero

在这个发行版本中，Velero 已从 1.9.2 升级到 1.9.5 版本。

Restic

在本发行版本中，Restic 从 0.13.1 升级到 0.14.0 版本。

4.2.3.7.2. 已解决的问题

本发行版本中解决了以下问题：

- OADP-1150
- OADP-290
- OADP-1056

4.2.3.7.3. 已知问题

这个版本有以下已知问题：

- OADP 目前不支持使用 Velero (**OADP-778**) 中的 restic 备份和恢复 AWS EFS 卷。
- CSI 备份可能会因为每个 PVC 的 **VolumeSnapshotContent** 快照限制而失败。
  您可以创建同一持久性卷声明(PVC)的许多快照，但无法调度定期创建快照：
  - 对于 CephFS，您可以为每个 PVC 创建最多 100 个快照。(**OADP-804**)
  - 对于 RADOS 块设备 (RBD)，您可以为每个 PVC 创建最多 512 个快照。(**OADP-975**)

  如需更多信息，请参阅卷快照。

4.2.3.8. OADP 1.1.1 发行注记

OADP 1.1.1 发行注记包括产品建议和已知问题的描述。

4.2.3.8.1. 产品建议

在安装 OADP 1.1.1 前，建议安装 VolSync 0.5.1 或升级到它。

4.2.3.8.2. 已知问题

这个版本有以下已知问题：

- 启用的多个 HTTP/2 的 Web 服务器容易受到 DDoS 攻击(Rapid Reset Attack) 的影响
  HTTP/2 协议易受拒绝服务攻击的影响，因为请求可以快速重置多个流。服务器需要在没有达到

每个连接的最大活跃流数量在服务器端的限制的情况下，设置和处理流。这会导致因为服务器的资源被耗尽而出现拒绝服务的问题。有关与此 CVE 关联的所有 OADP 问题列表，请查看以下 JIRA 列表。

建议升级到 OADP 1.1.7 或 1.2.3，从而解决了这个问题。

如需更多信息，请参阅 CVE-2023-39325 (Rapid Reset Attack) 。

- OADP 目前不支持使用 Velero (**OADP-778**) 中的 restic 备份和恢复 AWS EFS 卷。
- CSI 备份可能会因为每个 PVC 的 **VolumeSnapshotContent** 快照限制而失败。
  您可以创建同一持久性卷声明(PVC)的许多快照，但无法调度定期创建快照：
  - 对于 CephFS，您可以为每个 PVC 创建最多 100 个快照。
  - 对于 RADOS 块设备 (RBD)，您可以为每个 PVC 创建最多 512 个快照。(**OADP-804**) 和 (**OADP-975**)
    如需更多信息，请参阅卷快照。

4.3. OADP 功能和插件

OpenShift API 用于数据保护(OADP)功能，提供用于备份和恢复应用的选项。

默认插件使 Velero 能够与某些云供应商集成，并备份和恢复 OpenShift Container Platform 资源。

4.3.1. OADP 功能

OpenShift API 用于数据保护(OADP)支持以下功能：

Backup

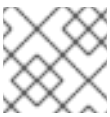
您可以使用 OADP 备份 OpenShift Platform 中的所有应用程序，或者您可以根据类型、命名空间或标签过滤资源。
OADP 通过将 Kubernetes 对象和内部镜像保存为对象存储上的存档文件来备份 Kubernetes 对象和内部镜像。OADP 使用原生云快照 API 或通过容器存储接口(CSI)创建快照来备份持久性卷(PV)。对于不支持快照的云供应商，OADP 使用 Restic 备份资源和 PV 数据。

注意

您必须从应用程序的备份中排除 Operator，以便成功备份和恢复。

恢复

您可以从备份中恢复资源和 PV。您可以恢复备份中的所有对象，或者根据命名空间、PV 或标签过滤对象。



注意

您必须从应用程序的备份中排除 Operator，以便成功备份和恢复。

调度

您可以通过指定的间隔调度备份。

钩子

您可以使用 hook 在 pod 上的容器中运行命令，如 **fsfreeze** 以冻结文件系统。您可以将 hook 配置为在备份或恢复之前或之后运行。恢复 hook 可以在 init 容器或应用程序容器中运行。

4.3.2. OADP 插件

用于数据保护(OADP)的 OpenShift API 提供了与存储供应商集成的默认 Velero 插件，以支持备份和恢复操作。您可以根据 Velero 插件创建自定义插件。

OADP 还为 OpenShift Container Platform 资源备份、OpenShift Virtualization 资源备份和 Container Storage Interface(CSI)快照提供了插件。

表 4.1. OADP 插件

OADP 插件	功能	存储位置
aws	备份和恢复 Kubernetes 对象。	AWS S3
	使用快照备份和恢复卷。	AWS EBS
azure	备份和恢复 Kubernetes 对象。	Microsoft Azure Blob 存储
	使用快照备份和恢复卷。	Microsoft Azure 管理的磁盘
gcp	备份和恢复 Kubernetes 对象。	Google Cloud Storage
	使用快照备份和恢复卷。	Google Compute Engine 磁盘
openshift	备份和恢复 OpenShift Container Platform 资源。[1]	对象存储
kubevirt	备份和恢复 OpenShift Virtualization 资源。[2]	对象存储
csi	使用 CSI 快照备份和恢复卷。[3]	支持 CSI 快照的云存储
vsm	VolumeSnapshotMover 将快照从集群重新定位到一个在恢复过程中使用的对象存储中以恢复有状态的应用程序，例如集群删除的情况。[4]	对象存储

1. 必需。
2. 虚拟机磁盘使用 CSI 快照或 Restic 备份。
3. **csi** 插件使用 Kubernetes CSI 快照 API。
   - OADP 1.1 或更高版本使用 **snapshot.storage.k8s.io/v1**
   - OADP 1.0 使用 **snapshot.storage.k8s.io/v1beta1**

4. 仅限 OADP 1.2。

4.3.3. 关于 OADP Velero 插件

安装 Velero 时，您可以配置两种类型的插件：

- 默认云供应商插件
- 自定义插件

两种类型的插件都是可选的，但大多数用户都会至少配置一个云供应商插件。

4.3.3.1. 默认 Velero 云供应商插件

当您在部署过程中配置 **oadp_v1alpha1_dpa.yaml** 文件时，您可以安装以下默认 Velero 云供应商插件：

- **aws** (Amazon Web Services)
- **gcp** (Google Cloud Platform)
- **azure** (Microsoft Azure)
- **openshift** (OpenShift Velero plugin)
- **csi** (Container Storage Interface)
- **kubevirt** (KubeVirt)

在部署过程中，您可以在 **oadp_v1alpha1_dpa.yaml** 文件中指定所需的默认插件。

示例文件

以下 **.yaml** 文件会安装 **openshift**、**aws**、**azure** 和 **gcp** 插件：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - aws
      - azure
      - gcp
```

4.3.3.2. 自定义 Velero 插件

您可在部署期间配置 **oadp_v1alpha1_dpa.yaml** 文件时，通过指定插件 **镜像**和**名称**来安装自定义 Velero 插件。

在部署过程中，您可以在 **oadp_v1alpha1_dpa.yaml** 文件中指定所需的自定义插件。

示例文件

以下 **.yaml** 文件会安装默认的 **openshift**、**azure** 和 **gcp** 插件，以及一个自定义插件，其名称为 **custom-plugin-example** 和镜像 **quay.io/example-repo/custom-velero-plugin** ：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: dpa-sample
spec:
 configuration:
   velero:
     defaultPlugins:
     - openshift
     - azure
     - gcp
     customPlugins:
     - name: custom-plugin-example
       image: quay.io/example-repo/custom-velero-plugin
```

4.3.3.3. Velero 插件返回 "received EOF, stop recv loop" 信息

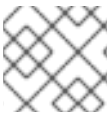
注意

Velero 插件作为单独的进程启动。当 Velero 操作完成后，无论是否成功，它们都会退出。接收到 **received EOF, stopping recv loop** 消息表示插件操作已完成。这并不意味着发生了错误。

4.3.4. OADP 支持的构架

OpenShift API for Data Protection (OADP) 支持以下构架：

- AMD64
- ARM64
- PPC64le
- s390x



注意

OADP 1.2.0 及更新版本支持 ARM64 架构。

4.3.5. OADP 支持 IBM Power 和 IBM Z

OpenShift API for Data Protection (OADP) 是一个平台中立的平台。以下的信息只与 IBM Power® 和 IBM Z® 相关。

- OADP 1.1.7 对于 IBM Power® 和 IBM Z® 均针对 OpenShift Container Platform 4.11 进行了成功测试。以下章节提供了在这些系统的备份位置上 OADP 1.1.7 的测试和支持信息。
- OADP 1.2.3 对于 IBM Power® 和 IBM Z®，针对 OpenShift Container Platform 4.12、4.13、4.14 和 4.15 进行了成功测试。以下章节提供了在这些系统的备份位置上 OADP 1.2.3 的测试和支持信息。

- OADP 1.3.1 对于 IBM Power® 和 IBM Z®，针对 OpenShift Container Platform 4.13、4.14 和 4.15 进行了成功测试。以下章节提供了在这些系统的备份位置上 OADP 1.3.1 的测试和支持信息。

4.3.5.1. OADP 支持使用 IBM Power 的目标备份位置

- 在 IBM Power® 中运行 OpenShift Container Platform 4.11 和 4.12，以及 OpenShift API for Data Protection (OADP) 1.1.7 已针对 AWS S3 备份位置目标成功进行了测试。虽然测试只涉及一个 AWS S3 目标，但红帽也支持在 OpenShift Container Platform 4.11 和 4.12 中运行 IBM Power®，以及针对所有 S3 备份位置目标（不是 AWS）运行 OADP 1.1.7。

- 使用 OpenShift Container Platform 4.12、4.13、4.14 和 4.15 和 OADP 1.2.3 运行的 IBM Power® 已针对 AWS S3 备份位置目标成功进行了测试。虽然测试只涉及一个 AWS S3 目标，但红帽也支持针对所有 S3 备份位置目标，使用 OpenShift Container Platform 4.12、4.13. 4.14 和 4.15 和 OADP 1.2.3 运行 IBM Power®。

- 使用 OpenShift Container Platform 4.13、4.14 和 4.15 和 OADP 1.3.1 运行的 IBM Power® 已针对 AWS S3 备份位置目标成功进行了测试。虽然测试只涉及一个 AWS S3 目标，但红帽也支持针对所有的 S3 备份位置目标（不是 AWS），在 IBM Power® 中使用 OpenShift Container Platform 4.13、4.14 和 4.15 和 OADP 1.3.1。

4.3.5.2. OADP 测试并支持使用 IBM Z 的目标备份位置

- 在 IBM Z® 中运行 OpenShift Container Platform 4.11 和 4.12，以及 OpenShift API for Data Protection (OADP) 1.1.7 已针对 AWS S3 备份位置目标成功进行了测试。虽然测试只涉及一个 AWS S3 目标，但红帽也支持在 OpenShift Container Platform 4.11 和 4.12 中运行 IBM Z®，以及针对所有的 S3 备份位置目标（不是 AWS）运行 OADP 1.1.7。

- 使用 OpenShift Container Platform 4.12、4.13、4.14 和 4.15 和 OADP 1.2.3 运行的 IBM Z® 已针对 AWS S3 备份位置目标成功进行了测试。虽然测试只涉及一个 AWS S3 目标，但红帽也支持针对所有的 S3 备份位置目标，使用 OpenShift Container Platform 4.12、4.13、4.14 和 4.15 和 OADP 1.2.3 运行 IBM Z®。

- 使用 OpenShift Container Platform 4.13、4.14 和 4.15 和 OADP 1.3.1 运行的 IBM Z® 已针对 AWS S3 备份位置目标成功进行了测试。虽然测试只涉及一个 AWS S3 目标，但红帽也支持针对所有的 S3 备份位置目标（不是 AWS），在 IBM Z® 中使用 OpenShift Container Platform 4.13、4.14 和 4.15 和 OADP 1.3.1。

4.3.5.2.1. 使用 IBM Power (R) 和 IBM Z (R) 平台的 OADP 已知问题

- 目前，在 IBM Power® 和 IBM Z® 平台上部署的单节点 OpenShift 集群的备份方法限制。目前，只有 NFS 存储与这些平台上的单节点 OpenShift 集群兼容。另外，只支持文件系统备份 (FSB) 方法，如 Kopia 和 Restic 用于备份和恢复操作。当前没有解决此问题的方法。

4.3.6. OADP 插件的已知问题

以下章节介绍了 OpenShift API for Data Protection (OADP) 插件中的已知问题：

4.3.6.1. 因为缺少 secret，Velero 插件在镜像流备份过程中会出现错误

当在数据保护应用程序(DPA)范围外管理备份和备份存储位置(BSL)时，OADP 控制器，这意味着 DPA 协调不会创建相关的 **oadp-<bsl_name>-<bl_provider>-registry-secret**。

当备份运行时，OpenShift Velero 插件在镜像流备份中出现错误，并显示以下错误：

```
024-02-27T10:46:50.028951744Z time="2024-02-27T10:46:50Z" level=error msg="Error backing up
```

```
item"
backup=openshift-adp/<backup name> error="error executing custom action
(groupResource=imagestreams.image.openshift.io,
namespace=<BSL Name>, name=postgres): rpc error: code = Aborted desc = plugin panicked:
runtime error: index out of range with length 1, stack trace: goroutine 94…
```

4.3.6.1.1. 临时解决方案以避免出现错误

要避免 Velero 插件 panic 错误，请执行以下步骤：

1. 使用相关标签标记自定义 BSL：

```
$ oc label BackupStorageLocation <bsl_name> app.kubernetes.io/component=bsl
```

2. 在标记 BSL 后，等待 DPA 协调。

注意

您可以通过对 DPA 本身进行任何更改来强制进行协调。

3. 当 DPA 协调时，确认相关的 **oadp-<bsl_name>-<bsl_provider>-registry-secret** 已被填充到其中：

```
$ oc -n openshift-adp get secret/oadp-<bsl_name>-<bsl_provider>-registry-secret -o json | jq
-r '.data'
```

4.3.6.2. OpenShift ADP Controller 分段错误

如果您在同时启用了 **cloudstorage** 和 **restic** 的情况下配置 DPA，**openshift-adp-controller-manager** pod 会无限期重复崩溃和重启过程，直到 pod 出现一个崩溃循环分段错误为止。

您只能定义 **velero** 或 **cloudstorage**，它们是互斥的字段。

- 如果您同时定义了 **velero** 和 **cloudstorage**，**openshift-adp-controller-manager** 会失败。
- 如果 **velero** 和 **cloudstorage** 都没有定义，**openshift-adp-controller-manager** 也将失败。

有关此问题的更多信息，请参阅 OADP-1054。

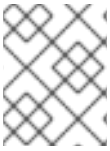
4.3.6.2.1. OpenShift ADP Controller 分段错误临时解决方案

在配置一个 DPA 时，您必须定义 **velero** 或 **cloudstorage**。如果您在 DPA 中同时定义了这两个 API，**openshift-adp-controller-manager** pod 会失败，并显示崩溃循环分段错误。

4.4. 安装和配置 OADP

4.4.1. 关于安装 OADP

作为集群管理员，您可以通过安装 OADP Operator 来为数据保护(OADP)安装 OpenShift API。OADP Operator 安装 Velero 1.12。



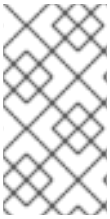
注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

要备份 Kubernetes 资源和内部镜像，必须将对象存储用作备份位置，如以下存储类型之一：

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- 多云对象网关
- IBM Cloud® Object Storage S3
- AWS S3 兼容对象存储，如 Multicloud 对象网关或 MinIO

您可以为每个单独的 OADP 部署在同一命名空间中配置多个备份存储位置。



注意

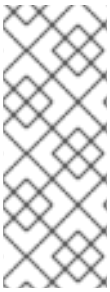
除非另有指定，"NooBaa" 指的是提供轻量级对象存储的开源项目，而 "Multicloud Object Gateway (MCG) " 是指 NooBaa 的红帽发行版本。

如需有关 MCG 的更多信息，请参阅使用应用程序访问多云对象网关。

重要

CloudStorage API（它自动为对象存储创建一个存储桶）只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。



注意

当使用 **CloudStorage** 对象，并希望 OADP 使用 **CloudStorage** API 自动创建 S3 存储桶以用作 **BackupStorageLocation** 时，**CloudStorage** API 只是一个技术预览功能。

CloudStorage API 支持通过指定一个现有的 S3 存储桶来手动创建 **BackupStorageLocation** 对象。自动创建 S3 存储桶的 **CloudStorage** API 目前只为 AWS S3 存储启用。

您可以使用快照或文件系统备份 (FSB) 备份持久性卷 (PV)。

要使用快照备份 PV，您必须有一个支持原生快照 API 或 Container Storage Interface(CSI)快照的云供应商，如以下云供应商之一：

- Amazon Web Services
- Microsoft Azure

- Google Cloud Platform
- 启用 CSI 快照的云供应商，如 OpenShift Data Foundation

注意

如果要在 OCP 4.11 及之后的版本中使用 CSI 备份，请安装 OADP 1.1.*x*。

OADP 1.0.*x* 不支持 OCP 4.11 及更高版本上的 CSI 备份。OADP 1.0. *x* 包括 Velero 1.7.*x*，并需要 API 组 **snapshot.storage.k8s.io/v1beta1**，这在 OCP 4.11 及更高版本中不存在。

如果您的云供应商不支持快照，或者您的存储是 NFS，您可以在对象存储中使用文件系统备份对应用程序进行备份：Kopia 或 Restic 来备份应用程序。

您可以创建一个默认 **Secret**，然后安装数据保护应用程序。

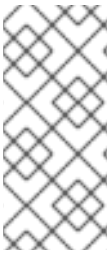
4.4.1.1. AWS S3 兼容备份存储供应商

OADP 与许多对象存储供应商兼容，用于不同的备份和恢复操作。一些对象存储供应商被完全支持，一些不被支持但可以正常工作，另外一些有已知的限制。

4.4.1.1.1. 支持的备份存储供应商

通过 AWS 插件，以下 AWS S3 兼容对象存储供应商被 OADP 完全支持作为备份存储：

- MinIO
- 多云对象网关 (MCG)
- Amazon Web Services (AWS) S3
- IBM Cloud® Object Storage S3



注意

支持以下兼容对象存储供应商，并有自己的 Velero 对象存储插件：

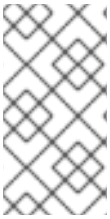
- Google Cloud Platform (GCP)
- Microsoft Azure

4.4.1.1.2. 不支持的备份存储供应商

通过 AWS 插件，以下 AWS S3 兼容对象存储供应商可以与 Velero 一起正常工作作为备份存储，但它们不被支持，且还没有经过红帽测试：

- Oracle Cloud
- DigitalOcean
- NooBaa，除非使用 Multicloud Object Gateway (MCG) 安装
- Tencent Cloud
- Ceph RADOS v12.2.7

- Quobyte
- Cloudian HyperStore



注意

除非另有指定，"NooBaa" 指的是提供轻量级对象存储的开源项目，而 "Multicloud Object Gateway (MCG) " 是指 NooBaa 的红帽发行版本。

如需有关 MCG 的更多信息，请参阅使用应用程序访问多云对象网关。

4.4.1.1.3. 带有已知限制的备份存储供应商

通过 AWS 插件，以下 AWS S3 兼容对象存储供应商可以与 Velero 搭配使用，但有一些已知的限制：

- Swift - 它可以作为备份存储的备份存储位置，但对于基于文件系统的卷备份和恢复，它与 Restic 不兼容。

4.4.1.2. 为 OpenShift Data Foundation 上的灾难恢复配置多云对象网关 (MCG)

如果您在 OpenShift Data Foundation 上为 MCG bucket **backupStorageLocation** 使用集群存储，请将 MCG 配置为外部对象存储。

警告

将 MCG 配置为外部对象存储可能会导致备份不可用。

注意

除非另有指定，"NooBaa" 指的是提供轻量级对象存储的开源项目，而 "Multicloud Object Gateway (MCG) " 是指 NooBaa 的红帽发行版本。

如需有关 MCG 的更多信息，请参阅使用应用程序访问多云对象网关。

流程

- 将 MCG 配置为外部对象存储，如为混合或多云添加存储资源中所述。

其他资源

- Velero 文档中的备份和恢复位置概述

4.4.1.3. 关于 OADP 更新频道

安装 OADP Operator 时，您可以选择 *更新频道*。这个频道决定到您接收到的 OADP Operator 和 Velero 的哪些升级。您可以随时切换频道。

可用的更新频道如下：

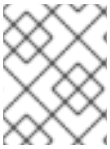
- **stable** 频道现已弃用。**stable** 频道包含 **oadp.v1.1.z** 和目 **oadp.v1.0.z** 的更老版本的 OADP **ClusterServiceVersion** 的补丁 (z-stream 更新)。
- **stable-1.0** 频道包含 **oadp.v1.0.*z***，它是最新的 OADP 1.0 **ClusterServiceVersion**。
- **stable-1.1** 频道包含 **oadp.v1.1.*z***，它是最新的 OADP 1.1 **ClusterServiceVersion**。
- **stable-1.2** 频道包括 **oadp.v1.2.*z***，最新的 OADP 1.2 **ClusterServiceVersion**。
- **stable-1.3** 频道包含 **oadp.v1.3.*z***，它是最新的 OADP 1.3 **ClusterServiceVersion**。

哪个更新频道适合您？

- **stable** 频道现已弃用。如果您已使用 stable 频道，则继续从 **oadp.v1.1.*z*** 获取更新。
- 选择 **stable-1.*y*** 更新频道来安装 OADP 1.*y*，并继续为其接受补丁。如果您选择此频道，您将收到版本 1.*y.z* 的所有 z-stream。

何时需要切换更新频道？

- 如果您安装了 OADP 1.*y*，并且只想接收那个 y-stream 的补丁，则必须从 **stable** 更新频道切换到 **stable-1.*y*** 更新频道。然后，您将收到版本 1.*y.z* 的所有 z-stream 补丁。
- 如果您安装了 OADP 1.0，希望升级到 OADP 1.1，然后只接收 OADP 1.1 的补丁，则必须从 **stable-1.0** 更新频道切换到 **stable-1.1** 更新频道。然后，您将收到版本 1.1.*z* 的所有 z-stream 补丁。
- 如果您安装了 OADP 1.*y*，且 *y* 大于 0，并且希望切换到 OADP 1.0，则必须 *卸载* OADP Operator，然后使用 **stable-1.0** 更新频道重新安装。然后，您将收到 1.0.*z* 版本的所有 z-stream 补丁。



注意

您无法通过切换更新频道从 OADP 1.*y* 切换到 OADP 1.0。您必须卸载 Operator，然后重新安装它。

4.4.1.4. 在多个命名空间中安装 OADP

您可以将 OpenShift API for Data Protection (OADP) 安装到同一集群中的多个命名空间中，以便多个项目所有者可以管理自己的 OADP 实例。这个用例已通过文件系统备份 (FSB) 和 Container Storage Interface (CSI) 进行验证。

您可以根据本文档中包含的每个平台流程指定安装每个 OADP 实例，并有以下额外的要求：

- 同一集群中的所有 OADP 部署都必须相同版本，如 1.1.4。**不支持**在同一集群中安装 OADP 的不同版本。
- 每个 OADP 部署都必须具有一组唯一的凭证和唯一的 **BackupStorageLocation** 配置。您还可以在同一命名空间中使用多个 **BackupStorageLocation** 配置。
- 默认情况下，每个 OADP 部署在不同的命名空间中都有集群级别的访问权限。OpenShift Container Platform 管理员需要仔细检查安全性和 RBAC 设置，并对它们进行任何更改，以确保每个 OADP 实例都有正确的权限。

其他资源

- 集群服务版本

4.4.1.5. 基于收集到的数据的 Velero CPU 和内存要求

以下建议基于在扩展和性能实验室中观察到的性能。备份和恢复资源可能会受到插件类型、备份或恢复所需的资源数量，以及与这些资源相关的持久性卷 (PV) 中包含的相应数据。

4.4.1.5.1. 配置的 CPU 和内存要求

配置类型	[1] 平均用量	[2] 大使用	resourceTimeouts
CSI	Velero: CPU- Request 200m, Limits 1000m 内存 - Request 256Mi, Limits 1024Mi	Velero: CPU- Request 200m, Limits 2000m 内存- Request 256Mi, Limits 2048Mi	N/A
Restic	[3] Restic: CPU- Request 1000m, Limits 2000m 内存 - Request 16Gi, Limits 32Gi	[4] Restic: CPU - Request 2000m, Limits 8000m 内存 - Request 16Gi, Limits 40Gi	900m
[5] 数据 Mover	N/A	N/A	10m - 平均使用 60m - 大型使用

1. 平均使用 - 将这些设置用于大多数使用情况。
2. 大型使用 - 使用这些设置进行大型使用情况，如大型 PV (500GB 使用情况)、多个命名空间 (100+)或单个命名空间中的多个 pod (2000 pods+)，以及对涉及大型数据集进行备份和恢复的最佳性能。
3. Restic 资源使用量与数据的数量和数据类型对应。例如，许多小文件或大量数据都可能导致 Restic 使用大量资源。在 Velero 文档中 500m 是默认设置，但在我们的大多数测试中，我们认为 200m request 和 1000m limit 是比较适当的设置。如 Velero 文档中所述，除了环境限制外，具体的 CPU 和内存用量还取决于文件和目录的规模。
4. 增加 CPU 会对改进备份和恢复时间有重大影响。
5. Data Mover - Data Mover 默认 resourceTimeout 为 10m。我们的测试显示恢复大型 PV (500GB 使用量)，需要将 resourceTimeout 增加到 60m。

注意

本指南中列出的资源要求仅用于平均使用。对于大型用途，请按照上表所述调整设置。

4.4.1.5.2. 用于大用量的 NodeAgent CPU

测试显示，在使用 OpenShift API for Data Protection (OADP) 时，增加 **NodeAgent** CPU 可以显著提高备份和恢复的时间。

重要

因为 Kopia 会以激进的方式消耗资源，因此不建议在没有在运行生产负载的节点上进行限制的环境中使用 Kopia。但是，如果运行 Kopia 时有太低的限制会导致 CPU 的限制，并减慢备份和恢复的速度。测试显示，在具有 20 个内核和 32 Gi 内存的环境中运行 Kopia，支持在跨多个命名空间或在一个命名空间中的 2000 个 pod 中对 100 GB 数据进行备份和恢复操作。

在具有这样配置的环境中的测试中没有出现 CPU 限制或内存饱和的问题。

您可以按照 更改 rook-ceph pod 上的 CPU 和内存资源 中的步骤在 Ceph MDS pod 中设置这些限制。

您需要在存储集群自定义资源 (CR) 中添加以下行来设置限制：

```
resources:
  mds:
    limits:
      cpu: "3"
      memory: 128Gi
    requests:
      cpu: "3"
      memory: 8Gi
```

4.4.2. 安装 OADP Operator

您可以使用 Operator Lifecycle Manager (OLM) 在 OpenShift Container Platform 4.16 上安装 Data Protection (OADP) Operator 的 OpenShift API。

OADP Operator 安装 Velero 1.12。

先决条件

- 您必须以具有 **cluster-admin** 权限的用户身份登录。

流程

1. 在 OpenShift Container Platform Web 控制台中，点击 **Operators → OperatorHub**。
2. 使用 **Filter by keyword** 字段查找 **OADP Operator**。
3. 选择 **OADP Operator** 并点 **Install**。
4. 点 **Install** 在 **openshift-adp** 项目中安装 Operator。
5. 点 **Operators → Installed Operators** 来验证安装。

4.4.2.1. OADP-Velero-OpenShift Container Platform 版本关系

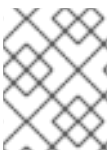
OADP 版本	Velero 版本	OpenShift Container Platform 版本
1.1.0	1.9	4.9 及更新的版本

OADP 版本	Velero 版本	OpenShift Container Platform 版本
1.1.1	1.9	4.9 及更新的版本
1.1.2	1.9	4.9 及更新的版本
1.1.3	1.9	4.9 及更新的版本
1.1.4	1.9	4.9 及更新的版本
1.1.5	1.9	4.9 及更新的版本
1.1.6	1.9	4.11 及更新的版本
1.1.7	1.9	4.11 及更新的版本
1.2.0	1.11	4.11 及更新的版本
1.2.1	1.11	4.11 及更新的版本
1.2.2	1.11	4.11 及更新的版本
1.2.3	1.11	4.11 及更新的版本
1.3.0	1.12	4.12 及更新的版本

4.4.3. 为 AWS S3 兼容存储的数据保护配置 OpenShift API

您可以通过安装 OADP Operator，使用 Amazon Web Services (AWS) S3 兼容存储安装 OpenShift API for Data Protection (OADP)。Operator 会安装 Velero 1.12。

IBM Cloud® S3 支持作为 AWS S3 兼容备份存储供应商。



注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

您可以为 Velero 配置 AWS，创建一个默认 **Secret**，然后安装数据保护应用程序。如需了解更多详细信息，请参阅安装 OADP Operator。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

4.4.3.1. 配置 Amazon Web Services

您可以为 OpenShift API 配置 Amazon Web Services(AWS)以进行数据保护(OADP)。

先决条件

- 已安装 AWS CLI。

流程

1. 设置 **BUCKET** 变量：

```
$ BUCKET=<your_bucket>
```

2. 设置 **REGION** 变量：

```
$ REGION=<your_region>
```

3. 创建 AWS S3 存储桶：

```
$ aws s3api create-bucket \
    --bucket $BUCKET \
    --region $REGION \
    --create-bucket-configuration LocationConstraint=$REGION 1
```

1 **us-east-1** 不支持 **LocationConstraint**。如果您的区域是 **us-east-1**，忽略 **--create-bucket-configuration LocationConstraint=$REGION**。

4. 创建一个 IAM 用户：

```
$ aws iam create-user --user-name velero 1
```

1 如果要使用 Velero 备份具有多个 S3 存储桶的集群，请为每个集群创建一个唯一用户名。

5. 创建 **velero-policy.json** 文件：

```
$ cat > velero-policy.json <<EOF
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:DescribeVolumes",
                "ec2:DescribeSnapshots",
                "ec2:CreateTags",
                "ec2:CreateVolume",
                "ec2:CreateSnapshot",
                "ec2:DeleteSnapshot"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "s3:GetObject",
                "s3:DeleteObject",
```

```
            "s3:PutObject",
            "s3:AbortMultipartUpload",
            "s3:ListMultipartUploadParts"
        ],
        "Resource": [
            "arn:aws:s3:::${BUCKET}/*"
        ]
    },
    {
        "Effect": "Allow",
        "Action": [
            "s3:ListBucket",
            "s3:GetBucketLocation",
            "s3:ListBucketMultipartUploads"
        ],
        "Resource": [
            "arn:aws:s3:::${BUCKET}"
        ]
    }
  ]
}
EOF
```

6. 附加策略，为 **velero** 用户提供所需的最低权限：

```
$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero \
  --policy-document file://velero-policy.json
```

7. 为 **velero** 用户创建访问密钥：

```
$ aws iam create-access-key --user-name velero
```

输出示例

```
{
  "AccessKey": {
        "UserName": "velero",
        "Status": "Active",
        "CreateDate": "2017-07-31T22:24:41.576Z",
        "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>,
        "AccessKeyId": <AWS_ACCESS_KEY_ID>
  }
}
```

8. 创建 **credentials-velero** 文件：

```
$ cat << EOF > ./credentials-velero
[default]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
EOF
```

在安装数据保护应用程序前，您可以使用 **credentials-velero** 文件为 AWS 创建 **Secret** 对象。

4.4.3.2. 关于备份和恢复位置及其 secret

您可以在 **DataProtectionApplication** 自定义资源(CR)中指定备份和快照位置及其 secret。

备份位置
您可以将 AWS S3 兼容对象存储（如 Multicloud Object Gateway 或 MinIO）指定为备份位置。

Velero 将 OpenShift Container Platform 资源、Kubernetes 对象和内部镜像备份为对象存储上的存档文件。

快照位置
如果使用云供应商的原生快照 API 备份持久性卷，您必须将云供应商指定为快照位置。

如果使用 Container Storage Interface(CSI)快照，则不需要指定快照位置，因为您要创建一个 **VolumeSnapshotClass** CR 来注册 CSI 驱动程序。

如果您使用文件系统备份 (FSB)，则不需要指定快照位置，因为 FSB 在对象存储上备份文件系统。

Secrets
如果备份和快照位置使用相同的凭证，或者不需要快照位置，请创建一个默认 **Secret**。

如果备份和恢复位置使用不同的凭证，您可以创建两个 secret 对象：

- 您在 **DataProtectionApplication** CR 中指定的备份位置的自定义 **Secret**。
- 快照位置的默认 **Secret**，在 **DataProtectionApplication** CR 中没有引用。

重要

数据保护应用程序需要一个默认的 **Secret**。否则，安装将失败。

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。

4.4.3.2.1. 创建默认 Secret

如果您的备份和快照位置使用相同的凭证，或者不需要快照位置，则创建一个默认 **Secret**。

Secret 的默认名称为 **cloud-credentials**。

注意

DataProtectionApplication 自定义资源(CR)需要一个默认的 **Secret**。否则，安装将失败。如果没有指定备份位置 **Secret** 的名称，则会使用默认名称。

如果您不想在安装过程中使用备份位置凭证，您可以使用空 **credentials-velero** 文件创建带有默认名称的 **Secret**。

先决条件

- 您的对象存储和云存储（若有）必须使用相同的凭证。
- 您必须为 Velero 配置对象存储。

- 您必须以适当的格式为对象存储创建一个 **credentials-velero** 文件。

流程

- 使用默认名称创建 **Secret** ：

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

在安装 Data Protection Application 时，**secret** 会在 **DataProtectionApplication** CR 的 **spec.backupLocations.credential** 块中引用。

4.4.3.2.2. 为不同凭证创建配置集

如果您的备份和快照位置使用不同的凭证，您可以在 **credentials-velero** 文件中创建单独的配置集。

然后，您可以创建一个 **Secret** 对象并在 **DataProtectionApplication** 自定义资源(CR)中指定配置集。

流程

1. 使用备份和快照位置的独立配置集创建一个 **credentials-velero** 文件，如下例所示：

```
[backupStorage]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>

[volumeSnapshot]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
```

2. 使用 **credentials-velero** 文件创建 **Secret** 对象：

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero (1)
```

3. 在 **DataProtectionApplication** CR 中添加配置集，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
        config:
          region: us-east-1
```

```
      profile: "backupStorage"
    credential:
      key: cloud
      name: cloud-credentials
snapshotLocations:
  - velero:
      provider: aws
      config:
        region: us-west-2
        profile: "volumeSnapshot"
```

4.4.3.3. 配置数据保护应用程序

您可以通过设置 Velero 资源分配或启用自签名 CA 证书来配置数据保护应用程序。

4.4.3.3.1. 设置 Velero CPU 和内存分配

您可以通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为 **Velero** pod 设置 CPU 和内存分配。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.podConfig.ResourceAllocations** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations: 2
          limits:
            cpu: "1"
            memory: 1024Mi
          requests:
            cpu: 200m
            memory: 256Mi
```

1 1 指定要提供给 Velero podSpec 的节点选择器。

2 列出的 **resourceAllocations** 用于平均使用。

注意

Kopia 是 OADP 1.3 及之后的版本中的一个选项。您可以使用 Kopia 进行文件系统备份，Kopia 是 Data Mover 的唯一选择，并带有内置数据 Mover。

和 Restic 相比，Kopia 需要更多资源，您可能需要相应地调整 CPU 和内存要求。

4.4.3.3.2. 启用自签名 CA 证书

您必须通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为对象存储启用自签名 CA 证书，以防止**由未知颁发机构签名的**证书。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.backupLocations.velero.objectStorage.caCert** 参数和 **spec.backupLocations.velero.config** 参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
# ...
```

1. 指定以 Base64 编码的 CA 证书字符串。
2. **insecureSkipTLSVerify** 配置可以设置为 **"true"** 或 **"false** "。如果设置为 **"true"**，则禁用 SSL/TLS 安全性。如果设置为 **"false"**，则启用 SSL/TLS 安全性。

4.4.3.3.2.1. 使用带有用于 velero 部署的 velero 命令的 CA 证书

如果您望使用 Velero CLI 而又不想在您的系统中安装它，可以为它创建一个别名。

先决条件

- 您必须使用具有 **cluster-admin** 角色的用户登录到 OpenShift Container Platform 集群。

- 已安装 OpenShift CLI (**oc**)。

1. 要使用别名的 Velero 命令，请运行以下命令：

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

2. 运行以下命令检查别名是否正常工作：

Example

```
$ velero version
Client:
 Version: v1.12.1-OADP
 Git commit: -
Server:
 Version: v1.12.1-OADP
```

3. 要使用这个命令来使用 CA 证书，您可以运行以下命令在 Velero 部署中添加证书：

```
$ CA_CERT=$(oc -n openshift-adp get dataprotectionapplications.oadp.openshift.io
<dpa-name> -o jsonpath='{.spec.backupLocations[0].velero.objectStorage.caCert}')

$ [[ -n $CA_CERT ]] && echo "$CA_CERT" | base64 -d | oc exec -n openshift-adp -i
deploy/velero -c velero -- bash -c "cat > /tmp/your-cacert.txt" || echo "DPA BSL has no
caCert"
```

```
$ velero describe backup <backup_name> --details --cacert /tmp/<your_cacert>.txt
```

4. 要获取备份日志，请运行以下命令：

```
$ velero backup logs  <backup_name>  --cacert /tmp/<your_cacert.txt>
```

您可以使用这些日志来查看无法备份的资源的失败和警告。

5. 如果 Velero pod 重启，**/tmp/your-cacert.txt** 文件会消失，您需要通过重新运行上一步中的命令来重新创建 **/tmp/your-cacert.txt** 文件。

6. 您可以运行以下命令来检查 **/tmp/your-cacert.txt** 文件是否存在（在存储它的文件位置中）：

```
$ oc exec -n openshift-adp -i deploy/velero -c velero -- bash -c "ls /tmp/your-cacert.txt"
/tmp/your-cacert.txt
```

在以后的 OpenShift API for Data Protection (OADP) 发行版本中，我们计划将证书挂载到 Velero pod，以便不需要这一步。

4.4.3.4. 安装数据保护应用程序 1.2 及更早版本

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

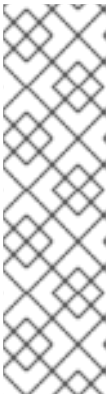
先决条件

- 您必须安装 OADP Operator。

- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，则必须使用默认名称 **cloud-credentials** 创建一个 **Secret**，其中包含备份和快照位置凭证的独立配置集。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。



注意

Velero 在 OADP 命名空间中创建一个名为 **velero-repo-credentials** 的 secret，其中包含默认的备份存储库密码。在运行第一个面向备份存储库的备份**之前**，您可以使用自己的密码更新 secret，以 base64 编码。要更新的键值是 **Data[repository-password]**。

创建 DPA 后，第一次运行指向备份存储库的备份时，Velero 会创建一个备份存储库，其 secret 为 **velero-repo-credentials**，其中包含默认密码或您替换它。如果在首次备份**之后**更新 secret 密码，新密码将与 **velero-repo-credentials** 中的密码不匹配，因此 Velero 将无法与旧的备份连接。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - openshift (1)
        - aws
      resourceTimeout: 10m (2)
    restic:
      enable: true (3)
      podConfig:
        nodeSelector: <node_selector> (4)
  backupLocations:
    - name: default
      velero:
        provider: aws
```

```
      default: true
      objectStorage:
        bucket: <bucket_name> 5
        prefix: <prefix> 6
      config:
        region: <region>
        profile: "default"
        s3ForcePathStyle: "true" 7
        s3Url: <s3_url> 8
      credential:
        key: cloud
        name: cloud-credentials 9
  snapshotLocations: 10
   - velero:
      provider: aws
      config:
        region: <region> 11
        profile: "default"
```

1. **openshift** 插件是必需的。
2. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
3. 如果要禁用 Restic 安装，则将此值设置为 **false**。Restic 部署一个守护进程集，这意味着 Restic pod 在每个工作节点上运行。在 OADP 版本 1.2 及更高版本中，您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置 Restic 进行备份。在 OADP 版本 1.1 中，将 **spec.defaultVolumesToRestic: true** 添加到 **Backup** CR 中。
4. 指定 Restic 在哪些节点上可用。默认情况下，Restic 在所有节点上运行。
5. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
6. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。
7. 指定是否为 S3 对象强制路径风格 URL (Boolean)。AWS S3 不需要。只适用于 S3 兼容存储。
8. 指定您用于存储备份的对象存储的 URL。AWS S3 不需要。只适用于 S3 兼容存储。
9. 指定您创建的 **Secret** 对象的名称。如果没有指定这个值，则使用默认值 **cloud-credentials**。如果您指定了自定义名称，则使用自定义名称进行备份位置。
10. 指定快照位置，除非您使用 CSI 快照或 Restic 备份 PV。
11. 快照位置必须与 PV 位于同一区域。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                 TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                         READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                       1/1     1            1           96s

NAME                                                    DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                       1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.3.5. 安装数据保护应用程序 1.3

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，则必须使用默认名称 **cloud-credentials** 创建一个 **Secret**，其中包含备份和快照位置凭证的独立配置集。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create** 实例。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - openshift (2)
        - aws
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
      podConfig:
        nodeSelector: <node_selector> (7)
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket_name> (8)
          prefix: <prefix> (9)
        config:
          region: <region>
          profile: "default"
          s3ForcePathStyle: "true" (10)
```

```
      s3Url: <s3_url> 11
    credential:
      key: cloud
      name: cloud-credentials 12
snapshotLocations: 13
 - name: default
   velero:
     provider: aws
     config:
       region: <region> 14
       profile: "default"
```

1. OADP 的默认命名空间是 **openshift-adp**。命名空间是一个变量，可配置。
2. **openshift** 插件是必需的。
3. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
4. 将管理请求路由到服务器的管理代理。
5. 如果要启用 **nodeAgent** 并执行文件系统备份，则将此值设置为 **true**。
6. 输入 **kopia** 或 **restic** 作为您的上传者。您不能在安装后更改选择。对于 Built-in DataMover，您必须使用 Kopia。**nodeAgent** 部署守护进程集，这意味着 **nodeAgent** pod 在每个工作节点上运行。您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置文件系统备份。
7. 指定 Kopia 或 Restic 可用的节点。默认情况下，Kopia 或 Restic 在所有节点上运行。
8. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
9. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。
10. 指定是否为 S3 对象强制路径风格 URL (Boolean)。AWS S3 不需要。只适用于 S3 兼容存储。
11. 指定您用于存储备份的对象存储的 URL。AWS S3 不需要。只适用于 S3 兼容存储。
12. 指定您创建的 **Secret** 对象的名称。如果没有指定这个值，则使用默认值 **cloud-credentials**。如果您指定了自定义名称，则使用自定义名称进行备份位置。
13. 指定快照位置，除非您使用 CSI 快照或文件系统备份 (FSB)备份 PV。
14. 快照位置必须与 PV 位于同一区域。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                1/1     Running   0          94s
pod/node-agent-m4lts                                1/1     Running   0          94s
pod/node-agent-pv4kr                                1/1     Running   0          95s
pod/velero-588db7f655-n842v                         1/1     Running   0          95s

NAME                                                TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc                   ClusterIP   172.30.10.0      <none>
8085/TCP   8h

NAME                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                      DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                         1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.3.5.1. 在 DataProtectionApplication CR 中启用 CSI

您可以在 **DataProtectionApplication** 自定义资源(CR)中启用 Container Storage Interface(CSI)来备份持久性卷，以使用 CSI 快照备份持久性卷。

先决条件

- 云供应商必须支持 CSI 快照。

流程

- 编辑 **DataProtectionApplication** CR，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

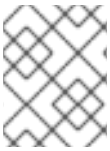
(1) 添加 **csi** 默认插件。

其他资源

- 使用 **kubevirt** 和 **openshift** 插件安装数据保护应用程序

4.4.4. 为 Microsoft Azure 的数据保护配置 OpenShift API

您可以通过安装 OADP Operator，使用 Microsoft Azure 安装 OpenShift API for Data Protection (OADP)。Operator 会安装 Velero 1.12。



注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

您可以为 Velero 配置 Azure，创建一个默认 **Secret**，然后安装数据保护应用程序。如需了解更多详细信息，请参阅安装 OADP Operator。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

4.4.4.1. 配置 Microsoft Azure

您可以为 OpenShift API 配置 Microsoft Azure for Data Protection (OADP)。

先决条件

- 已安装 Azure CLI。

流程

1. 登录到 Azure:

```
$ az login
```

2. 设置 **AZURE_RESOURCE_GROUP** 变量：

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

3. 创建 Azure 资源组：

```
$ az group create -n $AZURE_RESOURCE_GROUP --location CentralUS (1)
```

(1) 指定位置。

4. 设置 **AZURE_STORAGE_ACCOUNT_ID** 变量：

```
$ AZURE_STORAGE_ACCOUNT_ID="velero$(uuidgen | cut -d '-' -f5 | tr '[A-Z]' '[a-z]')"
```

5. 创建 Azure 存储帐户：

```
$ az storage account create \
    --name $AZURE_STORAGE_ACCOUNT_ID \
    --resource-group $AZURE_RESOURCE_GROUP \
    --sku Standard_GRS \
    --encryption-services blob \
    --https-only true \
    --kind BlobStorage \
    --access-tier Hot
```

6. 设置 **BLOB_CONTAINER** 变量：

```
$ BLOB_CONTAINER=velero
```

7. 创建 Azure Blob 存储容器：

```
$ az storage container create \
  -n $BLOB_CONTAINER \
  --public-access off \
  --account-name $AZURE_STORAGE_ACCOUNT_ID
```

8. 为 **velero** 创建服务主体和凭证：

```
$ AZURE_SUBSCRIPTION_ID=`az account list --query '[?isDefault].id' -o tsv` \
  AZURE_TENANT_ID=`az account list --query '[?isDefault].tenantId' -o tsv` \
  AZURE_CLIENT_SECRET=`az ad sp create-for-rbac --name "velero" \
  --role "Contributor" --query 'password' -o tsv` \
  AZURE_CLIENT_ID=`az ad sp list --display-name "velero" \
  --query '[0].appId' -o tsv`
```

9. 在 **credentials-velero** 文件中保存服务主体的凭证：

```
$ cat << EOF > ./credentials-velero
AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
AZURE_TENANT_ID=${AZURE_TENANT_ID}
AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
```

```
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

您可以使用 **credentials-velero** 文件将 Azure 添加为复制存储库。

10. 获取存储帐户访问密钥：

```
$ AZURE_STORAGE_ACCOUNT_ACCESS_KEY=`az storage account keys list \
  --account-name $AZURE_STORAGE_ACCOUNT_ID \
  --query "[?keyName == 'key1'].value" -o tsv`
```

11. 创建具有最低所需权限的自定义角色：

```
AZURE_ROLE=Velero
az role definition create --role-definition '{
   "Name": "'$AZURE_ROLE'",
   "Description": "Velero related permissions to perform backups, restores and deletions",
   "Actions": [
      "Microsoft.Compute/disks/read",
      "Microsoft.Compute/disks/write",
      "Microsoft.Compute/disks/endGetAccess/action",
      "Microsoft.Compute/disks/beginGetAccess/action",
      "Microsoft.Compute/snapshots/read",
      "Microsoft.Compute/snapshots/write",
      "Microsoft.Compute/snapshots/delete",
      "Microsoft.Storage/storageAccounts/listkeys/action",
      "Microsoft.Storage/storageAccounts/regeneratekey/action"
   ],
   "AssignableScopes": ["/subscriptions/'$AZURE_SUBSCRIPTION_ID'"]
   }'
```

12. 创建 **credentials-velero** 文件：

```
$ cat << EOF > ./credentials-velero
AZURE_SUBSCRIPTION_ID=${AZURE_SUBSCRIPTION_ID}
AZURE_TENANT_ID=${AZURE_TENANT_ID}
AZURE_CLIENT_ID=${AZURE_CLIENT_ID}
AZURE_CLIENT_SECRET=${AZURE_CLIENT_SECRET}
AZURE_RESOURCE_GROUP=${AZURE_RESOURCE_GROUP}
AZURE_STORAGE_ACCOUNT_ACCESS_KEY=${AZURE_STORAGE_ACCOUNT_ACCES
S_KEY} (1)
AZURE_CLOUD_NAME=AzurePublicCloud
EOF
```

(1) 必需。如果 **credentials-velero** 文件只包含服务主体凭证，则无法备份内部镜像。

在安装 Data Protection 应用前，您可以使用 **credentials-velero** 文件为 Azure 创建 **Secret** 对象。

4.4.4.2. 关于备份和恢复位置及其 secret

您可以在 **DataProtectionApplication** 自定义资源(CR)中指定备份和快照位置及其 secret。

备份位置
您可以将 AWS S3 兼容对象存储（如 Multicloud Object Gateway 或 MinIO）指定为备份位置。

Velero 将 OpenShift Container Platform 资源、Kubernetes 对象和内部镜像备份为对象存储上的存档文件。

快照位置
如果使用云供应商的原生快照 API 备份持久性卷，您必须将云供应商指定为快照位置。

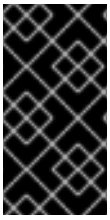
如果使用 Container Storage Interface(CSI)快照，则不需要指定快照位置，因为您要创建一个 **VolumeSnapshotClass** CR 来注册 CSI 驱动程序。

如果您使用文件系统备份 (FSB)，则不需要指定快照位置，因为 FSB 在对象存储上备份文件系统。

Secrets
如果备份和快照位置使用相同的凭证，或者不需要快照位置，请创建一个默认 **Secret**。

如果备份和恢复位置使用不同的凭证，您可以创建两个 secret 对象：

- 您在 **DataProtectionApplication** CR 中指定的备份位置的自定义 **Secret**。
- 快照位置的默认 **Secret**，在 **DataProtectionApplication** CR 中没有引用。



重要

数据保护应用程序需要一个默认的 **Secret**。否则，安装将失败。

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。

4.4.4.2.1. 创建默认 Secret

如果您的备份和快照位置使用相同的凭证，或者不需要快照位置，则创建一个默认 **Secret**。

Secret 的默认名称为 **cloud-credentials-azure**。

注意

DataProtectionApplication 自定义资源(CR)需要一个默认的 **Secret**。否则，安装将失败。如果没有指定备份位置 **Secret** 的名称，则会使用默认名称。

如果您不想在安装过程中使用备份位置凭证，您可以使用空 **credentials-velero** 文件创建带有默认名称的 **Secret**。

先决条件

- 您的对象存储和云存储（若有）必须使用相同的凭证。
- 您必须为 Velero 配置对象存储。
- 您必须以适当的格式为对象存储创建一个 **credentials-velero** 文件。

流程

- 使用默认名称创建 **Secret** ：

```
$ oc create secret generic cloud-credentials-azure -n openshift-adp --from-file
cloud=credentials-velero
```

在安装 Data Protection Application 时，**secret** 会在 **DataProtectionApplication** CR 的 **spec.backupLocations.credential** 块中引用。

4.4.4.2.2. 为不同凭证创建 secret

如果您的备份和恢复位置使用不同的凭证，您必须创建两个 **Secret** 对象：

- 具有自定义名称的备份位置 **Secret**。自定义名称在 **DataProtectionApplication** 自定义资源(CR)的 **spec.backupLocations** 块中指定。
- 带有默认名称 **cloud-credentials-azure** 的快照位置 **Secret**。此 **Secret** 不在 **DataProtectionApplication** CR 中指定。

流程

1. 为您的云供应商为快照位置创建一个 **credentials-velero** 文件。
2. 使用默认名称为快照位置创建 **Secret** ：

```
$ oc create secret generic cloud-credentials-azure -n openshift-adp --from-file
cloud=credentials-velero
```

3. 为您的对象存储创建一个用于备份位置的 **credentials-velero** 文件。
4. 使用自定义名称为备份位置创建 **Secret** ：

```
$ oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-
velero
```

5. 将带有自定义名称的 **Secret** 添加到 **DataProtectionApplication** CR 中，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          storageAccount: <azure_storage_account_id>
          subscriptionId: <azure_subscription_id>
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: <custom_secret> 1
        provider: azure
        default: true
        objectStorage:
```

```
        bucket: <bucket_name>
        prefix: <prefix>
    snapshotLocations:
      - velero:
          config:
            resourceGroup: <azure_resource_group>
            subscriptionId: <azure_subscription_id>
            incremental: "true"
          provider: azure
```

1 具有自定义名称的备份位置 **Secret**。

4.4.4.3. 配置数据保护应用程序

您可以通过设置 Velero 资源分配或启用自签名 CA 证书来配置数据保护应用程序。

4.4.4.3.1. 设置 Velero CPU 和内存分配

您可以通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为 **Velero** pod 设置 CPU 和内存分配。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.podConfig.ResourceAllocations** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations: 2
          limits:
            cpu: "1"
            memory: 1024Mi
          requests:
            cpu: 200m
            memory: 256Mi
```

1 指定要提供给 Velero podSpec 的节点选择器。

2 列出的 **resourceAllocations** 用于平均使用。

注意

Kopia 是 OADP 1.3 及之后的版本中的一个选项。您可以使用 Kopia 进行文件系统备份，Kopia 是 Data Mover 的唯一选择，并带有内置数据 Mover。

和 Restic 相比，Kopia 需要更多资源，您可能需要相应地调整 CPU 和内存要求。

4.4.4.3.2. 启用自签名 CA 证书

您必须通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为对象存储启用自签名 CA 证书，以防止**由未知颁发机构签名的**证书。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.backupLocations.velero.objectStorage.caCert** 参数和 **spec.backupLocations.velero.config** 参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
# ...
```

1. 指定以 Base64 编码的 CA 证书字符串。
2. **insecureSkipTLSVerify** 配置可以设置为 **"true"** 或 **"false** "。如果设置为 **"true"**，则禁用 SSL/TLS 安全性。如果设置为 **"false"**，则启用 SSL/TLS 安全性。

4.4.4.3.2.1. 使用带有用于 velero 部署的 velero 命令的 CA 证书

如果您望使用 Velero CLI 而又不想在您的系统中安装它，可以为它创建一个别名。

先决条件

- 您必须使用具有 **cluster-admin** 角色的用户登录到 OpenShift Container Platform 集群。

- 已安装 OpenShift CLI (**oc**)。

1. 要使用别名的 Velero 命令，请运行以下命令：

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

2. 运行以下命令检查别名是否正常工作：

Example

```
$ velero version
Client:
 Version: v1.12.1-OADP
 Git commit: -
Server:
 Version: v1.12.1-OADP
```

3. 要使用这个命令来使用 CA 证书，您可以运行以下命令在 Velero 部署中添加证书：

```
$ CA_CERT=$(oc -n openshift-adp get dataprotectionapplications.oadp.openshift.io
<dpa-name> -o jsonpath='{.spec.backupLocations[0].velero.objectStorage.caCert}')

$ [[ -n $CA_CERT ]] && echo "$CA_CERT" | base64 -d | oc exec -n openshift-adp -i
deploy/velero -c velero -- bash -c "cat > /tmp/your-cacert.txt" || echo "DPA BSL has no
caCert"
```

```
$ velero describe backup <backup_name> --details --cacert /tmp/<your_cacert>.txt
```

4. 要获取备份日志，请运行以下命令：

```
$ velero backup logs  <backup_name>  --cacert /tmp/<your_cacert.txt>
```

您可以使用这些日志来查看无法备份的资源的失败和警告。

5. 如果 Velero pod 重启，**/tmp/your-cacert.txt** 文件会消失，您需要通过重新运行上一步中的命令来重新创建 **/tmp/your-cacert.txt** 文件。

6. 您可以运行以下命令来检查 **/tmp/your-cacert.txt** 文件是否存在（在存储它的文件位置中）：

```
$ oc exec -n openshift-adp -i deploy/velero -c velero -- bash -c "ls /tmp/your-cacert.txt"
/tmp/your-cacert.txt
```

在以后的 OpenShift API for Data Protection (OADP) 发行版本中，我们计划将证书挂载到 Velero pod，以便不需要这一步。

4.4.4.4. 安装数据保护应用程序 1.2 及更早版本

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

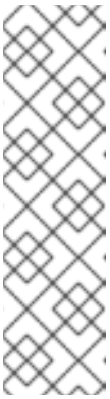
先决条件

- 您必须安装 OADP Operator。

- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials-azure** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，您必须创建两个 **Secret** ：
  - 带有备份位置的自定义名称的 **secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。
  - 带有快照位置的另一个自定义名称的 **Secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。



注意

Velero 在 OADP 命名空间中创建一个名为 **velero-repo-credentials** 的 secret，其中包含默认的备份存储库密码。在运行第一个面向备份存储库的备份**之前**，您可以使用自己的密码更新 secret，以 base64 编码。要更新的键值是 **Data[repository-password]**。

创建 DPA 后，第一次运行指向备份存储库的备份时，Velero 会创建一个备份存储库，其 secret 为 **velero-repo-credentials**，其中包含默认密码或您替换它。如果在首次备份**之后**更新 secret 密码，新密码将与 **velero-repo-credentials** 中的密码不匹配，因此 Velero 将无法与旧的备份连接。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create** 实**例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - azure
        - openshift (1)
      resourceTimeout: 10m (2)
    restic:
      enable: true (3)
```

```
    podConfig:
      nodeSelector: <node_selector> 4
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group> 5
          storageAccount: <azure_storage_account_id> 6
          subscriptionId: <azure_subscription_id> 7
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: cloud-credentials-azure 8
        provider: azure
        default: true
        objectStorage:
          bucket: <bucket_name> 9
          prefix: <prefix> 10
  snapshotLocations: 11
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          subscriptionId: <azure_subscription_id>
          incremental: "true"
        name: default
        provider: azure
```

1. **openshift** 插件是必需的。
2. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
3. 如果要禁用 Restic 安装，则将此值设置为 **false**。Restic 部署一个守护进程集，这意味着 Restic pod 在每个工作节点上运行。在 OADP 版本 1.2 及更高版本中，您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置 Restic 进行备份。在 OADP 版本 1.1 中，将 **spec.defaultVolumesToRestic: true** 添加到 **Backup** CR 中。
4. 指定 Restic 在哪些节点上可用。默认情况下，Restic 在所有节点上运行。
5. 指定 Azure 资源组。
6. 指定 Azure 存储帐户 ID。
7. 指定 Azure 订阅 ID。
8. 如果没有指定这个值，则使用默认值 **cloud-credentials-azure**。如果您指定了自定义名称，则使用自定义名称进行备份位置。
9. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
10. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。
11. 如果您使用 CSI 快照或 Restic 备份 PV，则不需要指定快照位置。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                    TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                          1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.4.5. 安装数据保护应用程序 1.3

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI)快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials-azure** 的 **Secret**。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - azure
        - openshift (2)
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
      podConfig:
        nodeSelector: <node_selector> (7)
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group> (8)
          storageAccount: <azure_storage_account_id> (9)
          subscriptionId: <azure_subscription_id> (10)
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: cloud-credentials-azure (11)
```

```
      provider: azure
      default: true
      objectStorage:
        bucket: <bucket_name> 12
        prefix: <prefix> 13
  snapshotLocations: 14
   - velero:
      config:
        resourceGroup: <azure_resource_group>
        subscriptionId: <azure_subscription_id>
        incremental: "true"
      name: default
      provider: azure
```

1. OADP 的默认命名空间是 **openshift-adp**。命名空间是一个变量，可配置。
2. **openshift** 插件是必需的。
3. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
4. 将管理请求路由到服务器的管理代理。
5. 如果要启用 **nodeAgent** 并执行文件系统备份，则将此值设置为 **true**。
6. 输入 **kopia** 或 **restic** 作为您的上传者。您不能在安装后更改选择。对于 Built-in DataMover，您必须使用 Kopia。**nodeAgent** 部署守护进程集，这意味着 **nodeAgent** pod 在每个工作节点上运行。您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置文件系统备份。
7. 指定 Kopia 或 Restic 可用的节点。默认情况下，Kopia 或 Restic 在所有节点上运行。
8. 指定 Azure 资源组。
9. 指定 Azure 存储帐户 ID。
10. 指定 Azure 订阅 ID。
11. 如果没有指定这个值，则使用默认值 **cloud-credentials-azure**。如果您指定了自定义名称，则使用自定义名称进行备份位置。
12. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
13. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。
14. 如果您使用 CSI 快照或 Restic 备份 PV，则不需要指定快照位置。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                1/1     Running   0          94s
pod/node-agent-m4lts                                1/1     Running   0          94s
pod/node-agent-pv4kr                                1/1     Running   0          95s
pod/velero-588db7f655-n842v                         1/1     Running   0          95s

NAME                                                TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc                   ClusterIP   172.30.10.0      <none>
8085/TCP   8h

NAME                                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                        1/1     1            1           96s

NAME                                                     DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                        1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.4.5.1. 在 DataProtectionApplication CR 中启用 CSI

您可以在 **DataProtectionApplication** 自定义资源(CR)中启用 Container Storage Interface(CSI)来备份持久性卷，以使用 CSI 快照备份持久性卷。

先决条件

- 云供应商必须支持 CSI 快照。

流程

- 编辑 **DataProtectionApplication** CR，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

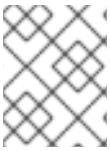
(1) 添加 **csi** 默认插件。

其他资源

- 使用 **kubevirt** 和 **openshift** 插件安装数据保护应用程序

4.4.5. 配置 OpenShift API 以进行 Google Cloud Platform 的数据保护

您可以通过安装 OADP Operator，使用 Google Cloud Platform (GCP) 安装 OpenShift API for Data Protection (OADP)。Operator 会安装 Velero 1.12。



注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

您可以为 Velero 配置 GCP，创建一个默认 **Secret**，然后安装数据保护应用程序。如需了解更多详细信息，请参阅安装 OADP Operator。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

4.4.5.1. 配置 Google Cloud Platform

对于数据保护(OADP)，您可以为 OpenShift API 配置 Google Cloud Platform(GCP)。

先决条件

- 您必须安装了 **gcloud** 和 **gsutil** CLI 工具。详情请查看 Google 云文档。

流程

1. 登录到 GCP：

```
$ gcloud auth login
```

2. 设置 **BUCKET** 变量：

```
$ BUCKET=<bucket> 1
```

1 指定存储桶名称。

3. 创建存储桶：

```
$ gsutil mb gs://$BUCKET/
```

4. 将 **PROJECT_ID** 变量设置为您的活跃项目：

```
$ PROJECT_ID=$(gcloud config get-value project)
```

5. 创建服务帐户：

```
$ gcloud iam service-accounts create velero \
    --display-name "Velero service account"
```

6. 列出服务帐户：

```
$ gcloud iam service-accounts list
```

7. 设置 **SERVICE_ACCOUNT_EMAIL** 变量，使其与 **email** 值匹配：

```
$ SERVICE_ACCOUNT_EMAIL=$(gcloud iam service-accounts list \
    --filter="displayName:Velero service account" \
    --format 'value(email)')
```

8. 附加策略，为 **velero** 用户提供所需的最低权限：

```
$ ROLE_PERMISSIONS=(
    compute.disks.get
    compute.disks.create
    compute.disks.createSnapshot
    compute.snapshots.get
    compute.snapshots.create
    compute.snapshots.useReadOnly
    compute.snapshots.delete
    compute.zones.get
    storage.objects.create
    storage.objects.delete
    storage.objects.get
    storage.objects.list
    iam.serviceAccounts.signBlob
)
```

9. 创建 **velero.server** 自定义角色：

```
$ gcloud iam roles create velero.server \
    --project $PROJECT_ID \
    --title "Velero Server" \
```

```
    --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"
```

10. 为项目添加 IAM 策略绑定：

```
$ gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member serviceAccount:$SERVICE_ACCOUNT_EMAIL \
    --role projects/$PROJECT_ID/roles/velero.server
```

11. 更新 IAM 服务帐户：

```
$ gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://${BUCKET}
```

12. 将 IAM 服务帐户的密钥保存到当前目录中的 **credentials-velero** 文件中：

```
$ gcloud iam service-accounts keys create credentials-velero \
    --iam-account $SERVICE_ACCOUNT_EMAIL
```

在安装 Data Protection Application 前，您可以使用 **credentials-velero** 文件为 GCP 创建 **Secret** 对象。

4.4.5.2. 关于备份和恢复位置及其 secret

您可以在 **DataProtectionApplication** 自定义资源(CR)中指定备份和快照位置及其 secret。

备份位置
您可以将 AWS S3 兼容对象存储（如 Multicloud Object Gateway 或 MinIO）指定为备份位置。

Velero 将 OpenShift Container Platform 资源、Kubernetes 对象和内部镜像备份为对象存储上的存档文件。

快照位置
如果使用云供应商的原生快照 API 备份持久性卷，您必须将云供应商指定为快照位置。

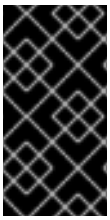
如果使用 Container Storage Interface(CSI)快照，则不需要指定快照位置，因为您要创建一个 **VolumeSnapshotClass** CR 来注册 CSI 驱动程序。

如果您使用文件系统备份 (FSB)，则不需要指定快照位置，因为 FSB 在对象存储上备份文件系统。

Secrets
如果备份和快照位置使用相同的凭证，或者不需要快照位置，请创建一个默认 **Secret**。

如果备份和恢复位置使用不同的凭证，您可以创建两个 secret 对象：

- 您在 **DataProtectionApplication** CR 中指定的备份位置的自定义 **Secret**。
- 快照位置的默认 **Secret**，在 **DataProtectionApplication** CR 中没有引用。



重要

数据保护应用程序需要一个默认的 **Secret**。否则，安装将失败。

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。

4.4.5.2.1. 创建默认 Secret

如果您的备份和快照位置使用相同的凭证，或者不需要快照位置，则创建一个默认 **Secret**。

Secret 的默认名称为 **cloud-credentials-gcp**。

注意

DataProtectionApplication 自定义资源(CR)需要一个默认的 **Secret**。否则，安装将失败。如果没有指定备份位置 **Secret** 的名称，则会使用默认名称。

如果您不想在安装过程中使用备份位置凭证，您可以使用空 **credentials-velero** 文件创建带有默认名称的 **Secret**。

先决条件

- 您的对象存储和云存储（若有）必须使用相同的凭证。
- 您必须为 Velero 配置对象存储。
- 您必须以适当的格式为对象存储创建一个 **credentials-velero** 文件。

流程

- 使用默认名称创建 **Secret** ：

```
$ oc create secret generic cloud-credentials-gcp -n openshift-adp --from-file
cloud=credentials-velero
```

在安装 Data Protection Application 时，**secret** 会在 **DataProtectionApplication** CR 的 **spec.backupLocations.credential** 块中引用。

4.4.5.2.2. 为不同凭证创建 secret

如果您的备份和恢复位置使用不同的凭证，您必须创建两个 **Secret** 对象：

- 具有自定义名称的备份位置 **Secret**。自定义名称在 **DataProtectionApplication** 自定义资源(CR)的 **spec.backupLocations** 块中指定。
- 带有默认名称 **cloud-credentials-gcp** 的快照位置 **Secret**。此 **Secret** 不在 **DataProtectionApplication** CR 中指定。

流程

1. 为您的云供应商为快照位置创建一个 **credentials-velero** 文件。
2. 使用默认名称为快照位置创建 **Secret** ：

```
$ oc create secret generic cloud-credentials-gcp -n openshift-adp --from-file
cloud=credentials-velero
```

3. 为您的对象存储创建一个用于备份位置的 **credentials-velero** 文件。
4. 使用自定义名称为备份位置创建 **Secret** ：

```
$ oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-velero
```

5. 将带有自定义名称的 **Secret** 添加到 **DataProtectionApplication** CR 中，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud
          name: <custom_secret> 1
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
  snapshotLocations:
    - velero:
        provider: gcp
        default: true
        config:
          project: <project>
          snapshotLocation: us-west1
```

1 具有自定义名称的备份位置 **Secret**。

4.4.5.3. 配置数据保护应用程序

您可以通过设置 Velero 资源分配或启用自签名 CA 证书来配置数据保护应用程序。

4.4.5.3.1. 设置 Velero CPU 和内存分配

您可以通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为 **Velero** pod 设置 CPU 和内存分配。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.podConfig.ResourceAllocations** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
```

```
  name: <dpa_sample>
spec:
# ...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations: 2
          limits:
            cpu: "1"
            memory: 1024Mi
          requests:
            cpu: 200m
            memory: 256Mi
```

1 指定要提供给 Velero podSpec 的节点选择器。

2 列出的 **resourceAllocations** 用于平均使用。

注意

Kopia 是 OADP 1.3 及之后的版本中的一个选项。您可以使用 Kopia 进行文件系统备份，Kopia 是 Data Mover 的唯一选择，并带有内置数据 Mover。

和 Restic 相比，Kopia 需要更多资源，您可能需要相应地调整 CPU 和内存要求。

4.4.5.3.2. 启用自签名 CA 证书

您必须通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为对象存储启用自签名 CA 证书，以防止**由未知颁发机构签名的**证书。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.backupLocations.velero.objectStorage.caCert** 参数和 **spec.backupLocations.velero.config** 参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
```

```
      prefix: <prefix>
      caCert: <base64_encoded_cert_string> 1
    config:
      insecureSkipTLSVerify: "false" 2
# ...
```

1. 指定以 Base64 编码的 CA 证书字符串。
2. **insecureSkipTLSVerify** 配置可以设置为 **"true"** 或 **"false** "。如果设置为 **"true"**，则禁用 SSL/TLS 安全性。如果设置为 **"false"**，则启用 SSL/TLS 安全性。

4.4.5.3.2.1. 使用带有用于 velero 部署的 velero 命令的 CA 证书

如果您望使用 Velero CLI 而又不想在您的系统中安装它，可以为它创建一个别名。

先决条件

- 您必须使用具有 **cluster-admin** 角色的用户登录到 OpenShift Container Platform 集群。
- 已安装 OpenShift CLI (**oc**)。

1. 要使用别名的 Velero 命令，请运行以下命令：

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

2. 运行以下命令检查别名是否正常工作：

Example

```
$ velero version
Client:
 Version: v1.12.1-OADP
 Git commit: -
Server:
 Version: v1.12.1-OADP
```

3. 要使用这个命令来使用 CA 证书，您可以运行以下命令在 Velero 部署中添加证书：

```
$ CA_CERT=$(oc -n openshift-adp get dataprotectionapplications.oadp.openshift.io <dpa-name> -o jsonpath='{.spec.backupLocations[0].velero.objectStorage.caCert}')

$ [[ -n $CA_CERT ]] && echo "$CA_CERT" | base64 -d | oc exec -n openshift-adp -i deploy/velero -c velero -- bash -c "cat > /tmp/your-cacert.txt" || echo "DPA BSL has no caCert"
```

```
$ velero describe backup <backup_name> --details --cacert /tmp/<your_cacert>.txt
```

4. 要获取备份日志，请运行以下命令：

```
$ velero backup logs  <backup_name>  --cacert /tmp/<your_cacert.txt>
```

您可以使用这些日志来查看无法备份的资源的失败和警告。

5. 如果 Velero pod 重启，**/tmp/your-cacert.txt** 文件会消失，您需要通过重新运行上一步中的命令来重新创建 **/tmp/your-cacert.txt** 文件。

6. 您可以运行以下命令来检查 **/tmp/your-cacert.txt** 文件是否存在（在存储它的文件位置中）：

```
$ oc exec -n openshift-adp -i deploy/velero -c velero -- bash -c "ls /tmp/your-cacert.txt"
/tmp/your-cacert.txt
```

在以后的 OpenShift API for Data Protection (OADP) 发行版本中，我们计划将证书挂载到 Velero pod，以便不需要这一步。

4.4.5.4. 安装数据保护应用程序 1.2 及更早版本

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials-gcp** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，您必须创建两个 **Secret** ：
  - 带有备份位置的自定义名称的 **secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。
  - 带有快照位置的另一个自定义名称的 **Secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

注意

Velero 在 OADP 命名空间中创建一个名为 **velero-repo-credentials** 的 secret，其中包含默认的备份存储库密码。在运行第一个面向备份存储库的备份**之前**，您可以使用自己的密码更新 secret，以 base64 编码。要更新的键值是 **Data[repository-password]**。

创建 DPA 后，第一次运行指向备份存储库的备份时，Velero 会创建一个备份存储库，其 secret 为 **velero-repo-credentials**，其中包含默认密码或您替换它。如果在首次备份**之后**更新 secret 密码，新密码将与 **velero-repo-credentials** 中的密码不匹配，因此 Velero 将无法与旧的备份连接。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - gcp
        - openshift (1)
      resourceTimeout: 10m (2)
    restic:
      enable: true (3)
      podConfig:
        nodeSelector: <node_selector> (4)
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud (5)
          name: cloud-credentials-gcp (6)
        objectStorage:
          bucket: <bucket_name> (7)
          prefix: <prefix> (8)
  snapshotLocations: (9)
    - velero:
        provider: gcp
        default: true
        config:
          project: <project>
          snapshotLocation: us-west1 (10)
```

(1) **openshift** 插件是必需的。

(2) 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。

(3) 如果要禁用 Restic 安装，则将此值设置为 **false**。Restic 部署一个守护进程集，这意味着 Restic pod 在每个工作节点上运行。在 OADP 版本 1.2 及更高版本中，您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置 Restic 进行备份。在 OADP 版本 1.1 中，将 **spec.defaultVolumesToRestic: true** 添加到 **Backup** CR 中。

(4) 指定 Restic 在哪些节点上可用。默认情况下，Restic 在所有节点上运行。

(5) 包含凭证的 secret 密钥。对于 Google 工作负载身份联邦云身份验证，请使用 **service_account.json**。

6 包含凭证的 secret 名称。如果没有指定这个值，则使用默认值 **cloud-credentials-gcp**。

7 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。

8 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。

9 指定快照位置，除非您使用 CSI 快照或 Restic 备份 PV。

10 快照位置必须与 PV 位于同一区域。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                    TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                          READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                        1/1     1            1           96s

NAME                                                   DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                      1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输**出示例**

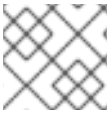
```
NAME          PHASE      LAST VALIDATED   AGE     DEFAULT
dpa-sample-1  Available  1s               3d16h   true
```

4.4.5.5. Google 工作负载**身份**联**邦云身份**验证

在 Google Cloud 外部运行的应用程序使用服务帐户密钥（如用户名和密码）来访问 Google Cloud 资源。如果没有正确管理，则这些服务帐户密钥可能会成为安全风险。

使用 Google 的工作负载身份联邦，您可以使用 Identity and Access Management (IAM)提供 IAM 角色，包括模拟服务帐户到外部身份的功能。这消除了与服务帐户密钥相关的维护和安全风险。

工作负载联邦处理加密和解密证书、提取用户属性和验证。身份联邦对身份验证进行外部化，将其传递给安全令牌服务 (STS)，并减少对各个开发人员的需求。授权和控制对资源的访问保持应用的职责。



注意

Google 工作负载身份联邦可用于 OADP 1.3.x 及更新的版本。

在备份卷时，使用 Google 工作负载身份联邦身份验证的 GCP 上 OADP 仅支持 CSI 快照。

在带有 Google 工作负载身份联邦身份验证的 GCP 上的 OADP 不支持卷快照位置 (VSL) 备份。如需了解更多详细信息，请参阅 Google 工作负载身份联邦已知问题。

如果不使用 Google 工作负载身份联邦云身份验证，请继续*安装数据保护应用程序*。

先决条件

- 您已以手动模式安装集群，并配置了 GCP Workload Identity。
- 您可以访问 Cloud Credential Operator 实用程序 (**ccoctl**) 以及关联的工作负载身份池。

流程

1. 运行以下命令，创建一个 **oadp-credrequest** 目录：

```
$ mkdir -p oadp-credrequest
```

2. 创建 **CredentialsRequest.yaml** 文件，如下所示：

```
echo 'apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: oadp-operator-credentials
  namespace: openshift-cloud-credential-operator
spec:
```

```
    providerSpec:
      apiVersion: cloudcredential.openshift.io/v1
      kind: GCPProviderSpec
      permissions:
      - compute.disks.get
      - compute.disks.create
      - compute.disks.createSnapshot
      - compute.snapshots.get
      - compute.snapshots.create
      - compute.snapshots.useReadOnly
      - compute.snapshots.delete
      - compute.zones.get
      - storage.objects.create
      - storage.objects.delete
      - storage.objects.get
      - storage.objects.list
      - iam.serviceAccounts.signBlob
      skipServiceCheck: true
    secretRef:
      name: cloud-credentials-gcp
      namespace: <OPERATOR_INSTALL_NS>
    serviceAccountNames:
    - velero
  ' > oadp-credrequest/credrequest.yaml
```

3. 运行以下命令，使用 **ccoctl** 实用程序处理 **oadp-credrequest** 目录中的 **CredentialsRequest** 对象：

```
$ ccoctl gcp create-service-accounts \
    --name=<name> \
    --project=<gcp_project_id> \
    --credentials-requests-dir=oadp-credrequest \
    --workload-identity-pool=<pool_id> \
    --workload-identity-provider=<provider_id>
```

manifests/openshift-adp-cloud-credentials-gcp-credentials.yaml 文件现在可用于以下步骤。

4. 运行以下命令来创建命名空间：

```
$ oc create namespace <OPERATOR_INSTALL_NS>
```

5. 运行以下命令，将凭证应用到命名空间：

```
$ oc apply -f manifests/openshift-adp-cloud-credentials-gcp-credentials.yaml
```

4.4.5.5.1. Google 工作负载身份联邦已知问题

- 在配置了 GCP 工作负载身份联邦时，卷快照位置(VSL) 备份会以一个 **PartiallyFailed** 阶段完成。Google 工作负载身份联邦身份验证不支持 VSL 备份。

4.4.5.6. 安装数据保护应用程序 1.3

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials-gcp** 的 **Secret**。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: <OPERATOR_INSTALL_NS> (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - gcp
        - openshift (2)
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
      podConfig:
        nodeSelector: <node_selector> (7)
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud (8)
          name: cloud-credentials-gcp (9)
        objectStorage:
          bucket: <bucket_name> (10)
          prefix: <prefix> (11)
  snapshotLocations: (12)
    - velero:
        provider: gcp
```

```
      default: true
      config:
        project: <project>
        snapshotLocation: us-west1 13
  backupImages: true 14
```

1. OADP 的默认命名空间是 **openshift-adp**。命名空间是一个变量，可配置。
2. **openshift** 插件是必需的。
3. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
4. 将管理请求路由到服务器的管理代理。
5. 如果要启用 **nodeAgent** 并执行文件系统备份，则将此值设置为 **true**。
6. 输入 **kopia** 或 **restic** 作为您的上传者。您不能在安装后更改选择。对于 Built-in DataMover，您必须使用 Kopia。**nodeAgent** 部署守护进程集，这意味着 **nodeAgent** pod 在每个工作节点上运行。您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置文件系统备份。
7. 指定 Kopia 或 Restic 可用的节点。默认情况下，Kopia 或 Restic 在所有节点上运行。
8. 包含凭证的 secret 密钥。对于 Google 工作负载身份联邦云身份验证，请使用 **service_account.json**。
9. 包含凭证的 secret 名称。如果没有指定这个值，则使用默认值 **cloud-credentials-gcp**。
10. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
11. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。
12. 指定快照位置，除非您使用 CSI 快照或 Restic 备份 PV。
13. 快照位置必须与 PV 位于同一区域。
14. Google 工作负载身份联邦支持内部镜像备份。如果您不想使用镜像备份，请将此字段设置为 **false**。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
```

```
pod/node-agent-pv4kr                                1/1     Running   0          95s
pod/velero-588db7f655-n842v                         1/1     Running   0          95s

NAME                                                TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc            ClusterIP   172.30.10.0      <none>
8085/TCP   8h

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                  READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                             DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                              1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输**出示例**

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输**出示例**

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.5.6.1. 在 DataProtectionApplication CR 中启用 CSI

您可以在 **DataProtectionApplication** 自定义资源(CR)中启用 Container Storage Interface(CSI)来备份持久性卷，以使用 CSI 快照备份持久性卷。

先决条件

- 云供应商必须支持 CSI 快照。

流程

- 编辑 **DataProtectionApplication** CR，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
```

(1) 添加 **csi** 默认插件。

其他资源

- 使用 **kubevirt** 和 **openshift** 插件安装数据保护应用程序

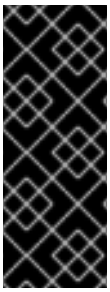
4.4.6. 为使用多云对象网关的数据保护配置 OpenShift API

您可以通过安装 OADP Operator，使用 Multicloud Object Gateway (MCG) 安装 OpenShift API for Data Protection (OADP)。Operator 会安装 Velero 1.12。

注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

您可以将 Multicloud 对象网关 配置为备份位置。MCG 是 OpenShift Data Foundation 的一个组件。您可以将 MCG 配置为 **DataProtectionApplication** 自定义资源(CR)中的备份位置。



重要

CloudStorage API（它自动为对象存储创建一个存储桶）只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

为备份位置创建一个 **Secret**，然后安装数据保护应用程序。如需了解更多详细信息，请参阅安装 OADP Operator。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

4.4.6.1. 检索多云对象网关凭证

您必须检索 Multicloud Object Gateway(MCG)凭证，以便为 OpenShift API 创建用于数据保护(OADP)的 **Secret** 自定义资源(CR)。

MCG 是 OpenShift Data Foundation 的一个组件。

先决条件

- 请根据相关的 OpenShift Data Foundation 部署指南 部署 OpenShift Data Foundation。

流程

1. 通过对 **NooBaa** 自定义资源运行 **describe** 命令，获取 S3 端点、**AWS_ACCESS_KEY_ID** 和 **AWS_SECRET_ACCESS_KEY**。
2. 创建 **credentials-velero** 文件：

```
$ cat << EOF > ./credentials-velero
[default]
aws_access_key_id=<AWS_ACCESS_KEY_ID>
aws_secret_access_key=<AWS_SECRET_ACCESS_KEY>
EOF
```

在安装 Data Protection Application 时，您可以使用 **credentials-velero** 文件创建 **Secret** 对象。

4.4.6.2. 关于备份和恢复位置及其 secret

您可以在 **DataProtectionApplication** 自定义资源(CR)中指定备份和快照位置及其 secret。

备份位置
您可以将 AWS S3 兼容对象存储（如 Multicloud Object Gateway 或 MinIO）指定为备份位置。

Velero 将 OpenShift Container Platform 资源、Kubernetes 对象和内部镜像备份为对象存储上的存档文件。

快照位置
如果使用云供应商的原生快照 API 备份持久性卷，您必须将云供应商指定为快照位置。

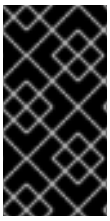
如果使用 Container Storage Interface(CSI)快照，则不需要指定快照位置，因为您要创建一个 **VolumeSnapshotClass** CR 来注册 CSI 驱动程序。

如果您使用文件系统备份 (FSB)，则不需要指定快照位置，因为 FSB 在对象存储上备份文件系统。

Secrets
如果备份和快照位置使用相同的凭证，或者不需要快照位置，请创建一个默认 **Secret**。

如果备份和恢复位置使用不同的凭证，您可以创建两个 secret 对象：

- 您在 **DataProtectionApplication** CR 中指定的备份位置的自定义 **Secret**。
- 快照位置的默认 **Secret**，在 **DataProtectionApplication** CR 中没有引用。



重要

数据保护应用程序需要一个默认的 **Secret**。否则，安装将失败。

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。

4.4.6.2.1. 创建默认 Secret

如果您的备份和快照位置使用相同的凭证，或者不需要快照位置，则创建一个默认 **Secret**。

Secret 的默认名称为 **cloud-credentials**。

注意

DataProtectionApplication 自定义资源(CR)需要一个默认的 **Secret**。否则，安装将失败。如果没有指定备份位置 **Secret** 的名称，则会使用默认名称。

如果您不想在安装过程中使用备份位置凭证，您可以使用空 **credentials-velero** 文件创建带有默认名称的 **Secret**。

先决条件

- 您的对象存储和云存储（若有）必须使用相同的凭证。
- 您必须为 Velero 配置对象存储。
- 您必须以适当的格式为对象存储创建一个 **credentials-velero** 文件。

流程

- 使用默认名称创建 **Secret** ：

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

在安装 Data Protection Application 时，**secret** 会在 **DataProtectionApplication** CR 的 **spec.backupLocations.credential** 块中引用。

4.4.6.2.2. 为不同凭证创建 secret

如果您的备份和恢复位置使用不同的凭证，您必须创建两个 **Secret** 对象：

- 具有自定义名称的备份位置 **Secret**。自定义名称在 **DataProtectionApplication** 自定义资源(CR)的 **spec.backupLocations** 块中指定。
- 带有默认名称 **cloud-credentials** 的快照位置 **Secret**。此 **Secret** 不在 **DataProtectionApplication** CR 中指定。

流程

1. 为您的云供应商为快照位置创建一个 **credentials-velero** 文件。
2. 使用默认名称为快照位置创建 **Secret** ：

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

3. 为您的对象存储创建一个用于备份位置的 **credentials-velero** 文件。
4. 使用自定义名称为备份位置创建 **Secret** ：

```
$ oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-velero
```

5. 将带有自定义名称的 **Secret** 添加到 **DataProtectionApplication** CR 中，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url>
          insecureSkipTLSVerify: "true"
          s3ForcePathStyle: "true"
        provider: aws
        default: true
        credential:
          key: cloud
          name:  <custom_secret> (1)
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
```

(1) 具有自定义名称的备份位置 **Secret**。

4.4.6.3. 配置数据保护应用程序

您可以通过设置 Velero 资源分配或启用自签名 CA 证书来配置数据保护应用程序。

4.4.6.3.1. 设置 Velero CPU 和内存分配

您可以通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为 **Velero** pod 设置 CPU 和内存分配。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

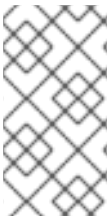
- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.podConfig.ResourceAllocations** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  configuration:
    velero:
```

```
    podConfig:
      nodeSelector: <node selector> 1
      resourceAllocations: 2
        limits:
          cpu: "1"
          memory: 1024Mi
        requests:
          cpu: 200m
          memory: 256Mi
```

1 指定要提供给 Velero podSpec 的节点选择器。

2 列出的 **resourceAllocations** 用于平均使用。



注意

Kopia 是 OADP 1.3 及之后的版本中的一个选项。您可以使用 Kopia 进行文件系统备份，Kopia 是 Data Mover 的唯一选择，并带有内置数据 Mover。

和 Restic 相比，Kopia 需要更多资源，您可能需要相应地调整 CPU 和内存要求。

4.4.6.3.2. 启用自签名 CA 证书

您必须通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为对象存储启用自签名 CA 证书，以防止**由未知颁发机构签名的**证书。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.backupLocations.velero.objectStorage.caCert** 参数和 **spec.backupLocations.velero.config** 参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
# ...
```

1 指定以 Base64 编码的 CA 证书字符串。

2 **insecureSkipTLSVerify** 配置可以设置为 **"true"** 或 **"false** "。如果设置为 **"true"**，则禁用 SSL/TLS 安全性。如果设置为 **"false"**，则启用 SSL/TLS 安全性。

4.4.6.3.2.1. 使用带有用于 velero 部署的 velero 命令的 CA 证书

如果您望使用 Velero CLI 而又不想在您的系统中安装它，可以为它创建一个别名。

先决条件

- 您必须使用具有 **cluster-admin** 角色的用户登录到 OpenShift Container Platform 集群。
- 已安装 OpenShift CLI (**oc**)。

1. 要使用别名的 Velero 命令，请运行以下命令：

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

2. 运行以下命令检查别名是否正常工作：

Example

```
$ velero version
Client:
 Version: v1.12.1-OADP
 Git commit: -
Server:
 Version: v1.12.1-OADP
```

3. 要使用这个命令来使用 CA 证书，您可以运行以下命令在 Velero 部署中添加证书：

```
$ CA_CERT=$(oc -n openshift-adp get dataprotectionapplications.oadp.openshift.io
<dpa-name> -o jsonpath='{.spec.backupLocations[0].velero.objectStorage.caCert}')

$ [[ -n $CA_CERT ]] && echo "$CA_CERT" | base64 -d | oc exec -n openshift-adp -i
deploy/velero -c velero -- bash -c "cat > /tmp/your-cacert.txt" || echo "DPA BSL has no
caCert"
```

```
$ velero describe backup <backup_name> --details --cacert /tmp/<your_cacert>.txt
```

4. 要获取备份日志，请运行以下命令：

```
$ velero backup logs  <backup_name>  --cacert /tmp/<your_cacert.txt>
```

您可以使用这些日志来查看无法备份的资源的失败和警告。

5. 如果 Velero pod 重启，**/tmp/your-cacert.txt** 文件会消失，您需要通过重新运行上一步中的命令来重新创建 **/tmp/your-cacert.txt** 文件。

6. 您可以运行以下命令来检查 **/tmp/your-cacert.txt** 文件是否存在（在存储它的文件位置中）：

```
$ oc exec -n openshift-adp -i deploy/velero -c velero -- bash -c "ls /tmp/your-cacert.txt"
/tmp/your-cacert.txt
```

在以后的 OpenShift API for Data Protection (OADP) 发行版本中，我们计划将证书挂载到 Velero pod，以便不需要这一步。

4.4.6.4. 安装数据保护应用程序 1.2 及更早版本

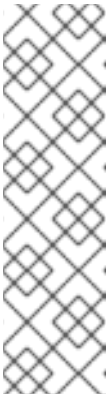
您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，您必须创建两个 **Secret** ：
  - 带有备份位置的自定义名称的 **secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。
  - 带有快照位置的另一个自定义名称的 **Secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。



注意

Velero 在 OADP 命名空间中创建一个名为 **velero-repo-credentials** 的 secret，其中包含默认的备份存储库密码。在运行第一个面向备份存储库的备份**之前**，您可以使用自己的密码更新 secret，以 base64 编码。要更新的键值是 **Data[repository-password]**。

创建 DPA 后，第一次运行指向备份存储库的备份时，Velero 会创建一个备份存储库，其 secret 为 **velero-repo-credentials**，其中包含默认密码或您替换它。如果在首次备份**之后**更新 secret 密码，新密码将与 **velero-repo-credentials** 中的密码不匹配，因此 Velero 将无法与旧的备份连接。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
```

```
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - aws
        - openshift (1)
      resourceTimeout: 10m (2)
    restic:
      enable: true (3)
      podConfig:
        nodeSelector: <node_selector> (4)
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url> (5)
          insecureSkipTLSVerify: "true"
          s3ForcePathStyle: "true"
        provider: aws
        default: true
        credential:
          key: cloud
          name: cloud-credentials (6)
        objectStorage:
          bucket: <bucket_name> (7)
          prefix: <prefix> (8)
```

(1) **openshift** 插件是必需的。

(2) 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。

(3) 如果要禁用 Restic 安装，则将此值设置为 **false**。Restic 部署一个守护进程集，这意味着 Restic pod 在每个工作节点上运行。在 OADP 版本 1.2 及更高版本中，您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置 Restic 进行备份。在 OADP 版本 1.1 中，将 **spec.defaultVolumesToRestic: true** 添加到 **Backup** CR 中。

(4) 指定 Restic 在哪些节点上可用。默认情况下，Restic 在所有节点上运行。

(5) 指定 S3 端点的 URL。

(6) 如果没有指定这个值，则使用默认值 **cloud-credentials**。如果您指定了自定义名称，则使用自定义名称进行备份位置。

(7) 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。

(8) 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                     1/1     Running   0          94s
pod/restic-m4lts                                     1/1     Running   0          94s
pod/restic-pv4kr                                     1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                    TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                          1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.6.5. 安装数据保护应用程序 1.3

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - aws
        - openshift (2)
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
      podConfig:
        nodeSelector: <node_selector> (7)
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url> (8)
          insecureSkipTLSVerify: "true"
          s3ForcePathStyle: "true"
        provider: aws
        default: true
        credential:
```

```
      key: cloud
      name: cloud-credentials 9
    objectStorage:
      bucket: <bucket_name> 10
      prefix: <prefix> 11
```

1. OADP 的默认命名空间是 **openshift-adp**。命名空间是一个变量，可配置。
2. **openshift** 插件是必需的。
3. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
4. 将管理请求路由到服务器的管理代理。
5. 如果要启用 **nodeAgent** 并执行文件系统备份，则将此值设置为 **true**。
6. 输入 **kopia** 或 **restic** 作为您的上传者。您不能在安装后更改选择。对于 Built-in DataMover，您必须使用 Kopia。**nodeAgent** 部署守护进程集，这意味着 **nodeAgent** pod 在每个工作节点上运行。您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置文件系统备份。
7. 指定 Kopia 或 Restic 可用的节点。默认情况下，Kopia 或 Restic 在所有节点上运行。
8. 指定 S3 端点的 URL。
9. 如果没有指定这个值，则使用默认值 **cloud-credentials**。如果您指定了自定义名称，则使用自定义名称进行备份位置。
10. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
11. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
```

```
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc                ClusterIP   172.30.10.0      <none>
8085/TCP   8h

NAME                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                    READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                               DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                             1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输**出示例**

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输**出示例**

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.6.5.1. 在 DataProtectionApplication CR 中启用 CSI

您可以在 **DataProtectionApplication** 自定义资源(CR)中启用 Container Storage Interface(CSI)来备份持久性卷，以使用 CSI 快照备份持久性卷。

先决条件

- 云供应商必须支持 CSI 快照。

流程

- 编辑 **DataProtectionApplication** CR，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
```

```
configuration:
  velero:
    defaultPlugins:
    - openshift
    - csi (1)
```

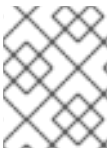
(1) 添加 **csi** 默认插件。

其他资源

- 多云对象网关的性能调节指南.
- 使用 **kubevirt** 和 **openshift** 插件安装数据保护应用程序

4.4.7. 为 OpenShift Data Foundation 的数据保护配置 OpenShift API

您可以通过安装 OADP Operator 并配置备份位置和快照位置，在 OpenShift Data Foundation 中安装 OpenShift API for Data Protection (OADP)。然后，您要安装数据保护应用程序。



注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

您可以将 Multicloud 对象网关 或任何 AWS S3 兼容对象存储配置为备份位置。

重要

CloudStorage API（它自动为对象存储创建一个存储桶）只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

为备份位置创建一个 **Secret**，然后安装数据保护应用程序。如需了解更多详细信息，请参阅安装 OADP Operator。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

4.4.7.1. 关于备份和恢复位置及其 secret

您可以在 **DataProtectionApplication** 自定义资源(CR)中指定备份和快照位置及其 secret。

备份位置
您可以将 AWS S3 兼容对象存储（如 Multicloud Object Gateway 或 MinIO）指定为备份位置。

Velero 将 OpenShift Container Platform 资源、Kubernetes 对象和内部镜像备份为对象存储上的存档文件。

快照位置
如果使用云供应商的原生快照 API 备份持久性卷，您必须将云供应商指定为快照位置。

如果使用 Container Storage Interface(CSI)快照，则不需要指定快照位置，因为您要创建一个 **VolumeSnapshotClass** CR 来注册 CSI 驱动程序。

如果您使用文件系统备份 (FSB)，则不需要指定快照位置，因为 FSB 在对象存储上备份文件系统。

Secrets
如果备份和快照位置使用相同的凭证，或者不需要快照位置，请创建一个默认 **Secret**。

如果备份和恢复位置使用不同的凭证，您可以创建两个 secret 对象：

- 您在 **DataProtectionApplication** CR 中指定的备份位置的自定义 **Secret**。
- 快照位置的默认 **Secret**，在 **DataProtectionApplication** CR 中没有引用。

重要

数据保护应用程序需要一个默认的 **Secret**。否则，安装将失败。

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。

其他资源

- 使用 OpenShift Web 控制台创建对象 Bucket 声明

4.4.7.1.1. 创建默认 Secret

如果您的备份和快照位置使用相同的凭证，或者不需要快照位置，则创建一个默认 **Secret**。

Secret 的默认名称为 **cloud-credentials**，除非备份存储供应商有一个默认插件，如 **aws**、**azure** 或 **gcp**。在这种情况下，默认名称是在特定于供应商的 OADP 安装过程中指定。

注意

DataProtectionApplication 自定义资源(CR)需要一个默认的 **Secret**。否则，安装将失败。如果没有指定备份位置 **Secret** 的名称，则会使用默认名称。

如果您不想在安装过程中使用备份位置凭证，您可以使用空 **credentials-velero** 文件创建带有默认名称的 **Secret**。

先决条件

- 您的对象存储和云存储（若有）必须使用相同的凭证。
- 您必须为 Velero 配置对象存储。
- 您必须以适当的格式为对象存储创建一个 **credentials-velero** 文件。

流程

- 使用默认名称创建 **Secret** ：

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

在安装 Data Protection Application 时，**secret** 会在 **DataProtectionApplication** CR 的 **spec.backupLocations.credential** 块中引用。

4.4.7.1.2. 为不同凭证创建 secret

如果您的备份和恢复位置使用不同的凭证，您必须创建两个 **Secret** 对象：

- 具有自定义名称的备份位置 **Secret**。自定义名称在 **DataProtectionApplication** 自定义资源(CR)的 **spec.backupLocations** 块中指定。
- 带有默认名称 **cloud-credentials** 的快照位置 **Secret**。此 **Secret** 不在 **DataProtectionApplication** CR 中指定。

流程

1. 为您的云供应商为快照位置创建一个 **credentials-velero** 文件。
2. 使用默认名称为快照位置创建 **Secret** ：

```
$ oc create secret generic cloud-credentials -n openshift-adp --from-file cloud=credentials-velero
```

3. 为您的对象存储创建一个用于备份位置的 **credentials-velero** 文件。
4. 使用自定义名称为备份位置创建 **Secret** ：

```
$ oc create secret generic <custom_secret> -n openshift-adp --from-file cloud=credentials-velero
```

5. 将带有自定义名称的 **Secret** 添加到 **DataProtectionApplication** CR 中，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        provider: <provider>
        default: true
        credential:
          key: cloud
          name: <custom_secret> (1)
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
```

(1) 具有自定义名称的备份位置 **Secret**。

4.4.7.2. 配置数据保护应用程序

您可以通过设置 Velero 资源分配或启用自签名 CA 证书来配置数据保护应用程序。

4.4.7.2.1. 设置 Velero CPU 和内存分配

您可以通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为 **Velero** pod 设置 CPU 和内存分配。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.podConfig.ResourceAllocations** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations: 2
          limits:
            cpu: "1"
            memory: 1024Mi
          requests:
            cpu: 200m
            memory: 256Mi
```

1. 指定要提供给 Velero podSpec 的节点选择器。
2. 列出的 **resourceAllocations** 用于平均使用。

注意

Kopia 是 OADP 1.3 及之后的版本中的一个选项。您可以使用 Kopia 进行文件系统备份，Kopia 是 Data Mover 的唯一选择，并带有内置数据 Mover。

和 Restic 相比，Kopia 需要更多资源，您可能需要相应地调整 CPU 和内存要求。

4.4.7.2.1.1. 根据收集的数据调整 Ceph CPU 和内存要求

以下建议基于在扩展和性能实验室中观察到的性能。更改与 Red Hat OpenShift Data Foundation (ODF) 相关。如果使用 ODF，请参阅相关的调优指南来了解官方的建议。

4.4.7.2.1.1.1. 配置的 CPU 和内存要求

备份和恢复操作需要大量 CephFS **PersistentVolume** (PV)。为了避免 Ceph MDS pod 重启并带有 **out-of-memory** (OOM) 错误，建议以下配置：

配置类型	Request（请求）	最大限制
CPU	请求改为 3	最大限制为 3
内存	请求改为 8 Gi	最大限制为 128 Gi

4.4.7.2.2. 启用自签名 CA 证书

您必须通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为对象存储启用自签名 CA 证书，以防止由未知颁发机构签名的证书。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.backupLocations.velero.objectStorage.caCert** 参数和 **spec.backupLocations.velero.config** 参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
# ...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket>
          prefix: <prefix>
          caCert: <base64_encoded_cert_string> 1
        config:
          insecureSkipTLSVerify: "false" 2
# ...
```

1. 指定以 Base64 编码的 CA 证书字符串。
2. **insecureSkipTLSVerify** 配置可以设置为 **"true"** 或 **"false** "。如果设置为 **"true"**，则禁用 SSL/TLS 安全性。如果设置为 **"false"**，则启用 SSL/TLS 安全性。

4.4.7.2.2.1. 使用带有用于 velero 部署的 velero 命令的 CA 证书

如果您望使用 Velero CLI 而又不想在您的系统中安装它，可以为它创建一个别名。

先决条件

- 您必须使用具有 **cluster-admin** 角色的用户登录到 OpenShift Container Platform 集群。
- 已安装 OpenShift CLI (**oc**)。

1. 要使用别名的 Velero 命令，请运行以下命令：

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

2. 运行以下命令检查别名是否正常工作：

Example

```
$ velero version
Client:
 Version: v1.12.1-OADP
 Git commit: -
Server:
 Version: v1.12.1-OADP
```

3. 要使用这个命令来使用 CA 证书，您可以运行以下命令在 Velero 部署中添加证书：

```
$ CA_CERT=$(oc -n openshift-adp get dataprotectionapplications.oadp.openshift.io
<dpa-name> -o jsonpath='{.spec.backupLocations[0].velero.objectStorage.caCert}')

$ [[ -n $CA_CERT ]] && echo "$CA_CERT" | base64 -d | oc exec -n openshift-adp -i
deploy/velero -c velero -- bash -c "cat > /tmp/your-cacert.txt" || echo "DPA BSL has no
caCert"
```

```
$ velero describe backup <backup_name> --details --cacert /tmp/<your_cacert>.txt
```

4. 要获取备份日志，请运行以下命令：

```
$ velero backup logs  <backup_name>  --cacert /tmp/<your_cacert.txt>
```

您可以使用这些日志来查看无法备份的资源的失败和警告。

5. 如果 Velero pod 重启，**/tmp/your-cacert.txt** 文件会消失，您需要通过重新运行上一步中的命令来重新创建 **/tmp/your-cacert.txt** 文件。

6. 您可以运行以下命令来检查 **/tmp/your-cacert.txt** 文件是否存在（在存储它的文件位置中）：

```
$ oc exec -n openshift-adp -i deploy/velero -c velero -- bash -c "ls /tmp/your-cacert.txt"
/tmp/your-cacert.txt
```

在以后的 OpenShift API for Data Protection (OADP) 发行版本中，我们计划将证书挂载到 Velero pod，以便不需要这一步。

4.4.7.3. 安装数据保护应用程序 1.2 及更早版本

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，您必须创建两个 **Secret** ：
  - 带有备份位置的自定义名称的 **secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。
  - 带有快照位置的另一个自定义名称的 **Secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。

注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

注意

Velero 在 OADP 命名空间中创建一个名为 **velero-repo-credentials** 的 secret，其中包含默认的备份存储库密码。在运行第一个面向备份存储库的备份**之前**，您可以使用自己的密码更新 secret，以 base64 编码。要更新的键值是 **Data[repository-password]**。

创建 DPA 后，第一次运行指向备份存储库的备份时，Velero 会创建一个备份存储库，其 secret 为 **velero-repo-credentials**，其中包含默认密码或您替换它。如果在首次备份**之后**更新 secret 密码，新密码将与 **velero-repo-credentials** 中的密码不匹配，因此 Velero 将无法与旧的备份连接。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - kubevirt (1)
        - gcp (2)
        - csi (3)
```

```
      - openshift 4
    resourceTimeout: 10m 5
  restic:
    enable: true 6
    podConfig:
      nodeSelector: <node_selector> 7
backupLocations:
  - velero:
      provider: gcp 8
      default: true
      credential:
        key: cloud
        name: <default_secret> 9
      objectStorage:
        bucket: <bucket_name> 10
        prefix: <prefix> 11
```

1. 可选： **kubevirt** 插件用于 OpenShift Virtualization。
2. 为备份供应商指定默认插件，如 **gcp** （如果适用）。
3. 如果使用 CSI 快照备份 PV，请指定 **csi** 默认插件。**csi** 插件使用 Velero CSI beta 快照 API。您不需要配置快照位置。
4. **openshift** 插件是必需的。
5. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
6. 如果要禁用 Restic 安装，则将此值设置为 **false**。Restic 部署一个守护进程集，这意味着 Restic pod 在每个工作节点上运行。在 OADP 版本 1.2 及更高版本中，您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置 Restic 进行备份。在 OADP 版本 1.1 中，将 **spec.defaultVolumesToRestic: true** 添加到 **Backup** CR 中。
7. 指定 Restic 在哪些节点上可用。默认情况下，Restic 在所有节点上运行。
8. 指定备份供应商。
9. 如果备份供应商使用一个默认插件，为 **Secret** 指定正确的默认名称，如 **cloud-credentials-gcp**。如果指定了一个自定义名称，则使用自定义名称用于备份位置。如果没有指定 **Secret** 名称，则使用默认名称。
10. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
11. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                  READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/restic-9cq4q                                      1/1     Running   0          94s
pod/restic-m4lts                                      1/1     Running   0          94s
pod/restic-pv4kr                                      1/1     Running   0          95s
pod/velero-588db7f655-n842v                           1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s

NAME                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/restic   3         3         3       3            3           <none>          96s

NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                  DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                             1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

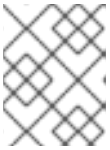
4.4.7.4. 安装数据保护应用程序 1.3

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。

- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。



注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - kubevirt (2)
        - gcp (3)
        - csi (4)
        - openshift (5)
      resourceTimeout: 10m (6)
    nodeAgent: (7)
      enable: true (8)
      uploaderType: kopia (9)
      podConfig:
        nodeSelector: <node_selector> (10)
  backupLocations:
    - velero:
        provider: gcp (11)
        default: true
        credential:
          key: cloud
          name: <default_secret> (12)
        objectStorage:
          bucket: <bucket_name> (13)
          prefix: <prefix> (14)
```

(1) OADP 的默认命名空间是 **openshift-adp**。命名空间是一个变量，可配置。

(2) 可选： **kubevirt** 插件用于 OpenShift Virtualization。

3. 为备份供应商指定默认插件，如 **gcp** （如果适用）。
4. 如果使用 CSI 快照备份 PV，请指定 **csi** 默认插件。**csi** 插件使用 Velero CSI beta 快照 API。您不需要配置快照位置。
5. **openshift** 插件是必需的。
6. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
7. 将管理请求路由到服务器的管理代理。
8. 如果要启用 **nodeAgent** 并执行文件系统备份，则将此值设置为 **true**。
9. 输入 **kopia** 或 **restic** 作为您的上传者。您不能在安装后更改选择。对于 Built-in DataMover，您必须使用 Kopia。**nodeAgent** 部署守护进程集，这意味着 **nodeAgent** pod 在每个工作节点上运行。您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置文件系统备份。
10. 指定 Kopia 或 Restic 可用的节点。默认情况下，Kopia 或 Restic 在所有节点上运行。
11. 指定备份供应商。
12. 如果备份供应商使用一个默认插件，为 **Secret** 指定正确的默认名称，如 **cloud-credentials-gcp**。如果指定了一个自定义名称，则使用自定义名称用于备份位置。如果没有指定 **Secret** 名称，则使用默认名称。
13. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
14. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                     TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc                 ClusterIP   172.30.10.0      <none>
8085/TCP   8h
```

```
NAME                 DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                 READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                            DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                              1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输**出示例**

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输**出示例**

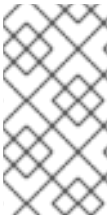
```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

4.4.7.4.1. 为 OpenShift Data Foundation 上的灾难恢复创建对象 Bucket 声明

如果您在 OpenShift Data Foundation 上将集群存储用于 Multicloud 对象网关 (MCG) 存储桶 **backupStorageLocation**，请使用 OpenShift Web 控制台创建一个对象 Bucket 声明 (OBC)。

警告

未能配置对象 Bucket 声明 (OBC) 可能会导致备份不可用。



注意

除非另有指定，"NooBaa" 指的是提供轻量级对象存储的开源项目，而 "Multicloud Object Gateway (MCG) " 是指 NooBaa 的红帽发行版本。

如需有关 MCG 的更多信息，请参阅使用应用程序访问多云对象网关。

流程

- 使用 OpenShift Web 控制台创建对象 Bucket 声明 (OBC)，如使用 OpenShift Web 控制台创建对象 Bucket 声明中所述。

4.4.7.4.2. 在 DataProtectionApplication CR 中启用 CSI

您可以在 **DataProtectionApplication** 自定义资源(CR)中启用 Container Storage Interface(CSI)来备份持久性卷，以使用 CSI 快照备份持久性卷。

先决条件

- 云供应商必须支持 CSI 快照。

流程

- 编辑 **DataProtectionApplication** CR，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi 1
```

1 添加 **csi** 默认插件。

其他资源

- 使用 **kubevirt** 和 **openshift** 插件安装数据保护应用程序

4.4.8. 为 OpenShift Virtualization 的数据保护配置 OpenShift API

您可以通过安装 OADP Operator 并配置备份位置，使用 OpenShift Virtualization 安装 OpenShift API for Data Protection (OADP)。然后，您可以安装数据保护应用程序。

使用 OpenShift API for Data Protection 来备份和恢复虚拟机。

注意

OpenShift API for Data Protection with OpenShift Virtualization 支持以下备份和恢复存储选项：

- 容器存储接口 (CSI) 备份
- 使用 DataMover 进行容器存储接口 (CSI) 备份

排除以下存储选项：

- 文件系统备份和恢复
- 卷快照备份和恢复

如需更多信息，请参阅使用文件系统备份备份应用程序： Kopia 或 Restic。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

4.4.8.1. 使用 OpenShift Virtualization 安装和配置 OADP

作为集群管理员，您可以通过安装 OADP Operator 来安装 OADP。

OADP Operator 的最新版本会安装 Velero 1.12。

先决条件

- 使用具有 **cluster-admin** 角色的用户访问集群。

流程

1. 根据您的存储供应商说明安装 OADP Operator。
2. 使用 **kubevirt** 和 **openshift** OADP 插件安装数据保护应用程序(DPA)。
3. 通过创建 **Backup** 自定义资源(CR) 来备份虚拟机。

警告

红帽支持仅限于以下选项：

- CSI 备份
- 使用 DataMover 的 CSI 备份。

您可以通过创建一个 **Restore** CR来恢复 **Backup** CR。

其他资源

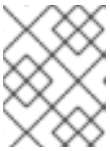
- OADP 插件
- **Backup** 自定义资源 (CR)
- **Restore** CR
- 在受限网络中使用 Operator Lifecycle Manager

4.4.8.2. 安装数据保护应用程序 1.3

您可以通过创建 **DataProtectionApplication** API 的实例来安装数据保护应用程序(DPA)。

先决条件

- 您必须安装 OADP Operator。
- 您必须将对象存储配置为备份位置。
- 如果使用快照来备份 PV，云供应商必须支持原生快照 API 或 Container Storage Interface(CSI) 快照。
- 如果备份和快照位置使用相同的凭证，您必须创建带有默认名称 **cloud-credentials** 的 **Secret**。
- 如果备份和快照位置使用不同的凭证，您必须创建两个 **Secret** ：
  - 带有备份位置的自定义名称的 **secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。
  - 带有快照位置的另一个自定义名称的 **Secret**。您可以将此 **Secret** 添加到 **DataProtectionApplication** CR 中。



注意

如果您不想在安装过程中指定备份或快照位置，您可以使用空 **credentials-velero** 文件创建默认 **Secret**。如果没有默认 **Secret**，安装将失败。

流程

1. 点 **Operators → Installed Operators** 并选择 OADP Operator。
2. 在 **Provided APIs** 下，点 **DataProtectionApplication** 框中的 **Create 实例**。
3. 点 **YAML View** 并更新 **DataProtectionApplication** 清单的参数：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - kubevirt (2)
        - gcp (3)
```

```
      - csi 4
      - openshift 5
    resourceTimeout: 10m 6
  nodeAgent: 7
    enable: true 8
    uploaderType: kopia 9
    podConfig:
      nodeSelector: <node_selector> 10
backupLocations:
  - velero:
      provider: gcp 11
      default: true
      credential:
        key: cloud
        name: <default_secret> 12
      objectStorage:
        bucket: <bucket_name> 13
        prefix: <prefix> 14
```

1. OADP 的默认命名空间是 **openshift-adp**。命名空间是一个变量，可配置。
2. OpenShift Virtualization 需要 **kubevirt** 插件。
3. 为备份供应商指定插件，如 **gcp** （如果存在）。
4. **csi** 插件是使用 CSI 快照备份 PV 所必需的。 **csi** 插件使用 Velero CSI beta 快照 API。您不需要配置快照位置。
5. **openshift** 插件是必需的。
6. 指定在超时发生前等待多个 Velero 资源的分钟，如 Velero CRD 可用、volumeSnapshot 删除和备份存储库可用。默认值为 10m。
7. 将管理请求路由到服务器的管理代理。
8. 如果要启用 **nodeAgent** 并执行文件系统备份，则将此值设置为 **true**。
9. 输入 **kopia** 作为您的上传程序，以使用 Built-in DataMover。 **nodeAgent** 部署守护进程集，这意味着 **nodeAgent** pod 在每个工作节点上运行。您可以通过在 **Backup** CR 中添加 **spec.defaultVolumesToFsBackup: true** 来配置文件系统备份。
10. 指定 Kopia 可用的节点。默认情况下，Kopia 在所有节点上运行。
11. 指定备份供应商。
12. 如果备份供应商使用一个默认插件，为 **Secret** 指定正确的默认名称，如 **cloud-credentials-gcp**。如果指定了一个自定义名称，则使用自定义名称用于备份位置。如果没有指定 **Secret** 名称，则使用默认名称。
13. 指定存储桶作为备份存储位置。如果存储桶不是 Velero 备份的专用存储桶，您必须指定一个前缀。
14. 如果存储桶用于多个目的，请为 Velero 备份指定一个前缀，如 **velero**。

4. 点 **Create**。

验证

1. 运行以下命令，查看 OpenShift API for Data Protection (OADP) 资源来验证安装：

```
$ oc get all -n openshift-adp
```

输出示例

```
NAME                                                 READY   STATUS    RESTARTS   AGE
pod/oadp-operator-controller-manager-67d9494d47-6l8z8   2/2     Running   0          2m8s
pod/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

NAME                                                    TYPE        CLUSTER-IP       EXTERNAL-IP
PORT(S)    AGE
service/oadp-operator-controller-manager-metrics-service   ClusterIP   172.30.70.140
<none>        8443/TCP   2m8s
service/openshift-adp-velero-metrics-svc                ClusterIP   172.30.10.0      <none>
8085/TCP   8h

NAME                  DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE
SELECTOR   AGE
daemonset.apps/node-agent   3         3         3       3            3           <none>          96s

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/oadp-operator-controller-manager   1/1     1            1           2m9s
deployment.apps/velero                             1/1     1            1           96s

NAME                                                       DESIRED   CURRENT   READY   AGE
replicaset.apps/oadp-operator-controller-manager-67d9494d47   1         1         1       2m9s
replicaset.apps/velero-588db7f655                          1         1         1       96s
```

2. 运行以下命令，验证 **DataProtectionApplication** (DPA) 是否已协调：

```
$ oc get dpa dpa-sample -n openshift-adp -o jsonpath='{.status}'
```

输出示例

```
{"conditions":[{"lastTransitionTime":"2023-10-27T01:23:57Z","message":"Reconcile
complete","reason":"Complete","status":"True","type":"Reconciled"}]}
```

3. 验证 **type** 被设置为 **Reconciled**。

4. 运行以下命令，验证备份存储位置并确认 **PHASE** 为 **Available** ：

```
$ oc get backupStorageLocation -n openshift-adp
```

输出示例

```
NAME           PHASE       LAST VALIDATED   AGE     DEFAULT
dpa-sample-1   Available   1s               3d16h   true
```

重要

红帽只支持 OADP 版本 1.3.0 及更新的版本，以及 OpenShift Virtualization 版本 4.14 及更新的版本。

在 1.3.0 前 OADP 版本不支持备份和恢复 OpenShift Virtualization。

4.5. 卸载 OADP

4.5.1. 为数据保护卸载 OpenShift API

您可以通过删除 OADP Operator 来卸载 OpenShift API for Data Protection(OADP)。详情请参阅从集群中删除 Operator。

4.6. OADP 备份

4.6.1. 备份应用程序

您可以通过创建一个 **Backup** 自定义资源 (CR) 来备份应用程序。请参阅 创建备份 CR。

- **Backup** CR 为 Kubernetes 资源和 S3 对象存储上的内部镜像创建备份文件。
- 如果您的云供应商有原生快照 API 或支持 CSI 快照，则 **Backup** CR 通过创建快照来备份持久性卷 (PV)。有关使用 CSI 快照的更多信息，请参阅使用 CSI 快照备份持久性卷。

有关 CSI 卷快照的更多信息，请参阅 CSI 卷快照。

重要

CloudStorage API（它自动为对象存储创建一个存储桶）只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

注意

当使用 **CloudStorage** 对象，并希望 OADP 使用 **CloudStorage** API 自动创建 S3 存储桶以用作 **BackupStorageLocation** 时，**CloudStorage** API 只是一个技术预览功能。

CloudStorage API 支持通过指定一个现有的 S3 存储桶来手动创建 **BackupStorageLocation** 对象。自动创建 S3 存储桶的 **CloudStorage** API 目前只为 AWS S3 存储启用。

- 如果您的云供应商不支持快照，或者应用程序位于 NFS 数据卷中，您可以使用 Kopia 或 Restic 创建备份。请参阅使用文件系统备份备份应用程序：Kopia 或 Restic。

重要

OpenShift API for Data Protection (OADP) 不支持对由其他软件创建的卷快照进行备份。

您可以创建备份 hook，以便在备份操作之前或之后运行命令。请参阅创建备份 hook。

您可以通过创建一个 **Schedule** CR 而不是 **Backup** CR 来调度备份。请参阅使用 Schedule CR] 调度备份。

4.6.1.1. 已知问题

OpenShift Container Platform 4.16 强制执行一个 pod 安全准入 (PSA) 策略，该策略可能会在 Restic 恢复过程中阻止 pod 的就绪度。

这个问题已在 OADP 1.1.6 和 OADP 1.2.2 版本中解决，因此建议用户升级到这些版本。

如需更多信息，请参阅因为更改 PSA 策略，在 OCP 4.15 上进行 Restic 恢复部分失败 。

其他资源

- 在集群上为管理员安装 Operator
- 在命名空间中为非管理员用户安装 Operator

4.6.2. 创建备份 CR

您可以通过创建 **Backup** 备份自定义资源(CR)来备份 Kubernetes 镜像、内部镜像和持久性卷(PV)。

先决条件

- 您必须安装用于数据保护(OADP)Operator 的 OpenShift API。
- **DataProtectionApplication** CR 必须处于 **Ready** 状态。
- 备份位置先决条件：
  - 您必须为 Velero 配置 S3 对象存储。
  - 您必须在 **DataProtectionApplication** CR 中配置了一个备份位置。
- 快照位置先决条件：
  - 您的云供应商必须具有原生快照 API 或支持 Container Storage Interface(CSI)快照。
  - 对于 CSI 快照，您必须创建一个 **VolumeSnapshotClass** CR 来注册 CSI 驱动程序。
  - 您必须在 **DataProtectionApplication** CR 中配置了一个卷位置。

流程

1. 输入以下命令来检索 **backupStorageLocations** CR：

```
$ oc get backupStorageLocations -n openshift-adp
```

输**出示例**

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   velero-sample-1   Available   11s              31m
```

2. 创建一个 **Backup** CR，如下例所示：

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  hooks: {}
  includedNamespaces:
  - <namespace> 1
  includedResources: [] 2
  excludedResources: [] 3
  storageLocation: <velero-sample-1> 4
  ttl: 720h0m0s
  labelSelector: 5
    matchLabels:
      app: <label_1>
      app: <label_2>
      app: <label_3>
  orLabelSelectors: 6
  - matchLabels:
      app: <label_1>
      app: <label_2>
      app: <label_3>
```

1. 指定要备份的命名空间数组。
2. 可选：指定一个要包含在备份中的资源的数组。资源可以是缩写方式（例如，'po' 代表 'pods'）或完全限定的方式。如果未指定，则会包含所有资源。
3. 可选：指定要从备份中排除的资源数组。资源可以是缩写方式（例如，'po' 代表 'pods'）或完全限定的方式。
4. 指定 **backupStorageLocations** CR 的名称。
5. 具有**所有**指定标签的备份资源的 {key,value} 对映射。
6. 具有**一个或多个**指定标签的备份资源的 {key,value} 对映射。

3. 验证 **Backup** CR 的状态是否为 **Completed** ：

```
$ oc get backup -n openshift-adp <backup> -o jsonpath='{.status.phase}'
```

4.6.3. 使用 CSI 快照备份持久性卷

在创建 **Backup** CR 前，您可以编辑云存储的 **VolumeSnapshotClass** 自定义资源(CR)来备份持久性卷(CSI)快照，请参阅 CSI 卷快照。

如需更多信息，请参阅创建备份 CR。

先决条件

- 云供应商必须支持 CSI 快照。
- 您必须在 **DataProtectionApplication** CR 中启用 CSI。

流程

- 将 **metadata.labels.velero.io/csi-volumesnapshot-class: "true"** 键值对添加到 **VolumeSnapshotClass** CR：

  配置文件示例

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true" (1)
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: true (2)
driver: <csi_driver>
deletionPolicy: <deletion_policy_type> (3)
```

(1) 必须设置为 **true**。

(2) 必须设置为 **true**。

(3) OADP 支持 CSI 和 Data Mover 备份和恢复的 **Retain** 和 **Delete** 删除策略类型。对于 OADP 1.2 Data Mover，将删除策略类型设置为 **Retain**。

后续步骤

- 现在，您可以创建一个 **Backup** CR。

4.6.4. 使用文件系统备份对应用程序进行备份：Kopia 或 Restic

您可以使用 OADP 从卷的文件系统备份和恢复附加到 pod 的 Kubernetes 卷。这个过程称为文件系统备份 (FSB) 或 Pod 卷备份 (PVB)。它通过使用来自开源备份工具 Restic 或 Kopia 的模块来完成。

如果您的云供应商不支持快照，或者应用程序位于 NFS 数据卷中，您可以使用 FSB 创建备份。

注意

默认情况下，Restic 由 OADP Operator 安装。如果您希望安装 Kopia。

FSB 与 OADP 集成提供了一种解决方案，用于备份和恢复几乎任何类型的 Kubernetes 卷。这个集成是 OADP 的一个额外功能，不是现有功能的替代品。

您可以通过编辑 **Backup** 备份自定义资源 (CR) 来使用 Kopia 或 Restic 对 Kubernetes 资源、内部镜像和持久性卷备份。

您不需要在 **DataProtectionApplication** CR 中指定快照位置。

注意

在 OADP 版本 1.3 及更高版本中，您可以使用 Kopia 或 Restic 备份应用程序。

对于 Built-in DataMover，您必须使用 Kopia。

在 OADP 版本 1.2 及更早版本中，您只能使用 Restic 备份应用程序。

重要

FSB 不支持对 **hostPath** 卷进行备份。如需更多信息，请参阅 FSB 限制。

先决条件

- 您必须安装用于数据保护(OADP)Operator 的 OpenShift API。
- 您不能通过将 **DataProtectionApplication** CR 中的 **spec.configuration.nodeAgent.enable** to **false** 来禁用默认的 **nodeAgent** 安装。
- 您必须在 **DataProtectionApplication** CR 中将 **spec.configuration.nodeAgent.uploaderType** 设置为 **kopia** 或 **restic** 来选择 Kopia 或 Restic 作为 uploader。
- **DataProtectionApplication** CR 必须处于 **Ready** 状态。

流程

- 创建 **Backup** CR，如下例所示：

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  defaultVolumesToFsBackup: true 1
...
```

1 在 OADP 版本 1.2 及更高版本中，在 **spec** 块中添加 **defaultVolumesToFsBackup: true** 设置。在 OADP 版本 1.1 中，添加 **defaultVolumesToRestic: true**。

4.6.5. 创建备份 hook

在执行备份时，可以根据正在备份的 pod，指定在 pod 内要执行的一个或多个命令。

可将命令配置为在任何自定义操作处理(*Pre* hook)或所有自定义操作完成后执行，且由自定义操作指定的任何其他项目都已备份（*Post* hook）。

您可以通过编辑备份自定义资源(CR)来创建 **Backup** hook 以在 pod 中运行的容器中运行命令。

流程

- 在 **Backup** CR 的 **spec.hooks** 块中添加 hook，如下例所示：

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces: 2
        - <namespace>
        includedResources: []
        - pods 3
        excludedResources: [] 4
        labelSelector: 5
          matchLabels:
            app: velero
            component: server
        pre: 6
          - exec:
              container: <container> 7
              command:
              - /bin/uname 8
              - -a
              onError: Fail 9
              timeout: 30s 10
        post: 11
...
```

1. 可选：您可以指定 hook 应用的命名空间。如果没有指定这个值，则 hook 适用于所有命名空间。
2. 可选：您可以指定 hook 不应用到的命名空间。
3. 目前，pod 是唯一可以应用 hook 的支持的资源。
4. 可选：您可以指定 hook 不应用到的资源。
5. 可选：此 hook 仅适用于与标签匹配的对象。如果没有指定这个值，则 hook 适用于所有对象。
6. 备份前要运行的 hook 数组。
7. 可选：如果没有指定容器，该命令将在 pod 的第一个容器中运行。
8. 这是添加 **init** 容器的入口点。
9. 错误处理允许的值是 **Fail** 和 **Continue**。默认值为 **Fail**。
10. 可选：等待命令运行的时间。默认值为 **30s**。
11. 此块定义了在备份后运行的一组 hook，其参数与 pre-backup hook 相同。

4.6.6. 使用 Schedule CR 调度备份

调度操作允许您在特定时间创建由 Cron 表达式指定的数据的备份。

您可以通过创建 **Schedule** 自定义资源(CR)而不是 **Backup** CR 来调度备份。

> **警告**
>
> 在您的备份调度中留有足够的时间，以便在创建另一个备份前完成了当前的备份。
>
> 例如，如果对一个命名空间进行备份通常需要 10 分钟才能完成，则调度的备份频率不应该超过每 15 分钟一次。

先决条件

- 您必须安装用于数据保护(OADP)Operator 的 OpenShift API。
- **DataProtectionApplication** CR 必须处于 **Ready** 状态。

流程

1. 检索 **backupStorageLocations** CR：

```
$ oc get backupStorageLocations -n openshift-adp
```

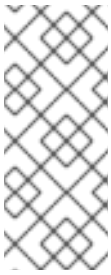
输出示例

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   velero-sample-1   Available   11s              31m
```

2. 创建一个 **Schedule** CR，如下例所示：

```
$ cat << EOF | oc apply -f -
apiVersion: velero.io/v1
kind: Schedule
metadata:
  name: <schedule>
  namespace: openshift-adp
spec:
  schedule: 0 7 * * * 1
  template:
    hooks: {}
    includedNamespaces:
    - <namespace> 2
    storageLocation: <velero-sample-1> 3
    defaultVolumesToFsBackup: true 4
    ttl: 720h0m0s
EOF
```

1. 调度备份的 **cron** 表达式，例如 **0 7 * * *** 代表在每天 7:00 执行备份。



注意

要以特定间隔调度备份，以以下格式输入 **<duration_in_minutes>**：

```
schedule: "*/10 * * * *"
```

在引号 (" ") 之间输入分钟值。

2. 要备份的命名空间数组。
3. **backupStorageLocations** CR 的名称。
4. 可选：在 OADP 版本 1.2 及更高版本中，在使用 Restic 进行卷备份时，将 **defaultVolumesToFsBackup: true** 键值对添加到您的配置中。在 OADP 版本 1.1 中，在使用 Restic 备份卷时添加 **defaultVolumesToRestic: true** 键值对。

    1. 在调度的备份运行后验证 **Schedule** CR 的状态是否为 **Completed** ：

```
$ oc get schedule -n openshift-adp <schedule> -o jsonpath='{.status.phase}'
```

4.6.7. 删除备份

您可以通过删除 **Backup** 自定义资源 (CR) 来删除备份文件。

警告

删除 **Backup** CR 和关联的对象存储数据后，您无法恢复删除的数据。

先决条件

- 您创建了 **Backup** CR。
- 您知道 **Backup** CR 的名称以及包含它的命名空间。
- 下载 Velero CLI 工具。
- 您可以访问集群中的 Velero 二进制文件。

流程

- 选择以下操作之一来删除 **Backup** CR：
    - 要删除 **Backup** CR 并保留关联的对象存储数据，请运行以下命令：

```
$ oc delete backup <backup_CR_name> -n <velero_namespace>
```

- 要删除 **Backup** CR 并删除关联的对象存储数据，请运行以下命令：

```
$ velero backup delete <backup_CR_name> -n <velero_namespace>
```

其中：

<backup_CR_name>
: **Backup** 自定义资源的名称。

<velero_namespace>
: 包含 **Backup** 自定义资源的命名空间。

4.6.8. 关于 Kopia

Kopia 是一个快速安全的开源备份和恢复工具，可让您创建数据的加密快照，并将快照保存到您选择的远程或云存储中。

Kopia 支持网络和本地存储位置，以及许多云或远程存储位置，包括：

- Amazon S3 以及与 S3 兼容的任何云存储
- Azure Blob Storage
- Google Cloud Storage 平台

Kopia 对快照使用可内容访问的存储：

- 快照始终是以增量方式进行的；已包含在之前快照中的数据不会重新上传到存储库。仅当文件被修改时，文件才会再次上传到存储库。
- 存储的数据会被去除重复数据；如果存在同一文件的多个副本，则仅存储其中一个文件。
- 如果文件被移动或重命名，Kopia 可以识别它们具有相同的内容，且不会重新上传它们。

4.6.8.1. OADP 与 Kopia 集成

除了 Restic 外，OADP 1.3 还支持 Kopia 作为 pod 卷备份的备份机制。您需要在安装时通过在 **DataProtectionApplication** 自定义资源(CR) 中设置 **uploaderType** 字段来选择其中一个。可能的值为 **restic** 或 **kopia**。如果没有指定 **uploaderType**，OADP 1.3 默认为使用 Kopia 作为备份机制。数据会从一个统一的存储库中读取或写入。

以下示例显示了配置了使用 Kopia 的 **DataProtectionApplication** CR：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    nodeAgent:
      enable: true
      uploaderType: kopia
# ...
```

4.7. OADP 恢复

4.7.1. 恢复应用程序

您可以通过创建一个 **Restore** 自定义资源 (CR) 来恢复应用程序备份。请参阅 创建 Restore CR。

您可以通过编辑 **Restore** CR 创建恢复 hook，以便在 pod 中的容器中运行命令。请参阅 创建恢复 hook。

4.7.1.1. 创建恢复 CR

您可以通过创建一个 **Restore** CR 来恢复 **Backup** 自定义资源(CR)。

先决条件

- 您必须安装用于数据保护(OADP)Operator 的 OpenShift API。
- **DataProtectionApplication** CR 必须处于 **Ready** 状态。
- 您必须具有 Velero **Backup** CR。
- 持久性卷 (PV) 容量必须与备份时请求的大小匹配。如果需要，调整请求的大小。

流程

1. 创建一个 **Restore** CR，如下例所示：

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  backupName: <backup> 1
  includedResources: [] 2
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  restorePVs: true 3
```

1. **备份 CR 的名称。**
2. 可选：指定要包含在恢复过程中的资源数组。资源可以是缩写方式（例如，**po** 代表 **pods**）或完全限定的方式。如果未指定，则会包含所有资源。
3. 可选： **restorePV** 参数可以设置为 **false**，以从 Container Storage Interface (CSI) 快照的 **VolumeSnapshot** 或配置 **VolumeSnapshotLocation** 时从原生快照中恢复 **PersistentVolume**。

2. 输入以下命令验证 **Restore** CR 的状态是否为 **Completed** ：

```
$ oc get restore -n openshift-adp <restore> -o jsonpath='{.status.phase}'
```

3. 输入以下命令验证备份资源是否已恢复：

```
$ oc get all -n <namespace> 1
```

1 备份的命名空间。

4. 如果您使用卷恢复 **DeploymentConfig**，或使用 post-restore hook，请输入以下命令运行 **dc-post-restore.sh** cleanup 脚本：

```
$ bash dc-restic-post-restore.sh -> dc-post-restore.sh
```

注意

在恢复过程中，OADP Velero 插件会缩减 **DeploymentConfig** 对象，并将 pod 恢复为独立 pod。这是为了防止集群在恢复时立即删除恢复的 **DeploymentConfig** pod，并允许 restore 和 post-restore hook 在恢复的 pod 上完成其操作。下面显示的清理脚本会删除这些断开连接的 pod，并将任何 **DeploymentConfig** 对象扩展至适当的副本数。

例 4.1. dc-restic-post-restore.sh → dc-post-restore.sh cleanup 脚本

```
#!/bin/bash
set -e

# if sha256sum exists, use it to check the integrity of the file
if command -v sha256sum >/dev/null 2>&1; then
  CHECKSUM_CMD="sha256sum"
else
  CHECKSUM_CMD="shasum -a 256"
fi

label_name () {
    if [ "${#1}" -le "63" ]; then
 echo $1
 return
    fi
    sha=$(echo -n $1|$CHECKSUM_CMD)
    echo "${1:0:57}${sha:0:6}"
}

OADP_NAMESPACE=${OADP_NAMESPACE:=openshift-adp}

if [[ $# -ne 1 ]]; then
    echo "usage: ${BASH_SOURCE} restore-name"
    exit 1
fi

echo using OADP Namespace $OADP_NAMESPACE
echo restore: $1

label=$(label_name $1)
```

```
echo label: $label

echo Deleting disconnected restore pods
oc delete pods -l oadp.openshift.io/disconnected-from-dc=$label

for dc in $(oc get dc --all-namespaces -l oadp.openshift.io/replicas-modified=$label -o
jsonpath='{range .items[*]}{.metadata.namespace}{","}{.metadata.name}{","}
{.metadata.annotations.oadp\.openshift\.io/original-replicas}{","}
{.metadata.annotations.oadp\.openshift\.io/original-paused}{"\n"}')
do
    IFS=',' read -ra dc_arr <<< "$dc"
    if [ ${#dc_arr[0]} -gt 0 ]; then
 echo Found deployment ${dc_arr[0]}/${dc_arr[1]}, setting replicas: ${dc_arr[2]}, paused:
${dc_arr[3]}
 cat <<EOF | oc patch dc  -n ${dc_arr[0]} ${dc_arr[1]} --patch-file /dev/stdin
spec:
  replicas: ${dc_arr[2]}
  paused: ${dc_arr[3]}
EOF
    fi
done
```

4.7.1.2. 创建恢复 hook

您可以通过编辑 **Restore** 自定义资源 (CR)创建恢复 hook，以便在 pod 中的容器中运行命令。

您可以创建两种类型的恢复 hook：

- **init** hook 将 init 容器添加到 pod，以便在应用程序容器启动前执行设置任务。
  如果您恢复 Restic 备份，则会在恢复 hook init 容器前添加 **restic-wait** init 容器。
- **exec** hook 在恢复的 pod 的容器中运行命令或脚本。

流程

- 在 **Restore** CR 的 **spec.hooks** 块中添加 hook，如下例所示：

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces:
        - <namespace>
        includedResources:
        - pods 2
        excludedResources: []
        labelSelector: 3
```

```
      matchLabels:
        app: velero
        component: server
    postHooks:
    - init:
        initContainers:
        - name: restore-hook-init
          image: alpine:latest
          volumeMounts:
          - mountPath: /restores/pvc1-vm
            name: pvc1-vm
          command:
          - /bin/ash
          - -c
        timeout: 4
    - exec:
        container: <container> 5
        command:
        - /bin/bash 6
        - -c
        - "psql < /backup/backup.sql"
        waitTimeout: 5m 7
        execTimeout: 1m 8
        onError: Continue 9
```

1. 可选： hook 应用的命名空间数组。如果没有指定这个值，则 hook 适用于所有命名空间。
2. 目前，pod 是唯一可以应用 hook 的支持的资源。
3. 可选：此 hook 仅适用于与标签选择器匹配的对象。
4. 可选：超时指定 Velero 等待 **initContainers** 完成的最大时间长度。
5. 可选：如果没有指定容器，该命令将在 pod 的第一个容器中运行。
6. 这是正在添加的 init 容器的入口点。
7. 可选：等待容器就绪的时间。这应该足够长，以便容器可以启动，在相同容器中的任何以前的 hook 可以完成。如果没有设置，恢复过程会无限期等待。
8. 可选：等待命令运行的时间。默认值为 **30s**。
9. 错误处理的允许值为 **Fail** 和 **Continue**：
   - **Continue**: 只记录命令失败。
   - **Fail**: 任何 pod 中的任何容器中没有更多恢复 hook 运行。 **Restore** CR 的状态将是 **PartiallyFailed**。

4.8. OADP 和 ROSA

4.8.1. 使用 OADP 在 ROSA 集群上备份应用程序

您可以使用 OpenShift API for Data Protection (OADP)与 Red Hat OpenShift Service on AWS (ROSA) 集群来备份和恢复应用程序数据。

ROSA 是一个完全管理的一站式应用平台，允许您通过构建和部署应用程序来为客户提供价值。

ROSA 提供与各种 Amazon Web Services（AWS）计算、数据库、分析、机器学习、网络、移动和其他服务无缝集成，以加快为您的客户构建和交付不同体验。

您可以直接从 AWS 帐户订阅该服务。

创建集群后，您可以使用 OpenShift Container Platform Web 控制台或 Red Hat OpenShift Cluster Manager 来运行集群。您还可以在 OpenShift API 和命令行界面 (CLI) 工具中使用 ROSA。

有关 ROSA 安装的更多信息，请参阅在 AWS (ROSA)上安装 Red Hat OpenShift Service 。

在为数据保护(OADP)安装 OpenShift API 前，您必须为 OADP 设置角色和策略凭证，以便它可以使用 Amazon Web Services API。

这个过程在以下两个阶段执行：

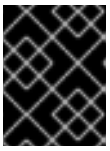
1. 准备 AWS 凭证
2. 安装 OADP Operator 并为其提供 IAM 角色

4.8.1.1. 为 OADP 准备 AWS 凭证

必须准备 Amazon Web Services 帐户，并配置为接受 OpenShift API for Data Protection (OADP) 安装。

流程

1. 运行以下命令来创建以下环境变量：



重要

更改集群名称来匹配您的 ROSA 集群，并确保以管理员身份登录到集群。在继续操作前，确保所有字段被正常输出。

```
$ export CLUSTER_NAME=my-cluster (1)
  export ROSA_CLUSTER_ID=$(rosa describe cluster -c ${CLUSTER_NAME} --output json |
jq -r .id)
  export REGION=$(rosa describe cluster -c ${CLUSTER_NAME} --output json | jq -r
.region.id)
  export OIDC_ENDPOINT=$(oc get authentication.config.openshift.io cluster -o
jsonpath='{.spec.serviceAccountIssuer}' | sed 's|^https://||')
  export AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
  export CLUSTER_VERSION=$(rosa describe cluster -c ${CLUSTER_NAME} -o json | jq -r
.version.raw_id | cut -f -2 -d '.')
  export ROLE_NAME="${CLUSTER_NAME}-openshift-oadp-aws-cloud-credentials"
  export SCRATCH="/tmp/${CLUSTER_NAME}/oadp"
  mkdir -p ${SCRATCH}
  echo "Cluster ID: ${ROSA_CLUSTER_ID}, Region: ${REGION}, OIDC Endpoint:
  ${OIDC_ENDPOINT}, AWS Account ID: ${AWS_ACCOUNT_ID}"
```

(1) 将 **my-cluster** 替换为您的 ROSA 集群名称。

2. 在 AWS 帐户中，创建一个 IAM 策略以允许访问 AWS S3：

   a. 运行以下命令，检查策略是否存在：

```
$ POLICY_ARN=$(aws iam list-policies --query "Policies[?
PolicyName=='RosaOadpVer1'].{ARN:Arn}" --output text) 1
```

   1 将 **RosaOadp** 替换为您的策略名称。

   b. 输入以下命令来创建策略 JSON 文件，然后在 ROSA 中创建策略：

注意

如果没有找到策略 ARN，命令会创建策略。如果策略 ARN 已存在，则 **if** 语句会有意跳过策略创建。

```
$ if [[ -z "${POLICY_ARN}" ]]; then
  cat << EOF > ${SCRATCH}/policy.json 1
  {
  "Version": "2012-10-17",
  "Statement": [
   {
     "Effect": "Allow",
     "Action": [
       "s3:CreateBucket",
       "s3:DeleteBucket",
       "s3:PutBucketTagging",
       "s3:GetBucketTagging",
       "s3:PutEncryptionConfiguration",
       "s3:GetEncryptionConfiguration",
       "s3:PutLifecycleConfiguration",
       "s3:GetLifecycleConfiguration",
       "s3:GetBucketLocation",
       "s3:ListBucket",
       "s3:GetObject",
       "s3:PutObject",
       "s3:DeleteObject",
       "s3:ListBucketMultipartUploads",
       "s3:AbortMultipartUploads",
       "s3:ListMultipartUploadParts",
       "s3:DescribeSnapshots",
       "ec2:DescribeVolumes",
       "ec2:DescribeVolumeAttribute",
       "ec2:DescribeVolumesModifications",
       "ec2:DescribeVolumeStatus",
       "ec2:CreateTags",
       "ec2:CreateVolume",
       "ec2:CreateSnapshot",
       "ec2:DeleteSnapshot"
     ],
     "Resource": "*"
   }
  ]}
EOF
```

```
  POLICY_ARN=$(aws iam create-policy --policy-name "RosaOadpVer1" \
  --policy-document file:///${SCRATCH}/policy.json --query Policy.Arn \
  --tags Key=rosa_openshift_version,Value=${CLUSTER_VERSION}
Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-oadp Key=operator_name,Value=openshift-
oadp \
  --output text)
  fi
```

1 **SCRATCH** 是为环境变量创建的临时目录的名称。

c. 运行以下命令来查看策略 ARN：

```
$ echo ${POLICY_ARN}
```

3. 为集群创建 IAM 角色信任策略：

a. 运行以下命令来创建信任策略文件：

```
$ cat <<EOF > ${SCRATCH}/trust-policy.json
 {
    "Version":2012-10-17",
    "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-
provider/${OIDC_ENDPOINT}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_ENDPOINT}:sub": [
            "system:serviceaccount:openshift-adp:openshift-adp-controller-manager",
            "system:serviceaccount:openshift-adp:velero"]
        }
      }
    }]
 }
EOF
```

b. 运行以下命令来创建角色：

```
$ ROLE_ARN=$(aws iam create-role --role-name \
  "${ROLE_NAME}" \
  --assume-role-policy-document file://${SCRATCH}/trust-policy.json \
--tags Key=rosa_cluster_id,Value=${ROSA_CLUSTER_ID}
Key=rosa_openshift_version,Value=${CLUSTER_VERSION}
Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-adp Key=operator_name,Value=openshift-
oadp \
   --query Role.Arn --output text)
```

c. 运行以下命令来查看角色 ARN：

```
$ echo ${ROLE_ARN}
```

4. 运行以下命令，将 IAM 策略附加到 IAM 角色：

```
$ aws iam attach-role-policy --role-name "${ROLE_NAME}" \
  --policy-arn ${POLICY_ARN}
```

4.8.1.2. 安装 OADP Operator 并提供 IAM 角色

AWS 安全令牌服务 (AWS STS) 是一个全局 Web 服务，它为 IAM 或联邦用户提供简短凭证。带有 STS 的 OpenShift Container Platform (ROSA) 是 ROSA 集群的建议凭证模式。本文档论述了如何使用 AWS STS 在 ROSA 上安装 OpenShift API for Data Protection (OADP)。

重要

使用 AWS STS 环境的 ROSA 上的 OADP 不支持 Restic 和 Kopia。验证 Restic 和 Kopia 节点代理是否已禁用。对于备份卷，使用 AWS STS 的 ROSA 上 OADP 仅支持原生快照和 Container Storage Interface (CSI) 快照。

在使用 STS 验证的 Amazon ROSA 集群中，不支持在不同的 AWS 区域中恢复备份数据。

目前，ROSA 集群不支持 Data Mover 功能。您可以使用原生 AWS S3 工具移动数据。

先决条件

- 具有所需访问权限和令牌的 OpenShift Container Platform ROSA 集群。具体步骤请查看 *为 OADP 准备 AWS 凭证*。如果您计划使用两个不同的集群来备份和恢复，您必须为每个集群准备 AWS 凭证，包括 **ROLE_ARN**。

流程

1. 输入以下命令，从 AWS 令牌文件创建 OpenShift Container Platform secret：

   a. 创建凭证文件：

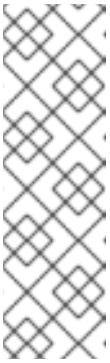
```
$ cat <<EOF > ${SCRATCH}/credentials
  [default]
  role_arn = ${ROLE_ARN}
  web_identity_token_file = /var/run/secrets/openshift/serviceaccount/token
EOF
```

   b. 为 OADP 创建命名空间：

```
$ oc create namespace openshift-adp
```

   c. 创建 OpenShift Container Platform secret：

```
$ oc -n openshift-adp create secret generic cloud-credentials \
  --from-file=${SCRATCH}/credentials
```



注意

在 OpenShift Container Platform 版本 4.14 及更新的版本中，OADP Operator 通过 Operator Lifecycle Manager (OLM) 和 Cloud Credentials Operator (CCO)支持新的标准化 STS 工作流。在此工作流中，您不需要创建上述 secret，您需要在使用 OpenShift Container Platform Web 控制台安装 OLM 管理的 Operator 时提供角色 ARN，请参阅使用 *Web 控制台从 OperatorHub 安装*。

前面的 secret 由 CCO 自动创建。

2. 安装 OADP Operator：

   a. 在 OpenShift Container Platform Web 控制台中，浏览 **Operators → OperatorHub**。

   b. 搜索 **OADP Operator**。

   c. 在 **role_ARN** 字段中，粘贴之前创建的 role_arn，再点 **Install**。

3. 输入以下命令，使用 AWS 凭证创建 AWS 云存储：

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: CloudStorage
  metadata:
    name: ${CLUSTER_NAME}-oadp
    namespace: openshift-adp
  spec:
    creationSecret:
      key: credentials
      name: cloud-credentials
    enableSharedConfig: true
    name: ${CLUSTER_NAME}-oadp
    provider: aws
    region: $REGION
EOF
```

4. 输入以下命令检查应用程序的存储默认存储类：

```
$ oc get pvc -n <namespace>
```

输**出示例**

```
NAME     STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS   AGE
applog   Bound    pvc-351791ae-b6ab-4e8b-88a4-30f73caf5ef8   1Gi        RWO            gp3-
csi        4d19h
mysql    Bound    pvc-16b8e009-a20a-4379-accc-bc81fedd0621   1Gi        RWO            gp3-
csi        4d19h
```

5. 运行以下命令来获取存储类：

```
$ oc get storageclass
```

输**出示例**

```
NAME              PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
gp2               kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   true
4d21h
gp2-csi           ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3               ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3-csi (default)   ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
```

注意

以下存储类可以正常工作：

- gp3-csi
- gp2-csi
- gp3
- gp2

如果要备份的应用程序或应用程序都使用带有 Container Storage Interface (CSI) 的持久性卷 (PV)，建议在 OADP DPA 配置中包含 CSI 插件。

6. 创建 **DataProtectionApplication** 资源，以配置存储备份和卷快照的存储的连接：

   a. 如果您只使用 CSI 卷，请输入以下命令部署数据保护应用程序：

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true 1
    features:
      dataMover:
        enable: false
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
```

```
      velero:
        defaultPlugins:
        - openshift
        - aws
        - csi
      restic:
        enable: false
EOF
```

1 ROSA 支持内部镜像备份。如果您不想使用镜像备份，请将此字段设置为 **false**。

a. 如果使用 CSI 或非 CSI 卷，请输入以下命令来部署数据保护应用程序：

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true 1
    features:
      dataMover:
        enable: false
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
      velero:
        defaultPlugins:
        - openshift
        - aws
      nodeAgent: 2
        enable: false
        uploaderType: restic
    snapshotLocations:
      - velero:
          config:
            credentialsFile: /tmp/credentials/openshift-adp/cloud-credentials-credentials 3
            enableSharedConfig: "true" 4
            profile: default 5
            region: ${REGION} 6
          provider: aws
EOF
```

1 ROSA 支持内部镜像备份。如果您不想使用镜像备份，请将此字段设置为 false。

2 请参阅有关 **nodeAgent** 属性的重要备注。

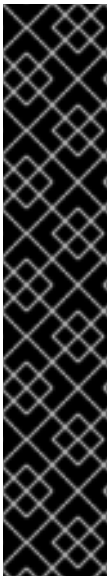
3 **credentialsFile** 字段是 pod 上存储桶凭证的挂载位置。

4 **enableSharedConfig** 字段允许 **snapshotLocations** 共享或重复使用为存储桶定义的凭证。

5 使用 AWS 凭证文件中设置的配置集名称。

6 将 **region** 指定为您的 AWS 区域。这必须与集群区域相同。

现在，您可以备份和恢复 OpenShift Container Platform 应用程序，如 *备份应用程序* 中所述。



重要

此配置中的 **restic** 的 **enable** 参数设置为 **false**，因为 OADP 不支持 ROSA 环境中的 Restic。

如果使用 OADP 1.2，请替换此配置：

```
nodeAgent:
  enable: false
  uploaderType: restic
```

使用以下配置：

```
restic:
  enable: false
```

如果要使用两个不同的集群来备份和恢复，则两个集群必须在云存储 CR 和 OADP **DataProtectionApplication** 配置中具有相同的 AWS S3 存储名称。

其他资源

- 使用 Web 控制台 从 OperatorHub 安装。
- 备份应用程序

4.8.1.3. 示例：使用可选清理在 OADP ROSA STS 上备份工作负载

4.8.1.3.1. 使用 OADP 和 ROSA STS 执行备份

以下示例 **hello-world** 应用没有附加持久性卷 (PV)。使用 OpenShift API 对 Red Hat OpenShift Service on AWS (ROSA) STS 进行数据保护 (OADP) 进行备份。

数据保护应用程序 (DPA) 配置都将正常工作。

1. 运行以下命令，创建一个工作负载来备份：

```
$ oc create namespace hello-world
```

```
$ oc new-app -n hello-world --image=docker.io/openshift/hello-openshift
```

2. 运行以下命令来公开路由：

```
$ oc expose service/hello-openshift -n hello-world
```

3. 运行以下命令检查应用程序是否正常工作：

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

输出示例

```
Hello OpenShift!
```

4. 运行以下命令来备份工作负载：

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Backup
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    includedNamespaces:
    - hello-world
    storageLocation: ${CLUSTER_NAME}-dpa-1
    ttl: 720h0m0s
EOF
```

5. 等待备份完成，然后运行以下命令：

```
$ watch "oc -n openshift-adp get backup hello-world -o json | jq .status"
```

输出示例

```
{
  "completionTimestamp": "2022-09-07T22:20:44Z",
  "expiration": "2022-10-07T22:20:22Z",
  "formatVersion": "1.1.0",
  "phase": "Completed",
  "progress": {
    "itemsBackedUp": 58,
    "totalItems": 58
  },
  "startTimestamp": "2022-09-07T22:20:22Z",
  "version": 1
}
```

6. 运行以下命令来删除 demo 工作负载：

```
$ oc delete ns hello-world
```

7. 运行以下命令，从备份中恢复工作负载：

```
$ cat << EOF | oc create -f -
```

```
  apiVersion: velero.io/v1
  kind: Restore
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    backupName: hello-world
EOF
```

8. 运行以下命令等待 Restore 完成：

```
$ watch "oc -n openshift-adp get restore hello-world -o json | jq .status"
```

输**出示例**

```
{
  "completionTimestamp": "2022-09-07T22:25:47Z",
  "phase": "Completed",
  "progress": {
    "itemsRestored": 38,
    "totalItems": 38
  },
  "startTimestamp": "2022-09-07T22:25:28Z",
  "warnings": 9
}
```

9. 运行以下命令检查工作负载是否已恢复：

```
$ oc -n hello-world get pods
```

输**出示例**

```
NAME                          READY   STATUS    RESTARTS   AGE
hello-openshift-9f885f7c6-kdjpj   1/1     Running   0          90s
```

10. 运行以下命令来检查 JSONPath：

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

输**出示例**

```
Hello OpenShift!
```

注意

有关故障排除提示的信息，请参阅 OADP 团队的 故障排除文档。

4.8.1.3.2. 使用 OADP 和 ROSA STS 的备份后清理集群

如果您需要卸载 OpenShift API for Data Protection (OADP) Operator 以及本例中的备份和 S3 存储桶，请按照以下步骤操作。

流程

1. 运行以下命令来删除工作负载：

```
$ oc delete ns hello-world
```

2. 运行以下命令来删除数据保护应用程序 (DPA)：

```
$ oc -n openshift-adp delete dpa ${CLUSTER_NAME}-dpa
```

3. 运行以下命令来删除云存储：

```
$ oc -n openshift-adp delete cloudstorage ${CLUSTER_NAME}-oadp
```

警告

如果这个命令挂起，您可能需要通过运行以下命令来删除终结器：

```
$ oc -n openshift-adp patch cloudstorage ${CLUSTER_NAME}-oadp -p
'{"metadata":{"finalizers":null}}' --type=merge
```

4. 如果不再需要 Operator，请运行以下命令删除它：

```
$ oc -n openshift-adp delete subscription oadp-operator
```

5. 从 Operator 中删除命名空间：

```
$ oc delete ns openshift-adp
```

6. 如果不再需要备份和恢复资源，请运行以下命令从集群中删除它们：

```
$ oc delete backup hello-world
```

7. 要删除 AWS S3 中的备份、恢复和远程对象，请运行以下命令：

```
$ velero backup delete hello-world
```

8. 如果您不再需要自定义资源定义 (CRD)，请运行以下命令从集群中删除它们：

```
$ for CRD in `oc get crds | grep velero | awk '{print $1}'`; do oc delete crd $CRD; done
```

9. 运行以下命令来删除 AWS S3 存储桶：

```
$ aws s3 rm s3://${CLUSTER_NAME}-oadp --recursive
```

```
$ aws s3api delete-bucket --bucket ${CLUSTER_NAME}-oadp
```

10. 运行以下命令，将策略从角色分离：

```
$ aws iam detach-role-policy --role-name "${ROLE_NAME}"  --policy-arn "${POLICY_ARN}"
```

11. 运行以下命令来删除角色：

```
$ aws iam delete-role --role-name "${ROLE_NAME}"
```

4.9. OADP 和 AWS STS

4.9.1. 使用 OADP 在 AWS STS 上备份应用程序

您可以通过安装 OADP Operator，使用 Amazon Web Services (AWS) 安装 OpenShift API for Data Protection (OADP)。Operator 会安装 Velero 1.12。

注意

从 OADP 1.0.4 开始，所有 OADP 1.0.*z* 版本都只能用作 MTC Operator 的依赖项，且不适用于独立 Operator。

您可以为 Velero 配置 AWS，创建一个默认 **Secret**，然后安装数据保护应用程序。如需了解更多详细信息，请参阅安装 OADP Operator。

要在受限网络环境中安装 OADP Operator，您必须首先禁用默认的 OperatorHub 源并镜像 Operator 目录。详情请参阅 在受限网络中使用 Operator Lifecycle Manager。

您可以手动在 AWS 安全令牌服务 (AWS STS) 集群上安装 OADP。Amazon AWS 将 AWS STS 作为 Web 服务提供，可让您为用户请求临时的、带有有限权限的凭证。您可以使用 STS 通过 API 调用、AWS 控制台或 AWS 命令行界面(CLI)为可信用户提供临时访问资源。

在为数据保护(OADP)安装 OpenShift API 前，您必须为 OADP 设置角色和策略凭证，以便它可以使用 Amazon Web Services API。

这个过程在以下两个阶段执行：

1. 准备 AWS 凭证。
2. 安装 OADP Operator，并为它提供一个 IAM 角色。

4.9.1.1. 为 OADP 准备 AWS STS 凭证

必须准备 Amazon Web Services 帐户，并配置为接受 OpenShift API for Data Protection (OADP) 安装。使用以下步骤准备 AWS 凭证。

流程

1. 运行以下命令来定义 **cluster_name** 环境变量：

```
$ export CLUSTER_NAME= <AWS_cluster_name> 1
```

1 变量可以设置为任何值。

2. 运行以下命令，获取**cluster** 的详情，如 **AWS_ACCOUNT_ID, OIDC_ENDPOINT** ：

```
$ export CLUSTER_VERSION=$(oc get clusterversion version -o
jsonpath='{.status.desired.version}{"\n"}')

export AWS_CLUSTER_ID=$(oc get clusterversion version -o jsonpath='{.spec.clusterID}
{"\n"}')

export OIDC_ENDPOINT=$(oc get authentication.config.openshift.io cluster -o
jsonpath='{.spec.serviceAccountIssuer}' | sed 's|^https://||')

export REGION=$(oc get infrastructures cluster -o
jsonpath='{.status.platformStatus.aws.region}' --allow-missing-template-keys=false || echo
us-east-2)

export AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)

export ROLE_NAME="${CLUSTER_NAME}-openshift-oadp-aws-cloud-credentials"
```

3. 运行以下命令，创建一个临时目录来存储所有文件：

```
$ export SCRATCH="/tmp/${CLUSTER_NAME}/oadp"
mkdir -p ${SCRATCH}
```

4. 运行以下命令显示所有收集的详细信息：

```
$ echo "Cluster ID: ${AWS_CLUSTER_ID}, Region: ${REGION}, OIDC Endpoint:
${OIDC_ENDPOINT}, AWS Account ID: ${AWS_ACCOUNT_ID}"
```

5. 在 AWS 帐户中，创建一个 IAM 策略以允许访问 AWS S3：

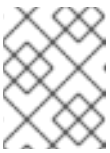
   a. 运行以下命令，检查策略是否存在：

```
$ export POLICY_NAME="OadpVer1" (1)
```

   (1) 变量可以设置为任何值。

```
$ POLICY_ARN=$(aws iam list-policies --query "Policies[?
PolicyName=='$POLICY_NAME'].{ARN:Arn}" --output text)
```

   b. 输入以下命令来创建策略 JSON 文件，然后创建策略：



注意

如果没有找到策略 ARN，命令会创建策略。如果策略 ARN 已存在，则 **if** 语句会有意跳过策略创建。

```
$ if [[ -z "${POLICY_ARN}" ]]; then
cat << EOF > ${SCRATCH}/policy.json
{
"Version": "2012-10-17",
"Statement": [
```

```
 {
  "Effect": "Allow",
  "Action": [
   "s3:CreateBucket",
   "s3:DeleteBucket",
   "s3:PutBucketTagging",
   "s3:GetBucketTagging",
   "s3:PutEncryptionConfiguration",
   "s3:GetEncryptionConfiguration",
   "s3:PutLifecycleConfiguration",
   "s3:GetLifecycleConfiguration",
   "s3:GetBucketLocation",
   "s3:ListBucket",
   "s3:GetObject",
   "s3:PutObject",
   "s3:DeleteObject",
   "s3:ListBucketMultipartUploads",
   "s3:AbortMultipartUpload",
   "s3:ListMultipartUploadParts",
   "ec2:DescribeSnapshots",
   "ec2:DescribeVolumes",
   "ec2:DescribeVolumeAttribute",
   "ec2:DescribeVolumesModifications",
   "ec2:DescribeVolumeStatus",
   "ec2:CreateTags",
   "ec2:CreateVolume",
   "ec2:CreateSnapshot",
   "ec2:DeleteSnapshot"
  ],
  "Resource": "*"
 }
]}
EOF

POLICY_ARN=$(aws iam create-policy --policy-name $POLICY_NAME \
--policy-document file:///${SCRATCH}/policy.json --query Policy.Arn \
--tags Key=openshift_version,Value=${CLUSTER_VERSION}
Key=operator_namespace,Value=openshift-adp Key=operator_name,Value=oadp \
--output text) 1
fi
```

1 **SCRATCH** 是为存储文件创建的临时目录的名称。

c. 运行以下命令来查看策略 ARN：

```
$ echo ${POLICY_ARN}
```

6. 为集群创建 IAM 角色信任策略：

a. 运行以下命令来创建信任策略文件：

```
$ cat <<EOF > ${SCRATCH}/trust-policy.json
{
   "Version": "2012-10-17",
   "Statement": [{
```

```
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-
provider/${OIDC_ENDPOINT}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_ENDPOINT}:sub": [
            "system:serviceaccount:openshift-adp:openshift-adp-controller-manager",
            "system:serviceaccount:openshift-adp:velero"]
        }
      }
    }]
}
EOF
```

b. 运行以下命令，为集群创建 IAM 角色信任策略：

```
$ ROLE_ARN=$(aws iam create-role --role-name \
  "${ROLE_NAME}" \
  --assume-role-policy-document file://${SCRATCH}/trust-policy.json \
  --tags Key=cluster_id,Value=${AWS_CLUSTER_ID}
Key=openshift_version,Value=${CLUSTER_VERSION}
Key=operator_namespace,Value=openshift-adp Key=operator_name,Value=oadp --
query Role.Arn --output text)
```

c. 运行以下命令来查看角色 ARN：

```
$ echo ${ROLE_ARN}
```

7. 运行以下命令，将 IAM 策略附加到 IAM 角色：

```
$ aws iam attach-role-policy --role-name "${ROLE_NAME}" --policy-arn ${POLICY_ARN}
```

4.9.1.1.1. 设置 Velero CPU 和内存分配

您可以通过编辑 **DataProtectionApplication** 自定义资源(CR)清单来为 **Velero** pod 设置 CPU 和内存分配。

先决条件

- 您必须安装了 OpenShift API for Data Protection(OADP)Operator。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.podConfig.ResourceAllocations** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
```

```
# ...
 configuration:
   velero:
     podConfig:
       nodeSelector: <node selector> 1
       resourceAllocations: 2
         limits:
           cpu: "1"
           memory: 1024Mi
         requests:
           cpu: 200m
           memory: 256Mi
```

1 指定要提供给 Velero podSpec 的节点选择器。

2 列出的 **resourceAllocations** 用于平均使用。

注意

Kopia 是 OADP 1.3 及之后的版本中的一个选项。您可以使用 Kopia 进行文件系统备份，Kopia 是 Data Mover 的唯一选择，并带有内置数据 Mover。

和 Restic 相比，Kopia 需要更多资源，您可能需要相应地调整 CPU 和内存要求。

4.9.1.2. 安装 OADP Operator 并提供 IAM 角色

AWS 安全令牌服务 (AWS STS) 是一个全局 Web 服务，它为 IAM 或联邦用户提供简短凭证。本文档论述了如何在 AWS STS 集群中手动安装 OpenShift API for Data Protection (OADP)。

重要

OADP AWS STS 环境中不支持 Restic 和 Kopia。验证 Restic 和 Kopia 节点代理是否已禁用。对于备份卷，AWS STS 上的 OADP 仅支持原生快照和 Container Storage Interface (CSI) 快照。

在使用 STS 验证的 AWS 集群中，不支持在不同的 AWS 区域中恢复备份数据。

AWS STS 集群目前不支持 Data Mover 功能。您可以使用原生 AWS S3 工具移动数据。

先决条件

- 具有所需访问权限和令牌的 OpenShift Container Platform AWS STS 集群。具体步骤请查看 *为 OADP 准备 AWS 凭证*。如果您计划使用两个不同的集群来备份和恢复，您必须为每个集群准备 AWS 凭证，包括 **ROLE_ARN**。

流程

1. 输入以下命令，从 AWS 令牌文件创建 OpenShift Container Platform secret：

   a. 创建凭证文件：

```
$ cat <<EOF > ${SCRATCH}/credentials
  [default]
  role_arn = ${ROLE_ARN}
```

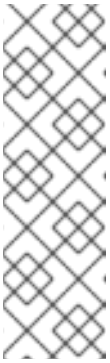
```
  web_identity_token_file = /var/run/secrets/openshift/serviceaccount/token
EOF
```

b. 为 OADP 创建命名空间：

```
$ oc create namespace openshift-adp
```

c. 创建 OpenShift Container Platform secret：

```
$ oc -n openshift-adp create secret generic cloud-credentials \
  --from-file=${SCRATCH}/credentials
```



注意

在 OpenShift Container Platform 版本 4.14 及更新的版本中，OADP Operator 通过 Operator Lifecycle Manager (OLM) 和 Cloud Credentials Operator (CCO)支持新的标准化 STS 工作流。在此工作流中，您不需要创建上述 secret，您需要在使用 OpenShift Container Platform Web 控制台安装 OLM 管理的 Operator 时提供角色 ARN，请参阅使用 *Web 控制台从 OperatorHub 安装*。

前面的 secret 由 CCO 自动创建。

2. 安装 OADP Operator：

a. 在 OpenShift Container Platform Web 控制台中，浏览 **Operators → OperatorHub**。

b. 搜索 **OADP Operator**。

c. 在 **role_ARN** 字段中，粘贴之前创建的 role_arn，再点 **Install**。

3. 输入以下命令，使用 AWS 凭证创建 AWS 云存储：

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: CloudStorage
  metadata:
    name: ${CLUSTER_NAME}-oadp
    namespace: openshift-adp
  spec:
    creationSecret:
      key: credentials
      name: cloud-credentials
    enableSharedConfig: true
    name: ${CLUSTER_NAME}-oadp
    provider: aws
    region: $REGION
EOF
```

4. 输入以下命令检查应用程序的存储默认存储类：

```
$ oc get pvc -n <namespace>
```

输**出示例**

```
NAME    STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS   AGE
applog   Bound    pvc-351791ae-b6ab-4e8b-88a4-30f73caf5ef8   1Gi        RWO            gp3-
csi        4d19h
mysql    Bound    pvc-16b8e009-a20a-4379-accc-bc81fedd0621   1Gi        RWO            gp3-
csi        4d19h
```

5. 运行以下命令来获取存储类：

```
$ oc get storageclass
```

输**出示例**

```
NAME              PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
gp2               kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   true
4d21h
gp2-csi           ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3               ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3-csi (default)   ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
```

注意

以下存储类可以正常工作：

- gp3-csi
- gp2-csi
- gp3
- gp2

如果要备份的应用程序或应用程序都使用带有 Container Storage Interface (CSI) 的持久性卷 (PV)，建议在 OADP DPA 配置中包含 CSI 插件。

6. 创建 **DataProtectionApplication** 资源，以配置存储备份和卷快照的存储的连接：

   a. 如果您只使用 CSI 卷，请输入以下命令部署数据保护应用程序：

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true (1)
    features:
      dataMover:
        enable: false
```

```
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
      velero:
        defaultPlugins:
        - openshift
        - aws
        - csi
      restic:
        enable: false
EOF
```

1 如果您不想使用镜像备份，请将此字段设置为 **false**。

a. 如果使用 CSI 或非 CSI 卷，请输入以下命令来部署数据保护应用程序：

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true 1
    features:
      dataMover:
        enable: false
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
      velero:
        defaultPlugins:
        - openshift
        - aws
      nodeAgent: 2
        enable: false
        uploaderType: restic
```

```
    snapshotLocations:
      - velero:
          config:
            credentialsFile: /tmp/credentials/openshift-adp/cloud-credentials-credentials 3
            enableSharedConfig: "true" 4
            profile: default 5
            region: ${REGION} 6
          provider: aws
EOF
```

1. 如果您不想使用镜像备份，请将此字段设置为 **false**。
2. 请参阅有关 **nodeAgent** 属性的重要备注。
3. **credentialsFile** 字段是 pod 上存储桶凭证的挂载位置。
4. **enableSharedConfig** 字段允许 **snapshotLocations** 共享或重复使用为存储桶定义的凭证。
5. 使用 AWS 凭证文件中设置的配置集名称。
6. 将 **region** 指定为您的 AWS 区域。这必须与集群区域相同。

现在，您可以备份和恢复 OpenShift Container Platform 应用程序，如 *备份应用程序* 中所述。

重要

如果使用 OADP 1.2，请替换此配置：

```
nodeAgent:
  enable: false
  uploaderType: restic
```

使用以下配置：

```
restic:
  enable: false
```

如果要使用两个不同的集群来备份和恢复，则两个集群必须在云存储 CR 和 OADP **DataProtectionApplication** 配置中具有相同的 AWS S3 存储名称。

其他资源

- 使用 Web 控制台从 OperatorHub 安装
- 备份应用程序

4.9.1.3. 备份 OADP AWS STS 上的工作负载，可以使用可选的清理

4.9.1.3.1. 使用 OADP 和 AWS STS 执行备份

以下示例 **hello-world** 应用没有附加持久性卷 (PV)。使用 OpenShift API 对 Amazon Web Services (AWS STS) 的数据保护 (OADP) 执行备份。

数据保护应用程序 (DPA) 配置都将正常工作。

1. 运行以下命令，创建一个工作负载来备份：

```
$ oc create namespace hello-world
```

```
$ oc new-app -n hello-world --image=docker.io/openshift/hello-openshift
```

2. 运行以下命令来公开路由：

```
$ oc expose service/hello-openshift -n hello-world
```

3. 运行以下命令检查应用程序是否正常工作：

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

输出示例

```
Hello OpenShift!
```

4. 运行以下命令来备份工作负载：

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Backup
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    includedNamespaces:
    - hello-world
    storageLocation: ${CLUSTER_NAME}-dpa-1
    ttl: 720h0m0s
EOF
```

5. 等待备份完成，然后运行以下命令：

```
$ watch "oc -n openshift-adp get backup hello-world -o json | jq .status"
```

输出示例

```
{
  "completionTimestamp": "2022-09-07T22:20:44Z",
  "expiration": "2022-10-07T22:20:22Z",
  "formatVersion": "1.1.0",
  "phase": "Completed",
  "progress": {
    "itemsBackedUp": 58,
    "totalItems": 58
  },
```

```
  "startTimestamp": "2022-09-07T22:20:22Z",
  "version": 1
}
```

6. 运行以下命令来删除 demo 工作负载：

```
$ oc delete ns hello-world
```

7. 运行以下命令，从备份中恢复工作负载：

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Restore
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    backupName: hello-world
EOF
```

8. 运行以下命令等待 Restore 完成：

```
$ watch "oc -n openshift-adp get restore hello-world -o json | jq .status"
```

输**出示例**

```
{
  "completionTimestamp": "2022-09-07T22:25:47Z",
  "phase": "Completed",
  "progress": {
    "itemsRestored": 38,
    "totalItems": 38
  },
  "startTimestamp": "2022-09-07T22:25:28Z",
  "warnings": 9
}
```

9. 运行以下命令检查工作负载是否已恢复：

```
$ oc -n hello-world get pods
```

输**出示例**

```
NAME                          READY   STATUS    RESTARTS   AGE
hello-openshift-9f885f7c6-kdjpj   1/1     Running   0          90s
```

10. 运行以下命令来检查 JSONPath：

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

输**出示例**

```
Hello OpenShift!
```

注意

有关故障排除提示的信息，请参阅 OADP 团队的 故障排除文档。

4.9.1.3.2. 在使用 OADP 和 AWS STS 备份后清理集群

如果您需要卸载 OpenShift API for Data Protection (OADP) Operator 以及本例中的备份和 S3 存储桶，请按照以下步骤操作。

流程

1. 运行以下命令来删除工作负载：

```
$ oc delete ns hello-world
```

2. 运行以下命令来删除数据保护应用程序 (DPA)：

```
$ oc -n openshift-adp delete dpa ${CLUSTER_NAME}-dpa
```

3. 运行以下命令来删除云存储：

```
$ oc -n openshift-adp delete cloudstorage ${CLUSTER_NAME}-oadp
```

重要

如果这个命令挂起，您可能需要通过运行以下命令来删除终结器：

```
$ oc -n openshift-adp patch cloudstorage ${CLUSTER_NAME}-oadp -p
'{"metadata":{"finalizers":null}}' --type=merge
```

4. 如果不再需要 Operator，请运行以下命令删除它：

```
$ oc -n openshift-adp delete subscription oadp-operator
```

5. 运行以下命令，从 Operator 中删除命名空间：

```
$ oc delete ns openshift-adp
```

6. 如果不再需要备份和恢复资源，请运行以下命令从集群中删除它们：

```
$ oc delete backup hello-world
```

7. 要删除 AWS S3 中的备份、恢复和远程对象，请运行以下命令：

```
$ velero backup delete hello-world
```

8. 如果您不再需要自定义资源定义 (CRD)，请运行以下命令从集群中删除它们：

```
$ for CRD in `oc get crds | grep velero | awk '{print $1}'`; do oc delete crd $CRD; done
```

9. 运行以下命令来删除 AWS S3 存储桶：

```
$ aws s3 rm s3://${CLUSTER_NAME}-oadp --recursive
```

```
$ aws s3api delete-bucket --bucket ${CLUSTER_NAME}-oadp
```

10. 运行以下命令，将策略从角色分离：

```
$ aws iam detach-role-policy --role-name "${ROLE_NAME}"  --policy-arn "${POLICY_ARN}"
```

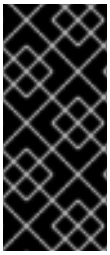
11. 运行以下命令来删除角色：

```
$ aws iam delete-role --role-name "${ROLE_NAME}"
```

4.10. OADP 1.2 DATA MOVER

4.10.1. OADP Data Mover 介绍

OADP Data Mover 允许您在故障、意外删除或集群崩溃时从存储中恢复有状态的应用程序。

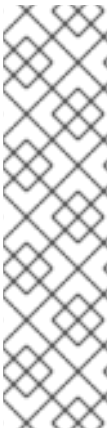


重要

OADP 1.2 Data Mover 是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

- 您可以使用 OADP Data Mover 将 Container Storage Interface (CSI) 卷快照备份到远程对象存储。对于 CSI 快照，请参阅使用数据 Mover。
- 您可以使用 OADP 1.2 Data Mover，为使用 CephFS、CephRBD 或这两者的集群备份和恢复应用程序数据。请参阅使用 OADP 1.2 数据与 Ceph 存储 。



注意

迁移后 hook 可能无法与 OADP 1.3 Data Mover 正常工作。

OADP 1.2 Data Movers 使用同步进程来备份和恢复应用程序数据。由于进程是同步的，因此用户只能在相关 pod 的持久性卷(PV)由 Data Mover 的持久性卷声明(PVC)发布后执行。

但是，OADP 1.3 Data Movers 使用异步过程。因此，在 Data Mover 的 PVC 发布相关的 PV 前，可能会调用后一个 post-restore hook。如果发生这种情况，pod 会一直处于 **Pending** 状态，且无法运行 hook。hook 尝试可能会在 pod 发布前超时，从而导致 **PartiallyFailed** 恢复操作。

4.10.1.1. OADP Data Mover 先决条件

- 您有一个有状态应用程序在单独的命名空间中运行。
- 已使用 Operator Lifecycle Manager (OLM) 安装 OADP Operator。
- 您已创建了适当的 **VolumeSnapshotClass** 和 **StorageClass**。
- 已使用 OLM 安装 VolSync operator。

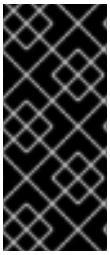
4.10.2. 对 CSI 快照使用 Data Mover

OADP Data Mover 可让客户将 Container Storage Interface (CSI) 卷快照备份到远程对象存储。启用 Data Mover 时，如果出现故障、意外删除或集群崩溃，您可以使用从对象存储中提取的 CSI 卷快照来恢复有状态的应用程序。

Data Mover 解决方案使用 VolSync 的 Restic 选项。

数据 Mover 支持 CSI 卷快照的备份和恢复。

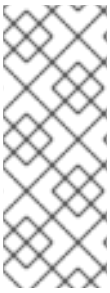
在 OADP 1.2 Data Mover 中，**VolumeSnapshotBackups** (VSBs) 和 **VolumeSnapshotRestores** (VSR) 使用 VolumeSnapshotMover (VSM) 排队。通过指定 VSB 和 VSR 同时处于 **InProgress** 的并发数量，可以提高 VSM 的性能。在所有异步插件操作都完成后，备份将标记为完成。



重要

OADP 1.2 Data Mover 是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。



注意

红帽建议使用 OADP 1.2 Data Mover 来备份和恢复 ODF CephFS 卷，升级或安装 OpenShift Container Platform 版本 4.12 或更高版本以提高性能。OADP Data Mover 可以利用 OpenShift Container Platform 版本 4.12 或更高版本中的 CephFS shouldow 卷，基于我们的测试，这可以提高备份时间的性能。

- CephFS ROX 详情

先决条件

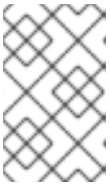
- 已确认 **StorageClass** 和 **VolumeSnapshotClass** 自定义资源 (CR) 支持 CSI。
- 您已确认只有一个 **VolumeSnapshotClass** CR 具有注解 **snapshot.storage.kubernetes.io/is-default-class: "true"**。

注意

在 OpenShift Container Platform 版本 4.12 或更高版本中，验证这是唯一的默认 **VolumeSnapshotClass**。

- 您已确认 **VolumeSnapshotClass** CR 的 **deletionPolicy** 被设置为 **Retain**。
- 您已确认只有一个 **StorageClass** CR 具有注解 **storageclass.kubernetes.io/is-default-class: "true"**。

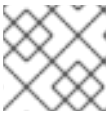
- 您已在 **VolumeSnapshotClass** CR 中包含标签 **velero.io/csi-volumesnapshot-class: "true"**。
- 您已确认 **OADP 命名空间具有注解 oc annotate --overwrite namespace/openshift-adp volsync.backube/privileged-movers="true"**。



注意

在 OADP 1.2 中，多数情况下不需要 **privileged-movers** 设置。恢复容器权限应该足以满足 Volsync 副本。在某些用户场景中，可能会有权限错误，**privileged-mover= true** 设置应被解析。

- 已使用 Operator Lifecycle Manager (OLM) 安装 VolSync Operator。



注意

使用 OADP Data Mover 需要 VolSync Operator。

- 已使用 OLM 安装 OADP operator。

流程

1. 通过创建一个 **.yaml** 文件来配置 Restic secret，如下所示：

```
apiVersion: v1
kind: Secret
metadata:
  name: <secret_name>
  namespace: openshift-adp
type: Opaque
stringData:
  RESTIC_PASSWORD: <secure_restic_password>
```

注意

默认情况下，Operator 会查找名为 **dm-credential** 的 secret。如果您使用其他名称，您需要使用 **dpa.spec.features.dataMover.credentialName** 通过 Data Protection Application (DPA) CR 指定名称。

2. 创建类似以下示例的 DPA CR。默认插件包括 CSI。

数据保护应用程序 (DPA) CR 示例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  backupLocations:
    - velero:
        config:
          profile: default
          region: us-east-1
```

```
      credential:
        key: cloud
        name: cloud-credentials
      default: true
      objectStorage:
        bucket: <bucket_name>
        prefix: <bucket-prefix>
      provider: aws
configuration:
  restic:
    enable: <true_or_false>
  velero:
    itemOperationSyncFrequency: "10s"
    defaultPlugins:
    - openshift
    - aws
    - csi
    - vsm
features:
  dataMover:
    credentialName: restic-secret
    enable: true
    maxConcurrentBackupVolumes: "3" 1
    maxConcurrentRestoreVolumes: "3" 2
    pruneInterval: "14" 3
    volumeOptions: 4
    sourceVolumeOptions:
      accessMode: ReadOnlyMany
      cacheAccessMode: ReadWriteOnce
      cacheCapacity: 2Gi
    destinationVolumeOptions:
      storageClass: other-storageclass-name
      cacheAccessMode: ReadWriteMany
snapshotLocations:
  - velero:
      config:
        profile: default
        region: us-west-2
      provider: aws
```

1. 可选：指定允许排队进行备份的快照数量的上限。默认值为 **10**。
2. 可选：指定允许排队进行恢复的快照数量的上限。默认值为 **10**。
3. 可选：指定存储库上运行的 Restic 修剪之间的天数。修剪操作会重新打包数据以释放可用的空间，但在操作过程中也会生成大量的 I/O 流量。设置此选项可在存储性能和使用成本间进行权衡。
4. 可选：指定用于备份和恢复的 VolumeSync 卷选项。

OADP Operator 安装两个自定义资源定义 (CRD)、**VolumeSnapshotBackup** 和 **VolumeSnapshotRestore**。

VolumeSnapshotBackup CRD 示例

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotBackup
metadata:
  name: <vsb_name>
  namespace: <namespace_name> 1
spec:
  volumeSnapshotContent:
    name: <snapcontent_name>
  protectedNamespace: <adp_namespace> 2
  resticSecretRef:
    name: <restic_secret_name>
```

1 指定卷快照所在的命名空间。

2 指定安装 OADP Operator 的命名空间。默认值为 **openshift-adp**。

VolumeSnapshotRestore CRD 示例

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotRestore
metadata:
  name: <vsr_name>
  namespace: <namespace_name> 1
spec:
  protectedNamespace: <protected_ns> 2
  resticSecretRef:
    name: <restic_secret_name>
  volumeSnapshotMoverBackupRef:
    sourcePVCData:
      name: <source_pvc_name>
      size: <source_pvc_size>
    resticrepository: <your_restic_repo>
    volumeSnapshotClassName: <vsclass_name>
```

1 指定卷快照所在的命名空间。

2 指定安装 OADP Operator 的命名空间。默认值为 **openshift-adp**。

3. 您可以执行以下步骤备份卷快照：

   a. 创建备份 CR：

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup_name>
  namespace: <protected_ns> 1
spec:
  includedNamespaces:
  - <app_ns> 2
  storageLocation: velero-sample-1
```

1. 指定安装 Operator 的命名空间。默认命名空间是 **openshift-adp**。
2. 指定要备份的应用程序命名空间。

b. 等待 10 分钟，并输入以下命令来检查 **VolumeSnapshotBackup** CR 状态是否为 **Completed**：

```
$ oc get vsb -n <app_ns>
```

```
$ oc get vsb <vsb_name> -n <app_ns> -o jsonpath="{.status.phase}"
```

在对象存储中创建快照是在 DPA 中配置。

注意

如果 **VolumeSnapshotBackup** CR 的状态变为 **Failed**，请参阅 Velero 日志进行故障排除。

4. 您可以执行以下步骤来恢复卷快照：

a. 删除由 Velero CSI 插件创建的 application 命名空间和 **VolumeSnapshotContent**。

b. 创建 **Restore** CR，并将 **restorePV** 设置为 **true**。

Restore CR 示例

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore_name>
  namespace: <protected_ns>
spec:
  backupName: <previous_backup_name>
  restorePVs: true
```

c. 等待 10 分钟，并通过输入以下命令来检查 **VolumeSnapshotRestore** CR 状态是否为 **Completed** ：

```
$ oc get vsr -n <app_ns>
```

```
$ oc get vsr <vsr_name> -n <app_ns> -o jsonpath="{.status.phase}"
```

d. 检查您的应用程序数据和资源是否已恢复。

注意

如果 **VolumeSnapshotRestore** CR 的状态变成 'Failed'，请参阅 Velero 日志进行故障排除。

4.10.3. 使用带有 Ceph 存储的 OADP 1.2 Data Mover

您可以使用 OADP 1.2 Data Mover，为使用 CephFS、CephRBD 或这两者的集群备份和恢复应用程序数据。

OADP 1.2 Data Mover 会利用支持大规模环境的 Ceph 功能。其中之一是 shallow copy 方法，可用于 OpenShift Container Platform 4.12 及更新的版本。此功能支持备份和恢复源持久性卷声明 (PVC) 上找到的 **StorageClass** 和 **AccessMode** 资源。

重要

CephFS 浅复制功能是一个备份功能。它不是恢复操作的一部分。

4.10.3.1. 在 Ceph 存储中使用 OADP 1.2 Data Mover 的先决条件

以下先决条件适用于在使用 Ceph 存储的集群中通过 OpenShift API for Data Protection (OADP) 1.2 Data Mover 进行数据备份和恢复的操作：

- 已安装 OpenShift Container Platform 4.12 或更高版本。
- 已安装 OADP Operator。
- 您已在命名空间 **openshift-adp** 中创建了 secret **cloud-credentials**。
- 已安装 Red Hat OpenShift Data Foundation。
- 已使用 Operator Lifecycle Manager 安装最新的 VolSync Operator。

4.10.3.2. 定义用于 OADP 1.2 Data Mover 的自定义资源

安装 Red Hat OpenShift Data Foundation 时，它会自动创建默认的 CephFS 和 CephRBD **StorageClass** 和 **VolumeSnapshotClass** 自定义资源 (CR)。您必须定义这些 CR 以用于 OpenShift API for Data Protection (OADP) 1.2 Data Mover。

定义 CR 后，您必须对环境进行一些其他更改，然后才能执行备份和恢复操作。

4.10.3.2.1. 定义 CephFS 自定义资源以用于 OADP 1.2 Data Mover

安装 Red Hat OpenShift Data Foundation 时，它会自动创建默认的 CephFS **StorageClass** 自定义资源 (CR) 和默认的 CephFS **VolumeSnapshotClass** CR。您可以定义这些 CR 以用于 OpenShift API for Data Protection (OADP) 1.2 Data Mover。

流程

1. 定义 **VolumeSnapshotClass** CR，如下例所示：

VolumeSnapshotClass CR 示例

```
apiVersion: snapshot.storage.k8s.io/v1
deletionPolicy: <deletion_policy_type> 1
driver: openshift-storage.cephfs.csi.ceph.com
kind: VolumeSnapshotClass
metadata:
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: true 2
  labels:
    velero.io/csi-volumesnapshot-class: true 3
```

```
    name: ocs-storagecluster-cephfsplugin-snapclass
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/snapshotter-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
```

1. OADP 支持 CSI 和 Data Mover 备份和恢复的 **Retain** 和 **Delete** 删除策略类型。对于 OADP 1.2 Data Mover，将删除策略类型设置为 **Retain**。
2. 必须设置为 **true**。
3. 必须设置为 **true**。

2. 定义 **StorageClass** CR，如下例所示：

StorageClass CR 示例

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ocs-storagecluster-cephfs
  annotations:
    description: Provides RWO and RWX Filesystem volumes
    storageclass.kubernetes.io/is-default-class: true (1)
provisioner: openshift-storage.cephfs.csi.ceph.com
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-cephfs-node
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  fsName: ocs-storagecluster-cephfilesystem
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

1. 必须设置为 **true**。

4.10.3.2.2. 定义 CephRBD 自定义资源以用于 OADP 1.2 Data Mover

安装 Red Hat OpenShift Data Foundation 时，它会自动创建默认的 CephRBD **StorageClass** 自定义资源 (CR) 和默认的 CephRBD **VolumeSnapshotClass** CR。您可以定义这些 CR 以用于 OpenShift API for Data Protection (OADP) 1.2 Data Mover。

流程

1. 定义 **VolumeSnapshotClass** CR，如下例所示：

VolumeSnapshotClass CR 示例

```
apiVersion: snapshot.storage.k8s.io/v1
```

```
deletionPolicy: <deletion_policy_type> 1
driver: openshift-storage.rbd.csi.ceph.com
kind: VolumeSnapshotClass
metadata:
  labels:
    velero.io/csi-volumesnapshot-class: true 2
  name: ocs-storagecluster-rbdplugin-snapclass
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/snapshotter-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
```

1. OADP 支持 CSI 和 Data Mover 备份和恢复的 **Retain** 和 **Delete** 删除策略类型。对于 OADP 1.2 Data Mover，将删除策略类型设置为 **Retain**。

2. 必须设置为 **true**。

2. 定义 **StorageClass** CR，如下例所示：

StorageClass CR 示例

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ocs-storagecluster-ceph-rbd
  annotations:
    description: 'Provides RWO Filesystem volumes, and RWO and RWX Block volumes'
provisioner: openshift-storage.rbd.csi.ceph.com
parameters:
  csi.storage.k8s.io/fstype: ext4
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-rbd-node
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-rbd-provisioner
  imageFormat: '2'
  clusterID: openshift-storage
  imageFeatures: layering
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  pool: ocs-storagecluster-cephblockpool
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

4.10.3.2.3. 定义用于 OADP 1.2 Data Mover 的额外自定义资源

在重新定义默认 **StorageClass** 和 CephRBD **VolumeSnapshotClass** 自定义资源 (CR) 后，您必须创建以下 CR：

- 定义为使用 shallow 复制功能的 CephFS **StorageClass** CR
- Restic **Secret** CR

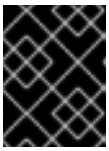
流程

1. 创建 CephFS **StorageClass** CR，并将 **backingSnapshot** 参数设置为 **true**，如下例所示：

将 backingSnapshot 设置为 true 的 CephFS StorageClass CR 示例

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ocs-storagecluster-cephfs-shallow
  annotations:
    description: Provides RWO and RWX Filesystem volumes
    storageclass.kubernetes.io/is-default-class: false
provisioner: openshift-storage.cephfs.csi.ceph.com
parameters:
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-cephfs-node
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-cephfs-provisioner
  clusterID: openshift-storage
  fsName: ocs-storagecluster-cephfilesystem
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  backingSnapshot: true (1)
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

(1) 必须设置为 **true**。



重要

确保 CephFS **VolumeSnapshotClass** 和 **StorageClass** CR 对 **provisioner** 有相同的值。

2. 配置 Restic **Secret** CR，如下例所示：

Restic Secret CR 示例

```
apiVersion: v1
kind: Secret
metadata:
  name: <secret_name>
  namespace: <namespace>
type: Opaque
stringData:
  RESTIC_PASSWORD: <restic_password>
```

4.10.3.3. 使用 OADP 1.2 Data Mover 和 CephFS 存储备份和恢复数据

您可以通过启用 CephFS 的 shallow copy 功能，使用 OpenShift API for Data Protection (OADP) 1.2 Data Mover 来备份和恢复使用 CephFS 存储的数据。

先决条件

- 有状态应用在单独的命名空间中运行，并将 CephFS 用作置备程序的持久性卷声明 (PVC)。
- 为 CephFS 和 OADP 1.2 Data Mover 定义 **StorageClass** 和 **VolumeSnapshotClass** 自定义资源 (CR)。
- **openshift-adp** 命名空间中有一个 secret **cloud-credentials**。

4.10.3.3.1. 创建用于 CephFS 存储的 DPA

在通过 OpenShift API for Data Protection (OADP) 1.2 Data Mover 来备份和存储使用 CephFS 存储的数据时，需要先创建一个 Data Protection Application (DPA) CR。

流程

1. 对于 OADP 1.2 Data Mover，您必须运行以下命令来验证 **VolumeSnapshotClass** CR 的 **deletionPolicy** 字段是否已设置为 **Retain** ：

```
$ oc get volumesnapshotclass -A  -o jsonpath='{range .items[*]}{"Name: "}{.metadata.name}{"  "}{"Retention Policy: "}{.deletionPolicy}{"\n"}{end}'
```

2. 运行以下命令，验证 **VolumeSnapshotClass** CR 的标签是否已设置为 **true** ：

```
$ oc get volumesnapshotclass -A  -o jsonpath='{range .items[*]}{"Name: "}{.metadata.name}{"  "}{"labels: "}{.metadata.labels}{"\n"}{end}'
```

3. 运行以下命令，验证 **StorageClass** CR 的 **storageclass.kubernetes.io/is-default-class** 注解是否已设置为 **true** ：

```
$ oc get storageClass -A  -o jsonpath='{range .items[*]}{"Name: "}{.metadata.name}{"  "}{"annotations: "}{.metadata.annotations}{"\n"}{end}'
```

4. 创建一个类似以下示例的 Data Protection Application (DPA) CR：

DPA CR 示例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  backupLocations:
    - velero:
        config:
          profile: default
          region: us-east-1
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: <my_bucket>
          prefix: velero
        provider: aws
```

```
configuration:
  restic:
    enable: false (1)
  velero:
    defaultPlugins:
      - openshift
      - aws
      - csi
      - vsm
features:
  dataMover:
    credentialName: <restic_secret_name> (2)
    enable: true (3)
    volumeOptionsForStorageClasses: (4)
      ocs-storagecluster-cephfs:
        sourceVolumeOptions:
          accessMode: ReadOnlyMany
          cacheAccessMode: ReadWriteMany
          cacheStorageClassName: ocs-storagecluster-cephfs
          storageClassName: ocs-storagecluster-cephfs-shallow
```

1. **enable** 字段没有默认值。有效值为 **true** 或者 **false**。
2. 使用您在准备环境时创建的 Restic **Secret**，以用于 OADP 1.2 Data Mover 和 Ceph。如果没有使用 Restic **Secret**，则 CR 会将默认值 **dm-credential** 用于此参数。
3. **enable** 字段没有默认值。有效值为 **true** 或者 **false**。
4. 可选参数。您可以为每个 **storageClass** 卷定义不同的 **VolumeOptionsForStorageClass** 标签集合。此配置为具有不同供应商的卷提供备份。可选的 **VolumeOptionsForStorageClass** 参数通常与 CephFS 一起使用，但可用于任何存储类型。

4.10.3.3.2. 使用 OADP 1.2 Data Mover 和 CephFS 存储备份数据

您可以通过启用 CephFS 存储的 shallow copy 功能，使用 OpenShift API for Data Protection (OADP) 1.2 Data Mover 来备份使用 CephFS 存储的数据。

流程

1. 如以下示例所示，创建一个 **Backup** CR：

Backup CR 示例

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup_name>
  namespace: <protected_ns>
spec:
  includedNamespaces:
  - <app_ns>
  storageLocation: velero-sample-1
```

2. 通过完成以下步骤来监控 **VolumeSnapshotBackup** CR 的进度：

   a. 要检查所有 **VolumeSnapshotBackup** CR 的进度，请运行以下命令：

   ```
   $ oc get vsb -n <app_ns>
   ```

   b. 要检查特定 **VolumeSnapshotBackup** CR 的进度，请运行以下命令：

   ```
   $ oc get vsb <vsb_name> -n <app_ns> -ojsonpath="{.status.phase}`
   ```

3. 等待几分钟，直到 **VolumeSnapshotBackup** CR 的状态为 **Completed**。

4. 验证 Restic **Secret** 中至少有一个快照在对象存储中提供。您可以在带有前缀 **/<OADP_namespace>** 的目标 **BackupStorageLocation** 存储供应商中检查这个快照。

4.10.3.3.3. 使用 OADP 1.2 Data Mover 和 CephFS 存储恢复数据

如果备份过程启用了 CephFS 存储的 shallow copy 功能，您可以使用 OpenShift API for Data Protection (OADP) 1.2 Data Mover 来恢复使用 CephFS 存储的数据。修剪复制功能没有在恢复过程中使用。

流程

1. 运行以下命令来删除应用程序命名空间：

   ```
   $ oc delete vsb -n <app_namespace> --all
   ```

2. 运行以下命令，删除在备份过程中创建的 **VolumeSnapshotContent** CR：

   ```
   $ oc delete volumesnapshotcontent --all
   ```

3. 创建一个 **Restore** CR，如下例所示：

   Restore CR 示例

   ```
   apiVersion: velero.io/v1
   kind: Restore
   metadata:
     name: <restore_name>
     namespace: <protected_ns>
   spec:
     backupName: <previous_backup_name>
   ```

4. 通过执行以下操作来监控 **VolumeSnapshotRestore** CR 的进度：

   a. 要检查所有 **VolumeSnapshotRestore** CR 的进度，请运行以下命令：

   ```
   $ oc get vsr -n <app_ns>
   ```

   b. 要检查特定 **VolumeSnapshotRestore** CR 的进度，请运行以下命令：

   ```
   $ oc get vsr <vsr_name> -n <app_ns> -ojsonpath="{.status.phase}
   ```

5. 运行以下命令验证您的应用程序数据是否已恢复：

```
$ oc get route <route_name> -n <app_ns> -ojsonpath="{.spec.host}"
```

4.10.3.4. 使用 OADP 1.2 Data Mover 和 split 卷 (CephFS 和 Ceph RBD) 备份和恢复数据

您可以使用 OpenShift API for Data Protection (OADP) 1.2 Data Mover 在具有 *分割卷* 的环境中备份和恢复数据，即使用 CephFS 和 CephRBD 的环境。

先决条件

- 有状态应用在单独的命名空间中运行，并将 CephFS 用作置备程序的持久性卷声明 (PVC)。
- 为 CephFS 和 OADP 1.2 Data Mover 定义 **StorageClass** 和 **VolumeSnapshotClass** 自定义资源 (CR)。
- **openshift-adp** 命名空间中有一个 secret **cloud-credentials**。

4.10.3.4.1. 创建用于分割卷的 DPA

在使用 OpenShift API for Data Protection (OADP) 1.2 Data Mover 来使用分割卷来备份和恢复数据前，您必须创建一个数据保护应用程序 (DPA) CR。

流程

- 创建一个数据保护应用程序 (DPA) CR，如下例所示：

  带有分割卷的环境的 DPA CR 示例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  backupLocations:
    - velero:
        config:
          profile: default
          region: us-east-1
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: <my-bucket>
          prefix: velero
        provider: aws
  configuration:
    restic:
      enable: false
    velero:
      defaultPlugins:
        - openshift
        - aws
        - csi
```

```
      - vsm
  features:
    dataMover:
      credentialName: <restic_secret_name> 1
      enable: true
      volumeOptionsForStorageClasses: 2
        ocs-storagecluster-cephfs:
          sourceVolumeOptions:
            accessMode: ReadOnlyMany
            cacheAccessMode: ReadWriteMany
            cacheStorageClassName: ocs-storagecluster-cephfs
            storageClassName: ocs-storagecluster-cephfs-shallow
        ocs-storagecluster-ceph-rbd:
          sourceVolumeOptions:
            storageClassName: ocs-storagecluster-ceph-rbd
            cacheStorageClassName: ocs-storagecluster-ceph-rbd
          destinationVolumeOptions:
            storageClassName: ocs-storagecluster-ceph-rbd
            cacheStorageClassName: ocs-storagecluster-ceph-rbd
```

1. 使用您在准备环境时创建的 Restic **Secret**，以用于 OADP 1.2 Data Mover 和 Ceph。如果没有，则 CR 会将默认值 **dm-credential** 用于此参数。

2. 可以为每个 **storageClass** 卷定义不同的 **VolumeOptionsForStorageClass** 标签，从而允许备份具有不同供应商的卷。**VolumeOptionsForStorageClass** 参数用于 CephFS。但是，可选的 **VolumeOptionsForStorageClass** 参数可用于任何存储类型。

4.10.3.4.2. 使用 OADP 1.2 Data Mover 和 split 卷备份数据

您可以使用 OpenShift API 进行数据保护 (OADP) 1.2 Data Mover 在有分割卷的环境中备份数据。

流程

1. 如以下示例所示，创建一个 **Backup** CR：

   Backup CR 示例

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup_name>
  namespace: <protected_ns>
spec:
  includedNamespaces:
  - <app_ns>
  storageLocation: velero-sample-1
```

2. 通过完成以下步骤来监控 **VolumeSnapshotBackup** CR 的进度：

   a. 要检查所有 **VolumeSnapshotBackup** CR 的进度，请运行以下命令：

```
$ oc get vsb -n <app_ns>
```

b. 要检查特定 **VolumeSnapshotBackup** CR 的进度，请运行以下命令：

```
$ oc get vsb <vsb_name> -n <app_ns> -ojsonpath="{.status.phase}`
```

3. 等待几分钟，直到 **VolumeSnapshotBackup** CR 的状态为 **Completed**。

4. 验证 Restic **Secret** 中至少有一个快照在对象存储中提供。您可以在带有前缀 **/<OADP_namespace>** 的目标 **BackupStorageLocation** 存储供应商中检查这个快照。

4.10.3.4.3. 使用 OADP 1.2 Data Mover 和 split 卷恢复数据

如果备份过程启用了 CephFS 存储的 shallow copy 功能，您可以使用 OpenShift API 进行数据保护 (OADP) 1.2 Data Mover 在带有分割卷的环境中恢复数据。修剪复制功能没有在恢复过程中使用。

流程

1. 运行以下命令来删除应用程序命名空间：

```
$ oc delete vsb -n <app_namespace> --all
```

2. 运行以下命令，删除在备份过程中创建的 **VolumeSnapshotContent** CR：

```
$ oc delete volumesnapshotcontent --all
```

3. 创建一个 **Restore** CR，如下例所示：

Restore CR 示例

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore_name>
  namespace: <protected_ns>
spec:
  backupName: <previous_backup_name>
```

4. 通过执行以下操作来监控 **VolumeSnapshotRestore** CR 的进度：

a. 要检查所有 **VolumeSnapshotRestore** CR 的进度，请运行以下命令：

```
$ oc get vsr -n <app_ns>
```

b. 要检查特定 **VolumeSnapshotRestore** CR 的进度，请运行以下命令：

```
$ oc get vsr <vsr_name> -n <app_ns> -ojsonpath="{.status.phase}
```

5. 运行以下命令验证您的应用程序数据是否已恢复：

```
$ oc get route <route_name> -n <app_ns> -ojsonpath="{.spec.host}"
```

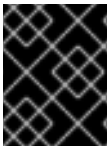
4.10.3.5. 删除 OADP 1.2 的策略

删除策略决定了从系统中删除数据的规则，指定根据保留周期、数据敏感度和合规要求等因素如何进行删除。它有效地管理数据删除，同时满足法规并保留宝贵的信息。

4.10.3.5.1. 删除 OADP 1.2 的策略指南

查看 OADP 1.2 的以下删除策略指南：

- 要使用 OADP 1.2.x Data Mover 来备份和恢复，请在 **VolumeSnapshotClass** 自定义资源(CR)中将 **deletionPolicy** 字段设置为 **Retain**。
- 在 OADP 1.2.x 中，要使用 CSI 备份和恢复，您可以在 **VolumeSnapshotClass** CR 中将 **deletionPolicy** 字段设置为 **Retain** 或 **Delete**。



重要

OADP 1.2.x Data Mover to backup 和 restore 是一个技术预览功能，它不被支持，没有支持例外。

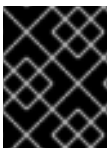
4.11. OADP 1.3 DATA MOVER

4.11.1. 关于 OADP 1.3 Data Mover

OADP 1.3 包含一个内置的 Data Mover，您可以使用它将 Container Storage Interface (CSI) 卷快照移到远程对象存储。如果发生故障、意外删除或损坏，内置的 Data Mover 可让您从远程对象存储中恢复有状态的应用程序。它使用 Kopia 作为上传程序机制来读取快照数据并写入统一存储库。

OADP 支持以下 CSI 快照：

- Red Hat OpenShift Data Foundation
- 使用支持 Kubernetes 卷快照 API 的 Container Storage Interface（CSI）驱动程序的任何其他云存储供应商



重要

OADP 内置 Data Mover（在 OADP 1.3 中作为技术预览引进）现在完全支持容器化和虚拟机工作负载。

4.11.1.1. 启用内置 Data Mover

要启用内置 Data Mover，您必须在 **DataProtectionApplication** 自定义资源 (CR) 中包含 CSI 插件并启用节点代理。节点代理是一个 Kubernetes daemonset，用于托管数据移动模块。这包括 Data Mover 控制器、上传程序和存储库。

DataProtectionApplication 清单示例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    nodeAgent:
      enable: true (1)
```

```
    uploaderType: kopia 2
  velero:
    defaultPlugins:
    - openshift
    - aws
    - csi 3
    defaultSnapshotMoveData: true
    defaultVolumesToFSBackup: 4
    featureFlags:
    - EnableCSI
# ...
```

1 启用节点代理的标记。

2 上传程序的类型。可能的值为 **restic** 或 **kopia**。内置 Data Mover 使用 Kopia 作为默认的上传程序机制，无论 **uploaderType** 字段的值是什么。

3 CSI 插件包含在默认插件列表中。

4 在 OADP 1.3.1 及更高版本中，如果您只对选择不使用 **fs-backup** 的卷使用 Data Mover，则设置为 **true**。如果卷默认使用 Data Mover，则设置为 **false**。

4.11.1.2. 内置数据管理控制器和自定义资源定义 (CRD)

内置的 Data Mover 功能引入了三个新的 API 对象，被定义为 CRD，用于管理备份和恢复：

- **DataDownload**: 代表卷快照的数据下载。CSI 插件为每个要恢复的卷创建一个 **DataDownload** 对象。**DataDownload** CR 包含有关目标卷的信息、指定的 Data Mover、当前数据下载的进度、指定的备份存储库以及进程完成后当前数据下载的结果。
- **DataUpload** ：代表卷快照的数据上传。CSI 插件为每个 CSI 快照创建一个 **DataUpload** 对象。**DataUpload** CR 包含有关指定快照的信息、指定的 Data Mover、指定的备份存储库、当前数据上传的进度，以及进程完成后当前数据上传的结果。
- **BackupRepository**: 代表和管理备份存储库的生命周期。当请求第一个 CSI 快照备份或恢复命名空间时，OADP 会为每个命名空间创建一个备份存储库。

4.11.2. 备份和恢复 CSI 快照数据移动

您可以使用 OADP 1.3 Data Mover 备份和恢复持久性卷。

4.11.2.1. 使用 CSI 快照备份持久性卷

您可以使用 OADP Data Mover 将 Container Storage Interface (CSI) 卷快照备份到远程对象存储。

先决条件

- 您可以使用 **cluster-admin** 角色访问集群。
- 已安装 OADP Operator。
- 您已在 **DataProtectionApplication** 自定义资源(CR) 中包含了 CSI 插件并启用了节点代理。
- 您有一个应用程序，其持久性卷在单独的命名空间中运行。

- 您已将 **metadata.labels.velero.io/csi-volumesnapshot-class: "true"** 键值对添加到 **VolumeSnapshotClass** CR。

流程

1. 为 **Backup** 对象创建一个 YAML 文件，如下例所示：

   Backup CR 示例

```
kind: Backup
apiVersion: velero.io/v1
metadata:
  name: backup
  namespace: openshift-adp
spec:
  csiSnapshotTimeout: 10m0s
  defaultVolumesToFsBackup: 1
  includedNamespaces:
  - mysql-persistent
  itemOperationTimeout: 4h0m0s
  snapshotMoveData: true 2
  storageLocation: default
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - dpa-sample-1
# ...
```

   1 如果您只对选择不使用 **fs-backup** 的卷使用 Data Mover，则设置为 **true**。如果卷默认使用 Data Mover，则设置为 **false**。

   2 设置为 **true** 以启用将 CSI 快照移到远程对象存储。

2. 应用清单：

```
$ oc create -f backup.yaml
```

   在快照创建完成后会创建一个 **DataUpload** CR。

验证

- 通过监控 **DataUpload** CR 的 **status.phase** 字段来验证快照数据是否已成功传送到远程对象存储。可能的值为 **In Progress**、**Completed**、**Failed** 或 **Canceled**。对象存储在 **DataProtectionApplication** CR 的 **backupLocations** 小节中配置。
  - 运行以下命令获取所有 **DataUpload** 对象的列表：

```
$ oc get datauploads -A
```

    输**出示例**

```
NAMESPACE       NAME              STATUS      STARTED   BYTES DONE   TOTAL
BYTES   STORAGE LOCATION   AGE     NODE
openshift-adp   backup-test-1-sw76b   Completed   9m47s     108104082    108104082
```

```
dpa-sample-1       9m47s   ip-10-0-150-57.us-west-2.compute.internal
openshift-adp   mongo-block-7dtpf     Completed   14m      1073741824   1073741824
dpa-sample-1       14m     ip-10-0-150-57.us-west-2.compute.internal
```

- 运行以下命令，检查特定 **DataUpload** 对象的 **status.phase** 字段的值：

```
$ oc get datauploads <dataupload_name> -o yaml
```

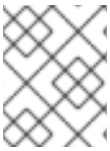
输出示例

```
apiVersion: velero.io/v2alpha1
kind: DataUpload
metadata:
  name: backup-test-1-sw76b
  namespace: openshift-adp
spec:
  backupStorageLocation: dpa-sample-1
  csiSnapshot:
    snapshotClass: ""
    storageClass: gp3-csi
    volumeSnapshot: velero-mysql-fq8sl
  operationTimeout: 10m0s
  snapshotType: CSI
  sourceNamespace: mysql-persistent
  sourcePVC: mysql
status:
  completionTimestamp: "2023-11-02T16:57:02Z"
  node: ip-10-0-150-57.us-west-2.compute.internal
  path: /host_pods/15116bac-cc01-4d9b-8ee7-
609c3bef6bde/volumes/kubernetes.io~csi/pvc-eead8167-556b-461a-b3ec-
441749e291c4/mount
  phase: Completed (1)
  progress:
    bytesDone: 108104082
    totalBytes: 108104082
  snapshotID: 8da1c5febf25225f4577ada2aeb9f899
  startTimestamp: "2023-11-02T16:56:22Z"
```

(1) 代表快照数据成功传输到远程对象存储。

4.11.2.2. 恢复 CSI 卷快照

您可以通过创建一个 **Restore** CR 来恢复卷快照。



注意

您不能使用 OAPD 1.3 内置数据 Mover 从 OADP 1.2 恢复 Volsync 备份。在升级到 OADP 1.3 之前，建议使用 Restic 对所有工作负载进行文件系统备份。

先决条件

- 您可以使用 **cluster-admin** 角色访问集群。

- 您有一个 OADP **Backup** CR，可从中恢复数据。

流程

1. 为 **Restore** CR 创建 YAML 文件，如下例所示：

   Restore CR 示例

   ```
   apiVersion: velero.io/v1
   kind: Restore
   metadata:
     name: restore
     namespace: openshift-adp
   spec:
     backupName: <backup>
   # ...
   ```

2. 应用清单：

   ```
   $ oc create -f restore.yaml
   ```

   恢复启动时会创建一个 **DataDownload** CR。

验证

- 您可以通过检查 **DataDownload** CR 的 **status.phase** 字段来监控恢复过程的状态。可能的值为 **In Progress**、**Completed**、**Failed** 或 **Canceled**。
  - 要获取所有 **DataDownload** 对象的列表，请运行以下命令：

    ```
    $ oc get datadownloads -A
    ```

    输出示例

    ```
    NAMESPACE       NAME                 STATUS      STARTED   BYTES DONE   TOTAL BYTES   STORAGE LOCATION   AGE     NODE
    openshift-adp   restore-test-1-sk7lg   Completed   7m11s     108104082    108104082     dpa-sample-1       7m11s   ip-10-0-150-57.us-west-2.compute.internal
    ```

  - 输入以下命令检查特定 **DataDownload** 对象的 **status.phase** 字段的值：

    ```
    $ oc get datadownloads <datadownload_name> -o yaml
    ```

    输出示例

    ```
    apiVersion: velero.io/v2alpha1
    kind: DataDownload
    metadata:
      name: restore-test-1-sk7lg
      namespace: openshift-adp
    spec:
      backupStorageLocation: dpa-sample-1
      operationTimeout: 10m0s
    ```

```
    snapshotID: 8da1c5febf25225f4577ada2aeb9f899
    sourceNamespace: mysql-persistent
    targetVolume:
      namespace: mysql-persistent
      pv: ""
      pvc: mysql
  status:
    completionTimestamp: "2023-11-02T17:01:24Z"
    node: ip-10-0-150-57.us-west-2.compute.internal
    phase: Completed (1)
    progress:
      bytesDone: 108104082
      totalBytes: 108104082
    startTimestamp: "2023-11-02T17:00:52Z"
```

(1) 表示 CSI 快照数据已被成功恢复。

4.11.2.3. 删除 OADP 1.3 的策略

删除策略决定了从系统中删除数据的规则，指定根据保留周期、数据敏感度和合规要求等因素如何进行删除。它有效地管理数据删除，同时满足法规并保留宝贵的信息。

4.11.2.3.1. 删除 OADP 1.3 的策略指南

查看 OADP 1.3 的以下删除策略指南：

- 在 OADP 1.3.x 中，当使用任何类型的备份和恢复方法时，您可以在 **VolumeSnapshotClass** 自定义资源(CR)中将 **deletionPolicy** 字段设置为 **Retain** 或 **Delete**。

4.12. 故障排除

您可以使用 OpenShift CLI 工具或 Velero CLI 工具调试 Velero 自定义资源(CR)。Velero CLI 工具提供更详细的日志和信息。

您可以检查 安装问题、备份和恢复 CR 问题，以及 Restic 问题。

您可以使用 **must-gather** 工具收集日志和 CR 信息。

您可以通过以下方法获取 Velero CLI 工具：

- 下载 Velero CLI 工具
- 访问集群中的 Velero 部署中的 Velero 二进制文件

4.12.1. 下载 Velero CLI 工具

您可以按照 Velero 文档页面中的说明下载并安装 Velero CLI 工具。

该页面包括：

- 使用 Homebrew 的 macOS
- GitHub

- 使用 Chocolatey 的 Windows

先决条件

- 您可以访问启用了 DNS 和容器网络的 Kubernetes 集群 v1.16 或更高版本。
- 您已在本地安装了 **kubectl**。

流程

1. 打开浏览器，进入到在 Velero 网站上的"安装 CLI"。
2. 按照 macOS、GitHub 或 Windows 的适当流程。
3. 下载适用于 OADP 和 OpenShift Container Platform 版本的 Velero 版本。

4.12.1.1. OADP-Velero-OpenShift Container Platform 版本关系

OADP 版本	Velero 版本	OpenShift Container Platform 版本
1.1.0	1.9	4.9 及更新的版本
1.1.1	1.9	4.9 及更新的版本
1.1.2	1.9	4.9 及更新的版本
1.1.3	1.9	4.9 及更新的版本
1.1.4	1.9	4.9 及更新的版本
1.1.5	1.9	4.9 及更新的版本
1.1.6	1.9	4.11 及更新的版本
1.1.7	1.9	4.11 及更新的版本
1.2.0	1.11	4.11 及更新的版本
1.2.1	1.11	4.11 及更新的版本
1.2.2	1.11	4.11 及更新的版本
1.2.3	1.11	4.11 及更新的版本
1.3.0	1.12	4.12 及更新的版本

4.12.2. 访问集群中的 Velero 部署中的 Velero 二进制文件

您可以使用 shell 命令访问集群中的 Velero 部署中的 Velero 二进制文件。

先决条件

- 您的 **DataProtectionApplication** 自定义资源的状态为 **Reconcile complete**。

流程

- 输入以下命令设定所需的别名：

```
$ alias velero='oc -n openshift-adp exec deployment/velero -c velero -it -- ./velero'
```

4.12.3. 使用 OpenShift CLI 工具调试 Velero 资源

您可以使用 OpenShift CLI 工具检查 Velero 自定义资源(CR)和 **Velero** pod 日志来调试失败的备份或恢复。

Velero CR
使用 **oc describe** 命令检索与 **Backup** 或 **Restore** CR 关联的警告和错误概述：

```
$ oc describe <velero_cr> <cr_name>
```

Velero pod 日志
使用 **oc logs** 命令检索 **Velero** pod 日志：

```
$ oc logs pod/<velero>
```

Velero pod 调试日志
您可以在 **DataProtectionApplication** 资源中指定 Velero 日志级别，如下例所示。

注意

这个选项可从 OADP 1.0.3 开始。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
spec:
  configuration:
    velero:
      logLevel: warning
```

可用的 **logLevel** 值如下：

- **trace**
- **debug**
- **info**
- **warning**

- **错误**
- **fatal**
- **panic**

对于多数日志，建议使用 **debug**。

4.12.4. 使用 Velero CLI 工具调试 Velero 资源

您可以调试 **Backup** 和 **Restore** 自定义资源(CR)并使用 Velero CLI 工具检索日志。

Velero CLI 工具比 OpenShift CLI 工具提供更详细的信息。

语法
使用 **oc exec** 命令运行 Velero CLI 命令：

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> <command> <cr_name>
```

Example

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

帮助选项
使用 **velero --help** 列出所有 Velero CLI 命令：

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  --help
```

describe 命令
使用 **velero describe** 命令检索与 **Backup** 或 **Restore** CR 关联的警告和错误概述：

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> describe <cr_name>
```

Example

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

velero describe 请求的输出中会显示以下类型的恢复错误和警告：

- **Velero**: 与 Velero 本身操作相关的信息列表，例如：与连接到云相关的信息，读取备份文件等
- **集群**：与备份和恢复集群范围的资源相关的消息列表
- **命名空间**：与备份或恢复存储在命名空间中资源相关的消息列表

这些类别中的一个或多个错误会导致 **Restore** 操作接收 **PartiallyFailed** 而不是 **Completed** 状态。警告不会造成完成状态的更改。

重要

- 对于特定于资源的错误，即 **Cluster** 和 **Namespaces** 错误，**restore describe --details** 输出包含一个资源列表，其中列出了 Velero 在恢复中成功的所有资源。对于具有此类错误的任何资源，请检查资源是否实际位于集群中。
- 如果存在 **Velero** 错误，但没有特定于资源的错误，在 **describe** 命令的输出中没有完成恢复，且没有恢复工作负载中的实际问题，但仔细验证恢复后应用程序。例如，如果输出包含 **PodVolumeRestore** 或节点代理相关的错误，请检查 **PodVolumeRestores** 和 **DataDownloads** 的状态。如果其中任何失败或仍在运行，则卷数据可能已被完全恢复。

logs 命令
使用 **velero logs** 命令检索 **Backup** 或 **Restore** CR 的日志：

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> logs <cr_name>
```

Example

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

4.12.5. 因内存不足或 CPU 造成 pod 崩溃或重启

如果 Velero 或 Restic pod 因为缺少内存或 CPU 而导致崩溃，您可以为其中任何一个资源设置特定的资源请求。

其他资源

- CPU 和内存要求

4.12.5.1. 为 Velero pod 设置资源请求

您可以使用 **oadp_v1alpha1_dpa.yaml** 文件中的 **configuration.velero.podConfig.resourceAllocations** 规格字段为 **Velero** pod 设置特定的资源请求。

流程

- 在 YAML 文件中设置 **cpu** 和 **memory** 资源请求：

  Velero 文件示例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
configuration:
  velero:
    podConfig:
      resourceAllocations: 1
        requests:
          cpu: 200m
          memory: 256Mi
```

1 列出的 **resourceAllocations** 用于平均使用。

4.12.5.2. 为 Restic pod 设置资源请求

您可以使用 **configuration.restic.podConfig.resourceAllocations** specification 字段为 **Restic** pod 设置特定的资源请求。

流程

- 在 YAML 文件中设置 **cpu** 和 **memory** 资源请求：

Restic 文件示例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
configuration:
  restic:
    podConfig:
      resourceAllocations: 1
        requests:
          cpu: 1000m
          memory: 16Gi
```

1 列出的 **resourceAllocations** 用于平均使用。

重要

资源请求字段的值必须遵循与 Kubernetes 资源要求相同的格式。另外，如果您没有指定 **configuration.velero.podConfig.resourceAllocations** 或 **configuration.restic.podConfig.resourceAllocations**，则 Velero pod 或 Restic pod 的默认 **resources** 规格如下：

```
requests:
  cpu: 500m
  memory: 128Mi
```

4.12.6. Velero 和准入 Webhook 的问题

Velero 在恢复过程中解决准入 Webhook 问题的能力有限。如果您的工作负载带有准入 webhook，您可能需要使用额外的 Velero 插件或更改如何恢复工作负载。

通常，带有准入 Webhook 的工作负载需要您首先创建特定类型的资源。如果您的工作负载具有子资源，因为准入 webhook 通常阻止子资源，则会出现这种情况。

例如，创建或恢复顶层对象，如 **service.serving.knative.dev** 通常会自动创建子资源。如果您首先这样做，则不需要使用 Velero 创建和恢复这些资源。这可避免由 Velero 可使用的准入 Webhook 阻断子资源的问题。

4.12.6.1. 为使用准入 webhook 的 Velero 备份恢复临时解决方案

本节介绍了使用准入 webhook 的一些类型的 Velero 备份恢复资源所需的额外步骤。

4.12.6.1.1. 恢复 Knative 资源

您可能会遇到使用 Velero 备份使用准入 webhook 的 Knative 资源的问题。

在备份和恢复使用准入 webhook 的 Knative 资源时，您可以通过首先恢复顶层 **Service** 资源来避免这个问题。

流程

- 恢复顶层 **service.serving.knavtive.dev Service** 资源：

```
$ velero restore <restore_name> \
  --from-backup=<backup_name> --include-resources \
  service.serving.knavtive.dev
```

4.12.6.1.2. 恢复 IBM AppConnect 资源

如果您使用 Velero 恢复具有准入 webhook 的 IBM® AppConnect 资源时遇到问题，您可以在此过程中运行检查。

流程

1. 检查集群中是否有 **kind: MutatingWebhookConfiguration** 的变异准入插件：

```
$ oc get mutatingwebhookconfigurations
```

2. 检查每个 **kind: MutatingWebhookConfiguration** 的 YAML 文件，以确保其没有规则块创建存在问题的对象。如需更多信息，请参阅官方 Kubernetes 文档。

3. 检查在备份时使用的 **type: Configuration.appconnect.ibm.com/v1beta1** 中的 **spec.version** 被已安装的 Operator 支持。

4.12.6.2. OADP 插件的已知问题

以下章节介绍了 OpenShift API for Data Protection (OADP) 插件中的已知问题：

4.12.6.2.1. 因为缺少 secret，Velero 插件在镜像流备份过程中会出现错误

当在数据保护应用程序(DPA)范围外管理备份和备份存储位置(BSL)时，OADP 控制器，这意味着 DPA 协调不会创建相关的 **oadp-<bsl_name>-<bl_provider>-registry-secret**。

当备份运行时，OpenShift Velero 插件在镜像流备份中出现错误，并显示以下错误：

```
024-02-27T10:46:50.028951744Z time="2024-02-27T10:46:50Z" level=error msg="Error backing up
item"
backup=openshift-adp/<backup name> error="error executing custom action
(groupResource=imagestreams.image.openshift.io,
namespace=<BSL Name>, name=postgres): rpc error: code = Aborted desc = plugin panicked:
runtime error: index out of range with length 1, stack trace: goroutine 94…
```

4.12.6.2.1.1. 临时解决方案以避免出现错误

要避免 Velero 插件 panic 错误，请执行以下步骤：

1. 使用相关标签标记自定义 BSL：

```
$ oc label BackupStorageLocation <bsl_name> app.kubernetes.io/component=bsl
```

2. 在标记 BSL 后，等待 DPA 协调。

注意

您可以通过对 DPA 本身进行任何更改来强制进行协调。

3. 当 DPA 协调时，确认相关的 **oadp-<bsl_name>-<bsl_provider>-registry-secret** 已被填充到其中：

```
$ oc -n openshift-adp get secret/oadp-<bsl_name>-<bsl_provider>-registry-secret -o json | jq -r '.data'
```

4.12.6.2.2. OpenShift ADP Controller 分段错误

如果您在同时启用了 **cloudstorage** 和 **restic** 的情况下配置 DPA，**openshift-adp-controller-manager** pod 会无限期重复崩溃和重启过程，直到 pod 出现一个崩溃循环分段错误为止。

您只能定义 **velero** 或 **cloudstorage**，它们是互斥的字段。

- 如果您同时定义了 **velero** 和 **cloudstorage**，**openshift-adp-controller-manager** 会失败。
- 如果 **velero** 和 **cloudstorage** 都没有定义，**openshift-adp-controller-manager** 也将失败。

有关此问题的更多信息，请参阅 OADP-1054。

4.12.6.2.2.1. OpenShift ADP Controller 分段错误临时解决方案

在配置一个 DPA 时，您必须定义 **velero** 或 **cloudstorage**。如果您在 DPA 中同时定义了这两个 API，**openshift-adp-controller-manager** pod 会失败，并显示崩溃循环分段错误。

4.12.6.3. Velero 插件返回 "received EOF, stop recv loop" 信息

注意

Velero 插件作为单独的进程启动。当 Velero 操作完成后，无论是否成功，它们都会退出。接收到 **received EOF, stopping recv loop** 消息表示插件操作已完成。这并不意味着发生了错误。

其他资源

- 准入插件
- Webhook 准入插件
- Webhook 准入插件类型

4.12.7. 安装问题

在安装数据保护应用程序时，您可能会遇到使用无效目录或不正确的凭证导致的问题。

4.12.7.1. 备份存储包含无效目录

Velero pod 日志显示错误消息，**备份存储包含无效的顶级目录**。

原因

对象存储包含不是 Velero 目录的顶级目录。

解决方案

如果对象存储不适用于 Velero，则必须通过设置 **DataProtectionApplication** 清单中的 **spec.backupLocations.velero.objectStorage.prefix** 参数为存储桶指定一个前缀。

4.12.7.2. AWS 凭证不正确

oadp-aws-registry pod 日志会显示错误消息 **InvalidAccessKeyId: The AWS Access Key Id you provided does not exist in our records.**

Velero pod 日志显示错误消息 **NoCredentialProviders: no valid provider in chain**。

原因

用于创建 **Secret** 对象的 **credentials-velero** 文件会错误地格式化。

解决方案

确保 **credentials-velero** 文件已正确格式化，如下例所示：

credentials-velero 文件示例

```
[default] 1
aws_access_key_id=AKIAIOSFODNN7EXAMPLE 2
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

1 AWS 默认配置集。

2 不用使用括号 (", ') 把值括起来。

4.12.8. OADP Operator 问题

OpenShift API for Data Protection (OADP) Operator 可能会遇到它无法解决的问题。

4.12.8.1. OADP Operator 静默失败

OADP Operator 的 S3 存储桶可能为空，但在运行 **oc get po -n <OADP_Operator_namespace>** 命令时，您会看到 Operator 的状态变为 **Running**。在这种情况下，Operator 被认为有*静默地失败*，因为它错误地报告它正在运行。

原因

这个问题是因为云凭证提供的权限不足。

解决方案

检索备份存储位置列表(BSL)，并检查每个 BSL 的清单是否有凭证问题。

流程

1. 运行以下命令之一以检索 BSL 列表：

   a. 使用 OpenShift CLI：

   ```
   $ oc get backupstoragelocation -A
   ```

   b. 使用 Velero CLI：

   ```
   $ velero backup-location get -n <OADP_Operator_namespace>
   ```

2. 使用 BSL 列表，运行以下命令来显示每个 BSL 的清单，并检查每个清单是否有错误。

   ```
   $ oc get backupstoragelocation -n <namespace> -o yaml
   ```

结**果示例**

```
apiVersion: v1
items:
- apiVersion: velero.io/v1
  kind: BackupStorageLocation
  metadata:
    creationTimestamp: "2023-11-03T19:49:04Z"
    generation: 9703
    name: example-dpa-1
    namespace: openshift-adp-operator
    ownerReferences:
    - apiVersion: oadp.openshift.io/v1alpha1
      blockOwnerDeletion: true
      controller: true
      kind: DataProtectionApplication
      name: example-dpa
      uid: 0beeeaff-0287-4f32-bcb1-2e3c921b6e82
    resourceVersion: "24273698"
    uid: ba37cd15-cf17-4f7d-bf03-8af8655cea83
  spec:
    config:
      enableSharedConfig: "true"
      region: us-west-2
    credential:
      key: credentials
      name: cloud-credentials
    default: true
    objectStorage:
      bucket: example-oadp-operator
      prefix: example
    provider: aws
  status:
    lastValidationTime: "2023-11-10T22:06:46Z"
    message: "BackupStorageLocation \"example-dpa-1\" is unavailable: rpc
      error: code = Unknown desc = WebIdentityErr: failed to retrieve credentials\ncaused
      by: AccessDenied: Not authorized to perform sts:AssumeRoleWithWebIdentity\n\tstatus
      code: 403, request id: d3f2e099-70a0-467b-997e-ff62345e3b54"
    phase: Unavailable
```

```
kind: List
metadata:
  resourceVersion: ""
```

4.12.9. OADP 超时

通过扩展超时，可以允许复杂的或资源密集型的进程在没有预先终止的情况下成功完成。此配置可减少错误、重试或失败的可能性。

确保您在扩展超时设置时符合正常的逻辑，，以便不会因为设置的超时时间太长导致隐藏了底层存在的问题。仔细考虑并监控超时设置，以符合相关进程的需求和整体系统性能要求。

以下是不同的 OADP 超时设置的信息：

4.12.9.1. Restic 超时

timeout 定义 Restic 超时。默认值为 **1h**。

在以下情况下使用 Restic **timeout** ：

- 对总 PV 数据使用量大于 500GB 的 Restic 备份。
- 如果备份超时并显示以下错误：

  ```
  level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all PodVolumeBackups to complete"
  ```

流程

- 编辑 **DataProtectionApplication** CR 清单中的 **spec.configuration.restic.timeout** 块的值，如下例所示：

  ```
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
   name: <dpa_name>
  spec:
    configuration:
      restic:
        timeout: 1h
  # ...
  ```

4.12.9.2. Velero 资源超时

resourceTimeout 定义在超时发生前等待 Velero 资源的时间，如等待 Velero 自定义资源定义 (CRD)可用、**volumeSnapshot** 删除和存储库可用。默认值为 **10m**。

在以下情况下使用 **resourceTimeout** ：

- 对总 PV 数据使用量大于 1TB 的备份。当在将备份标记为完成前，Velero 尝试清理或删除 Container Storage Interface (CSI)快照时使用此参数作为超时值。
  - 这个清理过程的一个子任务会尝试修补 VSC，此超时可用于该任务。

- 要创建或确保一个备份存储库已准备好用于 Restic 或 Kopia 的基于文件系统的备份。
- 在从备份中恢复自定义资源 (CR) 或资源前，检查集群中的 Velero CRD 是否可用。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.resourceTimeout** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: <dpa_name>
spec:
  configuration:
    velero:
      resourceTimeout: 10m
# ...
```

4.12.9.3. Data Mover timeout

timeout 是一个用户提供的、完成 **VolumeSnapshotBackup** 和 **VolumeSnapshotRestore** 的超时值。默认值为 **10m**。

在以下情况下使用 Data Mover **timeout** ：

- 如果创建 **VolumeSnapshotBackups** (VSBs) 和 **VolumeSnapshotRestores** (VSR)，则会在 10 分钟后超时。
- 对于总 PV 数据使用量超过 500GB 的大型环境。设置 **1h** 的超时时间。
- 使用 **VolumeSnapshotMover** (VSM) 插件。
- 只适用于 OADP 1.1.x。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.features.dataMover.timeout** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: <dpa_name>
spec:
  features:
    dataMover:
      timeout: 10m
# ...
```

4.12.9.4. CSI 快照超时

CSISnapshotTimeout 指定，在创建过程返回超时错误前，需要等待 **CSI VolumeSnapshot** 状态变为 **ReadyToUse** 的时间。默认值为 **10m**。

在以下情况下使用 **CSISnapshotTimeout** ：

- 使用 CSI 插件。
- 对于非常大型的存储卷，进行快照的时间可能会超过 10 分钟。如果在日志中出现超时信息，请调整此超时设置。

注意

通常，不需要调整 **CSISnapshotTimeout**，因为默认设置已考虑到大型存储卷的情况。

流程

- 编辑 **Backup** CR 清单的 **spec.csiSnapshotTimeout** 块中的值，如下例所示：

```
apiVersion: velero.io/v1
kind: Backup
metadata:
 name: <backup_name>
spec:
 csiSnapshotTimeout: 10m
# ...
```

4.12.9.5. Velero 默认项目操作超时

defaultItemOperationTimeout 定义在超时前等待异步 **BackupItemActions** 和 **RestoreItemActions** 所需的时间。默认值为 **1h**。

在以下情况下使用 **defaultItemOperationTimeout** ：

- 只有 Data Mover 1.2.x。
- 要指定一个特定备份或恢复应等待异步操作完成的时间长度。在 OADP 功能上下文中，这个值用于涉及 Container Storage Interface (CSI) Data Mover 功能的异步操作。
- 当使用 **defaultItemOperationTimeout** 在 Data Protection Application (DPA) 中定义 **defaultItemOperationTimeout** 时，它适用于备份和恢复操作。您可以使用 **itemOperationTimeout** 来只定义这些 CR 的备份过程或恢复过程，如以下 "Item operation timeout - restore" 和 "Item operation timeout - backup" 部分所述。

流程

- 编辑 **DataProtectionApplication** CR 清单的 **spec.configuration.velero.defaultItemOperationTimeout** 块中的值，如下例所示：

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: <dpa_name>
spec:
  configuration:
    velero:
      defaultItemOperationTimeout: 1h
# ...
```

4.12.9.6. 项目操作超时 - 恢复

ItemOperationTimeout 指定用于等待 **RestoreItemAction** 操作的时间。默认值为 **1h**。

在以下情况下，使用 restore **ItemOperationTimeout** ：

- 只有 Data Mover 1.2.x。
- 对于 Data Mover，上传到 **BackupStorageLocation** 或从其中下载。如果在达到超时时没有完成恢复操作，它将标记为失败。如果因为存储卷太大出现超时并导致数据 Data Mover 操作失败，则可能需要增加这个超时设置。

流程

- 编辑 **Restore** CR 清单的 **Restore.spec.itemOperationTimeout** 块中的值，如下例所示：

```
apiVersion: velero.io/v1
kind: Restore
metadata:
 name: <restore_name>
spec:
 itemOperationTimeout: 1h
 # ...
```

4.12.9.7. 项目操作超时 - 备份

ItemOperationTimeout 指定用于等待异步 **BackupItemAction** 操作的时间。默认值为 **1h**。

在以下情况下，使用 backup **ItemOperationTimeout** ：

- 只有 Data Mover 1.2.x。
- 对于 Data Mover，上传到 **BackupStorageLocation** 或从其中下载。如果在达到超时时没有完成备份操作，它将标记为失败。如果因为存储卷太大出现超时并导致数据 Data Mover 操作失败，则可能需要增加这个超时设置。

流程

- 编辑 **Backup** CR 清单的 **Backup.spec.itemOperationTimeout** 块中的值，如下例所示：

```
apiVersion: velero.io/v1
kind: Backup
metadata:
 name: <backup_name>
spec:
 itemOperationTimeout: 1h
 # ...
```

4.12.10. 备份和恢复 CR 问题

您可能会遇到 **Backup** 和 **Restore** 自定义资源(CR)的常见问题。

4.12.10.1. 备份 CR 无法检索卷

Backup CR 显示错误消息 **InvalidVolume.NotFound: The volume 'vol-xxxx' does not exist**。

原因

持久性卷(PV)和快照位置位于不同的区域。

解决方案

1. 编辑 **DataProtectionApplication** 清单中的 **spec.snapshotLocations.velero.config.region** 键的值，使快照位置位于与 PV 相同的区域。
2. 创建新的 **Backup** CR。

4.12.10.2. 备份 CR 状态在进行中

Backup CR 的状态保留在 **InProgress** 阶段，且未完成。

原因

如果备份中断，则无法恢复。

解决方案

1. 检索 **Backup** CR 的详细信息：

```
$ oc -n {namespace} exec deployment/velero -c velero -- ./velero \
  backup describe <backup>
```

2. 删除 **Backup** CR：

```
$ oc delete backup <backup> -n openshift-adp
```

   您不需要清理备份位置，因为正在进行中的 **Backup** CR 没有上传文件到对象存储。

3. 创建新的 **Backup** CR。

4.12.10.3. 备份 CR 状态处于 PartiallyFailed

在没有 Restic 使用时一个 **Backup** CR 的状态保留在 **PartiallyFailed** 阶段，且没有完成。从属 PVC 的快照没有创建。

原因

如果备份是基于 CSI 快照类创建的，但缺少标签，CSI 快照插件将无法创建快照。因此，**Velero** pod 会记录类似如下的错误：

```
time="2023-02-17T16:33:13Z" level=error msg="Error backing up item" backup=openshift-adp/user1-backup-check5 error="error executing custom action (groupResource=persistentvolumeclaims, namespace=busy1, name=pvc1-user1): rpc error: code = Unknown desc = failed to get volumesnapshotclass for storageclass ocs-storagecluster-ceph-rbd: failed to get volumesnapshotclass for provisioner openshift-storage.rbd.csi.ceph.com, ensure that the desired volumesnapshot class has the velero.io/csi-volumesnapshot-class label" logSource="/remote-source/velero/app/pkg/backup/backup.go:417" name=busybox-79799557b5-vprq
```

解决方案

1. 删除 **Backup** CR：

```
$ oc delete backup <backup> -n openshift-adp
```

2. 如果需要，清理 **BackupStorageLocation** 上存储的数据以释放空间。
3. 将标签 **velero.io/csi-volumesnapshot-class=true** 应用到 **VolumeSnapshotClass** 对象：

```
$ oc label volumesnapshotclass/<snapclass_name> velero.io/csi-volumesnapshot-class=true
```

4. 创建新的 **Backup** CR。

4.12.11. Restic 问题

在使用 Restic 备份应用程序时，您可能会遇到这些问题。

4.12.11.1. 启用了 root_squash 的 NFS 数据卷的 Restic 权限错误

Restic pod 日志显示错误消息, **controller=pod-volume-backup error="fork/exec/usr/bin/restic: permission denied"**。

原因

如果您的 NFS 数据卷启用了 **root_squash**，**Restic** 映射到 **nfsnobody**，且没有创建备份的权限。

解决方案

您可以通过为 **Restic** 创建补充组并将组 ID 添加到 **DataProtectionApplication** 清单中来解决这个问题：

1. 在 NFS 数据卷中为 **Restic** 创建补充组。
2. 在 NFS 目录上设置 **setgid** 位，以便继承组所有权。
3. 将 **spec.configuration.restic.supplementalGroups** 参数和组 ID 添加到 **DataProtectionApplication** 清单中，如下例所示：

```
spec:
  configuration:
    restic:
      enable: true
      supplementalGroups:
      - <group_id> 1
```

1 指定补充组 ID。

4. 等待 **Restic** pod 重启，以便应用更改。

4.12.11.2. 在存储桶被强制后重新创建 Restic Backup CR

如果您为命名空间创建 Restic **Backup** CR，请清空对象存储的存储桶，然后为同一命名空间重新创建 **Backup** CR，重新创建的 **Backup** CR 会失败。

velero pod 日志显示以下错误消息：**stderr=Fatal: unable to open config file: Stat: The specified key does not exist.\nIs there a repository at the following location?**。

原因

如果 Restic 目录从对象存储中删除，Velero 不会从 **ResticRepository** 清单重新创建或更新 Restic 存储库。如需更多信息，请参阅 Velero 问题 4421。

解决方案

- 运行以下命令，从命名空间中删除相关的 Restic 存储库：

```
$ oc delete resticrepository openshift-adp <name_of_the_restic_repository>
```

在以下错误日志中，**mysql-persistent** 是有问题的 Restic 存储库。存储库的名称会出现在其说明中。

```
time="2021-12-29T18:29:14Z" level=info msg="1 errors
encountered backup up item" backup=velero/backup65
logSource="pkg/backup/backup.go:431" name=mysql-7d99fc949-qbkds
time="2021-12-29T18:29:14Z" level=error msg="Error backing up item"
backup=velero/backup65 error="pod volume backup failed: error running
restic backup, stderr=Fatal: unable to open config file: Stat: The
specified key does not exist.\nIs there a repository at the following
location?\ns3:http://minio-minio.apps.mayap-oadp-
veleo-1234.qe.devcluster.openshift.com/mayapvelerooadp2/velero1/
restic/mysql-persistent\n: exit status 1" error.file="/remote-source/
src/github.com/vmware-tanzu/velero/pkg/restic/backupper.go:184"
error.function="github.com/vmware-tanzu/velero/
pkg/restic.(*backupper).BackupPodVolumes"
logSource="pkg/backup/backup.go:435" name=mysql-7d99fc949-qbkds
```

4.12.11.3. 因为更改了 PSA 策略，Restic 恢复部分在 OCP 4.14 上失败

OpenShift Container Platform 4.14 强制执行一个 pod 安全准入 (PSA) 策略，该策略可能会在 Restic 恢复过程中阻止 pod 的就绪度。

如果创建 Pod 时找不到 **SecurityContextConstraints** (SCC) 资源，并且 pod 上的 PSA 策略没有设置为满足所需的标准，则 Pod 准入将被拒绝。

造成这个问题的原因是 Velero 资源恢复的顺序。

错误**示例**

```
\"level=error\" in line#2273: time=\"2023-06-12T06:50:04Z\"
level=error msg=\"error restoring mysql-869f9f44f6-tp5lv: pods\\\
"mysql-869f9f44f6-tp5lv\\\" is forbidden: violates PodSecurity\\\
"restricted:v1.24\\\": privil eged (container \\\"mysql\\\
" must not set securityContext.privileged=true),
allowPrivilegeEscalation != false (containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.allowPrivilegeEscalation=false), unrestricted
capabilities (containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.capabilities.drop=[\\\"ALL\\\"]), seccompProfile
(pod or containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.seccompProfile.type to \\\
"RuntimeDefault\\\" or \\\"Localhost\\\")\" logSource=\"/remote-
source/velero/app/pkg/restore/restore.go:1388\" restore=openshift-adp/todolist-backup-0780518c-
08ed-11ee-805c-0a580a80e92c\n
```

```
velero container contains \"level=error\" in line#2447: time=\"2023-06-12T06:50:05Z\"
level=error msg=\"Namespace todolist-mariadb,
resource restore error: error restoring pods/todolist-mariadb/mysql-869f9f44f6-tp5lv: pods \\\
"mysql-869f9f44f6-tp5lv\\\" is forbidden: violates PodSecurity \\\"restricted:v1.24\\\": privileged
(container \\\
"mysql\\\" must not set securityContext.privileged=true),
allowPrivilegeEscalation != false (containers \\\
"restic-wait\\\",\\\"mysql\\\" must set securityContext.allowPrivilegeEscalation=false), unrestricted
capabilities (containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.capabilities.drop=[\\\"ALL\\\"]), seccompProfile
(pod or containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.seccompProfile.type to \\\
"RuntimeDefault\\\" or \\\"Localhost\\\")\"
logSource=\"/remote-source/velero/app/pkg/controller/restore_controller.go:510\"
restore=openshift-adp/todolist-backup-0780518c-08ed-11ee-805c-0a580a80e92c\n]",
```

解决方案

1. 在 DPA 自定义资源 (CR) 中，检查或设置 Velero 服务器上的 **restore-resource-priorities** 字段，以确保在资源列表的 **pod** 之前列出 **securitycontextconstraints** ：

```
$ oc get dpa -o yaml
```

DPA CR 示例

```
# ...
configuration:
  restic:
    enable: true
  velero:
    args:
      restore-resource-priorities:
'securitycontextconstraints,customresourcedefinitions,namespaces,storageclasses,volumesnaps
hotclass.snapshot.storage.k8s.io,volumesnapshotcontents.snapshot.storage.k8s.io,volumesnap
hots.snapshot.storage.k8s.io,datauploads.velero.io,persistentvolumes,persistentvolumeclaims,s
rviceaccounts,secrets,configmaps,limitranges,pods,replicasets.apps,clusterclasses.cluster.x-
k8s.io,endpoints,services,-,clusterbootstraps.run.tanzu.vmware.com,clusters.cluster.x-
k8s.io,clusterresourcesets.addons.cluster.x-k8s.io' (1)
    defaultPlugins:
    - gcp
    - openshift
```

(1) 如果您有一个现有的恢复资源优先级列表，请确保将现有列表与完整列表合并。

2. 确保应用程序 pod 的安全标准一致（如为部署修复 PodSecurity Admission 警告 中所述）以防止部署警告。如果应用程序与安全标准不一致，无论 SCC 是什么，都可能会出现错误。

注意

这个解决方案是临时的，永久解决方案正在讨论中。

其他资源

- 为部署修复 PodSecurity Admission 警告

4.12.12. 使用 must-gather 工具

您可以使用 **must-gather** 工具收集有关 OADP 自定义资源的日志、指标和信息。

must-gather 数据必须附加到所有客户案例。

您可以使用以下数据收集选项运行 **must-gather** 工具：

- 完全 **must-gather** 数据收集为安装 OADP Operator 的所有命名空间收集 Prometheus metrics、pod 日志和 Velero CR 信息。
- 基本的 **must-gather** 数据收集在特定持续时间内收集 pod 日志和 Velero CR 信息，例如一小时或 24 小时。Prometheus 指标和重复日志不包含在内。
- 使用超时的 **must-gather** 数据收集。如果有许多 **Backup** CR 失败，则数据收集需要很长时间。您可以通过设置超时值来提高性能。
- Prometheus 指标数据转储下载包含 Prometheus 收集的数据的存档文件。

先决条件

- 您必须使用具有 **cluster-admin** 角色的用户登录到 OpenShift Container Platform 集群。
- 已安装 OpenShift CLI (**oc**)。
- 对于 OADP 1.2，需要使用 Red Hat Enterprise Linux (RHEL) 8.x。
- 对于 OADP 1.3，需要使用 Red Hat Enterprise Linux (RHEL) 9.x。

流程

1. 进入存储 **must-gather** 数据的目录。
2. 为以下数据收集选项之一运行 **oc adm must-gather** 命令：
   - 完整的 **must-gather** 数据收集，包括 Prometheus 指标：
     a. 对于 OADP 1.2，运行以下命令：

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.2
     ```

     b. 对于 OADP 1.3，运行以下命令：

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3
     ```

     数据保存为 **must-gather/must-gather.tar.gz**。您可以将此文件上传到红帽客户门户网站中的支持问题单中。
   - 特定持续时间内，基本 **must-gather** 数据收集功能不进行 Prometheus 指标：

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.1 \
       -- /usr/bin/gather_<time>_essential 1
     ```

1 以小时为单位指定时间。允许的值是 **1h**、**6h**、**24h**、**72h** 或 **all**，例如 **gather_1h_essential** 或 **gather_all_essential**。

- 使用超时的 **must-gather** 数据收集：

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.1 \
  -- /usr/bin/gather_with_timeout <timeout> 1
```

1 以秒为单位指定超时值。

- Prometheus 指标数据转储：

a. 对于 OADP 1.2，运行以下命令：

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.2 --
/usr/bin/gather_metrics_dump
```

b. 对于 OADP 1.3，运行以下命令：

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 --
/usr/bin/gather_metrics_dump
```

此操作可能需要很长时间。数据保存为 **must-gather/metrics/prom_data.tar.gz**。

其他资源

- 收集集群数据

4.12.12.1. 使用带有不安全 TLS 连接的 must-gather

如果使用自定义 CA 证书，**must-gather** pod 无法获取 **velero logs/describe** 的输出。要将 **must-gather** 工具与不安全的 TLS 连接搭配使用，您可以将 **gather_without_tls** 标志传递给 **must-gather** 命令。

流程

- 使用以下命令，将 **gather_without_tls** 标志（值设为 **true**）传递给 **must-gather** 工具：

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 --
/usr/bin/gather_without_tls <true/false>
```

默认情况下，这个标志的值设为 **false**。将值设为 **true** 以允许不安全的 TLS 连接。

4.12.12.2. 使用 must-gather 工具合并选项

目前，无法组合 must-gather 脚本，例如指定超时阈值，同时允许不安全的 TLS 连接。在某些情况下，您可以通过在 must-gather 命令行中设置内部变量来解决这个限制，如下例所示：

```
oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 -- skip_tls=true
/usr/bin/gather_with_timeout <timeout_value_in_seconds>
```

在本例中，在运行 **gather_with_timeout** 脚本前设置 **skip_tls** 变量。结果是 **gather_with_timeout** 和 **gather_without_tls** 的组合。

您可以以这种方式指定的其他变量是如下：

- **logs_since**，默认值为 **72h**
- **request_timeout**，默认值为 **0s**

如果 **DataProtectionApplication** 自定义资源(CR)配置了 **s3Url** 和 **insecureSkipTLS: true**，则 CR 不会因为缺少 CA 证书而收集必要的日志。要收集这些日志，请使用以下选项运行 **must-gather** 命令：

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 --
/usr/bin/gather_without_tls true
```

4.12.13. OADP Monitoring

OpenShift Container Platform 提供了一个监控堆栈，允许用户和管理员有效地监控和管理其集群，并监控和分析集群中运行的用户应用程序和服务的工作负载性能，包括在事件发生时收到警报。

其他资源

- 监控堆栈

4.12.13.1. OADP 监控设置

OADP Operator 利用 OpenShift Monitoring Stack 提供的 OpenShift User Workload Monitoring 从 Velero 服务端点检索指标。监控堆栈允许使用 OpenShift Metrics 查询前端创建用户定义的 Alerting Rules 或查询指标。

启用 User Workload Monitoring 后，可以配置和使用任何与 Prometheus 兼容的第三方 UI （如 Grafana）来视觉化 Velero 指标。

监控指标需要为用户定义的项目启用监控，并创建一个 **ServiceMonitor** 资源，以便从位于 **openshift-adp** 命名空间中的已启用的 OADP 服务端点中提取这些指标。

先决条件

- 可以使用具有 **cluster-admin** 权限的账户访问 OpenShift Container Platform 集群。
- 您已创建了集群监控配置映射。

流程

1. 编辑 **openshift-monitoring** 命名空间中的 **cluster-monitoring-config ConfigMap** 对象：

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

2. 在 **data** 部分的 **config.yaml** 字段中添加或启用 **enableUserWorkload** 选项：

```
apiVersion: v1
data:
  config.yaml: |
    enableUserWorkload: true 1
kind: ConfigMap
metadata:
# ...
```

1 添加这个选项或设置为 **true**

3. 通过检查以下组件是否在 **openshift-user-workload-monitoring** 命名空间中运行，等待较短的时间段来验证 User Workload Monitoring Setup：

```
$ oc get pods -n openshift-user-workload-monitoring
```

输出示例

```
NAME                                   READY   STATUS    RESTARTS   AGE
prometheus-operator-6844b4b99c-b57j9   2/2     Running   0          43s
prometheus-user-workload-0             5/5     Running   0          32s
prometheus-user-workload-1             5/5     Running   0          32s
thanos-ruler-user-workload-0           3/3     Running   0          32s
thanos-ruler-user-workload-1           3/3     Running   0          32s
```

4. 验证 **openshift-user-workload-monitoring** 中是否存在 **user-workload-monitoring-config** ConfigMap。如果存在，请跳过这个过程中的剩余步骤。

```
$ oc get configmap user-workload-monitoring-config -n openshift-user-workload-monitoring
```

输出示例

```
Error from server (NotFound): configmaps "user-workload-monitoring-config" not found
```

5. 为 User Workload Monitoring 创建一个 **user-workload-monitoring-config ConfigMap** 对象，并将它保存为 **2_configure_user_workload_monitoring.yaml** 文件：

输出示例

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
```

6. 应用 **2_configure_user_workload_monitoring.yaml** 文件：

```
$ oc apply -f 2_configure_user_workload_monitoring.yaml
configmap/user-workload-monitoring-config created
```

4.12.13.2. 创建 OADP 服务监控器

OADP 提供了一个 **openshift-adp-velero-metrics-svc** 服务，它会在配置 DPA 时创建。用户工作负载监控使用的服务监控器必须指向定义的服务。

运行以下命令来获取该服务的详情：

流程

1. 确保 **openshift-adp-velero-metrics-svc** 服务存在。它应当包含 **app.kubernetes.io/name=velero** 标签，这些标签将用作 **ServiceMonitor** 对象的选择器。

```
$ oc get svc -n openshift-adp -l app.kubernetes.io/name=velero
```

输**出示例**

```
NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
openshift-adp-velero-metrics-svc   ClusterIP   172.30.38.244   <none>        8085/TCP   1h
```

2. 创建一个与现有 service 标签匹配的 **ServiceMonitor** YAML 文件，并将文件保存为 **3_create_oadp_service_monitor.yaml**。服务监控器在 **openshift-adp** 命名空间中创建，其中 **openshift-adp-velero-metrics-svc** 服务所在的位置。

ServiceMonitor 对**象示例**

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app: oadp-service-monitor
  name: oadp-service-monitor
  namespace: openshift-adp
spec:
  endpoints:
  - interval: 30s
    path: /metrics
    targetPort: 8085
    scheme: http
  selector:
    matchLabels:
      app.kubernetes.io/name: "velero"
```

3. 应用 **3_create_oadp_service_monitor.yaml** 文件：

```
$ oc apply -f 3_create_oadp_service_monitor.yaml
```

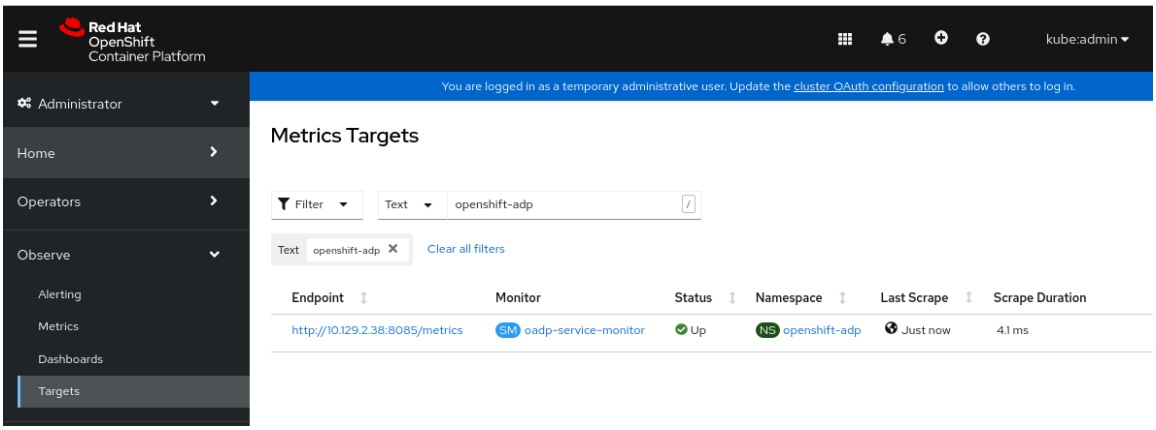
输**出示例**

```
servicemonitor.monitoring.coreos.com/oadp-service-monitor created
```

验证

- 使用 OpenShift Container Platform Web 控制台的 **Administrator** 视角确认新服务监控器处于 **Up** 状态：
  a. 进入到 **Observe → Targets** 页面。
  b. 确保没有选择 **Filter**，或选择了 **User** source，并在 **Text** 搜索字段中输入 **openshift-adp**。
  c. 验证服务监控器的 **Status** 的状态是否为 **Up**。

图 4.1. OADP 指标目标



4.12.13.3. 创建警报规则

OpenShift Container Platform 监控堆栈允许接收使用 Alerting Rules 配置的 Alerts。要为 OADP 项目创建 Alerting 规则，请使用用户工作负载监控提取的其中一个指标。

流程

1. 使用示例 **OADPBackupFailing** 警报创建一个 **PrometheusRule** YAML 文件，并将其保存为 **4_create_oadp_alert_rule.yaml**。

   OADPBackupFailing 警报示例

   ```
   apiVersion: monitoring.coreos.com/v1
   kind: PrometheusRule
   metadata:
     name: sample-oadp-alert
     namespace: openshift-adp
   spec:
     groups:
     - name: sample-oadp-backup-alert
       rules:
       - alert: OADPBackupFailing
         annotations:
           description: 'OADP had {{$value | humanize}} backup failures over the last 2 hours.'
           summary: OADP has issues creating backups
         expr: |
           increase(velero_backup_failure_total{job="openshift-adp-velero-metrics-svc"}[2h]) > 0
         for: 5m
         labels:
           severity: warning
   ```

   在本例中，Alert 在以下情况下显示：

   - 在最后 2 个小时内增加了新的故障备份（大于 0），且状态至少维持了 5 分钟。
   - 如果第一次增加的时间小于 5 分钟，则 Alert 将处于 **Pending** 状态，之后它将进入 **Firing** 状态。

2. 应用 **4_create_oadp_alert_rule.yaml** 文件，该文件在 **openshift-adp** 命名空间中创建 **PrometheusRule** 对象：

```
$ oc apply -f 4_create_oadp_alert_rule.yaml
```

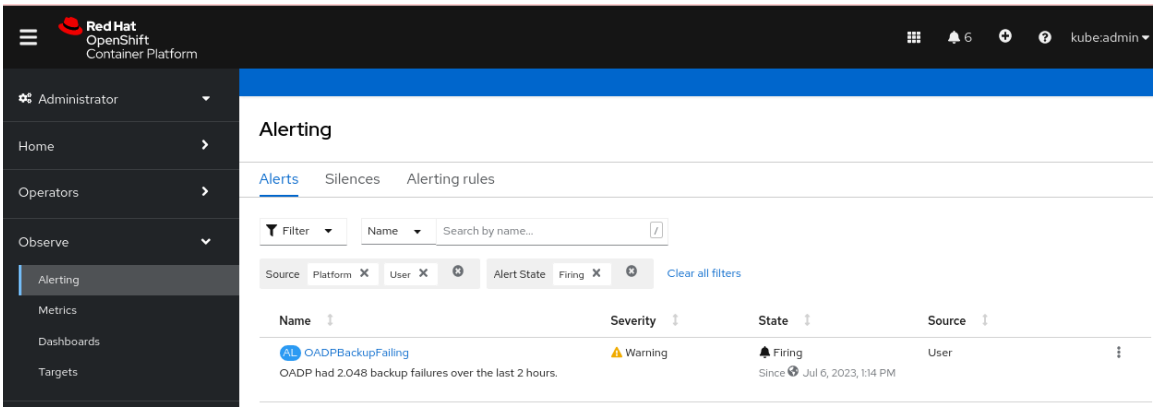
输出示例

```
prometheusrule.monitoring.coreos.com/sample-oadp-alert created
```

验证

- 在触发 Alert 后，您可以使用以下方法查看它：
  - 在 **Developer** 视角中，选择 **Observe** 菜单。
  - 在 **Observe → Alerting** 菜单下的 **Administrator** 视角中，在 **Filter** 框中选择 **User**。否则，默认只会显示 **Platform** Alerts。

图 4.2. OADP 备份失败警报



其他资源

- 管理警报

4.12.13.4. 可用指标列表

这些是 OADP 提供的指标列表，以及它们的 类型。

指标名称	描述	类型
kopia_content_cache_hit_bytes	从缓存检索的字节数	计数
kopia_content_cache_hit_count	从缓存检索内容的次数	计数
kopia_content_cache_malformed	从缓存中读取不正确的内容的次数	计数
kopia_content_cache_miss_count	没有在缓存中找到内容并获取它的次数	计数

指标名称	描述	类型
kopia_content_cache_missed_bytes	从底层存储检索的字节数	计数
kopia_content_cache_miss_error_count	在底层存储中无法找到内容的次数	计数
kopia_content_cache_store_error_count	无法保存到缓存中的次数	计数
kopia_content_get_bytes	使用 **GetContent()** 检索的字节数。	计数
kopia_content_get_count	**GetContent()** 被调用的次数	计数
kopia_content_get_error_count	调用 **GetContent()** 的次数，结果是错误	计数
kopia_content_get_not_found_count	调用 **GetContent()** 的次数，结果没有被找到	计数
kopia_content_write_bytes	传递给 **WriteContent()**的字节数。	计数
kopia_content_write_count	**WriteContent()** 被调用的次数	计数
velero_backup_attempt_total	试图备份的总数	计数
velero_backup_deletion_attempt_total	试图备份删除的总数	计数
velero_backup_deletion_failure_total	删除失败的备份总数	计数
velero_backup_deletion_success_total	成功删除备份的总数	计数
velero_backup_duration_seconds	完成备份所需的时间，以秒为单位	Histogram
velero_backup_failure_total	失败备份的总数	计数
velero_backup_items_errors	备份过程中遇到的错误总数	量表
velero_backup_items_total	备份的项目总数	量表

指标名称	描述	类型
velero_backup_last_status	备份的最后状态。值 1 代表成功，0。	量表
velero_backup_last_successful_timestamp	备份最后一次运行成功的时间，Unix 时间戳（以秒为单位）	量表
velero_backup_partial_failure_total	部分失败的备份总数	计数
velero_backup_success_total	成功备份的总数	计数
velero_backup_tarball_size_bytes	备份的大小，以字节为单位	量表
velero_backup_total	当前存在的备份数量	量表
velero_backup_validation_failure_total	验证失败的备份总数	计数
velero_backup_warning_total	警告备份的总数	计数
velero_csi_snapshot_attempt_total	CSI 试图卷快照的总数	计数
velero_csi_snapshot_failure_total	CSI 失败卷快照的总数	计数
velero_csi_snapshot_success_total	CSI 成功卷快照总数	计数
velero_restore_attempt_total	尝试恢复的总数	计数
velero_restore_failed_total	恢复的失败总数	计数
velero_restore_partial_failure_total	恢复部分失败的总数	计数
velero_restore_success_total	成功恢复的总数	计数
velero_restore_total	当前存在的恢复数量	量表
velero_restore_validation_failed_total	恢复失败验证的总数	计数

指标名称	描述	类型
velero_volume_snapshot_attempt_total	尝试的卷快照总数	计数
velero_volume_snapshot_failure_total	失败的卷快照总数	计数
velero_volume_snapshot_success_total	成功卷快照的总数	计数

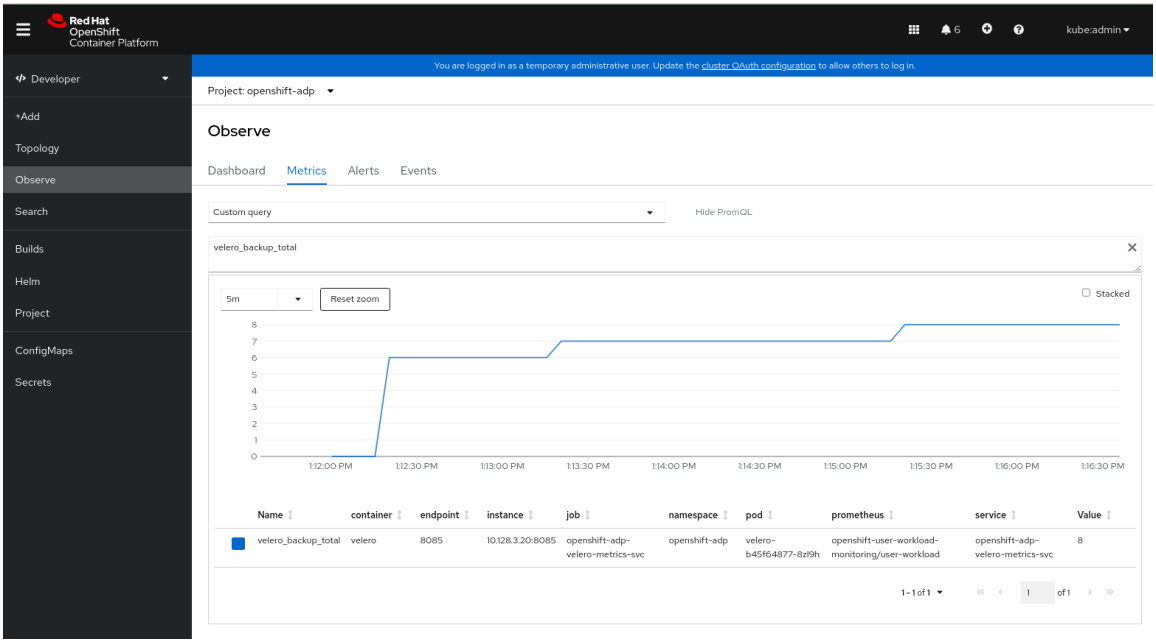
4.12.13.5. 使用 Observe UI 查看指标

您可以从 **Administrator** 或 **Developer** 视角查看 OpenShift Container Platform Web 控制台中的指标，该视角必须有权访问 **openshift-adp** 项目。

流程

- 进入到 **Observe** → **Metrics** 页面：
  - 如果使用 **Developer** 视角，请按照以下步骤执行：
    a. 选择 **Custom query**，或者点 **Show PromQL** 链接。
    b. 输入查询并点 **Enter**。
  - 如果使用 **Administrator** 视角，请在文本字段中输入表达式，然后选择 **Run Queries**。

图 4.3. OADP 指标查询



4.13. 与 OADP 一起使用的 API

本文档提供有关您可以在 OADP 一起使用的以下 API 的信息：

- Velero API

- OADP API

4.13.1. Velero API

Velero API 文档由 Velero 维护，而不是由红帽维护。它可在 Velero API 类型中找到。

4.13.2. OADP API

下表提供了 OADP API 的结构：

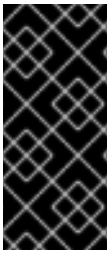
表 4.2. DataProtectionApplicationSpec

属性	类型	描述
backupLocations	[] **BackupLocation**	定义用于 **BackupStorageLocations** 的配置列表。
snapshotLocations	[] **SnapshotLocation**	定义 **VolumeSnapshotLocations** 使用的配置列表。
unsupportedOverrides	map [UnsupportedImageKey] string	可用于覆盖为开发而部署的依赖镜像。选项为 **veleroImageFqin**, **awsPluginImageFqin**, **openshiftPluginImageFqin**, **azurePluginImageFqin**, **gcpPluginImageFqin**, **csiPluginImageFqin**, **dataMoverImageFqin**, **resticRestoreImageFqin**, **kubevirtPluginImageFqin**, and **operator-type**。
podAnnotations	map [string] string	用于将注解添加到 Operator 部署的 pod。
podDnsPolicy	**DNSPolicy**	定义 Pod 的 DNS 的配置。
podDnsConfig	**PodDNSConfig**	定义除了由 **DNSPolicy** 生成的以外的 pod 的 DNS 参数。
backupImages	*bool	用于指定是否要部署 registry 以启用镜像的备份和恢复。
配置	***ApplicationConfig**	用于定义数据保护应用服务器配置。
功能	***特性**	定义 DPA 的配置以启用技术预览功能。

OADP API 的完整架构定义。

表 4.3. BackupLocation

属性	类型	描述
velero	*velero.BackupStorageLocationSpec	存储卷快照的位置，如备份存储位置所述。
bucket	*CloudStorageLocation	[技术预览] 在某些云存储供应商处自动创建存储桶，用作备份存储位置。



重要

bucket 参数只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

类型 **BackupLocation** 的完整 schema 定义。

表 4.4. SnapshotLocation

属性	类型	描述
velero	*VolumeSnapshotLocationSpec	用于存储卷快照的位置，如卷快照位置。

类型 **SnapshotLocation** 的完整 schema 定义。

表 4.5. ApplicationConfig

属性	类型	描述
velero	*VeleroConfig	定义 Velero 服务器配置。
restic	*ResticConfig	定义 Restic 服务器配置。

类型 **ApplicationConfig** 的完整 schema 定义。

表 4.6. VeleroConfig

属性	类型	描述
featureFlags	[] string	定义为 Velero 实例启用的功能列表。

属性	类型	描述
defaultPlugins	[] string	可以安装以下类型的默认 Velero 插件：**aws**、**azure**、**csi**、**gcp**、**kubevirt** 和 **openshift**。
customPlugins	[]CustomPlugin	用于安装自定义 Velero 插件。 默认的以及自定义的插件信息包括在OADP plug-ins 中
restoreResourcesVersionPriority	string	代表一个配置映射，它在定义与 **EnableAPIGroupVersions** 功能标记结合使用时会被创建。定义此字段会在 Velero 服务器功能标记中添加 **EnableAPIGroupVersions**。
noDefaultBackupLocation	bool	要在没有默认备份存储位置的情况下安装 Velero，您必须设置 **noDefaultBackupLocation** 标志来确认安装。
podConfig	***PodConfig**	定义 **Velero** pod 的配置。
logLevel	string	Velero 服务器日志级别（在最精细的日志中使用 **debug**，对 Velero 默认保留未设置）。有效选项包括 **trace**、**debug**、**info**、**warning**、**error**、**fatal** 和 **panic**。

类型为 **VeleroConfig** 的完整 schema 定义。

表 4.7. CustomPlugin

属性	类型	描述
name	string	自定义插件的名称。
image	string	自定义插件的镜像。

类型 **CustomPlugin** 的完整 schema 定义。

表 4.8. ResticConfig

属性	类型	描述

属性	类型	描述
enable	*bool	如果设置为 **true**，则使用 Restic 启用备份和恢复。如果设置为 **false**，则需要快照。
supplementalGroups	[]int64	定义要应用到 **Restic** pod 的 Linux 组。
timeout	string	定义 Restic 超时的用户提供的持续时间字符串。默认值为 **1hr**（1 小时）。一个代表时间段的字符串，可以是一组十进制数字序列，每个数字都可以带有一个可选的分数及单位后缀，如 **300ms**、-1.5h' 或 **2h45m**。有效时间单位是 **ns**、**us**（或 **µs**）、**ms**、**s**、**m** 和 **h**。
podConfig	***PodConfig**	定义 **Restic** pod 的配置。

类型为 **ResticConfig** 的完整 schema 定义。

表 4.9. PodConfig

属性	类型	描述
nodeSelector	map [string] string	定义要提供给 **Velero podSpec** 或 **Restic podSpec** 的 **nodeSelector**。
容限（tolerations）	[]Toleration	定义要应用到 Velero 部署或 Restic **daemonset** 的容限列表。
resourceAllocations	ResourceRequirements	为一个 **Velero** pod 或 **Restic** pod 设置特定的资源**限值**和**请求**，如设置 Velero CPU 和内存分配所述。
labels	map [string] string	要添加到 pod 的标签。

类型 **PodConfig** 的完整 schema 定义。

表 4.10. 功能

属性	类型	描述
dataMover	*DataMover	定义 Data Mover 的配置。

类型 **Features** 的完整 schema 定义。

表 4.11. DataMover

属性	类型	描述
enable	bool	如果设置为 **true**，请部署卷快照控制器和修改的 CSI Data Mover 插件。如果设置为 **false**，则不会部署它们。
credentialName	string	Data Mover 用户提供的 Restic **Secret** 名称。
timeout	string	要完成 **VolumeSnapshotBackup** 和 **VolumeSnapshotRestore** 的用户提供的持续时间字符串。默认值为 **10m**（10 分钟）。一个代表时间段的字符串，可以是一组十进制数字序列，每个数字都可以带有一个可选的分数及单位后缀，如 **300ms**、-1.5h' 或 **2h45m**。有效时间单位是 **ns**、**us**（或 **µs**）、**ms**、**s**、**m** 和 **h**。

OADP API 在 OADP Operator 中更为详细。

4.14. 高级 OADP 特性和功能

本文档提供有关 OpenShift API for Data Protection (OADP) 的高级功能。

4.14.1. 在同一集群中使用不同的 Kubernetes API 版本

4.14.1.1. 列出集群中的 Kubernetes API 组版本

源集群可能会提供多个 API 版本，其中的一个版本是首选的 API 版本。例如，带有名为 **Example** 的 API 的源集群可能包括在 **example.com/v1** 和 **example.com/v1beta2** API 组中。

如果您使用 Velero 备份和恢复这样的源集群，Velero 仅备份了使用 Kubernetes API 首选版本的该资源的版本。

要返回上例，如果 **example.com/v1** 是首选的 API，则 Velero 只备份使用 **example.com/v1** 的资源的版本。另外，目标集群需要 **example.com/v1** 在它的一组可用 API 资源中注册，以便 Velero 恢复目标集群上的资源。

因此，您需要在目标集群上生成 Kubernetes API 组版本列表，以确保在一组可用的 API 资源中注册了首选的 API 版本。

流程

- 输入以下命令：

```
$ oc api-resources
```

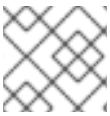
4.14.1.2. 关于启用 API 组版本

默认情况下，Velero 只备份使用 Kubernetes API 的首选版本的资源。但是，Velero 还包括一个启用 API 组版本功能，它解决了这个限制。当在源集群中启用时，这个功能会使 Velero 备份集群中支持的*所有* Kubernetes API 组版本，而不只是首选集群。当版本存储在备份 .tar 文件中被保存后，可以在目标集群上恢复它们。

例如，带有名为 **Example** 的 API 的源集群可能包括在 **example.com/v1** 和 **example.com/v1beta2** API 组中，**example.com/v1** 是首选 API。

如果没有启用 Enable API Group Versions 功能，Velero 仅备份 **Example** 的首选 API 组版本，即 **example.com/v1**。启用该功能后，Velero 还会备份 **example.com/v1beta2**。

当目标集群上启用了"启用 API 组版本"功能时，Velero 根据 API 组版本优先级顺序选择恢复的版本。



注意

启用 API 组版本仍处于测试阶段。

Velero 使用以下算法为 API 版本分配优先级，并将 **1** 作为最高优先级：

1. *destination* 集群的首选版本
2. source_ cluster 的首选版本
3. 带有最高 Kubernetes 版本优先级的通用非首选支持版本

其他资源

- 启用 API 组版本功能

4.14.1.3. 使用启用 API 组版本

您可以使用 Velero 的启用 API 组版本功能来备份集群中支持的*所有* Kubernetes API 组版本，而不只是首选版本。

注意

启用 API 组版本仍处于测试阶段。

流程

- 配置 **EnableAPIGroupVersions** 功能标记：

```
apiVersion: oadp.openshift.io/vialpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      featureFlags:
      - EnableAPIGroupVersions
```

其他资源

- 启用 API 组版本功能

4.14.2. 从一个集群中备份数据，并将其恢复到另一个集群

4.14.2.1. 关于从一个集群中备份数据，并在另一个集群中恢复数据

OpenShift API for Data Protection (OADP) 旨在在同一 OpenShift Container Platform 集群中备份和恢复应用程序数据。MTC (Migration Toolkit for Containers) 旨在将容器（包括应用程序数据）从一个 OpenShift Container Platform 集群迁移到另一个集群。

您可以使用 OADP 从一个 OpenShift Container Platform 集群中备份应用程序数据，并在另一个集群中恢复它。但是，这样做比使用 MTC 或使用 OADP 在同一集群中备份和恢复更为复杂。

要成功使用 OADP 从一个集群备份数据并将其恢复到另一个集群，除了使用 OADP 备份和恢复数据需要的先决条件和步骤外，还需要考虑以下因素：

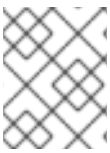
- Operator
- 使用 Velero
- UID 和 GID 范围

4.14.2.1.1. Operator

您必须从应用程序的备份中排除 Operator，以便成功备份和恢复。

4.14.2.1.2. 使用 Velero

Velero （基于 OADP 构建）不支持在云供应商间原生迁移持久性卷快照。要在云平台之间迁移卷快照数据，您*需要*启用 Velero Restic 文件系统备份选项，该选项会在文件系统级别备份卷内容，*或*使用 OADP Data Mover 进行 CSI 快照。



注意

在 OADP 1.1 及更早版本中，Velero Restic 文件系统备份选项被称为 **restic**。在 OADP 1.2 及更高版本中，Velero Restic 文件系统备份选项称为 **file-system-backup**。

- 您还必须使用 Velero 的 文件系统备份 在 AWS 区域或 Microsoft Azure 区域之间迁移数据。
- Velero 不支持将数据恢复到比源集群 *更早的* Kubernetes 版本的集群。

- 在理论上，可以将工作负载迁移到比源*更新的* Kubernetes 版本，但您必须考虑每个自定义资源的集群间 API 组的兼容性。如果 Kubernetes 版本升级会破坏内核或原生 API 组的兼容性，您必须首先更新受影响的自定义资源。

4.14.2.2. 关于确定要备份的 pod 卷

在使用文件系统备份(FSB)启动备份操作前，您必须指定包含要备份的卷的 pod。Velero 将此过程称为"发现"适当的 pod 卷。

Velero 支持两种方法确定 pod 卷：

- **opt-in 方法** ：opt-in 方法要求您主动表示您要包含 - *opt-in* - 一个卷在备份中。您可以通过标记包含要备份的卷的每个 pod 来达到此目的。当 Velero 找到持久性卷 (PV) 时，它会检查挂载卷的 pod。如果 pod 使用卷名称标记，Velero 会备份 pod。

- **opt-out 方法** ：使用 opt-out 方法，您必须主动指定您要从备份中排除卷。为此，您可以标记包含您不想备份的卷的每个 pod。当 Velero 找到 PV 时，它会检查挂载卷的 pod。如果 pod 使用卷的名称标记，Velero 不会备份 pod。

4.14.2.2.1. 限制

- FSB 不支持备份和恢复 **hostpath** 卷。但是 FSB 支持备份和恢复本地卷。

- Velero 对它创建的所有备份存储库使用静态通用加密密钥。**这个静态密钥意味着可以访问备份存储的任何人也可以解密您的备份数据**。务必要限制对备份存储的访问。

- 对于 PVC，每个增量备份链都会在 pod 重新调度之间维护。
  对于*不是* PVC 的 pod 卷，如 **emptyDir** 卷，如果一个 pod 被删除或重新创建（例如，通过 **ReplicaSet** 或一个部署），则这些卷的下一次备份将是完整备份，而不是增量备份。假设 pod 卷的生命周期由其 pod 定义。

- 虽然备份数据可能会以递增方式保存，备份大型文件（如数据库）可能需要很长时间。这是因为 FSB 使用 deduplication 来查找需要备份的区别。

- FSB 通过访问运行该 pod 的节点的文件系统来读取和写入卷中的数据。因此，FSB 只能备份从 pod 挂载的卷，而不直接从 PVC 进行挂载。有些 Velero 用户通过运行一个 staging pod （如 BusyBox 或 Alpine 容器）来解决这个限制，以便在执行 Velero 备份前挂载这些 PVC 和 PV 对。

- FSB 要求将卷挂载到 **<hostPath>/<pod UID>** 下，**<hostPath>** 可以被配置。有些 Kubernetes 系统（如 vCluster）不会在 **<pod UID>** 子目录下挂载卷，VFB 无法按预期工作。

4.14.2.2.2. 使用 opt-in 方法备份 pod 卷

您可以使用 opt-in 方法来指定需要由文件系统备份(FSB)备份哪些卷。您可以使用 **backup.velero.io/backup-volumes** 命令进行此操作。

流程

- 在每个包含您要备份的一个或多个卷的 pod 中，输入以下命令：

```
$ oc -n <your_pod_namespace> annotate pod/<your_pod_name> \
  backup.velero.io/backup-volumes=<your_volume_name_1>, \ <your_volume_name_2>>,...,
<your_volume_name_n>
```

  其中：

<your_volume_name_x>

指定 pod 规格中 xth 卷的名称。

4.14.2.2.3. 使用 opt-out 方法备份 pod 卷

使用 opt-out 方法时，所有 pod 卷都使用文件系统备份(FSB)备份，但有一些例外：

- 挂载默认服务帐户令牌、secret 和配置映射的卷。
- **hostPath** 卷

您可以使用 opt-out 方法指定**不要**备份的卷。您可以使用 **backup.velero.io/backup-volumes-excludes** 命令进行此操作。

流程

- 在包含您不想备份的一个或多个卷的 pod 中，运行以下命令：

```
$ oc -n <your_pod_namespace> annotate pod/<your_pod_name> \
  backup.velero.io/backup-volumes-excludes=<your_volume_name_1>, \
<your_volume_name_2>>,...,<your_volume_name_n>
```

其中：

<your_volume_name_x>

指定 pod 规格中 xth 卷的名称。

注意

您可以使用 **--default-volumes-to-fs-backup** 标志运行 **velero install** 命令，为所有 Velero 备份启用此行为。

4.14.2.3. UID 和 GID 范围

如果您从一个集群备份数据并将其恢复到另一个集群，则可能会出现 UID （用户 ID）和 GID （组 ID）范围的问题。下面的部分解释了这些潜在问题和缓解措施：

问题概述

命名空间 UID 和 GID 范围可能会因目标集群而异。OADP 不会备份和恢复 OpenShift UID 范围元数据。如果备份的应用程序需要特定的 UID，请确保范围是可用的。如需有关 OpenShift 的 UID 和 GID 范围的更多信息，请参阅 OpenShift 和 UID 的指南 。

问题详细描述

当您使用 **oc create namespace** 在 OpenShift Container Platform 中创建命名空间时，OpenShift Container Platform 会为命名空间分配一个唯一用户 ID (UID) 范围，即 Supplemental Group (GID)范围和唯一的 SELinux MCS 标签。此信息存储在集群的 **metadata.annotations** 字段中。此信息是安全性上下文约束(SCC)注解的一部分，它由以下组件组成：

- **openshift.io/sa.scc.mcs**
- **openshift.io/sa.scc.supplemental-groups**
- **openshift.io/sa.scc.uid-range**

当使用 OADP 恢复命名空间时，它会自动使用 **metadata.annotations** 中的信息，而无需为目标集群重置它。因此，如果满足以下条件，工作负载可能无法访问备份的数据：

- 存在一个带有其他 SCC 注解的现有命名空间，例如在另一个集群中。在这种情况下，OADP 在备份过程中使用现有命名空间，而不是您要恢复的命名空间。
- 备份过程中使用了标签选择器，但执行工作负载的命名空间没有标签。在这种情况下，OADP 不会备份命名空间，而是在恢复过程中创建一个新的命名空间，该命名空间不包含备份命名空间的注解。这会导致为命名空间分配一个新的 UID 范围。
  如果 OpenShift Container Platform 根据从持久性卷数据备份时更改的命名空间注解为 pod 为 **securityContext** UID，则可能会出现问题。
- 容器 UID 不再与文件所有者的 UID 匹配。
- 发生错误，因为 OpenShift Container Platform 没有修改目标集群的 UID 范围，以匹配备份集群的数据。因此，备份集群与目标集群的 UID 不同，这意味着应用程序无法向目标集群读取或写入数据。

  缓解方案

  您可以使用以下一个或多个缓解方案来解决 UID 和 GID 范围问题：

- 简单的缓解方案：
  - 如果您在 **Backup** CR 中使用标签选择器过滤要包含在备份中的对象，请确保将此标签选择器添加到包含工作区的命名空间中。
  - 在尝试恢复具有相同名称的命名空间前，请删除目标集群上任何已存在的命名空间版本。
- 高级缓解方案：
  - 迁移后，通过在 OpenShift 命名空间中解决重叠的 UID 范围来修复 UID 范围。第 1 步是可选的。

有关 OpenShift Container Platform 中 UID 和 GID 范围的详细讨论，重点放在一个集群中备份数据并在另一个集群中恢复数据时出现问题，请参阅 OpenShift 和 UID 的指南 。

4.14.2.4. 从一个集群中备份数据，并将其恢复到另一个集群

通常，您可以从一个 OpenShift Container Platform 集群备份数据，并以与将数据备份并恢复到同一集群的方式在另一个 OpenShift Container Platform 集群上恢复数据。但是，从一个 OpenShift Container Platform 集群备份数据时，会有一些额外的前提条件和不同之处，并在另一个集群中恢复它。

先决条件

- 所有在平台上备份和恢复的相关先决条件（如 AWS、Microsoft Azure、GCP 等），特别是数据保护应用程序(DPA) 的先决条件。

流程

- 在为您的平台提供的流程中添加以下内容：
  - 确保备份存储位置 (BSL) 和卷快照位置具有相同的名称和路径，以将资源恢复到另一个集群。
  - 在集群间共享相同的对象存储位置凭证。
  - 为获得最佳结果，请使用 OADP 在目标集群中创建命名空间。

- 如果您使用 Velero **file-system-backup** 选项，请运行以下命令启用 **--default-volumes-to-fs-backup** 标志以便在备份过程中使用：

```
$ velero backup create <backup_name> --default-volumes-to-fs-backup
<any_other_options>
```

注意

在 OADP 1.2 及更高版本中，Velero Restic 选项名为 **file-system-backup**。

4.14.3. OADP 存储类映射

4.14.3.1. 存储类映射

存储类映射允许您定义规则或策略，指定应将哪些存储类应用到不同类型的数据。此功能根据访问频率、数据重要性和成本注意事项自动确定存储类的过程。它通过确保数据存储在最合适的存储类中用于其特征和使用模式来优化存储效率和经济性。

您可以使用 **change-storage-class-config** 字段更改数据对象的存储类，这可让您通过在不同存储层之间移动数据（如从标准到归档存储）来优化成本和性能，例如，根据您的需要和访问模式来优化成本和性能。

4.14.3.1.1. 使用 MTC 的存储类映射

您可以使用 MTC 将容器（包括应用程序数据）从一个 OpenShift Container Platform 集群迁移到另一个集群以及存储类映射和转换。您可以在同一个集群中迁移持久性卷(PV)的存储类来转换它。要做到这一点，您必须在 MTC web 控制台中创建并运行迁移计划。

4.14.3.1.2. 使用 OADP 映射存储类

您可以在 Velero 插件 v1.1.0 及之后的版本中使用 OpenShift API 进行数据保护 (OADP)，在恢复过程中更改持久性卷 (PV) 的存储类，方法是在 Velero 命名空间中的配置映射中配置存储类映射。

要使用 OADP 部署 ConfigMap，请使用 **change-storage-class-config** 字段。您必须根据您的云供应商更改存储类映射。

流程

1. 运行以下命令来更改存储类映射：

```
$ cat change-storageclass.yaml
```

2. 在 Velero 命名空间中创建配置映射，如下例所示：

Example

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: change-storage-class-config
  namespace: openshift-adp
  labels:
    velero.io/plugin-config: ""
```

```
    velero.io/change-storage-class: RestoreItemAction
data:
  standard-csi: ssd-csi
```

3. 运行以下命令保存存储类映射首选项：

```
$ oc create -f change-storage-class-config
```

4.14.4. 其他资源

- 在同一集群中使用不同的 Kubernetes API 版本。
- 将 Data Mover 用于 CSI 快照。
- 使用文件系统备份备份应用程序：Kopia 或 Restic。
- 迁移转换存储类。

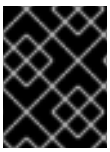
第 5 章 CONTROL PLANE 备份和恢复

5.1. 备份 ETCD

etcd 是 OpenShift Container Platform 的以"键-值"形式进行的存储，它会保留所有资源对象的状态。

定期备份集群的 etcd 数据，并在 OpenShift Container Platform 环境以外的安全位置保存备份数据。不要在第一个证书轮转完成前（安装后的 24 小时内）进行 etcd 备份，否则备份将包含过期的证书。另外，建议您在非高峰期使用 etcd 备份，因为 etcd 快照有较高的 I/O 成本。

确保升级集群后执行 etcd 备份。这很重要，因为当恢复集群时，必须使用从同一 z-stream 发行版本中获取的 etcd 备份。例如，OpenShift Container Platform 4.y.z 集群必须使用从 4.y.z 中获得的 etcd 备份。



重要

通过在 control plane 主机上执行一次备份脚本来备份集群的 etcd 数据。不要为每个 control plane 主机进行备份。

在进行了 etcd 备份后，就可以恢复到一个以前的集群状态。

5.1.1. 备份 etcd 数据

按照以下步骤，通过创建 etcd 快照并备份静态 pod 的资源来备份 etcd 数据。这个备份可以被保存，并在以后需要时使用它来恢复 etcd 数据。

重要

只保存单一 control plane 主机的备份。不要从集群中的每个 control plane 主机进行备份。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 您已检查是否启用了集群范围代理。

提示

您可以通过查看 **oc get proxy cluster -o yaml** 的输出来检查代理是否已启用。如果 **httpProxy**、**httpsProxy**和 **noProxy** 字段设置了值，则会启用代理。

流程

1. 以 root 用户身份为 control plane 节点启动一个 debug 会话：

```
$ oc debug --as-root node/<node_name>
```

2. 在 debug shell 中将根目录改为 **/host** ：

```
sh-4.4# chroot /host
```

3. 如果启用了集群范围的代理，请确定已导出了 **NO_PROXY**、**HTTP_PROXY**和 **HTTPS_PROXY** 环境变量。

4. 在 debug shell 中运行 **cluster-backup.sh** 脚本，并传递保存备份的位置。

提示

cluster-backup.sh 脚本作为 etcd Cluster Operator 的一个组件被维护，它是 **etcdctl snapshot save** 命令的包装程序（wrapper）。

```
sh-4.4# /usr/local/bin/cluster-backup.sh /home/core/assets/backup
```

脚本输出示例

```
found latest kube-apiserver: /etc/kubernetes/static-pod-resources/kube-apiserver-pod-6
found latest kube-controller-manager: /etc/kubernetes/static-pod-resources/kube-controller-manager-pod-7
found latest kube-scheduler: /etc/kubernetes/static-pod-resources/kube-scheduler-pod-6
found latest etcd: /etc/kubernetes/static-pod-resources/etcd-pod-3
ede95fe6b88b87ba86a03c15e669fb4aa5bf0991c180d3c6895ce72eaade54a1
etcdctl version: 3.4.14
API version: 3.4
{"level":"info","ts":1624647639.0188997,"caller":"snapshot/v3_snapshot.go:119","msg":"created temporary db file","path":"/home/core/assets/backup/snapshot_2021-06-25_190035.db.part"}
{"level":"info","ts":"2021-06-25T19:00:39.030Z","caller":"clientv3/maintenance.go:200","msg":"opened snapshot stream; downloading"}
{"level":"info","ts":1624647639.0301006,"caller":"snapshot/v3_snapshot.go:127","msg":"fetching snapshot","endpoint":"https://10.0.0.5:2379"}
{"level":"info","ts":"2021-06-25T19:00:40.215Z","caller":"clientv3/maintenance.go:208","msg":"completed snapshot read; closing"}
{"level":"info","ts":1624647640.6032252,"caller":"snapshot/v3_snapshot.go:142","msg":"fetched snapshot","endpoint":"https://10.0.0.5:2379","size":"114 MB","took":1.584090459}
{"level":"info","ts":1624647640.6047094,"caller":"snapshot/v3_snapshot.go:152","msg":"saved","path":"/home/core/assets/backup/snapshot_2021-06-25_190035.db"}
Snapshot saved at /home/core/assets/backup/snapshot_2021-06-25_190035.db
{"hash":3866667823,"revision":31407,"totalKey":12828,"totalSize":114446336}
snapshot db and kube resources are successfully saved to /home/core/assets/backup
```

在这个示例中，在 control plane 主机上的 **/home/core/assets/backup/** 目录中创建了两个文件：

- **snapshot_<datetimestamp>.db**：这个文件是 etcd 快照。**cluster-backup.sh** 脚本确认其有效。
- **static_kuberesources_<datetimestamp>.tar.gz**：此文件包含静态 pod 的资源。如果启用了 etcd 加密，它也包含 etcd 快照的加密密钥。

注意

如果启用了 etcd 加密，建议出于安全考虑，将第二个文件与 etcd 快照分开保存。但是，需要这个文件才能从 etcd 快照中进行恢复。

请记住，etcd 仅对值进行加密，而不对键进行加密。这意味着资源类型、命名空间和对象名称是不加密的。

5.1.2. 其他资源

- 恢复不健康的 etcd 集群

5.1.3. 创建自动的 etcd 备份

etcd 的自动备份功能支持重复备份和单一备份。重复备份会创建一个 cron 作业，该作业在每次作业触发时都启动一次备份。

重要

自动 etcd 备份是一个技术预览功能。技术预览功能不受红帽产品服务等级协议（SLA）支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅技术预览功能支持范围。

按照以下步骤为 etcd 启用自动备份。

警告

在集群中启用 **TechPreviewNoUpgrade** 功能集可防止次版本更新。**TechPreviewNoUpgrade** 功能集无法被禁用。不要在生产环境集群中启用此功能。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 您可以访问 OpenShift CLI(**oc**)。

流程

1. 使用以下内容创建名为 **enable-tech-preview-no-upgrade.yaml** 的 **FeatureGate** 自定义资源 (CR) 文件：

```
apiVersion: config.openshift.io/v1
kind: FeatureGate
metadata:
  name: cluster
spec:
  featureSet: TechPreviewNoUpgrade
```

2. 应用 CR 并启用自动备份：

```
$ oc apply -f enable-tech-preview-no-upgrade.yaml
```

3. 启用相关的 API 需要一些时间。运行以下命令，验证自定义资源定义 (CRD) 的创建：

```
$ oc get crd | grep backup
```

输出示例

```
backups.config.openshift.io 2023-10-25T13:32:43Z
etcdbackups.operator.openshift.io 2023-10-25T13:32:04Z
```

5.1.3.1. 创建单个 etcd 备份

按照以下步骤，通过创建并应用自定义资源 (CR) 来创建单个 etcd 备份。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 您可以访问 OpenShift CLI(**oc**)。

流程

- 如果动态置备的存储可用，请完成以下步骤以创建单个自动 etcd 备份：
  a. 创建名为 **etcd-backup-pvc.yaml** 的持久性卷声明 (PVC)，其内容如下：

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
  namespace: openshift-etcd
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 200Gi 1
  volumeMode: Filesystem
```

1 PVC 可用的存储量。根据您的要求调整这个值。

  b. 运行以下命令来应用 PVC：

```
$ oc apply -f etcd-backup-pvc.yaml
```

  c. 运行以下命令验证 PVC 的创建：

```
$ oc get pvc
```

输出示例

```
NAME              STATUS    VOLUME   CAPACITY   ACCESS MODES
STORAGECLASS   AGE
etcd-backup-pvc   Bound                                      51s
```

注意

动态 PVC 处于 **Pending** 状态，直到它们被挂载为止。

d. 创建名为 **etcd-single-backup.yaml** 的 CR 文件，其内容如下：

```
apiVersion: operator.openshift.io/v1alpha1
kind: EtcdBackup
metadata:
  name: etcd-single-backup
  namespace: openshift-etcd
spec:
  pvcName: etcd-backup-pvc (1)
```

(1) 保存备份的 PVC 名称。根据您的环境调整这个值。

e. 应用 CR 以启动单个备份：

```
$ oc apply -f etcd-single-backup.yaml
```

- 如果动态置备的存储不可用，请完成以下步骤来创建单个自动 etcd 备份：

a. 创建名为 **etcd-backup-local-storage.yaml** 的 **StorageClass** CR 文件，其内容如下：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: etcd-backup-local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: Immediate
```

b. 运行以下命令来应用 **StorageClass** CR：

```
$ oc apply -f etcd-backup-local-storage.yaml
```

c. 创建名为 **etcd-backup-pv-fs.yaml** 的 PV，其内容类似以下示例：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: etcd-backup-pv-fs
spec:
  capacity:
    storage: 100Gi (1)
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: etcd-backup-local-storage
  local:
    path: /mnt
  nodeAffinity:
    required:
```

```
      nodeSelectorTerms:
      - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
        - <example_master_node> 2
```

1 PV 可用的存储量。根据您的要求调整这个值。

2 将此值替换为将此 PV 附加到的节点。

d. 运行以下命令验证 PV 的创建：

```
$ oc get pv
```

输**出示例**

```
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
CLAIM   STORAGECLASS            REASON   AGE
etcd-backup-pv-fs   100Gi      RWO            Retain           Available           etcd-backup-
local-storage           10s
```

e. 创建名为 **etcd-backup-pvc.yaml** 的 PVC，其内容如下：

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
  namespace: openshift-etcd
spec:
  accessModes:
  - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 10Gi 1
```

1 PVC 可用的存储量。根据您的要求调整这个值。

f. 运行以下命令来应用 PVC：

```
$ oc apply -f etcd-backup-pvc.yaml
```

g. 创建名为 **etcd-single-backup.yaml** 的 CR 文件，其内容如下：

```
apiVersion: operator.openshift.io/v1alpha1
kind: EtcdBackup
metadata:
  name: etcd-single-backup
  namespace: openshift-etcd
spec:
  pvcName: etcd-backup-pvc 1
```

1 将备份保存到的持久性卷声明 (PVC) 的名称。根据您的环境调整这个值。

h. 应用 CR 以启动单个备份：

```
$ oc apply -f etcd-single-backup.yaml
```

5.1.3.2. 创建重复的 etcd 备份

按照以下步骤创建自动重复备份 etcd。

如果可能，使用动态置备的存储将创建的 etcd 备份数据保存在安全的外部位置。如果动态置备的存储不可用，请考虑将备份数据存储在 NFS 共享上，以便更易访问备份恢复。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 您可以访问 OpenShift CLI(**oc**)。

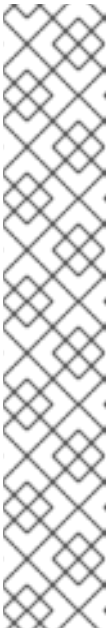
流程

1. 如果动态置备的存储可用，请完成以下步骤来创建自动重复备份：

   a. 创建名为 **etcd-backup-pvc.yaml** 的持久性卷声明 (PVC)，其内容如下：

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
  namespace: openshift-etcd
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 200Gi 1
  volumeMode: Filesystem
  storageClassName: etcd-backup-local-storage
```

1 PVC 可用的存储量。根据您的要求调整这个值。



注意

每个供应商都需要更改 **accessModes** 和 **storageClassName** 密钥：

供应商	accessModes 值	storageClassName value
带有 **versioned-installer-efc_operator-ci** 配置集的 AWS	**- ReadWriteMany**	**efs-sc**
Google Cloud Platform	**- ReadWriteMany**	**filestore-csi**
Microsoft Azure	**- ReadWriteMany**	**azurefile-csi**

b. 运行以下命令来应用 PVC：

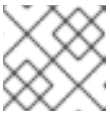
```
$ oc apply -f etcd-backup-pvc.yaml
```

c. 运行以下命令验证 PVC 的创建：

```
$ oc get pvc
```

输出示例

```
NAME              STATUS    VOLUME   CAPACITY   ACCESS MODES
STORAGECLASS   AGE
etcd-backup-pvc   Bound                                      51s
```



注意

动态 PVC 处于 **Pending** 状态，直到它们被挂载为止。

2. 如果动态置备的存储不可用，请完成以下步骤来创建本地存储 PVC：

警告

如果您删除或丢失对包含存储备份数据的节点的访问，可能会丢失数据。

a. 创建名为 **etcd-backup-local-storage.yaml** 的 **StorageClass** CR 文件，其内容如下：

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
```

```
  name: etcd-backup-local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: Immediate
```

b. 运行以下命令来应用 **StorageClass** CR：

```
$ oc apply -f etcd-backup-local-storage.yaml
```

c. 从应用的 **StorageClass** 中创建一个名为 **etcd-backup-pv-fs.yaml** 的 PV，其内容类似以下示例：

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: etcd-backup-pv-fs
spec:
  capacity:
    storage: 100Gi (1)
  volumeMode: Filesystem
  accessModes:
  - ReadWriteMany
  persistentVolumeReclaimPolicy: Delete
  storageClassName: etcd-backup-local-storage
  local:
    path: /mnt/
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
        - key: kubernetes.io/hostname
          operator: In
          values:
          - <example_master_node> (2)
```

(1) PV 可用的存储量。根据您的要求调整这个值。

(2) 将这个值替换为要附加此 PV 的 master 节点。

提示

运行以下命令来列出可用的节点：

```
$ oc get nodes
```

d. 运行以下命令验证 PV 的创建：

```
$ oc get pv
```

输出示例

```
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
CLAIM   STORAGECLASS              REASON   AGE
```

```
etcd-backup-pv-fs    100Gi    RWX          Delete          Available          etcd-backup-
local-storage          10s
```

e. 创建名为 **etcd-backup-pvc.yaml** 的 PVC，其内容如下：

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
spec:
  accessModes:
  - ReadWriteMany
  volumeMode: Filesystem
  resources:
    requests:
      storage: 10Gi (1)
  storageClassName: etcd-backup-local-storage
```

(1) PVC 可用的存储量。根据您的要求调整这个值。

f. 运行以下命令来应用 PVC：

```
$ oc apply -f etcd-backup-pvc.yaml
```

3. 创建名为 **etcd-recurring-backups.yaml** 的自定义资源定义 (CRD) 文件。创建的 CRD 的内容定义自动备份的调度和保留类型。
对于带有 15 个保留备份的 **RetentionNumber** 的默认保留类型，请使用类似以下示例的内容：

```
apiVersion: config.openshift.io/v1alpha1
kind: Backup
metadata:
  name: etcd-recurring-backup
spec:
  etcd:
    schedule: "20 4 * * *" (1)
    timeZone: "UTC"
    pvcName: etcd-backup-pvc
```

(1) 用于重复备份的 **CronTab** 调度。根据您的需要调整这个值。

要使用基于最大备份数的保留，请在 **etcd** 键中添加以下键值对：

```
spec:
  etcd:
    retentionPolicy:
      retentionType: RetentionNumber (1)
      retentionNumber:
        maxNumberOfBackups: 5 (2)
```

(1) 保留类型。如果未指定，则默认为 **RetentionNumber**。

(2) 要保留的最大备份数量。根据您的需要调整这个值。如果未指定，则默认为保留 15 个备份。

警告

已知问题会导致保留备份的数量大于配置的值。

要根据备份的文件大小保留，请使用：

```
spec:
  etcd:
    retentionPolicy:
      retentionType: RetentionSize
      retentionSize:
        maxSizeOfBackupsGb: 20 1
```

1 以 GB 为单位保留备份的最大文件大小。根据您的需要调整这个值。如果未指定，则默认为 10 GB。

警告

已知问题会导致保留备份的最大大小超过配置的值 10 GB。

4. 运行以下命令，创建 CRD 定义的 cron 作业：

```
$ oc create -f etcd-recurring-backup.yaml
```

5. 要查找创建的 cron 任务，请运行以下命令：

```
$ oc get cronjob -n openshift-etcd
```

5.2. 替换不健康的 ETCD 成员

本文档描述了替换一个不健康 etcd 成员的过程。

此过程取决于 etcd 成员不健康的原因，如机器没有运行，或节点未就绪，或 etcd pod 处于 crashlooping 状态。

注意

如果您丢失了大多数 control plane 主机，请按照灾难恢复流程恢复到以前的一个集群状态，而不是这个过程。

如果 control plane 证书在被替换的成员中无效，则必须遵循从已过期 control plane 证书中恢复的步骤，而不是此过程。

如果 control plane 节点丢失并且创建了一个新节点，etcd 集群 Operator 将处理生成新 TLS 证书并将节点添加为 etcd 成员。

5.2.1. 先决条件

- 在替换不健康的 etcd 成员，需要进行 etcd 备份。

5.2.2. 找出一个不健康的 etcd 成员

您可以识别集群是否有不健康的 etcd 成员。

先决条件

- 使用具有 **cluster-admin** 角色的用户访问集群。

流程

1. 使用以下命令检查 **EtcdMembersAvailable** 状态条件的状态：

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="EtcdMembersAvailable")]}{.message}{"\n"}'
```

2. 查看输出：

```
2 of 3 members are available, ip-10-0-131-183.ec2.internal is unhealthy
```

这个示例输出显示 **ip-10-0-131-183.ec2.internal** etcd 成员不健康。

5.2.3. 确定不健康的 etcd 成员的状态

替换不健康 etcd 成员的步骤取决于 etcd 的以下状态：

- 机器没有运行或者该节点未就绪
- etcd pod 处于 crashlooping 状态

此流程决定了 etcd 成员处于哪个状态。这可让您了解替换不健康的 etcd 成员要遵循的步骤。

注意

如果您知道机器没有运行或节点未就绪，但它们应该很快返回健康状态，那么您就不需要执行替换 etcd 成员的流程。当机器或节点返回一个健康状态时，etcd cluster Operator 将自动同步。

先决条件

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 您已找到不健康的 etcd 成员。

流程

1. 检查 **机器是否没有运行**:

```
$ oc get machines -A -ojsonpath='{range .items[*]}{@.status.nodeRef.name}{"\t"}
{@.status.providerStatus.instanceState}{"\n"}' | grep -v running
```

输**出示例**

```
ip-10-0-131-183.ec2.internal  stopped (1)
```

(1) 此输出列出了节点以及节点机器的状态。如果状态不是 **running**，则**代表机器没有运行**。

如果**机器没有运行**，按照 *替换机器没有运行或节点没有就绪的非健康 etcd 成员*过程进行操作。

2. 确定 **节点是否未就绪**。
如果以下任何一种情况是正确的，则代表**节点没有就绪**。

- 如果机器正在运行，检查节点是否不可访问：

```
$ oc get nodes -o jsonpath='{range .items[*]}{"\n"}{.metadata.name}{"\t"}{range
.spec.taints[*]}{.key}{" "}' | grep unreachable
```

输**出示例**

```
ip-10-0-131-183.ec2.internal node-role.kubernetes.io/master
node.kubernetes.io/unreachable node.kubernetes.io/unreachable (1)
```

(1) 如果节点带有 **unreachable** 污点，则**节点没有就绪**。

- 如果该节点仍然可访问，则检查该节点是否列为 **NotReady**:

```
$ oc get nodes -l node-role.kubernetes.io/master | grep "NotReady"
```

输**出示例**

```
ip-10-0-131-183.ec2.internal   NotReady   master   122m   v1.29.4 (1)
```

(1) 如果节点列表为 **NotReady**，则 **该节点没有就绪**。

如果**节点没有就绪**，按照 *替换机器没有运行或节点没有就绪的 etcd 成员*的步骤进行操作。

3. 确定 **etcd Pod 是否处于 crashlooping 状态**。
如果机器正在运行并且节点已就绪，请检查 etcd pod 是否处于 crashlooping 状态。

a. 验证所有 control plane 节点都列为 **Ready** ：

```
$ oc get nodes -l node-role.kubernetes.io/master
```

输出示例

```
NAME                          STATUS   ROLES    AGE     VERSION
ip-10-0-131-183.ec2.internal  Ready    master   6h13m   v1.29.4
ip-10-0-164-97.ec2.internal   Ready    master   6h13m   v1.29.4
ip-10-0-154-204.ec2.internal  Ready    master   6h13m   v1.29.4
```

b. 检查 etcd pod 的状态是否为 **Error** 或 **CrashLoopBackOff**:

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

输出示例

```
etcd-ip-10-0-131-183.ec2.internal   2/3   Error     7   6h9m  1
etcd-ip-10-0-164-97.ec2.internal    3/3   Running   0   6h6m
etcd-ip-10-0-154-204.ec2.internal   3/3   Running   0   6h6m
```

1 由于此 pod 的状态是 **Error**，因此 **etcd pod 为 crashlooping 状态**。

如果 **etcd pod 为 crashlooping 状态**，请按照 *替换 etcd pod 处于 crashlooping 状态的不健康的 etcd 成员*的步骤进行操作。

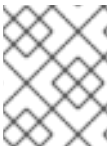
5.2.4. 替换不健康的 etcd 成员

根据不健康的 etcd 成员的状态，使用以下一个流程：

- 替换机器没有运行或节点未就绪的不健康 etcd 成员
- 替换其 etcd Pod 处于 crashlooping 状态的不健康 etcd 成员
- 替换不健康的裸机 etcd 成员

5.2.4.1. 替换机器没有运行或节点未就绪的不健康 etcd 成员

此流程详细介绍了替换因机器没有运行或节点未就绪造成不健康的 etcd 成员的步骤。



注意

如果您的集群使用 control plane 机器集，请参阅 "对控制平面机器集进行故障排除" 中的"恢复降级的 etcd Operator"。

先决条件

- 您已找出不健康的 etcd 成员。
- 您已确认机器没有运行，或者该节点未就绪。

重要

您必须等待其他 control plane 节点关闭。control plane 节点必须保持关闭状态，直到替换完不健康的 etcd 成员为止。

- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 已进行 etcd 备份。

重要

执行此流程前务必要进行 etcd 备份，以便在遇到任何问题时可以恢复集群。

流程

1. 删除不健康的成员。
   a. 选择一个*不在*受影响节点上的 pod:
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

   输出示例

```
etcd-ip-10-0-131-183.ec2.internal             3/3     Running     0          123m
etcd-ip-10-0-164-97.ec2.internal              3/3     Running     0          123m
etcd-ip-10-0-154-204.ec2.internal             3/3     Running     0          124m
```

   b. 连接到正在运行的 etcd 容器，传递没有在受影响节点上的 pod 的名称：
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

   c. 查看成员列表：

```
sh-4.2# etcdctl member list -w table
```

   输出示例

```
+------------------+---------+-----------------------------+---------------------------+---------------
-----------+
|        ID        | STATUS  |            NAME             |        PEER ADDRS         |      CLIENT
ADDRS       |
+------------------+---------+-----------------------------+---------------------------+---------------
-----------+
| 6fc1e7c9db35841d | started | ip-10-0-131-183.ec2.internal | https://10.0.131.183:2380 |
https://10.0.131.183:2379 |
| 757b6793e2408b6c | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| ca8c2990a0aa29d1 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380 |
https://10.0.154.204:2379 |
+------------------+---------+-----------------------------+---------------------------+---------------
-----------+
```

记录不健康的 etcd 成员的 ID 和名称，因为稍后需要这些值。**$ etcdctl endpoint health** 命令将列出已删除的成员，直到完成替换过程并添加了新成员。

d. 通过向 **etcdctl member remove** 命令提供 ID 来删除不健康的 etcd 成员 :

```
sh-4.2# etcdctl member remove 6fc1e7c9db35841d
```

输**出示例**

```
Member 6fc1e7c9db35841d removed from cluster ead669ce1fbfb346
```

e. 再次查看成员列表，并确认成员已被删除：

```
sh-4.2# etcdctl member list -w table
```

输**出示例**

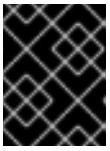
```
+------------------+---------+-----------------------------+--------------------------+----------------
-----------+
|        ID        | STATUS  |            NAME             |        PEER ADDRS        |      CLIENT
ADDRS       |
+------------------+---------+-----------------------------+--------------------------+----------------
-----------+
| 757b6793e2408b6c | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| ca8c2990a0aa29d1 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380 |
https://10.0.154.204:2379 |
+------------------+---------+-----------------------------+--------------------------+----------------
-----------+
```

现在您可以退出节点 shell。

2. 输入以下命令关闭仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

此命令可确保您可以成功重新创建机密并推出静态 pod。



重要

关闭仲裁保护后，在剩余的 etcd 实例进行重启以使配置改变生效期间，集群可能无法访问。

注意

在只使用两个成员运行时，etcd 将无法容忍任何成员失败。重启剩余的成员会破坏仲裁，并导致集群出现停机问题。由于可能导致停机的配置更改，仲裁保护可以防止 etcd 重启，因此必须禁用它才能完成这个过程。

3. 运行以下命令来删除受影响的节点：

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

4. 删除已删除的不健康 etcd 成员的旧 secret。

   a. 列出已删除的不健康 etcd 成员的 secret。

   ```
   $ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
   ```

   1 传递您之前在这个过程中记录的不健康 etcd 成员的名称。

   有一个对等的、服务和指标的 secret，如以下输出所示：

   输出示例

   ```
   etcd-peer-ip-10-0-131-183.ec2.internal             kubernetes.io/tls                     2      47m
   etcd-serving-ip-10-0-131-183.ec2.internal          kubernetes.io/tls                     2      47m
   etcd-serving-metrics-ip-10-0-131-183.ec2.internal  kubernetes.io/tls                     2
   47m
   ```

   b. 删除已删除的不健康 etcd 成员的 secret。

      i. 删除 peer（对等）secret:

      ```
      $ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
      ```

      ii. 删除 serving secret:

      ```
      $ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
      ```

      iii. 删除 metrics secret:

      ```
      $ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-
      183.ec2.internal
      ```

5. 删除并重新创建 control plane 机器。重新创建此机器后，会强制一个新修订版本并自动扩展 etcd。
   如果您正在运行安装程序置备的基础架构，或者您使用 Machine API 创建机器，请按照以下步骤执行。否则，您必须使用最初创建 master 时使用的相同方法创建新的 master。

   a. 获取不健康成员的机器。
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

   ```
   $ oc get machines -n openshift-machine-api -o wide
   ```

   输出示例

   ```
   NAME                                PHASE     TYPE        REGION      ZONE         AGE
   NODE                          PROVIDERID                          STATE
   clustername-8qw5l-master-0          Running   m4.xlarge   us-east-1   us-east-1a
   3h37m   ip-10-0-131-183.ec2.internal   aws:///us-east-1a/i-0ec2782f8287dfb7e   stopped
   1
   clustername-8qw5l-master-1          Running   m4.xlarge   us-east-1   us-east-1b
   3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
   ```

```
clustername-8qw5l-master-2                 Running   m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-east-
1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b
3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-
1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a
running
```

1 这是不健康节点的 control plane 机器 **ip-10-0-131-183.ec2.internal**。

b. 将机器配置保存到文件系统中的一个文件中：

```
$ oc get machine clustername-8qw5l-master-0 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1 为不健康的节点指定 control plane 机器的名称。

c. 编辑上一步中创建的 **new-master-machine.yaml** 文件，以分配新名称并删除不必要的字段。

i. 删除整个 **status** 部分：

```
status:
  addresses:
  - address: 10.0.131.183
    type: InternalIP
  - address: ip-10-0-131-183.ec2.internal
    type: InternalDNS
  - address: ip-10-0-131-183.ec2.internal
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Node
    name: ip-10-0-131-183.ec2.internal
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: awsproviderconfig.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: AWSMachineProviderStatus
```

ii. 将 **metadata.name** 字段更改为新名称。
建议您保留与旧机器相同的基础名称，并将结束号码改为下一个可用数字。在本例中，**clustername-8qw5l-master-0** 改为 **clustername-8qw5l-master-3**。

例如：

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. 删除 **spec.providerID** 字段：

```
  providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

d. 删除不健康成员的机器：

```
$ oc delete machine -n openshift-machine-api clustername-8qw5l-master-0 1
```

1 为不健康的节点指定 control plane 机器的名称。

e. 验证机器是否已删除：

```
$ oc get machines -n openshift-machine-api -o wide
```

输**出示例**

```
NAME                                PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1          Running   m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running   m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-east-
1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b
3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-
1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a
running
```

f. 使用 **new-master-machine.yaml** 文件创建新机器：

```
$ oc apply -f new-master-machine.yaml
```

g. 验证新机器是否已创建：

```
$ oc get machines -n openshift-machine-api -o wide
```

输出示例

```
NAME                                 PHASE          TYPE        REGION      ZONE       AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1           Running        m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2           Running        m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-master-3           Provisioning   m4.xlarge   us-east-1   us-east-1a
85s     ip-10-0-133-53.ec2.internal    aws:///us-east-1a/i-015b0888fe17bc2c8   running
(1)
clustername-8qw5l-worker-us-east-1a-wbtgd   Running        m4.large    us-east-1   us-
east-1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running        m4.large    us-east-1   us-east-
1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b
running
clustername-8qw5l-worker-us-east-1c-pkg26   Running        m4.large    us-east-1   us-
east-1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a
running
```

(1) 新机器 **clustername-8qw5l-master-3** 将被创建，并当阶段从 **Provisioning** 变为 **Running** 后就可以使用。

创建新机器可能需要几分钟时间。当机器或节点返回一个健康状态时，etcd cluster Operator 将自动同步。

6. 输入以下命令重新打开仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

7. 您可以输入以下命令验证 **unsupportedConfigOverrides** 部分是否已从对象中删除：

```
$ oc get etcd/cluster -oyaml
```

8. 如果使用单节点 OpenShift，请重启该节点。否则，您可能会在 etcd 集群 Operator 中遇到以下错误：

输出示例

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-
0": the object has been modified; please apply your changes to the latest version and try
again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been
modified; please apply your changes to the latest version and try again, Operation cannot be
fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply
your changes to the latest version and try again]
```

验证

1. 验证所有 etcd pod 是否都正常运行。
在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

输出示例

```
etcd-ip-10-0-133-53.ec2.internal             3/3     Running     0          7m49s
etcd-ip-10-0-164-97.ec2.internal             3/3     Running     0          123m
etcd-ip-10-0-154-204.ec2.internal            3/3     Running     0          124m
```

如果上一命令的输出只列出两个 pod，您可以手动强制重新部署 etcd。在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge (1)
```

(1) **forceRedeploymentReason** 值必须是唯一的，这就是为什么附加时间戳的原因。

2. 验证只有三个 etcd 成员。

   a. 连接到正在运行的 etcd 容器，传递没有在受影响节点上的 pod 的名称：
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

   ```
   $ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
   ```

   b. 查看成员列表：

   ```
   sh-4.2# etcdctl member list -w table
   ```

   输出示例

   ```
   +------------------+---------+----------------------------+---------------------------+---------------
   -----------+
   |        ID        | STATUS  |            NAME            |        PEER ADDRS         |      CLIENT
   ADDRS       |
   +------------------+---------+----------------------------+---------------------------+---------------
   -----------+
   | 5eb0d6b8ca24730c | started |  ip-10-0-133-53.ec2.internal |  https://10.0.133.53:2380 |
   https://10.0.133.53:2379 |
   | 757b6793e2408b6c | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
   https://10.0.164.97:2379 |
   | ca8c2990a0aa29d1 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380 |
   https://10.0.154.204:2379 |
   +------------------+---------+----------------------------+---------------------------+---------------
   -----------+
   ```

   如果上一命令的输出列出了超过三个 etcd 成员，您必须删除不需要的成员。

警告

确保删除正确的 etcd 成员；如果删除了正常的 etcd 成员则有可能会导致仲裁丢失。

其他资源

- 恢复降级的 etcd Operator

5.2.4.2. 替换其 etcd Pod 处于 crashlooping 状态的不健康 etcd 成员

此流程详细介绍了替换因 etcd pod 处于 crashlooping 状态造成不健康的 etcd 成员的步骤。

先决条件

- 您已找出不健康的 etcd 成员。
- 已确认 etcd pod 处于 crashlooping 状态。
- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 已进行 etcd 备份。

重要

执行此流程前务必要进行 etcd 备份，以便在遇到任何问题时可以恢复集群。

流程

1. 停止处于 crashlooping 状态的 etcd pod。
   a. 对处于 crashlooping 状态的节点进行调试。
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc debug node/ip-10-0-131-183.ec2.internal 1
```

   1 使用不健康节点的名称来替换它。

   b. 将您的根目录改为 **/host** ：

```
sh-4.2# chroot /host
```

   c. 将现有 etcd pod 文件从 Kubelet 清单目录中移出：

```
sh-4.2# mkdir /var/lib/etcd-backup
```

```
sh-4.2# mv /etc/kubernetes/manifests/etcd-pod.yaml /var/lib/etcd-backup/
```

d. 将 etcd 数据目录移到不同的位置：

```
sh-4.2# mv /var/lib/etcd/ /tmp
```

现在您可以退出节点 shell。

2. 删除不健康的成员。

a. 选择一个*不在*受影响节点上的 pod。
在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

输**出示例**

```
etcd-ip-10-0-131-183.ec2.internal             2/3     Error       7          6h9m
etcd-ip-10-0-164-97.ec2.internal              3/3     Running     0          6h6m
etcd-ip-10-0-154-204.ec2.internal             3/3     Running     0          6h6m
```

b. 连接到正在运行的 etcd 容器，传递没有在受影响节点上的 pod 的名称。
在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

c. 查看成员列表：

```
sh-4.2# etcdctl member list -w table
```

输**出示例**

```
+------------------+---------+-----------------------------+--------------------------+----------------
-----------+
|        ID        | STATUS  |            NAME             |        PEER ADDRS        |      CLIENT
ADDRS       |
+------------------+---------+-----------------------------+--------------------------+----------------
-----------+
| 62bcf33650a7170a | started | ip-10-0-131-183.ec2.internal | https://10.0.131.183:2380 |
https://10.0.131.183:2379 |
| b78e2856655bc2eb | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| d022e10b498760d5 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380
| https://10.0.154.204:2379 |
+------------------+---------+-----------------------------+--------------------------+----------------
-----------+
```

记录不健康的 etcd 成员的 ID 和名称，因为稍后需要这些值。

d. 通过向 **etcdctl member remove** 命令提供 ID 来删除不健康的 etcd 成员：

```
sh-4.2# etcdctl member remove 62bcf33650a7170a
```

输**出示例**

```
Member 62bcf33650a7170a removed from cluster ead669ce1fbfb346
```

e. 再次查看成员列表，并确认成员已被删除：

```
sh-4.2# etcdctl member list -w table
```

输出示例

```
+------------------+---------+--------------------------+--------------------------+----------------
----------+
|        ID        | STATUS  |          NAME            |       PEER ADDRS         |      CLIENT
ADDRS       |
+------------------+---------+--------------------------+--------------------------+----------------
----------+
| b78e2856655bc2eb | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| d022e10b498760d5 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380
| https://10.0.154.204:2379 |
+------------------+---------+--------------------------+--------------------------+----------------
----------+
```

现在您可以退出节点 shell。

3. 输入以下命令关闭仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

此命令可确保您可以成功重新创建机密并推出静态 pod。

4. 删除已删除的不健康 etcd 成员的旧 secret。

a. 列出已删除的不健康 etcd 成员的 secret。

```
$ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
```

1 传递您之前在这个过程中记录的不健康 etcd 成员的名称。

有一个对等的、服务和指标的 secret，如以下输出所示：

输出示例

```
etcd-peer-ip-10-0-131-183.ec2.internal             kubernetes.io/tls                     2      47m
etcd-serving-ip-10-0-131-183.ec2.internal          kubernetes.io/tls                     2      47m
etcd-serving-metrics-ip-10-0-131-183.ec2.internal  kubernetes.io/tls                     2
47m
```

b. 删除已删除的不健康 etcd 成员的 secret。

i. 删除 peer（对等）secret:

```
$ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
```

ii. 删除 serving secret:

```
$ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
```

iii. 删除 metrics secret:

```
$ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-183.ec2.internal
```

5. 强制 etcd 重新部署。
在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "single-master-recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge 1
```

1 **forceRedeploymentReason** 值必须是唯一的，这就是为什么附加时间戳的原因。

当 etcd 集群 Operator 执行重新部署时，它会确保所有 control plane 节点都有可正常工作的 etcd pod。

6. 输入以下命令重新打开仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

7. 您可以输入以下命令验证 **unsupportedConfigOverrides** 部分是否已从对象中删除：

```
$ oc get etcd/cluster -oyaml
```

8. 如果使用单节点 OpenShift，请重启该节点。否则，您可能会在 etcd 集群 Operator 中遇到以下错误：

输出示例

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

验证

- 确认新成员可用且健康。

a. 连接到正在运行的 etcd 容器。
在一个终端中使用 cluster-admin 用户连接到集群，运行以下命令：

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

b. 验证所有成员是否健康：

```
sh-4.2# etcdctl endpoint health
```

-

输出示例

```
https://10.0.131.183:2379 is healthy: successfully committed proposal: took =
16.671434ms
https://10.0.154.204:2379 is healthy: successfully committed proposal: took =
16.698331ms
https://10.0.164.97:2379 is healthy: successfully committed proposal: took =
16.621645ms
```

5.2.4.3. 替换机器没有运行或节点未就绪的不健康裸机 etcd 成员

此流程详细介绍了替换因机器没有运行或节点未就绪造成不健康的裸机 etcd 成员的步骤。

如果您正在运行安装程序置备的基础架构，或者您使用 Machine API 创建机器，请按照以下步骤执行。否则，您必须使用最初创建控制平面节点时使用的相同方法创建新的控制平面。

先决条件

- 您已找出不健康的裸机 etcd 成员。
- 您已确认机器没有运行，或者该节点未就绪。
- 您可以使用具有 **cluster-admin** 角色的用户访问集群。
- 已进行 etcd 备份。

重要

执行此流程前务必要进行 etcd 备份，以便在遇到任何问题时可以恢复集群。

流程

1. 验证并删除不健康的成员。
   a. 选择一个*不在*受影响节点上的 pod:
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd -o wide
```

   输出示例

```
etcd-openshift-control-plane-0  5/5  Running  11  3h56m  192.168.10.9  openshift-
control-plane-0  <none>          <none>
etcd-openshift-control-plane-1  5/5  Running  0   3h54m  192.168.10.10  openshift-
control-plane-1  <none>          <none>
etcd-openshift-control-plane-2  5/5  Running  0   3h58m  192.168.10.11  openshift-
control-plane-2  <none>          <none>
```

   b. 连接到正在运行的 etcd 容器，传递没有在受影响节点上的 pod 的名称：
   在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

c. 查看成员列表：

```
sh-4.2# etcdctl member list -w table
```

输出示例

```
+------------------+---------+------------------+--------------------------+--------------------------
-+--------------------+
| ID               | STATUS  | NAME                 | PEER ADDRS               | CLIENT
ADDRS                | IS LEARNER |
+------------------+---------+------------------+--------------------------+--------------------------
-+--------------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380/ |
https://192.168.10.11:2379/ | false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380/ |
https://192.168.10.10:2379/ | false |
| cc3830a72fc357f9 | started | openshift-control-plane-0 | https://192.168.10.9:2380/ |
https://192.168.10.9:2379/   | false |
+------------------+---------+------------------+--------------------------+--------------------------
-+--------------------+
```

记录不健康的 etcd 成员的 ID 和名称，因为稍后需要这些值。**etcdctl endpoint health** 命令将列出已删除的成员，直到完成替换过程并添加了新成员。

d. 通过向 **etcdctl member remove** 命令提供 ID 来删除不健康的 etcd 成员：

警告

确保删除正确的 etcd 成员；如果删除了正常的 etcd 成员则有可能会导致仲裁丢失。

```
sh-4.2# etcdctl member remove 7a8197040a5126c8
```

输出示例

```
Member 7a8197040a5126c8 removed from cluster b23536c33f2cdd1b
```

e. 再次查看成员列表，并确认成员已被删除：

```
sh-4.2# etcdctl member list -w table
```

输出示例

```
+------------------+---------+------------------+--------------------------+--------------------------
-+------------------------+
| ID               | STATUS  | NAME                 | PEER ADDRS               | CLIENT
ADDRS                | IS LEARNER |
+------------------+---------+------------------+--------------------------+--------------------------
```

```
-+------------------------+
| cc3830a72fc357f9 | started | openshift-control-plane-2 | https://192.168.10.11:2380/ |
https://192.168.10.11:2379/ | false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380/ |
https://192.168.10.10:2379/ | false |
+-----------------+---------+-------------------+--------------------------+------------------------
-+------------------------+
```

现在您可以退出节点 shell。

重要

删除成员后，在剩余的 etcd 实例重启时，集群可能无法访问。

2. 输入以下命令关闭仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

此命令可确保您可以成功重新创建机密并推出静态 pod。

3. 运行以下命令，删除已删除的不健康 etcd 成员的旧 secret。

   a. 列出已删除的不健康 etcd 成员的 secret。

   ```
   $ oc get secrets -n openshift-etcd | grep openshift-control-plane-2
   ```

   传递您之前在这个过程中记录的不健康 etcd 成员的名称。

   有一个对等的、服务和指标的 secret，如以下输出所示：

   ```
   etcd-peer-openshift-control-plane-2              kubernetes.io/tls   2   134m
   etcd-serving-metrics-openshift-control-plane-2  kubernetes.io/tls   2   134m
   etcd-serving-openshift-control-plane-2           kubernetes.io/tls   2   134m
   ```

   b. 删除已删除的不健康 etcd 成员的 secret。

      i. 删除 peer（对等）secret:

      ```
      $ oc delete secret etcd-peer-openshift-control-plane-2 -n openshift-etcd

      secret "etcd-peer-openshift-control-plane-2" deleted
      ```

      ii. 删除 serving secret:

      ```
      $ oc delete secret etcd-serving-metrics-openshift-control-plane-2 -n openshift-etcd

      secret "etcd-serving-metrics-openshift-control-plane-2" deleted
      ```

      iii. 删除 metrics secret:

      ```
      $ oc delete secret etcd-serving-openshift-control-plane-2 -n openshift-etcd

      secret "etcd-serving-openshift-control-plane-2" deleted
      ```

4. 删除 control plane 机器。
   如果您正在运行安装程序置备的基础架构，或者您使用 Machine API 创建机器，请按照以下步骤执行。否则，您必须使用最初创建控制平面节点时使用的相同方法创建新的控制平面。

   a. 获取不健康成员的机器。
      在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

      ```
      $ oc get machines -n openshift-machine-api -o wide
      ```

      输出示例

      ```
      NAME                              PHASE     TYPE   REGION   ZONE   AGE     NODE
      PROVIDERID                                                        STATE
      examplecluster-control-plane-0    Running                          3h11m   openshift-control-
      plane-0   baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-
      3ff2-41c5-b099-0aa41222964e   externally provisioned (1)
      examplecluster-control-plane-1    Running                          3h11m   openshift-control-
      plane-1   baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-
      329c-475e-8d81-03b20280a3e1   externally provisioned
      examplecluster-control-plane-2    Running                          3h11m   openshift-control-
      plane-2   baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-
      61d8-410f-be5b-6a395b056135   externally provisioned
      examplecluster-compute-0          Running                          165m    openshift-compute-0
      baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-4bb3-80ec-
      13a31858241f          provisioned
      examplecluster-compute-1          Running                          165m    openshift-compute-1
      baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-4241-91dc-
      e7ea72ab13b9          provisioned
      ```

      (1) 这是不健康节点的 control plane 机器，**examplecluster-control-plane-2**。

   b. 将机器配置保存到文件系统中的一个文件中：

      ```
      $ oc get machine examplecluster-control-plane-2 \ (1)
        -n openshift-machine-api \
        -o yaml \
        > new-master-machine.yaml
      ```

      (1) 为不健康的节点指定 control plane 机器的名称。

   c. 编辑上一步中创建的 **new-master-machine.yaml** 文件，以分配新名称并删除不必要的字段。

      i. 删除整个 **status** 部分：

      ```
      status:
        addresses:
        - address: ""
          type: InternalIP
        - address: fe80::4adf:37ff:feb0:8aa1%ens1f1.373
          type: InternalDNS
      ```

```
  - address: fe80::4adf:37ff:feb0:8aa1%ens1f1.371
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Machine
    name: fe80::4adf:37ff:feb0:8aa1%ens1f1.372
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: machine.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: Machine
```

5. 将 **metadata.name** 字段更改为新名称。
   建议您保留与旧机器相同的基础名称，并将结束号码改为下一个可用数字。在本例中，**examplecluster-control-plane-2** 改为 **examplecluster-control-plane-3**。

   例如：

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: examplecluster-control-plane-3
  ...
```

   a. 删除 **spec.providerID** 字段：

```
    providerID: baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-61d8-410f-be5b-6a395b056135
```

   b. 删除 **metadata.annotations** 和 **metadata.generation** 字段：

```
    annotations:
      machine.openshift.io/instance-state: externally provisioned
    ...
    generation: 2
```

   c. 删除 **spec.conditions**、**spec.lastUpdated**、**spec.nodeRef** 和 **spec.phase** 字段：

```
    lastTransitionTime: "2022-08-03T08:40:36Z"
  message: 'Drain operation currently blocked by: [{Name:EtcdQuorumOperator
  Owner:clusteroperator/etcd}]'
  reason: HookPresent
  severity: Warning
  status: "False"
```

```
type: Drainable
lastTransitionTime: "2022-08-03T08:39:55Z"
status: "True"
type: InstanceExists

lastTransitionTime: "2022-08-03T08:36:37Z"
status: "True"
type: Terminable
lastUpdated: "2022-08-03T08:40:36Z"
nodeRef:
kind: Node
name: openshift-control-plane-2
uid: 788df282-6507-4ea2-9a43-24f237ccbc3c
phase: Running
```

6. 运行以下命令，确保 Bare Metal Operator 可用：

```
$ oc get clusteroperator baremetal
```

输出示例

```
NAME       VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
baremetal  4.16.0    True        False         False      3d15h
```

7. 运行以下命令来删除旧的 **BareMetalHost** 对象：

```
$ oc delete bmh openshift-control-plane-2 -n openshift-machine-api
```

输出示例

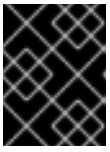
```
baremetalhost.metal3.io "openshift-control-plane-2" deleted
```

8. 运行以下命令来删除不健康成员的机器：

```
$ oc delete machine -n openshift-machine-api examplecluster-control-plane-2
```

删除 **BareMetalHost** 和 **Machine** 对象后，**Machine** Controller 会自动删除 **Node** 对象。

如果删除机器因任何原因或者命令被移动而延迟而延迟而延迟，您可以通过删除机器对象终结器字段来强制删除。



重要

不要通过按 **Ctrl+c** 中断机器删除。您必须允许命令继续完成。打开一个新的终端窗口来编辑并删除 finalizer 字段。

a. 运行以下命令来编辑机器配置：

```
$ oc edit machine -n openshift-machine-api examplecluster-control-plane-2
```

b. 删除 **Machine** 自定义资源中的以下字段，然后保存更新的文件：

```
-
```

```
  finalizers:
  - machine.machine.openshift.io
```

输出示例

```
machine.machine.openshift.io/examplecluster-control-plane-2 edited
```

9. 运行以下命令验证机器是否已删除：

```
$ oc get machines -n openshift-machine-api -o wide
```

输出示例

```
NAME                              PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                              STATE
examplecluster-control-plane-0    Running                          3h11m   openshift-control-plane-0
baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-3ff2-41c5-b099-
0aa41222964e   externally provisioned
examplecluster-control-plane-1    Running                          3h11m   openshift-control-plane-1
baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-329c-475e-8d81-
03b20280a3e1   externally provisioned
examplecluster-compute-0          Running                          165m    openshift-compute-0
baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-4bb3-80ec-
13a31858241f          provisioned
examplecluster-compute-1          Running                          165m    openshift-compute-1
baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-4241-91dc-
e7ea72ab13b9          provisioned
```

10. 运行以下命令验证节点是否已删除：

```
$ oc get nodes

NAME                       STATUS ROLES    AGE    VERSION
openshift-control-plane-0 Ready master 3h24m v1.29.4
openshift-control-plane-1 Ready master 3h24m v1.29.4
openshift-compute-0       Ready worker 176m v1.29.4
openshift-compute-1       Ready worker 176m v1.29.4
```

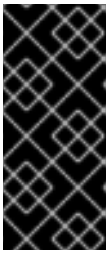
11. 创建新的 **BareMetalHost** 对象和 secret，以存储 BMC 凭证：

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: openshift-control-plane-2-bmc-secret
  namespace: openshift-machine-api
data:
  password: <password>
  username: <username>
type: Opaque
---
apiVersion: metal3.io/v1alpha1
kind: BareMetalHost
```

```
metadata:
  name: openshift-control-plane-2
  namespace: openshift-machine-api
spec:
  automatedCleaningMode: disabled
  bmc:
    address: redfish://10.46.61.18:443/redfish/v1/Systems/1
    credentialsName: openshift-control-plane-2-bmc-secret
    disableCertificateVerification: true
  bootMACAddress: 48:df:37:b0:8a:a0
  bootMode: UEFI
  externallyProvisioned: false
  online: true
  rootDeviceHints:
    deviceName: /dev/disk/by-id/scsi-<serial_number>
  userData:
    name: master-user-data-managed
    namespace: openshift-machine-api
EOF
```

注意

用户名和密码可从其他裸机主机的 secret 中找到。**bmc:address** 中使用的协议可以从其他 bmh 对象获取。



重要

如果您从现有 control plane 主机重复使用 **BareMetalHost** 对象定义，请不要将 external **Provisioned** 字段保留为 **true**。

如果 OpenShift Container Platform 安装程序置备，现有 control plane **BareMetalHost** 对象可能会将 **externallyProvisioned** 标记设为 **true**。

检查完成后，**BareMetalHost** 对象会被创建并可用置备。

12. 使用可用的 **BareMetalHost** 对象验证创建过程：

```
$ oc get bmh -n openshift-machine-api

NAME                 STATE                  CONSUMER                      ONLINE ERROR   AGE
openshift-control-plane-0 externally provisioned examplecluster-control-plane-0 true
4h48m
openshift-control-plane-1 externally provisioned examplecluster-control-plane-1 true
4h48m
openshift-control-plane-2 available          examplecluster-control-plane-3 true        47m
openshift-compute-0       provisioned        examplecluster-compute-0       true        4h48m
openshift-compute-1       provisioned        examplecluster-compute-1       true        4h48m
```

a. 使用 **new-master-machine.yaml** 文件创建新 control plane 机器：

```
$ oc apply -f new-master-machine.yaml
```

b. 验证新机器是否已创建：

```
$ oc get machines -n openshift-machine-api -o wide
```

输出示例

```
NAME                                PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                                STATE
examplecluster-control-plane-0      Running                          3h11m   openshift-control-
plane-0   baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-
3ff2-41c5-b099-0aa41222964e   externally provisioned   (1)
examplecluster-control-plane-1      Running                          3h11m   openshift-control-
plane-1   baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-
329c-475e-8d81-03b20280a3e1   externally provisioned
examplecluster-control-plane-2      Running                          3h11m   openshift-control-
plane-2   baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-
61d8-410f-be5b-6a395b056135   externally provisioned
examplecluster-compute-0            Running                          165m    openshift-compute-
0         baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-
4bb3-80ec-13a31858241f        provisioned
examplecluster-compute-1            Running                          165m    openshift-compute-
1         baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-
4241-91dc-e7ea72ab13b9        provisioned
```

(1) 新机器 **clustername-8qw5l-master-3** 会被创建，并在阶段从 **Provisioning** 变为 **Running** 后就绪。

创建新机器需要几分钟时间。当机器或节点返回一个健康状态时，etcd cluster Operator 将自动同步。

c. 运行以下命令验证裸机主机是否被置备，且没有报告的错误：

```
$ oc get bmh -n openshift-machine-api
```

输出示例

```
$ oc get bmh -n openshift-machine-api
NAME                 STATE                  CONSUMER                       ONLINE ERROR AGE
openshift-control-plane-0 externally provisioned examplecluster-control-plane-0 true
4h48m
openshift-control-plane-1 externally provisioned examplecluster-control-plane-1 true
4h48m
openshift-control-plane-2 provisioned          examplecluster-control-plane-3 true
47m
openshift-compute-0       provisioned          examplecluster-compute-0       true
4h48m
openshift-compute-1       provisioned          examplecluster-compute-1       true
4h48m
```

d. 运行以下命令验证新节点是否已添加并处于就绪状态：

```
$ oc get nodes
```

输出示例

```
-
```

```
$ oc get nodes
NAME                 STATUS ROLES   AGE   VERSION
openshift-control-plane-0 Ready master 4h26m v1.29.4
openshift-control-plane-1 Ready master 4h26m v1.29.4
openshift-control-plane-2 Ready master 12m   v1.29.4
openshift-compute-0       Ready worker 3h58m v1.29.4
openshift-compute-1       Ready worker 3h58m v1.29.4
```

13. 输入以下命令重新打开仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

14. 您可以输入以下命令验证 **unsupportedConfigOverrides** 部分是否已从对象中删除：

```
$ oc get etcd/cluster -oyaml
```

15. 如果使用单节点 OpenShift，请重启该节点。否则，您可能会在 etcd 集群 Operator 中遇到以下错误：

输出示例

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

验证

1. 验证所有 etcd pod 是否都正常运行。
在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

输出示例

```
etcd-openshift-control-plane-0     5/5   Running   0   105m
etcd-openshift-control-plane-1     5/5   Running   0   107m
etcd-openshift-control-plane-2     5/5   Running   0   103m
```

如果上一命令的输出只列出两个 pod，您可以手动强制重新部署 etcd。在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge (1)
```

(1) **forceRedeploymentReason** 值必须是唯一的，这就是为什么附加时间戳的原因。

要验证是否有完全有三个 etcd 成员，连接到正在运行的 etcd 容器，传递没有在受影响节点上的 pod 的名称。在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

2. 查看成员列表：

```
sh-4.2# etcdctl member list -w table
```

输出示例

```
+------------------+---------+--------------------+--------------------------+--------------------------+---
--------------+
|        ID        | STATUS  |        NAME        |        PEER ADDRS        |        CLIENT ADDRS
|    IS LEARNER    |
+------------------+---------+--------------------+--------------------------+--------------------------+---
--------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380 |
https://192.168.10.11:2379 |    false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380 |
https://192.168.10.10:2379 |    false |
| cc3830a72fc357f9 | started | openshift-control-plane-0 | https://192.168.10.9:2380 |
https://192.168.10.9:2379 |     false |
+------------------+---------+--------------------+--------------------------+--------------------------+---
--------------+
```

注意

如果上一命令的输出列出了超过三个 etcd 成员，您必须删除不需要的成员。

3. 运行以下命令，验证所有 etcd 成员是否健康：

```
# etcdctl endpoint health --cluster
```

输出示例

```
https://192.168.10.10:2379 is healthy: successfully committed proposal: took = 8.973065ms
https://192.168.10.9:2379 is healthy: successfully committed proposal: took = 11.559829ms
https://192.168.10.11:2379 is healthy: successfully committed proposal: took = 11.665203ms
```

4. 运行以下命令，验证所有节点是否处于最新的修订版本：

```
$ oc get etcd -o=jsonpath='{range.items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

```
AllNodesAtLatestRevision
```

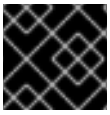
5.2.5. 其他资源

- 使用机器生命周期 hook 进行仲裁保护

5.3. 灾难恢复

5.3.1. 关于灾难恢复

灾难恢复文档为管理员提供了如何从 OpenShift Container Platform 集群可能出现的几个灾难情形中恢复的信息。作为管理员，您可能需要遵循以下一个或多个步骤将集群恢复为工作状态。



重要

灾难恢复要求您至少有一个健康的 control plane 主机。

恢复到一个以前的集群状态

如果您希望将集群恢复到一个以前的状态时（例如，管理员错误地删除了一些关键信息），则可以使用这个解决方案。这包括您丢失了大多数 control plane 主机并导致 etcd 仲裁丢失，且集群离线的情况。只要您执行了 etcd 备份，就可以按照这个步骤将集群恢复到之前的状态。
如果适用，可能还需要从过期的 control plane 证书中恢复。

警告

在一个正在运行的集群中恢复到以前的集群状态是破坏性的，而不稳定的操作。这仅应作为最后的手段使用。

在执行恢复前，请参阅关于恢复集群状态以了解有关对集群的影响的更多信息。

注意

如果大多数 master 仍可用，且仍有 etcd 仲裁，请按照以下步骤替换一个不健康的 etcd 成员。

从 control plane 证书已过期的情况下恢复

如果 control plane 证书已经过期，则可以使用这个解决方案。例如：在第一次证书轮转前（在安装后 24 小时内）关闭了集群，您的证书将不会被轮转，且会过期。可以按照以下步骤从已过期的 control plane 证书中恢复。

5.3.2. 恢复到一个以前的集群状态

为了将集群还原到以前的状态，您必须已通过创建快照来备份了 etcd 数据 。您将需要使用此快照来还原集群状态。

5.3.2.1. 关于恢复集群状态

您可以使用 etcd 备份将集群恢复到以前的状态。在以下情况中可以使用这个方法进行恢复：

- 集群丢失了大多数 control plane 主机（仲裁丢失）。
- 管理员删除了一些关键内容，必须恢复才能恢复集群。

警告

在一个正在运行的集群中恢复到以前的集群状态是破坏性的，而不稳定的操作。这仅应作为最后的手段使用。

如果您可以使用 Kubernetes API 服务器检索数据，则代表 etcd 可用，且您不应该使用 etcd 备份来恢复。

恢复 etcd 实际相当于把集群返回到以前的一个状态，所有客户端都会遇到一个有冲突的、并行历史记录。这会影响 kubelet、Kubernetes 控制器、SDN 控制器和持久性卷控制器等监视组件的行为。

当 etcd 中的内容与磁盘上的实际内容不匹配时，可能会导致 Operator churn，从而导致 Kubernetes API 服务器、Kubernetes 控制器管理器、Kubernetes 调度程序和 etcd 的 Operator 在磁盘上的文件与 etcd 中的内容冲突时卡住。这可能需要手动操作来解决问题。

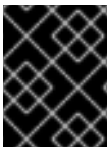
在极端情况下，集群可能会丢失持久性卷跟踪，删除已不存在的关键工作负载，重新镜像机器，以及重写带有过期证书的 CA 捆绑包。

5.3.2.2. 恢复到一个以前的集群状态

您可以使用保存的 **etcd** 备份来恢复以前的集群状态，或恢复丢失了大多数 control plane 主机的集群。

注意

如果您的集群使用 control plane 机器集，请参阅 "Troubleshooting control plane 机器集"来了解有关 **etcd** 恢复的过程。

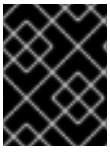


重要

恢复集群时，必须使用同一 z-stream 发行版本中获取的 **etcd** 备份。例如，OpenShift Container Platform 4.7.2 集群必须使用从 4.7.2 开始的 **etcd** 备份。

先决条件

- 通过一个基于证书的 **kubeconfig** 使用具有 **cluster-admin** 角色的用户访问集群，如安装期间的情况。
- 用作恢复主机的健康 control plane 主机。
- SSH 对 control plane 主机的访问。
- 包含从同一备份中获取的 etcd 快照和静态 **pod** 资源的备份目录。该目录中的文件名必须采用以下格式: **snapshot_<datetimestamp>.db** 和 **static_kuberesources_<datetimestamp>.tar.gz**。

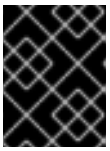


重要

对于非恢复 control plane 节点，不需要建立 SSH 连接或停止静态 pod。您可以逐个删除并重新创建其他非恢复 control plane 机器。

流程

1. 选择一个要用作恢复主机的 control plane 主机。这是您要在其中运行恢复操作的主机。
2. 建立到每个 control plane 节点（包括恢复主机）的 SSH 连接。
   恢复过程启动后，**kube-apiserver** 将无法访问，因此您无法访问 control plane 节点。因此，建议在一个单独的终端中建立到每个control plane 主机的 SSH 连接。



重要

如果没有完成这个步骤，将无法访问 control plane 主机来完成恢复过程，您将无法从这个状态恢复集群。

3. 将 **etcd** 备份目录复制到恢复 control plane 主机上。
   此流程假设您将 **backup** 目录（其中包含 **etcd** 快照和静态 pod 资源）复制到恢复 control plane 主机的 **/home/core/** 目录中。
4. 在任何其他 control plane 节点上停止静态 pod。

注意

您不需要停止恢复主机上的静态 pod。

   a. 访问不是恢复主机的 control plane 主机。
   b. 运行以下命令，将现有 etcd pod 文件从 kubelet 清单目录中移出：

```
$ sudo mv -v /etc/kubernetes/manifests/etcd-pod.yaml /tmp
```

   c. 使用以下命令验证 **etcd** pod 是否已停止：

```
$ sudo crictl ps | grep etcd | egrep -v "operator|etcd-guard"
```

   如果这个命令的输出不为空，请等待几分钟，然后再次检查。
   d. 运行以下命令，将现有 **kube-apiserver** 文件从 kubelet 清单目录中移出：

```
$ sudo mv -v /etc/kubernetes/manifests/kube-apiserver-pod.yaml /tmp
```

   e. 运行以下命令验证 **kube-apiserver** 容器是否已停止：

```
$ sudo crictl ps | grep kube-apiserver | egrep -v "operator|guard"
```

   如果这个命令的输出不为空，请等待几分钟，然后再次检查。
   f. 使用以下方法将现有 **kube-controller-manager** 文件从 kubelet 清单目录中移出：

```
$ sudo mv -v /etc/kubernetes/manifests/kube-controller-manager-pod.yaml /tmp
```

   g. 运行以下命令验证 **kube-controller-manager** 容器是否已停止：

```
$ sudo crictl ps | grep kube-controller-manager | egrep -v "operator|guard"
```

如果这个命令的输出不为空，请等待几分钟，然后再次检查。

h. 使用以下方法将现有 **kube-scheduler** 文件从 kubelet 清单目录中移出：

```
$ sudo mv -v /etc/kubernetes/manifests/kube-scheduler-pod.yaml /tmp
```

i. 使用以下命令验证 **kube-scheduler** 容器是否已停止：

```
$ sudo crictl ps | grep kube-scheduler | egrep -v "operator|guard"
```

如果这个命令的输出不为空，请等待几分钟，然后再次检查。

j. 使用以下示例将 **etcd** 数据目录移到不同的位置：

```
$ sudo mv -v /var/lib/etcd/ /tmp
```

k. 如果 **/etc/kubernetes/manifests/keepalived.yaml** 文件存在，且节点被删除，请按照以下步骤执行：

i. 将 **/etc/kubernetes/manifests/keepalived.yaml** 文件从 kubelet 清单目录中移出：

```
$ sudo mv -v /etc/kubernetes/manifests/keepalived.yaml /tmp
```

ii. 容器验证由 **keepalived** 守护进程管理的任何容器是否已停止：

```
$ sudo crictl ps --name keepalived
```

命令输出应该为空。如果它不是空的，请等待几分钟后再重新检查。

iii. 检查 control plane 是否已分配任何 Virtual IP (VIP)：

```
$ ip -o address | egrep '<api_vip>|<ingress_vip>'
```

iv. 对于每个报告的 VIP，运行以下命令将其删除：

```
$ sudo ip address del <reported_vip> dev <reported_vip_device>
```

l. 在其他不是恢复主机的 control plane 主机上重复此步骤。

5. 访问恢复 control plane 主机。

6. 如果使用 **keepalived** 守护进程，请验证恢复 control plane 节点是否拥有 VIP：

```
$ ip -o address | grep <api_vip>
```

如果存在 VIP 的地址（如果存在）。如果 VIP 没有设置或配置不正确，这个命令会返回一个空字符串。

7. 如果启用了集群范围的代理，请确定已导出了 **NO_PROXY**、**HTTP_PROXY**和 **HTTPS_PROXY** 环境变量。

提示

您可以通过查看 **oc get proxy cluster -o yaml** 的输出来检查代理是否已启用。如果 **httpProxy**、**httpsProxy**和 **noProxy** 字段设置了值，则会启用代理。

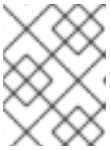
8. 在恢复 control plane 主机上运行恢复脚本，并传递到 **etcd** 备份目录的路径：

```
$ sudo -E /usr/local/bin/cluster-restore.sh /home/core/assets/backup
```

脚本输出示例

```
...stopping kube-scheduler-pod.yaml
...stopping kube-controller-manager-pod.yaml
...stopping etcd-pod.yaml
...stopping kube-apiserver-pod.yaml
Waiting for container etcd to stop
.complete
Waiting for container etcdctl to stop
............................complete
Waiting for container etcd-metrics to stop
complete
Waiting for container kube-controller-manager to stop
complete
Waiting for container kube-apiserver to stop
.....................................................................................complete
Waiting for container kube-scheduler to stop
complete
Moving etcd data-dir /var/lib/etcd/member to /var/lib/etcd-backup
starting restore-etcd static pod
starting kube-apiserver-pod.yaml
static-pod-resources/kube-apiserver-pod-7/kube-apiserver-pod.yaml
starting kube-controller-manager-pod.yaml
static-pod-resources/kube-controller-manager-pod-7/kube-controller-manager-pod.yaml
starting kube-scheduler-pod.yaml
static-pod-resources/kube-scheduler-pod-8/kube-scheduler-pod.yaml
```

cluster-restore.sh 脚本必须显示 **etcd**、**kube-apiserver**、**kube-controller-manager** 和 **kube-scheduler** pod 已停止，然后在恢复过程结束时启动。



注意

如果在上次 **etcd** 备份后更新了节点，则恢复过程可能会导致节点进入 **NotReady** 状态。

9. 检查节点以确保它们处于 **Ready** 状态。

   a. 运行以下命令:

```
$ oc get nodes -w
```

   输出示例

```
NAME              STATUS  ROLES        AGE    VERSION
```

```
host-172-25-75-28   Ready   master        3d20h   v1.29.4
host-172-25-75-38   Ready   infra,worker   3d20h   v1.29.4
host-172-25-75-40   Ready   master        3d20h   v1.29.4
host-172-25-75-65   Ready   master        3d20h   v1.29.4
host-172-25-75-74   Ready   infra,worker   3d20h   v1.29.4
host-172-25-75-79   Ready   worker        3d20h   v1.29.4
host-172-25-75-86   Ready   worker        3d20h   v1.29.4
host-172-25-75-98   Ready   infra,worker   3d20h   v1.29.4
```

所有节点都可能需要几分钟时间报告其状态。

b. 如果有任何节点处于 **NotReady** 状态，登录到节点，并从每个节点上的 **/var/lib/kubelet/pki** 目录中删除所有 PEM 文件。您可以 SSH 到节点，或使用 web 控制台中的终端窗口。

```
$  ssh -i <ssh-key-path> core@<master-hostname>
```

pki 目录示例

```
sh-4.4# pwd
/var/lib/kubelet/pki
sh-4.4# ls
kubelet-client-2022-04-28-11-24-09.pem  kubelet-server-2022-04-28-11-24-15.pem
kubelet-client-current.pem              kubelet-server-current.pem
```

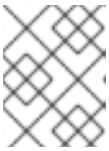
10. 在所有 control plane 主机上重启 kubelet 服务。

a. 在恢复主机中运行：

```
$ sudo systemctl restart kubelet.service
```

b. 在所有其他 control plane 主机上重复此步骤。

11. 批准待处理的证书签名请求 (CSR)：



注意

没有 worker 节点的集群（如单节点集群或由三个可调度的 control plane 节点组成的集群）不会批准任何待处理的 CSR。您可以跳过此步骤中列出的所有命令。

a. 运行以下命令获取当前 CSR 列表：

```
$ oc get csr
```

输出示例

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2s94x   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending 1
csr-4bd6t   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending 2
csr-4hl85   13m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
```

```
3
csr-zhhhp   3m8s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
4
...
```

1 2 一个待处理的 kubelet 服务 CSR，由 kubelet 服务端点请求。

3 4 一个待处理的 kubelet 客户端 CSR，使用 **node-bootstrapper** 节点 bootstrap 凭证请求。

b. 运行以下命令，查看 CSR 的详情以验证其是否有效：

```
$ oc describe csr <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

c. 运行以下命令来批准每个有效的 **node-bootstrapper** CSR：

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

d. 对于用户置备的安装，运行以下命令批准每个有效的 kubelet 服务 CSR：

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

12. 确认单个成员 control plane 已被成功启动。

a. 在恢复主机上，使用以下命令验证 **etcd** 容器是否正在运行：

```
$ sudo crictl ps | grep etcd | egrep -v "operator|etcd-guard"
```

输出示例

```
3ad41b7908e32
36f86e2eeaaffe662df0d21041eb22b8198e0e58abeeae8c743c3e6e977e8009
About a minute ago   Running             etcd                                    0
7c05f8af362f0
```

b. 在恢复主机上，使用以下命令验证 **etcd** pod 是否正在运行：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

输出示例

```
NAME                                READY   STATUS      RESTARTS   AGE
etcd-ip-10-0-143-125.ec2.internal           1/1     Running     1          2m47s
```

如果状态是 **Pending**，或者输出中列出了多个正在运行的 **etcd** pod，请等待几分钟，然后再次检查。

13. 如果使用 **OVNKubernetes** 网络插件，您必须重启 **ovnkube-controlplane** pod。

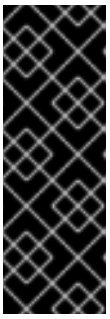
a. 运行以下命令删除所有 **ovnkube-controlplane** pod：

```
$ oc -n openshift-ovn-kubernetes delete pod -l app=ovnkube-control-plane
```

b. 使用以下命令验证所有 **ovnkube-controlplane** pod 是否已重新部署：

```
$ oc -n openshift-ovn-kubernetes get pod -l app=ovnkube-control-plane
```

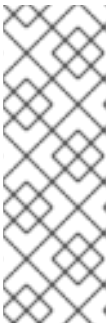
14. 如果使用 OVN-Kubernetes 网络插件，请逐个重启所有节点上的 Open Virtual Network (OVN) Kubernetes pod。使用以下步骤重启每个节点上的 OVN-Kubernetes pod：



重要

按照以下顺序重启 OVN-Kubernetes pod：

1. 恢复控制平面主机
2. 其他控制平面主机（如果可用）
3. 其他节点



注意

验证和变异准入 Webhook 可能会拒绝 pod。如果您添加了额外的 Webhook，其 **failurePolicy** 被设置为 **Fail** 的，则它们可能会拒绝 pod，恢复过程可能会失败。您可以通过在恢复集群状态时保存和删除 Webhook 来避免这种情况。成功恢复集群状态后，您可以再次启用 Webhook。

另外，您可以在恢复集群状态时临时将 **failurePolicy** 设置为 **Ignore**。成功恢复集群状态后，您可以将 **failurePolicy** 设置为 **Fail**。

a. 删除北向数据库 (nbdb) 和南向数据库 (sbdb)。使用 Secure Shell (SSH) 访问恢复主机和剩余的 control plane 节点，并运行：

```
$ sudo rm -f /var/lib/ovn-ic/etc/*.db
```

b. 重新启动 OpenVSwitch 服务。使用 Secure Shell (SSH) 访问节点，并运行以下命令：

```
$ sudo systemctl restart ovs-vswitchd ovsdb-server
```

c. 运行以下命令删除节点上的 **ovnkube-node** pod，将 **<node>** 替换为您要重启的节点的名称：

```
$ oc -n openshift-ovn-kubernetes delete pod -l app=ovnkube-node --field-
selector=spec.nodeName==<node>
```

d. 使用以下命令验证 **ovnkube-node** pod 已再次运行：

```
$ oc -n openshift-ovn-kubernetes get pod -l app=ovnkube-node --field-
selector=spec.nodeName==<node>
```

注意

pod 可能需要几分钟时间来重启。

15. 逐个删除并重新创建其他非恢复 control plane 机器。重新创建机器后，会强制一个新修订版本，**etcd** 会自动扩展。

- 如果使用用户置备的裸机安装，您可以使用最初创建它时使用的相同方法重新创建 control plane 机器。如需更多信息，请参阅"在裸机上安装用户置备的集群"。

警告

不要为恢复主机删除并重新创建机器。

- 如果您正在运行安装程序置备的基础架构，或者您使用 Machine API 创建机器，请按照以下步骤执行：

警告

不要为恢复主机删除并重新创建机器。

对于安装程序置备的基础架构上的裸机安装，不会重新创建 control plane 机器。如需更多信息，请参阅"替换裸机控制平面节点"。

a. 为丢失的 control plane 主机之一获取机器。
在一个终端中使用 cluster-admin 用户连接到集群，运行以下命令：

```
$ oc get machines -n openshift-machine-api -o wide
```

输出示例：

```
NAME                                PHASE   TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-0          Running m4.xlarge   us-east-1   us-east-1a
3h37m  ip-10-0-131-183.ec2.internal  aws:///us-east-1a/i-0ec2782f8287dfb7e
stopped (1)
clustername-8qw5l-master-1          Running m4.xlarge   us-east-1   us-east-1b
3h37m  ip-10-0-143-125.ec2.internal  aws:///us-east-1b/i-096c349b700a19631
running
clustername-8qw5l-master-2          Running m4.xlarge   us-east-1   us-east-1c
3h37m  ip-10-0-154-194.ec2.internal  aws:///us-east-1c/i-02626f1dba9ed5bba
running
clustername-8qw5l-worker-us-east-1a-wbtgd  Running m4.large   us-east-1   us-
east-1a  3h28m  ip-10-0-129-226.ec2.internal  aws:///us-east-1a/i-
010ef6279b4662ced  running
```

```
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a   running
```

1 这是用于丢失的 control plane 主机 **ip-10-0-131-183.ec2.internal** 的 control plane 机器。

b. 运行以下命令，将机器配置保存到文件系统中的一个文件中：

```
$ oc get machine clustername-8qw5l-master-0 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1 为丢失的 control plane 主机指定 control plane 机器的名称。

c. 编辑上一步中创建的 **new-master-machine.yaml** 文件，以分配新名称并删除不必要的字段。

i. 运行以下命令删除整个 **status** 部分：

```
status:
  addresses:
  - address: 10.0.131.183
    type: InternalIP
  - address: ip-10-0-131-183.ec2.internal
    type: InternalDNS
  - address: ip-10-0-131-183.ec2.internal
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Node
    name: ip-10-0-131-183.ec2.internal
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: awsproviderconfig.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: AWSMachineProviderStatus
```

ii. 运行以下命令，将 **metadata.name** 字段改为新名称：
建议您保留与旧机器相同的基础名称，并将结束号码改为下一个可用数字。在本例中，**clustername-8qw5l-master-0** 被改为 **clustername-8qw5l-master-3** ：

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. 运行以下命令来删除 **spec.providerID** 字段：

```
providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

iv. 运行以下命令，删除 **metadata.annotations** 和 **metadata.generation** 字段：

```
annotations:
  machine.openshift.io/instance-state: running
...
generation: 2
```

v. 运行以下命令，删除 **metadata.resourceVersion** 和 **metadata.uid** 字段：

```
resourceVersion: "13291"
uid: a282eb70-40a2-4e89-8009-d05dd420d31a
```

d. 运行以下命令，删除丢失的 control plane 主机的机器：

```
$ oc delete machine -n openshift-machine-api clustername-8qw5l-master-0 1
```

1 为丢失的 control plane 主机指定 control plane 机器的名称。

e. 运行以下命令验证机器是否已删除：

```
$ oc get machines -n openshift-machine-api -o wide
```

输出示例：

```
NAME                                PHASE   TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1          Running m4.xlarge   us-east-1   us-east-1b   3h37m   ip-10-0-143-125.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running m4.xlarge   us-east-1   us-east-1c   3h37m   ip-10-0-154-194.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd  Running m4.large   us-east-1   us-east-1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb  Running m4.large   us-east-1   us-east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26  Running m4.large   us-east-1   us-east-1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a   running
```

f. 运行以下命令，使用 **new-master-machine.yaml** 文件创建机器：

```
$ oc apply -f new-master-machine.yaml
```

g. 运行以下命令验证新机器是否已创建：

```
$ oc get machines -n openshift-machine-api -o wide
```

输出示例：

```
NAME                                PHASE         TYPE        REGION      ZONE
AGE   NODE                          PROVIDERID                          STATE
clustername-8qw5l-master-1          Running       m4.xlarge   us-east-1   us-east-
1b    3h37m   ip-10-0-143-125.ec2.internal   aws:///us-east-1b/i-096c349b700a19631
running
clustername-8qw5l-master-2          Running       m4.xlarge   us-east-1   us-east-
1c    3h37m   ip-10-0-154-194.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba
running
clustername-8qw5l-master-3          Provisioning  m4.xlarge   us-east-1   us-east-
1a    85s     ip-10-0-173-171.ec2.internal   aws:///us-east-1a/i-015b0888fe17bc2c8
running (1)
clustername-8qw5l-worker-us-east-1a-wbtgd   Running       m4.large    us-east-1
us-east-1a    3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-
010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running       m4.large    us-east-1   us-
east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-
0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running       m4.large    us-east-1
us-east-1c    3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-
06861c00007751b0a   running
```

(1) 新机器 **clustername-8qw5l-master-3** 会被创建，并在阶段从 **Provisioning** 变为 **Running** 后就绪。

创建新机器可能需要几分钟时间。当机器或节点返回到健康状态时，**etcd** 集群 Operator 将自动同步。

h. 对不是恢复主机的每个已丢失的 control plane 主机重复此步骤。

16. 输入以下内容关闭仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

此命令可确保您可以成功重新创建机密并推出静态 pod。

17. 在恢复主机中的一个单独的终端窗口中，运行以下命令导出恢复 **kubeconfig** 文件：

```
$ export KUBECONFIG=/etc/kubernetes/static-pod-resources/kube-apiserver-
certs/secrets/node-kubeconfigs/localhost-recovery.kubeconfig
```

18. 强制 **etcd** 重新部署。
在导出恢复 **kubeconfig** 文件的同一终端窗口中，运行：

-

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
```
(1)

(1) **forceRedeploymentReason** 值必须是唯一的，这就是为什么附加时间戳的原因。

当 **etcd** 集群 Operator 执行重新部署时，现有节点开始使用与初始 bootstrap 扩展类似的新 pod。

19. 输入以下内容重新打开仲裁保护：

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

20. 您可以运行以下命令来验证 **unsupportedConfigOverrides** 部分是否已从对象中删除：

```
$ oc get etcd/cluster -oyaml
```

21. 验证所有节点是否已更新至最新的修订版本。
在一个终端中使用 **cluster-admin** 用户连接到集群，请运行：

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

查看 **etcd** 的 **NodeInstallerProgressing** 状态条件，以验证所有节点是否处于最新的修订版本。在更新成功后，输出会显示 **AllNodesAtLatestRevision**：

```
AllNodesAtLatestRevision
3 nodes are at revision 7
```
(1)

(1) 在本例中，最新的修订版本号是 **7**。

如果输出包含多个修订号，如 **2 个节点为修订版本 6；1 个节点为修订版本 7**，这意味着更新仍在进行中。等待几分钟后重试。

22. 重新部署 **etcd** 后，为 control plane 强制进行新的推出部署。 **kube-apiserver** 将在其他节点上重新安装自己，因为 kubelet 使用内部负载均衡器连接到 API 服务器。
在一个终端中使用 **cluster-admin** 用户连接到集群，请运行：

    a. 为 **kube-apiserver** 强制进行新的推出部署：

    ```
    $ oc patch kubeapiserver cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
    ```

    验证所有节点是否已更新至最新的修订版本。

    ```
    $ oc get kubeapiserver -o=jsonpath='{range .items[0].status.conditions[?(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
    ```

    查看 **NodeInstallerProgressing** 状态条件，以验证所有节点是否处于最新版本。在更新成功后，输出会显示 **AllNodesAtLatestRevision**：

```
AllNodesAtLatestRevision
3 nodes are at revision 7 1
```

1 在本例中，最新的修订版本号是 **7**。

如果输出包含多个修订号，如 **2 个节点为修订版本 6；1 个节点为修订版本 7**，这意味着更新仍在进行中。等待几分钟后重试。

b. 运行以下命令，为 Kubernetes 控制器管理器强制进行新的推出部署：

```
$ oc patch kubecontrollermanager cluster -p='{"spec": {"forceRedeploymentReason":
"recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
```

运行以下命令，验证所有节点是否已更新至最新的修订版本：

```
$ oc get kubecontrollermanager -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

查看 **NodeInstallerProgressing** 状态条件，以验证所有节点是否处于最新版本。在更新成功后，输出会显示 **AllNodesAtLatestRevision**：

```
AllNodesAtLatestRevision
3 nodes are at revision 7 1
```

1 在本例中，最新的修订版本号是 **7**。

如果输出包含多个修订号，如 **2 个节点为修订版本 6；1 个节点为修订版本 7**，这意味着更新仍在进行中。等待几分钟后重试。

c. 运行以下命令，为 **kube-scheduler** 强制进行新的推出部署：

```
$ oc patch kubescheduler cluster -p='{"spec": {"forceRedeploymentReason": "recovery-
'"$( date --rfc-3339=ns )"'"}}' --type=merge
```

使用以下命令验证所有节点是否已更新至最新的修订版本：

```
$ oc get kubescheduler -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

查看 **NodeInstallerProgressing** 状态条件，以验证所有节点是否处于最新版本。在更新成功后，输出会显示 **AllNodesAtLatestRevision**：

```
AllNodesAtLatestRevision
3 nodes are at revision 7 1
```

1 在本例中，最新的修订版本号是 **7**。

如果输出包含多个修订号，如 **2 个节点为修订版本 6；1 个节点为修订版本 7**，这意味着更新仍在进行中。等待几分钟后重试。

23. 验证所有 control plane 主机是否已启动并加入集群。

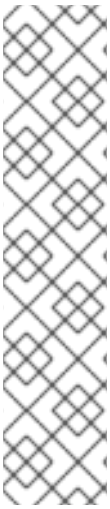
在一个终端中使用 **cluster-admin** 用户连接到集群，运行以下命令：

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

输出示例

```
etcd-ip-10-0-143-125.ec2.internal        2/2     Running     0          9h
etcd-ip-10-0-154-194.ec2.internal        2/2     Running     0          9h
etcd-ip-10-0-173-171.ec2.internal        2/2     Running     0          9h
```

为确保所有工作负载在恢复过程后返回到正常操作，请重启存储 **kube-apiserver** 信息的每个 pod。这包括 OpenShift Container Platform 组件，如路由器、Operator 和第三方组件。



注意

完成前面的流程步骤后，您可能需要等待几分钟，让所有服务返回到恢复的状态。例如，在重启 OAuth 服务器 pod 前，使用 **oc login** 进行身份验证可能无法立即正常工作。

考虑使用 **system:admin kubeconfig** 文件立即进行身份验证。这个方法基于 SSL/TLS 客户端证书作为 OAuth 令牌的身份验证。您可以发出以下命令来使用此文件进行身份验证：

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig
```

发出以下命令以显示您的验证的用户名：

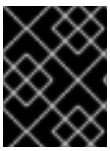
```
$ oc whoami
```

5.3.2.3. 其他资源

- 在裸机上安装用户置备的集群
- 创建堡垒主机以通过 SSH 访问 OpenShift Container Platform 实例和 control plane 节点。
- 替换裸机 control plane 节点

5.3.2.4. 恢复持久性存储状态的问题和解决方法

如果您的 OpenShift Container Platform 集群使用任何形式的持久性存储，集群的状态通常存储在 etcd 外部。它可能是在 pod 中运行的 Elasticsearch 集群，或者在 **StatefulSet** 对象中运行的数据库。从 etcd 备份中恢复时，还会恢复 OpenShift Container Platform 中工作负载的状态。但是，如果 etcd 快照是旧的，其状态可能无效或过期。



重要

持久性卷（PV）的内容绝不会属于 etcd 快照的一部分。从 etcd 快照恢复 OpenShift Container Platform 集群时，非关键工作负载可能会访问关键数据，反之亦然。

以下是生成过时状态的一些示例情况：

- MySQL 数据库在由 PV 对象支持的 pod 中运行。从 etcd 快照恢复 OpenShift Container Platform 不会使卷恢复到存储供应商上，且不会生成正在运行的 MySQL pod，尽管 pod 会重复尝试启动。您必须通过在存储供应商中恢复卷，然后编辑 PV 以指向新卷来手动恢复这个 pod。

- Pod P1 使用卷 A，它附加到节点 X。如果另一个 pod 在节点 Y 上使用相同的卷，则执行 etcd 恢复时，pod P1 可能无法正确启动，因为卷仍然被附加到节点 Y。OpenShift Container Platform 并不知道附加，且不会自动分离它。发生这种情况时，卷必须从节点 Y 手动分离，以便卷可以在节点 X 上附加，然后 pod P1 才可以启动。

- 在执行 etcd 快照后，云供应商或存储供应商凭证会被更新。这会导致任何依赖于这些凭证的 CSI 驱动程序或 Operator 无法正常工作。您可能需要手动更新这些驱动程序或 Operator 所需的凭证。

- 在生成 etcd 快照后，会从 OpenShift Container Platform 节点中删除或重命名设备。Local Storage Operator 会为从 **/dev/disk/by-id** 或 **/dev** 目录中管理的每个 PV 创建符号链接。这种情况可能会导致本地 PV 引用不再存在的设备。
  要解决这个问题，管理员必须：

  1. 手动删除带有无效设备的 PV。
  2. 从对应节点中删除符号链接。
  3. 删除 **LocalVolume** 或 **LocalVolumeSet** 对象（请参阅 *Storage → Configuring persistent storage → Persistent storage → Persistent storage → Deleting the Local Storage Operator Resources*）。

5.3.3. 从 control plane 证书已过期的情况下恢复

5.3.3.1. 从 control plane 证书已过期的情况下恢复

集群可以从过期的 control plane 证书中自动恢复。

但是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。对于用户置备的安装，您可能需要批准待处理的 kubelet 服务 CSR。

使用以下步骤批准待处理的 CSR：

流程

1. 获取当前 CSR 列表：

```
$ oc get csr
```

输出示例

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2s94x   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending 1
csr-4bd6t   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending
csr-4hl85   13m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending 2
csr-zhhhp   3m8s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
...
```

1 一个待处理的 kubelet 服务 CSR（用于用户置备的安装）。

2 一个待处理的 **node-bootstrapper** CSR。

2. 查看一个 CSR 的详细信息以验证其是否有效：

```
$ oc describe csr <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

3. 批准每个有效的 **node-bootstrapper** CSR：

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

4. 对于用户置备的安装，请批准每个有效的 kubelet 服务 CSR：

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