OpenShift Container Platform 4.10

バックアップおよび復元

OpenShift Container Platform クラスターのバックアップおよび復元

Last Updated: 2023-10-17

OpenShift Container Platform 4.10 バックアップおよび復元

OpenShift Container Platform クラスターのバックアップおよび復元

法律上の通知

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at
http://creativecommons.org/licenses/by-sa/3.0/
. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

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

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

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

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

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

Node.js ® is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

本書では、クラスターのデータのバックアップと、さまざまな障害関連のシナリオでの復旧方法について説明します。

目次

第1章 バックアップおよび復元

1.1. コントロールプレーンのバックアップおよび復元の操作

クラスター管理者は、OpenShift Container Platform クラスターを一定期間停止し、後で再起動する必要がある場合があります。クラスターを再起動する理由として、クラスターでメンテナンスを実行する必要がある、またはリソースコストを削減する必要がある、などが挙げられます。OpenShift Container Platform では、クラスターの正常なシャットダウン を実行して、後でクラスターを簡単に再起動できます。

クラスターをシャットダウンする前に etcd データをバックアップする 必要があります。etcd は OpenShift Container Platform のキーと値のストアであり、すべてのリソースオブジェクトの状態を保存します。etcd のバックアップは、障害復旧で重要なロールを果たします。OpenShift Container Platform では、正常でない etcd メンバーを置き換える こともできます。

クラスターを再度実行する場合は、クラスターを正常に再起動します。

注記

クラスターの証明書は、インストール日から 1 年後に有効期限が切れます。証明書が有効である間は、クラスターをシャットダウンし、正常に再起動することができます。クラスターは、期限切れのコントロールプレーン証明書を自動的に取得しますが、証明書署名要求 (CSR) を承認する 必要があります。

以下のように、OpenShift Container Platform が想定どおりに機能しないさまざまな状況に直面します。

- ノードの障害やネットワーク接続の問題などの予期しない状態により、再起動後にクラスターが機能しない。
- 誤ってクラスターで重要なものを削除した。
- 大多数のコントロールプレーンホストが失われたため、etcd のクォーラム (定足数) を喪失した。

保存した etcd スナップショットを使用して、クラスターを以前の状態に復元して、障害状況から常に回復できます。

1.2. アプリケーションのバックアップおよび復元の操作

クラスター管理者は、OpenShift API for Data Protection (OADP) を使用して、OpenShift Container Platform で実行しているアプリケーションをバックアップおよび復元できます。

OADP は、Velero CLI ツールのダウンロードの 表に従って、インストールする OADP のバージョンに適したバージョンの Velero を使用して、namespace の粒度で Kubernetes リソースと内部イメージをバックアップおよび復元します。OADP は、スナップショットまたは Restic を使用して、永続ボリューム (PV) をバックアップおよび復元します。詳細については、OADP の機能 を参照してください。

1.2.1. OADP 要件

OADP には以下の要件があります。

- **cluster-admin** ロールを持つユーザーとしてログインする必要があります。

- 次のストレージタイプのいずれかなど、バックアップを保存するためのオブジェクトストレージが必要です。
  - OpenShift Data Foundation
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - S3 と互換性のあるオブジェクトストレージ

重要

S3 ストレージ用の **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- スナップショットを使用して PV をバックアップするには、ネイティブスナップショット API を備えているか、次のプロバイダーなどの Container Storage Interface (CSI) スナップショットをサポートするクラウドストレージが必要です。
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - Ceph RBD や Ceph FS などの CSI スナップショット対応のクラウドストレージ

注記

スナップショットを使用して PV をバックアップしたくない場合は、デフォルトで OADP Operator によってインストールされる Restic を使用できます。

1.2.2. アプリケーションのバックアップおよび復元

Backup カスタムリソース (CR) を作成して、アプリケーションをバックアップします。バックアップ CR の作成 を参照してください。次のバックアップオプションを設定できます。

- バックアップ操作の前後にコマンドを実行するための バックアップフック
- スケジュールされたバックアップ
- Restic バックアップ

アプリケーションのバックアップを復元するには、**Restore** (CR) を作成します。復元 CR の作成 を参照してください。復元操作中に init コンテナーまたはアプリケーションコンテナーでコマンドを実行するように 復元フック を設定できます。

第2章 クラスターの正常なシャットダウン

本書では、クラスターを正常にシャットダウンするプロセスについて説明します。メンテナンスの目的で、またはリソースコストの節約のためにクラスターを一時的にシャットダウンする必要がある場合があります。

2.1. 前提条件

- クラスターをシャットダウンする前に etcd バックアップを作成します。

2.2. クラスターのシャットダウン

クラスターを正常な状態でシャットダウンし、後で再起動できるようにします。

注記

インストール日から1年までクラスターをシャットダウンして、正常に再起動することを期待できます。インストール日から1年後に、クラスター証明書が期限切れになります。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- etcd のバックアップを取得している。

重要

クラスターの再起動時に問題が発生した場合にクラスターを復元できるように、この手順を実行する前に etcd バックアップを作成しておくことは重要です。

たとえば、次の条件により、再起動したクラスターが誤動作する可能性があります。

- シャットダウン時の etcd データの破損
- ハードウェアが原因のノード障害
- ネットワーク接続の問題

クラスターが回復しない場合は、クラスターの以前の状態に復元する手順を実行します。

手順

1. クラスターを長期間シャットダウンする予定がある場合は、クラスター証明書の有効期限が切れる日付を決定します。
   証明書の有効期限が切れる前にクラスターを再起動する必要があります。クラスターの再起動時に、kubelet 証明書を回復するために保留中の証明書署名要求 (CSR) を手動で承認する必要がある場合があります。

   a. **kube-apiserver-to-kubelet-signer** CA 証明書の有効期限を確認します。

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

出力例

```
2023-08-05T14:37:50Z
```

b. kubelet 証明書の有効期限を確認します。

i. 次のコマンドを実行して、コントロールプレーンノードのデバッグセッションを開始します。

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

ii. 次のコマンドを実行して、ルートディレクトリーを **/host** に変更します。

```
sh-4.4# chroot /host
```

iii. 次のコマンドを実行して、kubelet クライアント証明書の有効期限を確認します。

```
sh-5.1# openssl x509 -in /var/lib/kubelet/pki/kubelet-client-current.pem -noout -
enddate
```

出力例

```
notAfter=Jun  6 10:50:07 2023 GMT
```

iv. 次のコマンドを実行して、kubelet サーバー証明書の有効期限を確認します。

```
sh-5.1# openssl x509 -in /var/lib/kubelet/pki/kubelet-server-current.pem -noout -
enddate
```

出力例

```
notAfter=Jun  6 10:50:07 2023 GMT
```

v. デバッグセッションを終了します。

vi. これらの手順を繰り返して、すべてのコントロールプレーンノードの証明書の有効期限を確認します。クラスターを正常に再起動できるようにするには、最も早い証明書の有効期限が切れる前にクラスターを再起動するように計画してください。

2. クラスターのすべてのノードをシャットダウンします。これは、クラウドプロバイダーの Web コンソールから実行したり、以下のループを実行できます。

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

1 **-h 1** は、コントロールプレーンノードがシャットダウンされるまで、このプロセスが継続する時間 (分単位) を示します。10 ノード以上の大規模なクラスターでは、まず始めにすべてのコンピュートノードをシャットダウンする時間を確保するために、10 分以上に設定します。

出力例

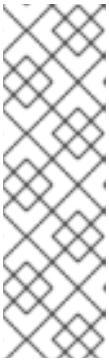
```
Starting pod/ip-10-0-130-169us-east-2computeinternal-debug ...
To use host binaries, run `chroot /host`
Shutdown scheduled for Mon 2021-09-13 09:36:17 UTC, use 'shutdown -c' to cancel.

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.
```

これらの方法のいずれかを使用してノードをシャットダウンすると、Pod は正常に終了するため、データが破損する可能性が低減します。

注記

大規模なクラスターでは、シャットダウン時間が長くなるように調整します。

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

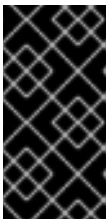


注記

シャットダウン前に OpenShift Container Platform に同梱される標準 Pod のコントロールプレーンノードをドレイン (解放) する必要はありません。

クラスター管理者は、クラスターの再起動後に独自のワークロードのクリーンな再起動を実行する必要があります。カスタムワークロードが原因でシャットダウン前にコントロールプレーンノードをドレイン (解放) した場合は、再起動後にクラスターが再び機能する前にコントロールプレーンノードをスケジュール可能としてマークする必要があります。

3. 外部ストレージや LDAP サーバーなど、不要になったクラスター依存関係をすべて停止します。この作業を行う前に、ベンダーのドキュメントを確認してください。



重要

クラスターをクラウドプロバイダープラットフォームにデプロイした場合は、関連するクラウドリソースをシャットダウン、一時停止、または削除しないでください。一時停止された仮想マシンのクラウドリソースを削除すると、OpenShift Container Platform が正常に復元されない場合があります。

2.3. 関連情報

- クラスターの正常な再起動
- クラスターの直前の状態への復元

第3章 クラスターの正常な再起動

本書では、正常なシャットダウン後にクラスターを再起動するプロセスについて説明します。

クラスターは再起動後に機能することが予想されますが、クラスターは以下の例を含む予期しない状態によって回復しない可能性があります。

- シャットダウン時の etcd データの破損
- ハードウェアが原因のノード障害
- ネットワーク接続の問題

クラスターが回復しない場合は、クラスターの以前の状態に復元する手順を実行します。

3.1. 前提条件

- クラスターを正常にシャットダウンしている。

3.2. クラスターの再起動

クラスターの正常なシャットダウン後にクラスターを再起動できます。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- この手順では、クラスターを正常にシャットダウンしていることを前提としています。

手順

1. 外部ストレージや LDAP サーバーなどのクラスターの依存関係すべてをオンにします。
2. すべてのクラスターマシンを起動します。
   クラウドプロバイダーの Web コンソールなどでマシンを起動するには、ご使用のクラウド環境に適した方法を使用します。

   約 10 分程度待機してから、コントロールプレーンノードのステータス確認に進みます。
3. すべてのコントロールプレーンノードが準備状態にあることを確認します。

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

   以下の出力に示されているように、コントロールプレーンノードはステータスが **Ready** の場合、準備状態にあります。

   ```
   NAME                          STATUS   ROLES    AGE   VERSION
   ip-10-0-168-251.ec2.internal   Ready    master   75m   v1.23.0
   ip-10-0-170-223.ec2.internal   Ready    master   75m   v1.23.0
   ip-10-0-211-16.ec2.internal    Ready    master   75m   v1.23.0
   ```
4. コントロールプレーンノードが準備状態に **ない** 場合、承認する必要がある保留中の証明書署名要求 (CSR) があるかどうかを確認します。

a. 現在の CSR の一覧を取得します。

```
$ oc get csr
```

b. CSR の詳細をレビューし、これが有効であることを確認します。

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

1 **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

c. それぞれの有効な CSR を承認します。

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

5. コントロールプレーンノードが準備状態になった後に、すべてのワーカーノードが準備状態にあることを確認します。

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

以下の出力に示されているように、ワーカーノードのステータスが **Ready** の場合、ワーカーノードは準備状態にあります。

```
NAME                         STATUS   ROLES    AGE   VERSION
ip-10-0-179-95.ec2.internal  Ready    worker   64m   v1.23.0
ip-10-0-182-134.ec2.internal  Ready    worker   64m   v1.23.0
ip-10-0-250-100.ec2.internal  Ready    worker   64m   v1.23.0
```

6. ワーカーノードが準備状態に **ない** 場合、承認する必要がある保留中の証明書署名要求 (CSR) があるかどうかを確認します。

a. 現在の CSR の一覧を取得します。

```
$ oc get csr
```

b. CSR の詳細をレビューし、これが有効であることを確認します。

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

1 **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

c. それぞれの有効な CSR を承認します。

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

7. クラスターが適切に起動していることを確認します。

a. パフォーマンスが低下したクラスター Operator がないことを確認します。

```
$ oc get clusteroperators
```

DEGRADED 条件が **True** に設定されているクラスター Operator がないことを確認します。

```
NAME                                       VERSION   AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.10.0    True        False         False      59m
cloud-credential                           4.10.0    True        False         False      85m
cluster-autoscaler                         4.10.0    True        False         False      73m
config-operator                            4.10.0    True        False         False      73m
console                                    4.10.0    True        False         False      62m
csi-snapshot-controller                    4.10.0    True        False         False      66m
dns                                        4.10.0    True        False         False      76m
etcd                                       4.10.0    True        False         False      76m
...
```

b. すべてのノードが **Ready** 状態にあることを確認します。

```
$ oc get nodes
```

すべてのノードのステータスが **Ready** であることを確認します。

```
NAME                           STATUS   ROLES    AGE   VERSION
ip-10-0-168-251.ec2.internal   Ready    master   82m   v1.23.0
ip-10-0-170-223.ec2.internal   Ready    master   82m   v1.23.0
ip-10-0-179-95.ec2.internal    Ready    worker   70m   v1.23.0
ip-10-0-182-134.ec2.internal   Ready    worker   70m   v1.23.0
ip-10-0-211-16.ec2.internal    Ready    master   82m   v1.23.0
ip-10-0-250-100.ec2.internal   Ready    worker   69m   v1.23.0
```

クラスターが適切に起動しなかった場合、etcd バックアップを使用してクラスターを復元する必要がある場合があります。

関連情報

- クラスターが再起動後に回復しない場合に etcd バックアップを使用して復元する方法については、クラスターの直前の状態への復元を参照してください。

第4章 アプリケーションのバックアップおよび復元

4.1. OADP リリースノート

OpenShift API for Data Protection (OADP) のリリースノートでは、新機能と拡張機能、非推奨の機能、製品の推奨事項、既知の問題、および解決された問題について説明します。

4.1.1. OADP 1.2.0 リリースノート

OADP 1.2.0 リリースノートには、新機能、バグ修正、既知の問題に関する情報が含まれています。

4.1.1.1. 新機能

リソースタイムアウト

新しい **resourceTimeout** オプションは、さまざまな Velero リソースを待機するタイムアウト期間を分単位で指定します。このオプションは、Velero CRD の可用性、**volumeSnapshot** の削除、バックアップリポジトリーの可用性などのリソースに適用されます。デフォルトの継続時間は 10 分です。

AWS S3 互換のバックアップストレージプロバイダー

AWS S3 互換プロバイダーでオブジェクトとスナップショットをバックアップできます。

4.1.1.1.1. テクニカルプレビュー機能

Data Mover

OADP Data Mover を使用すると、Container Storage Interface (CSI) ボリュームのスナップショットをリモートオブジェクトストアにバックアップできます。Data Mover を有効にすると、クラスターの偶発的な削除、クラスター障害、またはデータ破損が発生した場合に、オブジェクトストアから取得した CSI ボリュームスナップショットを使用してステートフルアプリケーションを復元できます。

重要

OADP Data Mover はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

4.1.1.2. バグが修正されました。

このリリースでは、次のバグが修正されています。

- OADP-144
- OADP-639
- OADP-1741
- OADP-1152

- OADP-1143
- OADP-1931
- OADP-148
- OADP-1067
- OADP-1332
- OADP-1164
- OADP-1105
- OADP-2009
- OADP-1370
- OADP-969
- OADP-1672
- OADP-1151
- OADP-988
- OADP-1941
- OADP-1830
- OADP-1821
- OADP-1783
- OADP-1719
- OADP-1833
- OADP-1872
- OADP-2047
- OADP-1932
- OADP-1844
- OADP-1182
- OADP-1183
- OADP-1798
- OADP-1726
- OADP-821
- OADP-1781

- OADP-697
- OADP-1281
- OADP-1077
- OADP-1076
- OADP-1670
- OADP-1307
- OADP-1640
- OADP-1987
- OADP-1934

4.1.1.3. 既知の問題

本リリースには既知の問題はありません。

4.1.2. OADP 1.1.4 リリースノート

OADP 1.1.4 リリースノートには、新機能、解決された問題とバグ、既知の問題がリストされています。

4.1.2.1. 新機能

このバージョンの OADP はサービスリリースです。このバージョンには新しい機能は追加されていません。

4.1.2.2. バグが修正されました。

このリリースでは、次のバグが修正されています。

- OADP-1557
- OADP-1822
- OADP-1511
- OADP-1642
- OADP-1398
- OADP-1267
- OADP-1390
- OADP-1650
- OADP-1487

4.1.2.3. 既知の問題

本リリースには、以下の既知の問題があります。

- アプリケーションがリストアされたクラスター上で UID/GID 範囲が変更された可能性があるため、OADP バックアップが失敗する可能性があり、その結果、OADP が OpenShift Container Platform の UID/GID 範囲メタデータをバックアップおよびリストアしません。この問題を回避するには、バックアップされたアプリケーションに特定の UUID が必要な場合、復元時にその範囲が使用可能であることを確認してください。追加の回避策として、OADP が復元操作で namespace を作成できるようにすることが挙げられます。

- ArgoCD の **app.kubernetes.io/instance** で使用されるラベルが原因で ArgoCD がプロセス中に使用されると、復元が失敗する場合があります。このラベルは、ArgoCD が管理する必要があるリソースを識別します。これにより、復元時にリソースを管理するための OADP の手順と競合が発生する可能性があります。この問題を回避するには、ArgoCD YAML の **.spec.resourceTrackingMethod** を **annotation+label** または **annotation** に設定します。問題が解決しない場合は、復元を開始する前に ArgoCD を無効にし、復元が完了したら再び有効にします。

4.1.3. OADP 1.1.2 リリースノート

OADP 1.1.2 リリースノートには、製品の推奨事項、修正されたバグのリスト、および既知の問題の説明が含まれています。

4.1.3.1. 製品の推奨事項

VolSync

VolSync 0.5.1 から VolSync **stable** チャネルから入手可能な最新バージョンへのアップグレードを準備するには、次のコマンドを実行して、このアノテーションを **openshift-adp** namespace に追加する必要があります。

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

Velero

このリリースでは、Velero がバージョン 1.9.2 からバージョン 1.9.5 にアップグレードされました。

Restic

このリリースでは、Restic がバージョン 0.13.1 からバージョン 0.14.0 にアップグレードされました。

4.1.3.2. バグが修正されました。

このリリースでは、次のバグが修正されています。

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

4.1.3.3. 既知の問題

本リリースには、以下の既知の問題があります。

- OADP は現在、Velero で restic を使用した AWS EFS ボリュームのバックアップと復元をサポートしていません (**OADP-778**)。

- PVC ごとの **VolumeSnapshotContent** スナップショットの Ceph 制限により、CSI バックアップが失敗する場合があります。
  同じ永続ボリューム要求 (PVC) のスナップショットを複数作成できますが、スナップショットの定期的な作成をスケジュールすることはできません。
  - CephFS の場合、PVC ごとに最大 100 スナップショットを作成できます。(OADP-804)
  - RADOS ブロックデバイス (RBD) の場合は、PVC ごとに最大 512 個のスナップショットを作成できます。(OADP-975)

  詳細は、ボリュームのスナップショット を参照してください。

4.1.4. OADP 1.1.1 リリースノート

OADP 1.1.1 リリースノートには、製品の推奨事項と既知の問題の説明が含まれています。

4.1.4.1. 製品の推奨事項

OADP 1.1.1 をインストールする前に、VolSync 0.5.1 をインストールするか、それにアップグレードすることをお勧めします。

4.1.4.2. 既知の問題

本リリースには、以下の既知の問題があります。

- OADP は現在、Velero で restic を使用した AWS EFS ボリュームのバックアップと復元をサポートしていません (OADP-778)。
- PVC ごとの **VolumeSnapshotContent** スナップショットの Ceph 制限により、CSI バックアップが失敗する場合があります。
  同じ永続ボリューム要求 (PVC) のスナップショットを複数作成できますが、スナップショットの定期的な作成をスケジュールすることはできません。
  - CephFS の場合、PVC ごとに最大 100 スナップショットを作成できます。
  - RADOS ブロックデバイス (RBD) の場合は、PVC ごとに最大 512 個のスナップショットを作成できます。(OADP-804) および (OADP-975)
    詳細は、ボリュームのスナップショット を参照してください。

4.2. OADP FEATURES AND PLUGINS

OpenShift API for Data Protection (OADP) 機能は、アプリケーションをバックアップおよび復元するためのオプションを提供します。

デフォルトのプラグインにより、Velero は特定のクラウドプロバイダーと統合し、OpenShift Container Platform リソースをバックアップおよび復元できます。

4.2.1. OADP の機能

OpenShift API for Data Protection (OADP) は、以下の機能をサポートします。

バックアップ

OADP を使用して OpenShift Platform 上のすべてのアプリケーションをバックアップしたり、タイプ、namespace、またはラベルでリソースをフィルターしたりできます。
OADP は、Kubernetes オブジェクトと内部イメージをアーカイブファイルとしてオブジェクトスト

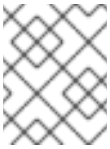
レージに保存することにより、それらをバックアップします。OADP は、ネイティブクラウドスナップショット API または Container Storage Interface (CSI) を使用してスナップショットを作成することにより、永続ボリューム (PV) をバックアップします。スナップショットをサポートしないクラウドプロバイダーの場合、OADP は Restic を使用してリソースと PV データをバックアップします。

注記

バックアップと復元を成功させるには、アプリケーションのバックアップから Operator を除外する必要があります。

復元

バックアップからリソースと PV を復元できます。バックアップ内のすべてのオブジェクトを復元することも、復元されたオブジェクトを namespace、PV、またはラベルでフィルタリングすることもできます。



注記

バックアップと復元を成功させるには、アプリケーションのバックアップから Operator を除外する必要があります。

スケジュール

指定した間隔でバックアップをスケジュールできます。

フック

フックを使用して、Pod 上のコンテナーでコマンドを実行できます。たとえば、**fsfreeze** を使用してファイルシステムをフリーズできます。バックアップまたは復元の前または後に実行するようにフックを設定できます。復元フックは、init コンテナーまたはアプリケーションコンテナーで実行できます。

4.2.2. OADP プラグイン

OpenShift API for Data Protection (OADP) は、バックアップおよびスナップショット操作をサポートするためにストレージプロバイダーと統合されたデフォルトの Velero プラグインを提供します。Velero プラグインに基づいて カスタムプラグイン を作成できます。

OADP は、OpenShift Container Platform リソースバックアップ、OpenShift Virtualization リソースバックアップ、および Container Storage Interface (CSI) スナップショット用のプラグインも提供します。

表4.1 OADP プラグイン

OADP プラグイン	機能	ストレージの場所
aws	Kubernetes オブジェクトをバックアップし、復元します。	AWS S3
	スナップショットを使用してボリュームをバックアップおよび復元します。	AWS EBS

OADP プラグイン	機能	ストレージの場所
azure	Kubernetes オブジェクトをバックアップし、復元します。	Microsoft Azure Blob ストレージ
	スナップショットを使用してボリュームをバックアップおよび復元します。	Microsoft Azure マネージドディスク
gcp	Kubernetes オブジェクトをバックアップし、復元します。	Google Cloud Storage
	スナップショットを使用してボリュームをバックアップおよび復元します。	Google Compute Engine ディスク
openshift	OpenShift Container Platform リソースをバックアップおよび復元します。[1]	オブジェクトストア
kubevirt	OpenShift Virtualization リソースをバックアップおよび復元します。[2]	オブジェクトストア
csi	CSI スナップショットを使用して、ボリュームをバックアップおよび復元します。[3]	CSI スナップショットをサポートするクラウドストレージ

1. 必須。
2. 仮想マシンディスクは CSI スナップショットまたは Restic でバックアップされます。
3. **csi** プラグインは、Velero CSI ベータスナップショット API を使用します。

4.2.3. OADP Velero プラグインについて

Velero のインストール時に、次の 2 種類のプラグインを設定できます。

- デフォルトのクラウドプロバイダープラグイン
- カスタムプラグイン

どちらのタイプのプラグインもオプションですが、ほとんどのユーザーは少なくとも 1 つのクラウドプロバイダープラグインを設定します。

4.2.3.1. デフォルトの Velero クラウドプロバイダープラグイン

デプロイメント中に **oadp_v1alpha1_dpa.yaml** ファイルを設定するときに、次のデフォルトの Velero クラウドプロバイダープラグインのいずれかをインストールできます。

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

デプロイメント中に **oadp_v1alpha1_dpa.yaml** ファイルで目的のデフォルトプラグインを指定します。

ファイルの例:

次の **.yaml** ファイルは、**openshift**、**aws**、**azure**、および **gcp** プラグインをインストールします。

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

4.2.3.2. カスタム Velero プラグイン

デプロイメント中に **oadp_v1alpha1_dpa.yaml** ファイルを設定するときに、プラグインの **image** と **name** を指定することにより、カスタム Velero プラグインをインストールできます。

デプロイメント中に **oadp_v1alpha1_dpa.yaml** ファイルで目的のカスタムプラグインを指定します。

ファイルの例:

次の **.yaml** ファイルは、デフォルトの **openshift**、**azure**、および **gcp** プラグインと、イメージ **quay.io/example-repo/custom-velero-plugin** を持つ **custom-plugin-example** という名前のカスタムプラグインをインストールします。

```
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. OADP のインストールおよび設定

4.3.1. OADP のインストールについて

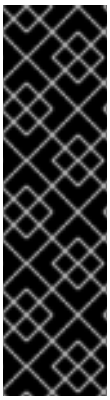
クラスター管理者は、OADP Operator をインストールして、OpenShift API for Data Protection (OADP) をインストールします。OADP オペレーターは Velero 1.7 をインストールします。

注記

OADP 1.0.4 以降、すべて OADP 1.0.**z** バージョンは、MTC Operator の依存関係としてのみ使用でき、スタンドアロン Operator としては使用できません。

Kubernetes リソースと内部イメージをバックアップするには、次のいずれかのストレージタイプなど、バックアップ場所としてオブジェクトストレージが必要です。

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Multicloud Object Gateway
- Noobaa や Minio などの AWS S3 互換のオブジェクトストレージ



重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

スナップショットまたは Restic を使用して、永続ボリューム (PV) をバックアップできます。

スナップショットを使用して PV をバックアップするには、ネイティブスナップショット API または Container Storage Interface (CSI) スナップショットのいずれかをサポートするクラウドプロバイダー (次のいずれかのクラウドプロバイダーなど) が必要です。

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- OpenShift Data Foundation などの CSI スナップショット対応クラウドプロバイダー

クラウドプロバイダーがスナップショットをサポートしていない場合、またはストレージが NFS の場合は、Restic バックアップ を使用してオブジェクトストレージにアプリケーションをバックアップできます。

デフォルトの **Secret** を作成し、次に、Data Protection Application をインストールします。

4.3.1.1. AWS S3 互換のバックアップストレージプロバイダー

OADP は、さまざまなバックアップおよびスナップショット操作で使用できる多数のオブジェクトストレージプロバイダーと互換性があります。いくつかのオブジェクトストレージプロバイダーは完全にサポートされていますが、いくつかはサポートされていないものの動作することがわかっており、一部には既知の制限があります。

4.3.1.1.1. サポートされているバックアップストレージプロバイダー

次の AWS S3 互換オブジェクトストレージプロバイダーは、バックアップストレージの場所として使用するために、AWS プラグインを介して OADP によって完全にサポートされています。

- MinIO
- NooBaa を備えた Multicloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3

注記

次の互換オブジェクトストレージプロバイダーはサポートされており、独自の Velero オブジェクトストアプラグインがあります。

- Google Cloud Platform (GCP)
- Microsoft Azure

4.3.1.1.2. サポートされていないバックアップストレージプロバイダー

次の AWS S3 互換オブジェクトストレージプロバイダーは、バックアップストレージの場所として使用するために、AWS プラグインを介して Velero と連携することが知られていますが、サポートされておらず、Red Hat によってテストされていません。

- IBM Cloud
- Oracle Cloud
- DigitalOcean
- NooBaa
- Tencent Cloud
- Ceph RADOS v12.2.7
- Quobyte
- Cloudian HyperStore

4.3.1.1.3. 既知の制限があるバックアップストレージプロバイダー

次の AWS S3 互換オブジェクトストレージプロバイダーは、限定された機能セットを備えた AWS プラグインを介して Velero と連携することが知られています。

- Swift - バックアップストレージのバックアップストレージ場所として使用できますが、ファイルシステムベースのボリュームバックアップおよび復元については Restic と互換性がありません。

4.3.1.2. OpenShift Data Foundation での障害復旧のための NooBaa の設定

OpenShift Data Foundation で NooBaa バケットの **backupStorageLocation** にクラスターストレージを使用する場合は、NooBaa を外部オブジェクトストアとして設定します。

警告

NooBaa を外部オブジェクトストアとして設定しないと、バックアップが利用できなくなる可能性があります。

手順

- ハイブリッドまたはマルチクラウドのストレージリソースの追加 の説明に従って、NooBaa を外部オブジェクトストアとして設定します。

関連情報

- Velero ドキュメント のバックアップ場所およびスナップショット場所の概要。

/// モジュールは、以下のアセンブリーに含まれています。

4.3.1.3. OADP 更新チャネルについて

OADP Operator をインストールするときに、**更新チャネル** を選択します。このチャネルにより、OADP Operator と Velero のどちらのアップグレードを受け取るかが決まります。いつでもチャンネルを切り替えることができます。

次の更新チャネルを利用できます。

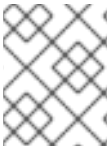
- **stable** チャネルは非推奨になりました。**stable** チャネルには、**oadp.v1.1.z** および **oadp.v1.0.z** の古いバージョン用の OADP **ClusterServiceVersion** のパッチ (z-stream 更新) が含まれています。
- **stable-1.0** チャネルには **oadp.v1.0.z**、OADP 1.0 **ClusterServiceVersion** が含まれています。
- **stable-1.1** チャネルには **oadp.v1.1.z**、最新の OADP 1.1 **ClusterServiceVersion** が含まれています。
- **stable-1.2** チャネルには、最新の OADP 1.2 **ClusterServiceVersion** の **oadp.v1.2.z** が含まれています。

適切な更新チャネルはどれですか?

- **stable** チャネルは非推奨になりました。すでに安定版チャネルを使用している場合は、引き続き、**oadp.v1.1.z** から更新を取得します。
- OADP 1.**y** をインストールする **stable-1.y** 更新チャネルを選択し、そのパッチを引き続き受け取ります。このチャネルを選択すると、バージョン 1.**y.z** のすべての z-stream パッチを受け取ります。

いつ更新チャネルを切り替える必要がありますか?

- OADP 1.**y** がインストールされていて、その y-stream のパッチのみを受け取りたい場合は、**stable** 更新チャネルから **stable-1.y** 更新チャネルに切り替える必要があります。その後、バージョン 1.**y.z** のすべての z-stream パッチを受け取ります。
- OADP 1.0 がインストールされていて、OADP 1.1 にアップグレードしたい場合、OADP 1.1 のみのパッチを受け取るには、**stable-1.0** 更新チャネルから **stable-1.1** 更新チャネルに切り替える必要があります。その後、バージョン 1.1.**z** のすべての z-stream パッチを受け取ります。
- OADP 1.**y** がインストールされていて、**y** が 0 より大きく、OADP 1.0 に切り替える場合は、OADP Operator を **アンインストール** してから、**stable-1.0** 更新チャネルを使用して再インストールする必要があります。その後、バージョン 1.0.**z** のすべての z-stream パッチを受け取ります。



注記

更新チャネルを切り替えて、OADP 1.**y** から OADP 1.0 に切り替えることはできません。Operator をアンインストールしてから再インストールする必要があります。

4.3.1.4. 複数の namespace への OADP のインストール

OADP を同じクラスター上の複数の namespace にインストールすると、複数のプロジェクト所有者が独自の OADP インスタンスを管理できるようになります。このユースケースは Restic および CSI で検証されています。

本書に含まれるプラットフォームごとの手順で指定されている OADP の各インスタンスを、以下の追加要件とともにインストールします。

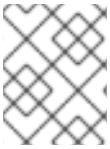
- 同じクラスター上のすべての OADP デプロイメントは、同じバージョン (1.1.4 など) である必要があります。同じクラスターに異なるバージョンの OADP をインストールすることはサポート**されていません**。
- OADP の個々のデプロイメントには、一意の認証情報のセットと一意の **BackupStorageLocation** 設定が必要です。
- デフォルトでは、各 OADP デプロイメントには namespace 全体でクラスターレベルのアクセス権があります。OpenShift Container Platform 管理者は、セキュリティーおよび RBAC 設定を注意深く確認し、必要な変更を加えて、各 OADP インスタンスに正しい権限があることを確認する必要があります。

関連情報

- クラスターサービスバージョン

4.3.2. Amazon Web Services を使用した OpenShift API for Data Protection のインストールおよび設定

OADP Operator をインストールすることで、Amazon Web Services (AWS) を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.7 をインストールします。



注記

OADP 1.0.4 以降、すべて OADP 1.0.**z** バージョンは、MTC Operator の依存関係としてのみ使用でき、スタンドアロン Operator としては使用できません。

Velero 向けに AWS を設定し、デフォルトの **Secret** を作成し、次に、Data Protection Application をインストールします。

制限されたネットワーク環境に OADP Operator をインストールするには、最初にデフォルトの Operator Hub ソースを無効にして、Operator カタログをミラーリングする必要があります。詳細は、ネットワークが制限された環境での Operator Lifecycle Manager の使用 を参照してください。

4.3.2.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.10 に OpenShift API for Data Protection (OADP) オペレーターをインストールします。

OADP オペレーターは Velero 1.7 をインストールします。

前提条件

- **cluster-admin** 権限を持つユーザーとしてログインしている。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** をクリックします。
2. **Filter by keyword** フィールドを使用して、**OADP Operator** を検索します。
3. **OADP Operator** を選択し、**Install** をクリックします。
4. **Install** をクリックして、**openshift-adp** プロジェクトに Operator をインストールします。
5. **Operators → Installed Operators** をクリックして、インストールを確認します。

4.3.2.2. Amazon Web Services の設定

OpenShift API for Data Protection (OADP) 用に Amazon Web Services (AWS) を設定します。

前提条件

- AWS CLI がインストールされていること。

手順

1. **BUCKET** 変数を設定します。

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

2. **REGION** 変数を設定します。

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

3. AWS S3 バケットを作成します。

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

1 **us-east-1** は **LocationConstraint** をサポートしていません。お住まいの地域が **us-east-1** の場合は、**--create-bucket-configuration LocationConstraint=$REGION** を省略してください。

4. IAM ユーザーを作成します。

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

1 Velero を使用して複数の S3 バケットを持つ複数のクラスターをバックアップする場合は、クラスターごとに一意のユーザー名を作成します。

5. **velero-policy.json** ファイルを作成します。

```
$ 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. ポリシーを添付して、**velero** ユーザーに必要最小限の権限を付与します。

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

7. **velero** ユーザーのアクセスキーを作成します。

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

出力例

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

8. **credentials-velero** ファイルを作成します。

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

Data Protection Application をインストールする前に、**credentials-velero** ファイルを使用して AWS の **Secret** オブジェクトを作成します。

4.3.2.3. バックアップおよびスナップショットの場所、ならびにそのシークレットについて

DataProtectionApplication カスタムリソース (CR) で、バックアップおよびスナップショットの場所、ならびにそのシークレットを指定します。

バックアップの場所

Multicloud Object Gateway、Noobaa、または Minio などの S3 互換オブジェクトストレージを、バックアップの場所として指定します。

Velero は、オブジェクトストレージのアーカイブファイルとして、OpenShift Container Platform リソース、Kubernetes オブジェクト、および内部イメージをバックアップします。

スナップショットの場所
クラウドプロバイダーのネイティブスナップショット API を使用して永続ボリュームをバックアップする場合、クラウドプロバイダーをスナップショットの場所として指定する必要があります。

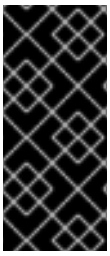
Container Storage Interface (CSI) スナップショットを使用する場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成するため、スナップショットの場所を指定する必要はありません。

Restic を使用する場合は、Restic がオブジェクトストレージにファイルシステムをバックアップするため、スナップショットの場所を指定する必要はありません。

シークレット
バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの secret オブジェクトを作成します。

- **DataProtectionApplication** CR で指定する、バックアップの場所用のカスタム **Secret**。
- **DataProtectionApplication** CR で参照されない、スナップショットの場所用のデフォルト **Secret**。



重要

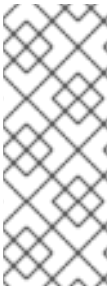
Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。

4.3.2.3.1. デフォルト Secret の作成

バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

Secret のデフォルト名は **cloud-credentials** です。



注記

DataProtectionApplication カスタムリソース (CR) にはデフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。バックアップの場所の **Secret** の名前が指定されていない場合は、デフォルトの名前が使用されます。

インストール時にバックアップの場所の認証情報を使用しない場合は、空の **credentials-velero** ファイルを使用してデフォルト名前で **Secret** を作成できます。

前提条件

- オブジェクトストレージとクラウドストレージがある場合は、同じ認証情報を使用する必要があります。
- Velero のオブジェクトストレージを設定する必要があります。
- オブジェクトストレージ用の **credentials-velero** ファイルを適切な形式で作成する必要があります。

手順

- デフォルト名で **Secret** を作成します。

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

Secret は、Data Protection Application をインストールするときに、**DataProtectionApplication** CR の **spec.backupLocations.credential** ブロックで参照されます。

4.3.2.3.2. 異なる認証情報のプロファイルの作成

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、**credentials-velero** ファイルに個別のプロファイルを作成します。

次に、**Secret** オブジェクトを作成し、**DataProtectionApplication** カスタムリソース (CR) でプロファイルを指定します。

手順

1. 次の例のように、バックアップとスナップショットの場所に別々のプロファイルを持つ **credentials-velero** ファイルを作成します。

```
[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. **credentials-velero** ファイルを使用して **Secret** オブジェクトを作成します。

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

3. 次の例のように、プロファイルを **DataProtectionApplication** CR に追加します。

```
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:
    - name: default
      velero:
        provider: aws
        config:
          region: us-west-2
          profile: "volumeSnapshot"
```

4.3.2.4. Data Protection Application の設定

Velero リソースの割り当てを設定するか、自己署名 CA 証明書を有効にして、Data Protection Application を設定できます。

4.3.2.4.1. Velero の CPU とメモリーのリソース割り当てを設定

DataProtectionApplication カスタムリソース (CR) マニフェストを編集して、**Velero** Pod の CPU およびメモリーリソースの割り当てを設定します。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR マニフェストの **spec.configuration.velero.podConfig.ResourceAllocations** ブロックの値を編集します。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> (1)
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
```

```
        requests:
          cpu: 500m
          memory: 256Mi
```

1 1 Velero podSpec に提供されるノードセレクターを指定します。

4.3.2.4.2. 自己署名 CA 証明書の有効化

certificate signed by unknown authority エラーを防ぐために、**DataProtectionApplication** カスタムリソース (CR) マニフェストを編集して、オブジェクトストレージの自己署名 CA 証明書を有効にする必要があります。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- **DataProtectionApplication** CR マニフェストの **spec.backupLocations.velero.objectStorage.caCert** パラメーターと **spec.backupLocations.velero.config** パラメーターを編集します。

```
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 Base46 でエンコードされた CA 証明書文字列を指定します。

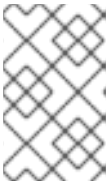
2 **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.3.2.5. Data Protection Application のインストール

DataProtectionApplication API のインスタンスを作成して、Data Protection Application (DPA) をインストールします。

前提条件

- OADP Operator をインストールする必要があります。
- オブジェクトストレージをバックアップ場所として設定する必要があります。
- スナップショットを使用して PV をバックアップする場合、クラウドプロバイダーはネイティブスナップショット API または Container Storage Interface (CSI) スナップショットのいずれかをサポートする必要があります。
- バックアップとスナップショットの場所で同じ認証情報を使用する場合は、デフォルトの名前である **cloud-credentials** を使用して **Secret** を作成する必要があります。
- バックアップとスナップショットの場所で異なる認証情報を使用する場合は、デフォルト名である **cloud-credentials** を使用して **Secret** を作成する必要があります。これには、バックアップとスナップショットの場所の認証情報用の個別のプロファイルが含まれます。



注記

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。デフォルトの **Secret** がない場合、インストールは失敗します。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。
2. **Provided APIs** で、**DataProtectionApplication** ボックスの **Create instance** をクリックします。
3. **YAML View** をクリックして、**DataProtectionApplication** マニフェストのパラメーターを更新します。

```
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"
      credential:
        key: cloud
        name: cloud-credentials 7
  snapshotLocations: 8
   - name: default
     velero:
       provider: aws
       config:
         region: <region> 9
         profile: "default"
```

1. **openshift** プラグインは必須です。
2. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
3. Restic のインストールを無効にする場合は、**false** に設定します。Restic はデーモンセットをデプロイします。これは、各ワーカーノードで **Restic** Pod が実行されていることを意味します。**spec.defaultVolumesToRestic: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。
4. Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。
5. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
6. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。
7. 作成した **Secret** オブジェクトの名前を指定します。この値を指定しない場合は、デフォルト名の **cloud-credentials** が使用されます。カスタム名を指定すると、バックアップの場所にカスタム名が使用されます。
8. CSI スナップショットまたは Restic を使用して PV をバックアップする場合を除き、スナップショットの場所を指定します。
9. スナップショットの場所は、PV と同じリージョンにある必要があります。

4. **Create** をクリックします。
5. OADP リソースを表示して、インストールを確認します。

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

出力例

```
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
```

4.3.2.5.1. DataProtectionApplication CR で CSI を有効にする

CSI スナップショットを使用して永続ボリュームをバックアップするには、**DataProtectionApplication** カスタムリソース (CR) で Container Storage Interface (CSI) を有効にします。

前提条件

- クラウドプロバイダーは、CSI スナップショットをサポートする必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR を編集します。

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

1 **csi** デフォルトプラグインを追加します。

4.3.3. Microsoft Azure を使用した OpenShift API for Data Protection のインストールおよび設定

OADP Operator をインストールすることで、Microsoft Azure を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.7 をインストールします。

注記

OADP 1.0.4 以降、すべて OADP 1.0.**z** バージョンは、MTC Operator の依存関係としてのみ使用でき、スタンドアロン Operator としては使用できません。

Velero 向けに Azure を設定し、デフォルトの **Secret** を作成し、次に、Data Protection Application をインストールします。

制限されたネットワーク環境に OADP Operator をインストールするには、最初にデフォルトの Operator Hub ソースを無効にして、Operator カタログをミラーリングする必要があります。詳細は、ネットワークが制限された環境での Operator Lifecycle Manager の使用 を参照してください。

4.3.3.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.10 に OpenShift API for Data Protection (OADP) オペレーターをインストールします。

OADP オペレーターは Velero 1.7 をインストールします。

前提条件

- **cluster-admin** 権限を持つユーザーとしてログインしている。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** をクリックします。
2. **Filter by keyword** フィールドを使用して、**OADP Operator** を検索します。
3. **OADP Operator** を選択し、**Install** をクリックします。
4. **Install** をクリックして、**openshift-adp** プロジェクトに Operator をインストールします。
5. **Operators → Installed Operators** をクリックして、インストールを確認します。

4.3.3.2. Microsoft Azure の設定

OpenShift API for Data Protection (OADP) 用に Microsoft Azure を設定します。

前提条件

- Azure CLI がインストールされていること。

手順

1. Azure にログインします。

```
$ az login
```

2. **AZURE_RESOURCE_GROUP** 変数を設定します。

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

3. Azure リソースグループを作成します。

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

(1) 場所を指定します。

4. **AZURE_STORAGE_ACCOUNT_ID** 変数を設定します。

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

5. Azure ストレージアカウントを作成します。

```
$ 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. **BLOB_CONTAINER** 変数を設定します。

```
$ BLOB_CONTAINER=velero
```

7. Azure Blob ストレージコンテナーを作成します。

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

8. ストレージアカウントのアクセスキーを取得します。

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

9. 必要最小限のパーミッションを持つカスタムロールを作成します。

```
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'"]
}'
```

10. **credentials-velero** ファイルを作成します。

```
$ 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) 必須。**credentials-velero** ファイルにサービスプリンシパル認証情報のみが含まれている場合は、内部イメージをバックアップすることはできません。

Data Protection Application をインストールする前に、**credentials-velero** ファイルを使用して Azure の **Secret** オブジェクトを作成します。

4.3.3.3. バックアップおよびスナップショットの場所、ならびにそのシークレットについて

DataProtectionApplication カスタムリソース (CR) で、バックアップおよびスナップショットの場所、ならびにそのシークレットを指定します。

バックアップの場所
Multicloud Object Gateway、Noobaa、または Minio などの S3 互換オブジェクトストレージを、バックアップの場所として指定します。

Velero は、オブジェクトストレージのアーカイブファイルとして、OpenShift Container Platform リソース、Kubernetes オブジェクト、および内部イメージをバックアップします。

スナップショットの場所
クラウドプロバイダーのネイティブスナップショット API を使用して永続ボリュームをバックアップする場合、クラウドプロバイダーをスナップショットの場所として指定する必要があります。

Container Storage Interface (CSI) スナップショットを使用する場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成するため、スナップショットの場所を指定する必要はありません。

Restic を使用する場合は、Restic がオブジェクトストレージにファイルシステムをバックアップするため、スナップショットの場所を指定する必要はありません。

シークレット
バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの secret オブジェクトを作成します。

- **DataProtectionApplication** CR で指定する、バックアップの場所用のカスタム **Secret**。
- **DataProtectionApplication** CR で参照されない、スナップショットの場所用のデフォルト **Secret**。

重要

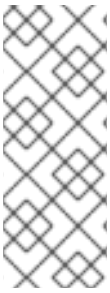
Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。

4.3.3.3.1. デフォルト Secret の作成

バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

Secret のデフォルト名は **cloud-credentials-azure** です。



注記

DataProtectionApplication カスタムリソース (CR) にはデフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。バックアップの場所の **Secret** の名前が指定されていない場合は、デフォルトの名前が使用されます。

インストール時にバックアップの場所の認証情報を使用しない場合は、空の **credentials-velero** ファイルを使用してデフォルト名前で **Secret** を作成できます。

前提条件

- オブジェクトストレージとクラウドストレージがある場合は、同じ認証情報を使用する必要があります。
- Velero のオブジェクトストレージを設定する必要があります。
- オブジェクトストレージ用の **credentials-velero** ファイルを適切な形式で作成する必要があります。

手順

- デフォルト名で **Secret** を作成します。

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

Secret は、Data Protection Application をインストールするときに、 **DataProtectionApplication** CR の **spec.backupLocations.credential** ブロックで参照されます。

4.3.3.3.2. 異なる認証情報のシークレットの作成

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの **Secret** オブジェクトを作成する必要があります。

- カスタム名を持つバックアップ場所の **Secret**。カスタム名は、**DataProtectionApplication** カスタムリソース (CR) の **spec.backupLocations** ブロックで指定されます。
- スナップショットの場所 **Secret** (デフォルト名は **cloud-credentials-azure**)。この **Secret** は、**DataProtectionApplication** で指定されていません。

手順

1. スナップショットの場所の **credentials-velero** ファイルをクラウドプロバイダーに適した形式で作成します。
2. デフォルト名でスナップショットの場所の **Secret** を作成します。

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

3. オブジェクトストレージに適した形式で、バックアップ場所の **credentials-velero** ファイルを作成します。
4. カスタム名を使用してバックアップ場所の **Secret** を作成します。

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

5. 次の例のように、カスタム名の **Secret** を **DataProtectionApplication** に追加します。

```
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"
    name: default
    provider: azure
```

1 カスタム名を持つバックアップ場所の **Secret**。

4.3.3.4. Data Protection Application の設定

Velero リソースの割り当てを設定するか、自己署名 CA 証明書を有効にして、Data Protection Application を設定できます。

4.3.3.4.1. Velero の CPU とメモリーのリソース割り当てを設定

DataProtectionApplication カスタムリソース (CR) マニフェストを編集して、**Velero** Pod の CPU およびメモリーリソースの割り当てを設定します。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR マニフェストの **spec.configuration.velero.podConfig.ResourceAllocations** ブロックの値を編集します。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 Velero podSpec に提供されるノードセレクターを指定します。

4.3.3.4.2. 自己署名 CA 証明書の有効化

certificate signed by unknown authority エラーを防ぐために、**DataProtectionApplication** カスタムリソース (CR) マニフェストを編集して、オブジェクトストレージの自己署名 CA 証明書を有効にする必要があります。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- **DataProtectionApplication** CR マニフェストの **spec.backupLocations.velero.objectStorage.caCert** パラメーターと **spec.backupLocations.velero.config** パラメーターを編集します。

```
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. Base46 でエンコードされた CA 証明書文字列を指定します。
2. **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.3.3.5. Data Protection Application のインストール

DataProtectionApplication API のインスタンスを作成して、Data Protection Application (DPA) をインストールします。

前提条件

- OADP Operator をインストールする必要があります。
- オブジェクトストレージをバックアップ場所として設定する必要があります。
- スナップショットを使用して PV をバックアップする場合、クラウドプロバイダーはネイティブスナップショット API または Container Storage Interface (CSI) スナップショットのいずれかをサポートする必要があります。
- バックアップとスナップショットの場所で同じ認証情報を使用する場合は、デフォルトの名前である **cloud-credentials-azure** を使用して **Secret** を作成する必要があります。

- バックアップとスナップショットの場所で異なる認証情報を使用する場合は、2 つの **Secrets** を作成する必要があります。
  - バックアップ場所のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。
  - スナップショットの場所のデフォルト名 **cloud-credentials-azure** の **Secret**。この **Secret** は、**DataProtectionApplication** CR では参照されません。

注記

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。デフォルトの **Secret** がない場合、インストールは失敗します。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。
2. **Provided APIs** で、**DataProtectionApplication** ボックスの **Create instance** をクリックします。
3. **YAML View** をクリックして、**DataProtectionApplication** マニフェストのパラメーターを更新します。

```
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** プラグインは必須です。
2. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
3. Restic のインストールを無効にする場合は、**false** に設定します。Restic はデーモンセットをデプロイします。これは、各ワーカーノードで **Restic** Pod が実行されていることを意味します。**spec.defaultVolumesToRestic: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。
4. Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。
5. Azure リソースグループを指定します。
6. Azure ストレージアカウント ID を指定します。
7. Azure サブスクリプション ID を指定します。
8. この値を指定しない場合は、デフォルト名の **cloud-credentials-azure** が使用されます。カスタム名を指定すると、バックアップの場所にカスタム名が使用されます。
9. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
10. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。
11. CSI スナップショットまたは Restic を使用して PV をバックアップする場合は、スナップショットの場所を指定する必要はありません。

4. **Create** をクリックします。
5. OADP リソースを表示して、インストールを確認します。

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

出力例

```
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
```

4.3.3.5.1. DataProtectionApplication CR で CSI を有効にする

CSI スナップショットを使用して永続ボリュームをバックアップするには、**DataProtectionApplication** カスタムリソース (CR) で Container Storage Interface (CSI) を有効にします。

前提条件

- クラウドプロバイダーは、CSI スナップショットをサポートする必要があります。

手順

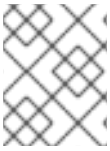
- 次の例のように、**DataProtectionApplication** CR を編集します。

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

1 **csi** デフォルトプラグインを追加します。

4.3.4. Google Cloud Platform を使用した OpenShift API for Data Protection のインストールおよび設定

OADP Operator をインストールすることで、Google Cloud Platform (GCP) を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.7 をインストールします。



注記

OADP 1.0.4 以降、すべて OADP 1.0.**z** バージョンは、MTC Operator の依存関係としてのみ使用でき、スタンドアロン Operator としては使用できません。

Velero 向けに GCP を設定し、デフォルトの **Secret** を作成し、次に、Data Protection Application をインストールします。

制限されたネットワーク環境に OADP Operator をインストールするには、最初にデフォルトの Operator Hub ソースを無効にして、Operator カタログをミラーリングする必要があります。詳細は、ネットワークが制限された環境での Operator Lifecycle Manager の使用 を参照してください。

4.3.4.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.10 に OpenShift API for Data Protection (OADP) オペレーターをインストールします。

OADP オペレーターは Velero 1.7 をインストールします。

前提条件

- **cluster-admin** 権限を持つユーザーとしてログインしている。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** をクリックします。
2. **Filter by keyword** フィールドを使用して、**OADP Operator** を検索します。
3. **OADP Operator** を選択し、**Install** をクリックします。
4. **Install** をクリックして、**openshift-adp** プロジェクトに Operator をインストールします。
5. **Operators → Installed Operators** をクリックして、インストールを確認します。

4.3.4.2. Google Cloud Provider の設定

OpenShift API for Data Protection (OADP) 用に Google Cloud Platform (GCP) を設定します。

前提条件

- **gcloud** および **gsutil** CLI ツールがインストールされている必要があります。詳細は、Google Cloud のドキュメント をご覧ください。

手順

1. GCP にログインします。

```
$ gcloud auth login
```

2. **BUCKET** 変数を設定します。

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

1 バケット名を指定します。

3. ストレージバケットを作成します。

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

4. **PROJECT_ID** 変数をアクティブなプロジェクトに設定します。

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

5. サービスアカウントを作成します。

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

6. サービスアカウントを一覧表示します。

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

7. **email** の値と一致するように **SERVICE_ACCOUNT_EMAIL** 変数を設定します。

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

8. ポリシーを添付して、**velero** ユーザーに必要最小限の権限を付与します。

```
$ 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. **velero.server** カスタムロールを作成します。

```
$ gcloud iam roles create velero.server \
    --project $PROJECT_ID \
    --title "Velero Server" \
    --permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"
```

10. IAM ポリシーバインディングをプロジェクトに追加します。

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

11. IAM サービスアカウントを更新します。

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

12. IAM サービスアカウントのキーを現在のディレクトリーにある **credentials-velero** ファイルに保存します。

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

Data Protection Application をインストールする前に、**credentials-velero** ファイルを使用して GCP の **Secret** オブジェクトを作成します。

4.3.4.3. バックアップおよびスナップショットの場所、ならびにそのシークレットについて

DataProtectionApplication カスタムリソース (CR) で、バックアップおよびスナップショットの場所、ならびにそのシークレットを指定します。

バックアップの場所
Multicloud Object Gateway、Noobaa、または Minio などの S3 互換オブジェクトストレージを、バックアップの場所として指定します。

Velero は、オブジェクトストレージのアーカイブファイルとして、OpenShift Container Platform リソース、Kubernetes オブジェクト、および内部イメージをバックアップします。

スナップショットの場所
クラウドプロバイダーのネイティブスナップショット API を使用して永続ボリュームをバックアップする場合、クラウドプロバイダーをスナップショットの場所として指定する必要があります。

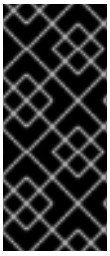
Container Storage Interface (CSI) スナップショットを使用する場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成するため、スナップショットの場所を指定する必要はありません。

Restic を使用する場合は、Restic がオブジェクトストレージにファイルシステムをバックアップするため、スナップショットの場所を指定する必要はありません。

シークレット
バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの secret オブジェクトを作成します。

- **DataProtectionApplication** CR で指定する、バックアップの場所用のカスタム **Secret**。
- **DataProtectionApplication** CR で参照されない、スナップショットの場所用のデフォルト **Secret**。



重要

Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。

4.3.4.3.1. デフォルト Secret の作成

バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

Secret のデフォルト名は **cloud-credentials-gcp** です。

注記

DataProtectionApplication カスタムリソース (CR) にはデフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。バックアップの場所の **Secret** の名前が指定されていない場合は、デフォルトの名前が使用されます。

インストール時にバックアップの場所の認証情報を使用しない場合は、空の **credentials-velero** ファイルを使用してデフォルト名前で **Secret** を作成できます。

前提条件

- オブジェクトストレージとクラウドストレージがある場合は、同じ認証情報を使用する必要があります。
- Velero のオブジェクトストレージを設定する必要があります。
- オブジェクトストレージ用の **credentials-velero** ファイルを適切な形式で作成する必要があります。

手順

- デフォルト名で **Secret** を作成します。

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

Secret は、Data Protection Application をインストールするときに、**DataProtectionApplication** CR の **spec.backupLocations.credential** ブロックで参照されます。

4.3.4.3.2. 異なる認証情報のシークレットの作成

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの **Secret** オブジェクトを作成する必要があります。

- カスタム名を持つバックアップ場所の **Secret**。カスタム名は、**DataProtectionApplication** カスタムリソース (CR) の **spec.backupLocations** ブロックで指定されます。
- スナップショットの場所 **Secret** (デフォルト名は **cloud-credentials-gcp**)。この **Secret** は、**DataProtectionApplication** で指定されていません。

手順

1. スナップショットの場所の **credentials-velero** ファイルをクラウドプロバイダーに適した形式で作成します。
2. デフォルト名でスナップショットの場所の **Secret** を作成します。

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

3. オブジェクトストレージに適した形式で、バックアップ場所の **credentials-velero** ファイルを作成します。
4. カスタム名を使用してバックアップ場所の **Secret** を作成します。

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

5. 次の例のように、カスタム名の **Secret** を **DataProtectionApplication** に追加します。

```
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 カスタム名を持つバックアップ場所の **Secret**。

4.3.4.4. Data Protection Application の設定

Velero リソースの割り当てを設定するか、自己署名 CA 証明書を有効にして、Data Protection Application を設定できます。

4.3.4.4.1. Velero の CPU とメモリーのリソース割り当てを設定

DataProtectionApplication カスタムリソース (CR) マニフェストを編集して、 **Velero** Pod の CPU およびメモリーリソースの割り当てを設定します。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR マニフェストの **spec.configuration.velero.podConfig.ResourceAllocations** ブロックの値を編集します。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> 1
        resourceAllocations:
          limits:
            cpu: "1"
            memory: 512Mi
          requests:
            cpu: 500m
            memory: 256Mi
```

1 Velero podSpec に提供されるノードセレクターを指定します。

4.3.4.4.2. 自己署名 CA 証明書の有効化

certificate signed by unknown authority エラーを防ぐために、**DataProtectionApplication** カスタムリソース (CR) マニフェストを編集して、オブジェクトストレージの自己署名 CA 証明書を有効にする必要があります。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- **DataProtectionApplication** CR マニフェストの **spec.backupLocations.velero.objectStorage.caCert** パラメーターと **spec.backupLocations.velero.config** パラメーターを編集します。

```
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 Base46 でエンコードされた CA 証明書文字列を指定します。

2 **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.3.4.5. Data Protection Application のインストール

DataProtectionApplication API のインスタンスを作成して、Data Protection Application (DPA) をインストールします。

前提条件

- OADP Operator をインストールする必要があります。
- オブジェクトストレージをバックアップ場所として設定する必要があります。
- スナップショットを使用して PV をバックアップする場合、クラウドプロバイダーはネイティブスナップショット API または Container Storage Interface (CSI) スナップショットのいずれかをサポートする必要があります。
- バックアップとスナップショットの場所で同じ認証情報を使用する場合は、デフォルトの名前である **cloud-credentials-gcp** を使用して **Secret** を作成する必要があります。
- バックアップとスナップショットの場所で異なる認証情報を使用する場合は、2 つの **Secrets** を作成する必要があります。
  - バックアップ場所のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。
  - スナップショットの場所として、デフォルト名 **cloud-credentials-gcp** の **Secret**。この **Secret** は、**DataProtectionApplication** CR では参照されません。

注記

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。デフォルトの **Secret** がない場合、インストールは失敗します。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。
2. **Provided APIs** で、**DataProtectionApplication** ボックスの **Create instance** をクリックします。
3. **YAML View** をクリックして、**DataProtectionApplication** マニフェストのパラメーターを更新します。

```
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
          name: cloud-credentials-gcp (5)
        objectStorage:
          bucket: <bucket_name> (6)
          prefix: <prefix> (7)
  snapshotLocations: (8)
    - velero:
        provider: gcp
        default: true
        config:
          project: <project>
          snapshotLocation: us-west1 (9)
```

(1) **openshift** プラグインは必須です。

(2) Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。

(3) Restic のインストールを無効にする場合は、**false** に設定します。Restic はデーモンセットをデプロイします。これは、各ワーカーノードで **Restic** Pod が実行されていることを意味します。**spec.defaultVolumesToRestic: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。

4 Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。

5 この値を指定しない場合は、デフォルトの名前である **cloud-credentials-gcp** が使用されます。カスタム名を指定すると、バックアップの場所にカスタム名が使用されます。

6 バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。

7 バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。

8 CSI スナップショットまたは Restic を使用して PV をバックアップする場合を除き、スナップショットの場所を指定します。

9 スナップショットの場所は、PV と同じリージョンにある必要があります。

4. **Create** をクリックします。

5. OADP リソースを表示して、インストールを確認します。

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

出力例

```
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
```

4.3.4.5.1. DataProtectionApplication CR で CSI を有効にする

CSI スナップショットを使用して永続ボリュームをバックアップするには、**DataProtectionApplication** カスタムリソース (CR) で Container Storage Interface (CSI) を有効にします。

前提条件

- クラウドプロバイダーは、CSI スナップショットをサポートする必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR を編集します。

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

(1) **csi** デフォルトプラグインを追加します。

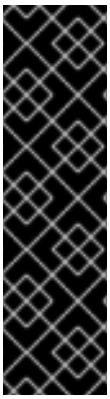
4.3.5. Multicloud Object Gateway を使用した OpenShift API for Data Protection のインストールおよび設定

OADP Operator をインストールすることで、Multicloud Object Gateway (MCG) を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.7 をインストールします。

注記

OADP 1.0.4 以降、すべて OADP 1.0.**z** バージョンは、MTC Operator の依存関係としてのみ使用でき、スタンドアロン Operator としては使用できません。

Multicloud Object Gateway をバックアップの場所として設定します。MCG は、OpenShift Data Foundation のコンポーネントです。MCG は、**DataProtectionApplication** カスタムリソース (CR) のバックアップ場所として設定します。



重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

バックアップの場所の **Secret** を作成し、次に、Data Protection Application をインストールします。

制限されたネットワーク環境に OADP Operator をインストールするには、最初にデフォルトの Operator Hub ソースを無効にして、Operator カタログをミラーリングする必要があります。詳細は、ネットワークが制限された環境での Operator Lifecycle Manager の使用 を参照してください。

4.3.5.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.10 に OpenShift API for Data Protection (OADP) オペレーターをインストールします。

OADP オペレーターは Velero 1.7 をインストールします。

前提条件

- **cluster-admin** 権限を持つユーザーとしてログインしている。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** をクリックします。
2. **Filter by keyword** フィールドを使用して、**OADP Operator** を検索します。
3. **OADP Operator** を選択し、**Install** をクリックします。
4. **Install** をクリックして、**openshift-adp** プロジェクトに Operator をインストールします。
5. **Operators → Installed Operators** をクリックして、インストールを確認します。

4.3.5.2. Multicloud Object Gateway の認証情報の取得

OpenShift API for Data Protection (OADP) の **Secret** カスタムリソース (CR) を作成するには、Multicloud Object Gateway (MCG) 認証情報を取得する必要があります。

MCG は、OpenShift Data Foundation のコンポーネントです。

前提条件

- 適切な OpenShift Data Foundation deployment guide を使用して、OpenShift Data Foundation をデプロイする必要があります。

手順

1. **NooBaa** カスタムリソースで **describe** コマンドを実行して、S3 エンドポイントである **AWS_ACCESS_KEY_ID** および **AWS_SECRET_ACCESS_KEY** を取得します。
2. **credentials-velero** ファイルを作成します。

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

Data Protection Application をインストールする際に、**credentials-velero** ファイルを使用して **Secret** オブジェクトを作成します。

4.3.5.3. バックアップおよびスナップショットの場所、ならびにそのシークレットについて

DataProtectionApplication カスタムリソース (CR) で、バックアップおよびスナップショットの場所、ならびにそのシークレットを指定します。

バックアップの場所
Multicloud Object Gateway、Noobaa、または Minio などの S3 互換オブジェクトストレージを、バックアップの場所として指定します。

Velero は、オブジェクトストレージのアーカイブファイルとして、OpenShift Container Platform リソース、Kubernetes オブジェクト、および内部イメージをバックアップします。

スナップショットの場所
クラウドプロバイダーのネイティブスナップショット API を使用して永続ボリュームをバックアップする場合、クラウドプロバイダーをスナップショットの場所として指定する必要があります。

Container Storage Interface (CSI) スナップショットを使用する場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成するため、スナップショットの場所を指定する必要はありません。

Restic を使用する場合は、Restic がオブジェクトストレージにファイルシステムをバックアップするため、スナップショットの場所を指定する必要はありません。

シークレット
バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの secret オブジェクトを作成します。

- **DataProtectionApplication** CR で指定する、バックアップの場所用のカスタム **Secret**。
- **DataProtectionApplication** CR で参照されない、スナップショットの場所用のデフォルト **Secret**。

重要

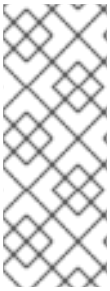
Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。

4.3.5.3.1. デフォルト Secret の作成

バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

Secret のデフォルト名は **cloud-credentials** です。



注記

DataProtectionApplication カスタムリソース (CR) にはデフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。バックアップの場所の **Secret** の名前が指定されていない場合は、デフォルトの名前が使用されます。

インストール時にバックアップの場所の認証情報を使用しない場合は、空の **credentials-velero** ファイルを使用してデフォルト名前で **Secret** を作成できます。

前提条件

- オブジェクトストレージとクラウドストレージがある場合は、同じ認証情報を使用する必要があります。
- Velero のオブジェクトストレージを設定する必要があります。
- オブジェクトストレージ用の **credentials-velero** ファイルを適切な形式で作成する必要があります。

手順

- デフォルト名で **Secret** を作成します。

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

Secret は、Data Protection Application をインストールするときに、 **DataProtectionApplication** CR の **spec.backupLocations.credential** ブロックで参照されます。

4.3.5.3.2. 異なる認証情報のシークレットの作成

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの **Secret** オブジェクトを作成する必要があります。

- カスタム名を持つバックアップ場所の **Secret**。カスタム名は、**DataProtectionApplication** カスタムリソース (CR) の **spec.backupLocations** ブロックで指定されます。
- スナップショットの場所 **Secret** (デフォルト名は **cloud-credentials**)。この **Secret** は、**DataProtectionApplication** で指定されていません。

手順

1. スナップショットの場所の **credentials-velero** ファイルをクラウドプロバイダーに適した形式で作成します。
2. デフォルト名でスナップショットの場所の **Secret** を作成します。

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

3. オブジェクトストレージに適した形式で、バックアップ場所の **credentials-velero** ファイルを作成します。
4. カスタム名を使用してバックアップ場所の **Secret** を作成します。

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

5. 次の例のように、カスタム名の **Secret** を **DataProtectionApplication** に追加します。

```
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) カスタム名を持つバックアップ場所の **Secret**。

4.3.5.4. Data Protection Application の設定

Velero リソースの割り当てを設定するか、自己署名 CA 証明書を有効にして、Data Protection Application を設定できます。

4.3.5.4.1. Velero の CPU とメモリーのリソース割り当てを設定

DataProtectionApplication カスタムリソース (CR) マニフェストを編集して、**Velero** Pod の CPU およびメモリーリソースの割り当てを設定します。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR マニフェストの **spec.configuration.velero.podConfig.ResourceAllocations** ブロックの値を編集します。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
spec:
...
  configuration:
    velero:
      podConfig:
        nodeSelector: <node selector> (1)
        resourceAllocations:
          limits:
```

```
        cpu: "1"
        memory: 512Mi
      requests:
        cpu: 500m
        memory: 256Mi
```

1 Velero podSpec に提供されるノードセレクターを指定します。

4.3.5.4.2. 自己署名 CA 証明書の有効化

certificate signed by unknown authority エラーを防ぐために、**DataProtectionApplication** カスタムリソース (CR) マニフェストを編集して、オブジェクトストレージの自己署名 CA 証明書を有効にする必要があります。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- **DataProtectionApplication** CR マニフェストの **spec.backupLocations.velero.objectStorage.caCert** パラメーターと **spec.backupLocations.velero.config** パラメーターを編集します。

```
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 Base46 でエンコードされた CA 証明書文字列を指定します。

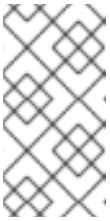
2 **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.3.5.5. Data Protection Application のインストール

DataProtectionApplication API のインスタンスを作成して、Data Protection Application (DPA) をインストールします。

前提条件

- OADP Operator をインストールする必要があります。
- オブジェクトストレージをバックアップ場所として設定する必要があります。
- スナップショットを使用して PV をバックアップする場合、クラウドプロバイダーはネイティブスナップショット API または Container Storage Interface (CSI) スナップショットのいずれかをサポートする必要があります。
- バックアップとスナップショットの場所で同じ認証情報を使用する場合は、デフォルトの名前である **cloud-credentials** を使用して **Secret** を作成する必要があります。
- バックアップとスナップショットの場所で異なる認証情報を使用する場合は、2 つの **Secrets** を作成する必要があります。
  - バックアップ場所のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。
  - スナップショットの場所のデフォルト名である **cloud-credentials** の **Secret**。この **Secret** は、**DataProtectionApplication** CR では参照されません。



注記

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。デフォルトの **Secret** がない場合、インストールは失敗します。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。
2. **Provided APIs** で、**DataProtectionApplication** ボックスの **Create instance** をクリックします。
3. **YAML View** をクリックして、**DataProtectionApplication** マニフェストのパラメーターを更新します。

```
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** プラグインは必須です。
2. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
3. Restic のインストールを無効にする場合は、**false** に設定します。Restic はデーモンセットをデプロイします。これは、各ワーカーノードで **Restic** Pod が実行されていることを意味します。**spec.defaultVolumesToRestic: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。
4. Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。
5. S3 エンドポイントの URL を指定します。
6. この値を指定しない場合は、デフォルト名の **cloud-credentials** が使用されます。カスタム名を指定すると、バックアップの場所にカスタム名が使用されます。
7. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
8. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。

4. **Create** をクリックします。
5. OADP リソースを表示して、インストールを確認します。

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

出力例

```
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
```

4.3.5.5.1. DataProtectionApplication CR で CSI を有効にする

CSI スナップショットを使用して永続ボリュームをバックアップするには、**DataProtectionApplication** カスタムリソース (CR) で Container Storage Interface (CSI) を有効にします。

前提条件

- クラウドプロバイダーは、CSI スナップショットをサポートする必要があります。

手順

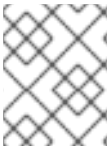
- 次の例のように、**DataProtectionApplication** CR を編集します。

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

(1) **csi** デフォルトプラグインを追加します。

4.3.6. OpenShift Data Foundation を使用した OpenShift API for Data Protection のインストールおよび設定

OpenShift Data Foundation を使用して Openshift API for Data Protection (OADP) をインストールするには、OADP Operator をインストールし、バックアップの場所とスナップショットロケーションを設定します。次に、Data Protection Application をインストールします。



注記

OADP 1.0.4 以降、すべて OADP 1.0.**z** バージョンは、MTC Operator の依存関係としてのみ使用でき、スタンドアロン Operator としては使用できません。

Multicloud Object Gateway または任意の S3 互換のオブジェクトストレージをバックアップの場所として設定できます。

重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

バックアップの場所の **Secret** を作成し、次に、Data Protection Application をインストールします。

制限されたネットワーク環境に OADP Operator をインストールするには、最初にデフォルトの Operator Hub ソースを無効にして、Operator カタログをミラーリングする必要があります。詳細は、ネットワークが制限された環境での Operator Lifecycle Manager の使用 を参照してください。

4.3.6.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.10 に OpenShift API for Data Protection (OADP) オペレーターをインストールします。

OADP オペレーターは Velero 1.7 をインストールします。

前提条件

- **cluster-admin** 権限を持つユーザーとしてログインしている。

手順

1. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** をクリックします。
2. **Filter by keyword** フィールドを使用して、**OADP Operator** を検索します。
3. **OADP Operator** を選択し、**Install** をクリックします。
4. **Install** をクリックして、**openshift-adp** プロジェクトに Operator をインストールします。
5. **Operators → Installed Operators** をクリックして、インストールを確認します。

4.3.6.2. バックアップおよびスナップショットの場所、ならびにそのシークレットについて

DataProtectionApplication カスタムリソース (CR) で、バックアップおよびスナップショットの場所、ならびにそのシークレットを指定します。

バックアップの場所
Multicloud Object Gateway、Noobaa、または Minio などの S3 互換オブジェクトストレージを、バックアップの場所として指定します。

Velero は、オブジェクトストレージのアーカイブファイルとして、OpenShift Container Platform リソース、Kubernetes オブジェクト、および内部イメージをバックアップします。

スナップショットの場所
クラウドプロバイダーのネイティブスナップショット API を使用して永続ボリュームをバックアップする場合、クラウドプロバイダーをスナップショットの場所として指定する必要があります。

Container Storage Interface (CSI) スナップショットを使用する場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成するため、スナップショットの場所を指定する必要はありません。

Restic を使用する場合は、Restic がオブジェクトストレージにファイルシステムをバックアップするため、スナップショットの場所を指定する必要はありません。

シークレット
バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの secret オブジェクトを作成します。

- **DataProtectionApplication** CR で指定する、バックアップの場所用のカスタム **Secret**。
- **DataProtectionApplication** CR で参照されない、スナップショットの場所用のデフォルト **Secret**。

重要

Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。

4.3.6.2.1. デフォルト Secret の作成

バックアップとスナップショットの場所が同じ認証情報を使用する場合、またはスナップショットの場所が必要ない場合は、デフォルトの **Secret** を作成します。

バックアップストレージプロバイダーに **aws**、**azure**、または **gcp** などのデフォルトのプラグインがない限り、**Secret** のデフォルト名は **cloud-credentials** です。その場合、プロバイダー固有の OADP インストール手順でデフォルト名が指定されています。

注記

DataProtectionApplication カスタムリソース (CR) にはデフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。バックアップの場所の **Secret** の名前が指定されていない場合は、デフォルトの名前が使用されます。

インストール時にバックアップの場所の認証情報を使用しない場合は、空の **credentials-velero** ファイルを使用してデフォルト名前で **Secret** を作成できます。

前提条件

- オブジェクトストレージとクラウドストレージがある場合は、同じ認証情報を使用する必要があります。
- Velero のオブジェクトストレージを設定する必要があります。
- オブジェクトストレージ用の **credentials-velero** ファイルを適切な形式で作成する必要があります。

手順

- デフォルト名で **Secret** を作成します。

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

Secret は、Data Protection Application をインストールするときに、 **DataProtectionApplication** CR の **spec.backupLocations.credential** ブロックで参照されます。

4.3.6.2.2. 異なる認証情報のシークレットの作成

バックアップとスナップショットの場所で異なる認証情報を使用する場合は、次の 2 つの **Secret** オブジェクトを作成する必要があります。

- カスタム名を持つバックアップ場所の **Secret**。カスタム名は、**DataProtectionApplication** カスタムリソース (CR) の **spec.backupLocations** ブロックで指定されます。
- スナップショットの場所 **Secret** (デフォルト名は **cloud-credentials**)。この **Secret** は、**DataProtectionApplication** で指定されていません。

手順

1. スナップショットの場所の **credentials-velero** ファイルをクラウドプロバイダーに適した形式で作成します。
2. デフォルト名でスナップショットの場所の **Secret** を作成します。

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

3. オブジェクトストレージに適した形式で、バックアップ場所の **credentials-velero** ファイルを作成します。
4. カスタム名を使用してバックアップ場所の **Secret** を作成します。

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

5. 次の例のように、カスタム名の **Secret** を **DataProtectionApplication** に追加します。

```
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) カスタム名を持つバックアップ場所の **Secret**。

4.3.6.3. Data Protection Application の設定

Velero リソースの割り当てを設定するか、自己署名 CA 証明書を有効にして、Data Protection Application を設定できます。

4.3.6.3.1. Velero の CPU とメモリーのリソース割り当てを設定

DataProtectionApplication カスタムリソース (CR) マニフェストを編集して、**Velero** Pod の CPU およびメモリーリソースの割り当てを設定します。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR マニフェストの **spec.configuration.velero.podConfig.ResourceAllocations** ブロックの値を編集します。

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

```
    podConfig:
      nodeSelector: <node selector> 1
      resourceAllocations:
        limits:
          cpu: "1"
          memory: 512Mi
        requests:
          cpu: 500m
          memory: 256Mi
```

1 Velero podSpec に提供されるノードセレクターを指定します。

4.3.6.3.2. 自己署名 CA 証明書の有効化

certificate signed by unknown authority エラーを防ぐために、**DataProtectionApplication** カスタムリソース (CR) マニフェストを編集して、オブジェクトストレージの自己署名 CA 証明書を有効にする必要があります。

前提条件

- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

- **DataProtectionApplication** CR マニフェストの **spec.backupLocations.velero.objectStorage.caCert** パラメーターと **spec.backupLocations.velero.config** パラメーターを編集します。

```
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 Base46 でエンコードされた CA 証明書文字列を指定します。

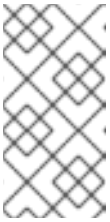
2 **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.3.6.4. Data Protection Application のインストール

DataProtectionApplication API のインスタンスを作成して、Data Protection Application (DPA) をインストールします。

前提条件

- OADP Operator をインストールする必要があります。
- オブジェクトストレージをバックアップ場所として設定する必要があります。
- スナップショットを使用して PV をバックアップする場合、クラウドプロバイダーはネイティブスナップショット API または Container Storage Interface (CSI) スナップショットのいずれかをサポートする必要があります。
- バックアップとスナップショットの場所で同じ認証情報を使用する場合は、デフォルトの名前である **cloud-credentials** を使用して **Secret** を作成する必要があります。
- バックアップとスナップショットの場所で異なる認証情報を使用する場合は、2 つの **Secrets** を作成する必要があります。
  - バックアップ場所のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。
  - スナップショットの場所のデフォルト名である **cloud-credentials** の **Secret**。この **Secret** は、**DataProtectionApplication** CR では参照されません。



注記

インストール中にバックアップまたはスナップショットの場所を指定したくない場合は、空の **credentials-velero** ファイルを使用してデフォルトの **Secret** を作成できます。デフォルトの **Secret** がない場合、インストールは失敗します。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。
2. **Provided APIs** で、**DataProtectionApplication** ボックスの **Create instance** をクリックします。
3. **YAML View** をクリックして、**DataProtectionApplication** マニフェストのパラメーターを更新します。

```
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. オプション: **kubevirt** プラグインは OpenShift Virtualization で使用されます。
2. 必要に応じて、バックアッププロバイダーのデフォルトのプラグイン (**gcp** など) を指定します。
3. CSI スナップショットを使用して PV をバックアップする場合は、**csi** のデフォルトプラグインを指定します。**csi** プラグインは、Velero CSI ベータスナップショット API を使用します。スナップショットの場所を設定する必要はありません。
4. **openshift** プラグインは必須です。
5. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
6. Restic のインストールを無効にする場合は、**false** に設定します。Restic はデーモンセットをデプロイします。これは、各ワーカーノードで **Restic** Pod が実行されていることを意味します。**spec.defaultVolumesToRestic: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。
7. Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。
8. バックアッププロバイダーを指定します。
9. バックアッププロバイダーにデフォルトのプラグインを使用する場合は、**Secret** の正しいデフォルト名を指定します (例: **cloud-credentials-gcp**)。カスタム名を指定すると、そのカスタム名がバックアップの場所に使用されます。**Secret** 名を指定しない場合は、デフォルトの名前が使用されます。
10. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
11. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。

4. **Create** をクリックします。

5. OADP リソースを表示して、インストールを確認します。

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

出力例

```
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
```

4.3.6.4.1. OpenShift Data Foundation での障害復旧のための NooBaa の設定

OpenShift Data Foundation で NooBaa バケットの **backupStorageLocation** にクラスターストレージを使用する場合は、NooBaa を外部オブジェクトストアとして設定します。

警告

NooBaa を外部オブジェクトストアとして設定しないと、バックアップが利用できなくなる可能性があります。

手順

- ハイブリッドまたはマルチクラウドのストレージリソースの追加 の説明に従って、NooBaa を外部オブジェクトストアとして設定します。

4.3.6.4.2. DataProtectionApplication CR で CSI を有効にする

CSI スナップショットを使用して永続ボリュームをバックアップするには、**DataProtectionApplication** カスタムリソース (CR) で Container Storage Interface (CSI) を有効にします。

前提条件

- クラウドプロバイダーは、CSI スナップショットをサポートする必要があります。

手順

- 次の例のように、**DataProtectionApplication** CR を編集します。

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

1 **csi** デフォルトプラグインを追加します。

4.3.7. OpenShift API for Data Protection のアンインストール

OpenShift API for Data Protection (OADP) をアンインストールするには、OADP Operator を削除します。詳細は、クラスターからの演算子の削除 を参照してください。

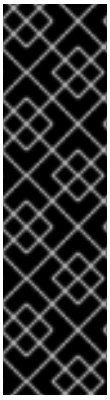
4.4. バックアップおよび復元

4.4.1. アプリケーションのバックアップ

Backup カスタムリソース (CR) を作成して、アプリケーションをバックアップします。バックアップ CR の作成 を参照してください。

Backup CR は、Kubernetes リソースや内部イメージのバックアップファイルを S3 オブジェクトストレージ上に作成し、クラウドプロバイダーが OpenShift Data Foundation 4 のようにスナップショットを作成するためにネイティブスナップショット API や Container Storage Interface (CSI) を使用している場合は、永続ボリューム (PV) のスナップショットを作成します。

CSI ボリュームスナップショットの詳細は、CSI ボリュームスナップショット を参照してください。

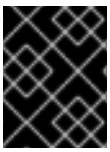


重要

S3 ストレージ用の **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

- クラウドプロバイダーがネイティブスナップショット API を備えている場合、または CSI スナップショットをサポートしている場合、**Backup** CR はスナップショットを作成することによって永続ボリューム (PV) をバックアップします。CSI スナップショットの操作の詳細は、CSI スナップショットを使用した永続ボリュームのバックアップ を参照してください。
- クラウドプロバイダーがスナップショットをサポートしていない場合、またはアプリケーションが NFS データボリューム上にある場合は、Restic を使用してバックアップを作成できます。Restic を使用したアプリケーションのバックアップ を参照してください。



重要

OpenShift API for Data Protection (OADP) は、他のソフトウェアで作成されたボリュームスナップショットのバックアップをサポートしていません。

バックアップ操作の前または後にコマンドを実行するためのバックアップフックを作成できます。バックアップフックの作成 を参照してください。

Backup CR の代わりに **Schedule** CR を作成することにより、バックアップをスケジュールできます。バックアップのスケジュール設定 を参照してください。

4.4.1.1. バックアップ CR の作成

Backup カスタムリソース (CR) を作成して、Kubernetes イメージ、内部イメージ、および永続ボリューム (PV) をバックアップします。

前提条件

- OpenShift API for Data Protection (OADP) Operator をインストールする必要があります。
- **DataProtectionApplication** CR は **Ready** 状態である必要があります。
- バックアップ場所の前提条件:
  - Velero 用に S3 オブジェクトストレージを設定する必要があります。
  - **DataProtectionApplication** CR でバックアップの場所を設定する必要があります。
- スナップショットの場所の前提条件:
  - クラウドプロバイダーには、ネイティブスナップショット API が必要であるか、Container Storage Interface (CSI) スナップショットをサポートしている必要があります。
  - CSI スナップショットの場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成する必要があります。

- **DataProtectionApplication** CR でボリュームの場所を設定する必要があります。

手順

1. 次のコマンドを入力して、**backupStorageLocations** CR を取得します。

```
$ oc get backupStorageLocations
```

出力例

```
NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
velero-sample-1   Available   11s              31m
```

2. 次の例のように、**Backup** CR を作成します。

```
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. バックアップする namespace の配列を指定します。
2. オプション: バックアップに含めるリソースの配列を指定します。リソースは、ショートカット (Pods は po など) または完全修飾の場合があります。指定しない場合、すべてのリソースが含まれます。
3. オプション: バックアップから除外するリソースの配列を指定します。リソースは、ショートカット (Pods は po など) または完全修飾の場合があります。
4. **backupStorageLocations** CR の名前を指定します。
5. 指定したラベルを **すべて** 持つバックアップリソースの {key,value} ペアのマップ。

6 指定したラベルを **1つ以上** 持つバックアップリソースの {key,value} ペアのマップ。

3. **Backup** CR のステータスが **Completed** したことを確認します。

```
$ oc get backup -n openshift-adp <backup> -o jsonpath='{.status.phase}'
```

4.4.1.2. CSI スナップショットを使用した永続ボリュームのバックアップ

Backup CR を作成する前に、クラウドストレージの **VolumeSnapshotClass** カスタムリソース (CR) を編集して、Container Storage Interface (CSI) スナップショットを使用して永続ボリュームをバックアップします。

前提条件

- クラウドプロバイダーは、CSI スナップショットをサポートする必要があります。
- **DataProtectionApplication** CR で CSI を有効にする必要があります。

手順

- **metadata.labels.velero.io/csi-volumesnapshot-class: "true"** のキー: 値ペアを **VolumeSnapshotClass** CR に追加します。

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true"
driver: <csi_driver>
deletionPolicy: Retain
```

これで、**Backup** CR を作成できます。

4.4.1.3. Restic を使用したアプリケーションのバックアップ

Backup カスタムリソース (CR) を編集して、Restic を使用して Kubernetes リソース、内部イメージ、および永続ボリュームをバックアップします。

DataProtectionApplication CR でスナップショットの場所を指定する必要はありません。

重要

Restic は、**hostPath** ボリュームのバックアップをサポートしません。詳細は、additional Rustic limitations を参照してください。

前提条件

- OpenShift API for Data Protection (OADP) Operator をインストールする必要があります。
- **DataProtectionApplication** CR で **spec.configuration.restic.enable** を **false** に設定して、デフォルトの Restic インストールを無効にしないでください。

- **DataProtectionApplication** CR は **Ready** 状態である必要があります。

手順

- 次の例のように、**Backup** CR を編集します。

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  defaultVolumesToRestic: true (1)
...
```

(1) **defaultVolumesToRestic: true** を **spec** ブロックに追加します。

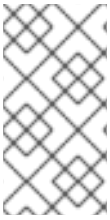
4.4.1.4. CSI スナップショットに Data Mover を使用する

重要

CSI スナップショット用の Data Mover は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

OADP 1.1.0 Data Mover を使用すると、顧客は Container Storage Interface (CSI) ボリュームスナップショットをリモートオブジェクトストアにバックアップできます。Data Mover が有効になっていると、クラスターの障害、誤った削除、または破損が発生した場合に、ストアからステートフルアプリケーションを復元できます。OADP 1.1.0 Data Mover ソリューションは、VolSync の Restic オプションを使用します。



注記

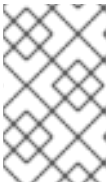
Data Mover は、CSI ボリュームスナップショットのバックアップとリストアのみをサポートします。

現在、Data Mover は Google Cloud Storage (GCS) バケットをサポートしていません。

前提条件

- **StorageClass** および **VolumeSnapshotClass** カスタムリソース (CR) が CSI をサポートしていることを確認しました。
- 注釈 **snapshot.storage.kubernetes.io/is-default-class: true** を持つ **volumeSnapshotClass** CR が 1 つだけであることを確認しました。

- 注釈 **storageclass.kubernetes.io/is-default-class: true** を持つ **storageClass** CR が 1 つだけであることを確認しました。
- **VolumeSnapshotClass** CR にラベル **velero.io/csi-volumesnapshot-class: 'true'** を含めました。
- Operator Lifecycle Manager (OLM) を使用して VolSync Operator をインストールしました。



注記

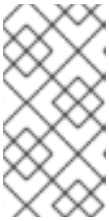
VolSync Operator は、テクノロジープレビューのデータ Mover との使用にのみ必要です。Operator は、OADP プロダクション機能を使用するために必要ではありません。

- OLM を使用して OADP Operator をインストールしました。

手順

1. 次のように **.yaml** ファイルを作成して、Restic シークレットを設定します。

```
apiVersion: v1
kind: Secret
metadata:
  name: <secret_name>
  namespace: openshift-adp
type: Opaque
stringData:
  RESTIC_PASSWORD: <secure_restic_password>
```



注記

デフォルトでは、Operator は **dm-credential** という名前のシークレットを探します。別の名前を使用している場合は、**dpa.spec.features.dataMover.credentialName** を使用して、データ保護アプリケーション (DPA) CR で名前を指定する必要があります。

2. 次の例のような DPA CR を作成します。デフォルトのプラグインには CSI が含まれています。

データ保護アプリケーション (DPA) CR の例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  features:
    dataMover:
      enable: true
      credentialName: <secret_name> 1
  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:
    defaultPlugins:
      - openshift
      - aws
      - csi
```

1 前のステップの Restic シークレット名を追加します。これが行われない場合、デフォルトのシークレット名 **dm-credential** が使用されます。

OADP Operator は、2 つのカスタムリソース定義 (CRD)、**VolumeSnapshotBackup** および **VolumeSnapshotRestore** をインストールします。

VolumeSnapshotBackup CRD の例

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotBackup
metadata:
  name: <vsb_name>
  namespace: <namespace_name> 1
spec:
  volumeSnapshotContent:
    name: <snapcontent_name>
  protectedNamespace: <adp_namespace>
  resticSecretRef:
    name: <restic_secret_name>
```

1 ボリュームスナップショットが存在する namespace を指定します。

VolumeSnapshotRestore CRD の例

```
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. ボリュームスナップショットが存在する namespace を指定します。
2. Operator がインストールされている namespace を指定します。デフォルトは **openshift-adp** です。

3. 次の手順を実行して、ボリュームスナップショットをバックアップできます。
   a. バックアップ CR を作成します。

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup_name>
  namespace: <protected_ns> 1
spec:
  includedNamespaces:
  - <app_ns>
  storageLocation: velero-sample-1
```

1. Operator がインストールされている namespace を指定します。デフォルトの namespace は **openshift-adp** です。

   b. 次のコマンドを入力して、最大 10 分待機し、**VolumeSnapshotBackup** CR のステータスが **Completed** かどうかを確認します。

```
$ oc get vsb -n <app_ns>
```

```
$ oc get vsb <vsb_name> -n <app_ns> -o jsonpath="{.status.phase}"
```

DPA で設定されたオブジェクトストアにスナップショットが作成されます。

注記

VolumeSnapshotBackup CR のステータスが **Failed** になった場合は、トラブルシューティングのために Velero ログを参照してください。

4. 次の手順を実行して、ボリュームスナップショットを復元できます。
   a. アプリケーションの namespace と、Velero CSI プラグインによって作成された **volumeSnapshotContent** を削除します。

   b. **Restore** CR を作成し、**restorePVs** を **true** に設定します。

Restore CR の例

```
apiVersion: velero.io/v1
kind: Restore
```

```
  metadata:
    name: <restore_name>
    namespace: <protected_ns>
  spec:
    backupName: <previous_backup_name>
    restorePVs: true
```

c. 最大 10 分間待機し、次のコマンドを入力して、**VolumeSnapshotRestore** CR ステータスが **Completed** であるかどうかを確認します。

```
$ oc get vsr -n <app_ns>
```

```
$ oc get vsr <vsr_name> -n <app_ns> -o jsonpath="{.status.phase}"
```

d. アプリケーションデータとリソースが復元されたかどうかを確認します。

注記

VolumeSnapshotRestore CR のステータスが失敗になった場合は、トラブルシューティングのために Velero ログを参照してください。

4.4.1.5. Data Mover と OADP 1.1 を使用したバックアップ後のクリーンアップ。

OADP 1.1 の場合、Data Mover のいずれかのバージョンを使用してバックアップを実行した後、データクリーンアップを実行する必要があります。

クリーンアップには、次のリソースの削除が含まれます。

- バケット内のスナップショット
- クラスターリソース
- スケジュールに従って実行されるか、繰り返し実行されるバックアップ手順の後のボリュームスナップショットバックアップ (VSB)

4.4.1.5.1. バケット内のスナップショットの削除

Data Mover は、バックアップ後に 1 つ以上のスナップショットをバケットに残す場合があります。すべてのスナップショットを削除することも、個々のスナップショットを削除することもできます。

手順

- バケット内のすべてのスナップショットを削除するには、データ保護アプリケーション (DPA) の **.spec.backupLocation.objectStorage.bucket** リソースで指定されている **/<protected_namespace>** フォルダーを削除します。
- 個々のスナップショットを削除するには、以下のようになりました。

  1. DPA **.spec.backupLocation.objectStorage.bucket** リソースで指定されている **/<protected_namespace>** フォルダーを参照します。
  2. **/<volumeSnapshotContent name>-pvc** という接頭辞が付いた適切なフォルダーを削除します。ここで、**<VolumeSnapshotContent_name>** は、Data Mover によって PVC ごとに作成された **VolumeSnapshotContent** です。

4.4.1.5.2. クラスターリソースの削除

Data Mover は、コンテナーストレージインターフェイス (CSI) ボリュームのスナップショットをリモートオブジェクトストアに正常にバックアップするかどうかに関係なく、クラスターリソースを残す場合があります。

4.4.1.5.2.1. Data Mover を使用したバックアップとリストアが成功した後のクラスターリソースの削除

Data Mover を使用したバックアップとリストアが成功した後、アプリケーションの namespace に残っている **VolumeSnapshotBackup** または **VolumeSnapshotRestore** CR を削除できます。

手順

1. Data Mover を使用したバックアップ後に、アプリケーションのnamespace (バックアップおよびリストアするアプリケーション PVC を含む namespace) に残っているクラスターリソースを削除します。

```
$ oc delete vsb -n <app_namespace> --all
```

2. Data Mover を使用するリストア後に残るクラスターリソースを削除します。

```
$ oc delete vsr -n <app_namespace> --all
```

3. 必要に応じて、Data Mover を使用するバックアップおよびリストア後に残っている **VolumeSnapshotContent** リソースを削除します。

```
$ oc delete volumesnapshotcontent --all
```

4.4.1.5.2.2. Data Mover を使用したバックアップとリストアが部分的に成功または失敗した後のクラスターリソースの削除

Data Mover を使用したバックアップおよびリストア操作が失敗するか、部分的にしか成功しない場合は、アプリケーションの namespace に存在する **VolumeSnapshotBackup** (VSB) または **VolumeSnapshotRestore** カスタムリソース定義 (CRD) をクリーンアップし、このコントローラーによって作成された余分なリソースをクリーンアップする必要があります。

手順

1. 次のコマンドを入力して、Data Mover を使用したバックアップ操作後に残ったクラスターリソースをクリーンアップします。

   a. アプリケーション namespace 上の VSB CRD を削除します。この namespace には、バックアップおよび復元するアプリケーション PVC が含まれています。

```
$ oc delete vsb -n <app_namespace> --all
```

   b. **VolumeSnapshot** CR を削除します。

```
$ oc delete volumesnapshot -A --all
```

   c. **VolumeSnapshotContent** CR を削除します。

```
$ oc delete volumesnapshotcontent --all
```

d. 保護された namespace (Operator がインストールされている namespace) 上の PVC をすべて削除します。

```
$ oc delete pvc -n <protected_namespace> --all
```

e. namespace 上の **ReplicationSource** リソースをすべて削除します。

```
$ oc delete replicationsource -n <protected_namespace> --all
```

2. 次のコマンドを入力して、Data Mover を使用したリストア操作後に残ったクラスターリソースをクリーンアップします。

a. VSR CRD を削除します。

```
$ oc delete vsr -n <app-ns> --all
```

b. **VolumeSnapshot** CR を削除します。

```
$ oc delete volumesnapshot -A --all
```

c. **VolumeSnapshotContent** CR を削除します。

```
$ oc delete volumesnapshotcontent --all
```

d. namespace 上の **ReplicationDestination** リソースをすべて削除します。

```
$ oc delete replicationdestination -n <protected_namespace> --all
```

関連情報

- 管理者向けのクラスターへの Operator のインストール
- 管理者以外の namespace に Operator をインストールする

4.4.1.6. バックアップフックの作成

Backup カスタムリソース (CR) を編集して、Pod 内のコンテナーでコマンドを実行するためのバックアップフックを作成します。

プレ フックは、Pod のバックアップが作成される前に実行します。**ポスト** フックはバックアップ後に実行します。

手順

- 次の例のように、**Backup** CR の **spec.hooks** ブロックにフックを追加します。

```
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. オプション: フックが適用される namespace を指定できます。この値が指定されていない場合、フックはすべてのネームスペースに適用されます。
2. オプション: フックが適用されない namespace を指定できます。
3. 現在、Pod は、フックを適用できる唯一のサポート対象リソースです。
4. オプション: フックが適用されないリソースを指定できます。
5. オプション: このフックは、ラベルに一致するオブジェクトにのみ適用されます。この値が指定されていない場合、フックはすべてのネームスペースに適用されます。
6. バックアップの前に実行するフックの配列。
7. オプション: コンテナーが指定されていない場合、コマンドは Pod の最初のコンテナーで実行されます。
8. これは、追加される init コンテナーのエントリーポイントです。
9. エラー処理に許可される値は、**Fail** と **Continue** です。デフォルトは **Fail** です。
10. オプション: コマンドの実行を待機する時間。デフォルトは **30s** です。
11. このブロックでは、バックアップ後に実行するフックの配列を、バックアップ前のフックと同じパラメーターで定義します。

4.4.1.7. バックアップのスケジュール

Backup CR の代わりに **Schedule** カスタムリソース (CR) を作成して、バックアップをスケジュールします。

警告

バックアップスケジュールでは、別のバックアップが作成される前にバックアップを数量するための時間を十分確保してください。

たとえば、namespace のバックアップに通常 10 分かかる場合は、15 分ごとよりも頻繁にバックアップをスケジュールしないでください。

前提条件

- OpenShift API for Data Protection (OADP) Operator をインストールする必要があります。
- **DataProtectionApplication** CR は **Ready** 状態である必要があります。

手順

1. **backupStorageLocations** CR を取得します。

```
$ oc get backupStorageLocations
```

出力例

```
NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
velero-sample-1   Available   11s              31m
```

2. 次の例のように、**Schedule** CR を作成します。

```
$ 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
    defaultVolumesToRestic: true 4
    ttl: 720h0m0s
EOF
```

1. バックアップをスケジュールするための **cron** 式。たとえば、毎日 7:00 にバックアップを実行する場合は **0 7 * * *** です。
2. バックアップを作成する namespace の配列。

3 **backupStorageLocations** CR の名前。

4 オプション: Restic を使用してボリュームをバックアップする場合は、キーと値のペア **defaultVolumesToRestic: true** を追加します。

3. スケジュールされたバックアップの実行後に、**Schedule** CR のステータスが **Completed** となっていることを確認します。

```
$ oc get schedule -n openshift-adp <schedule> -o jsonpath='{.status.phase}'
```

4.4.1.8. バックアップの削除

Backup カスタムリソース (CR) を削除することで、バックアップファイルを削除できます。

> **警告**
>
> **Backup** CR および関連するオブジェクトストレージデータを削除した後、削除したデータを復元することはできません。

前提条件

- **Backup** CR を作成した。
- **Backup** CR の名前とそれを含む namespace がわかっている。
- Velero CLI ツールをダウンロードした。
- クラスター内の Velero バイナリーにアクセスできる。

手順

- 次のいずれかのアクションを選択して、**Backup** CR を削除します。
  - **Backup** CR を削除し、関連するオブジェクトストレージデータを保持する場合は、次のコマンドを実行します。

    ```
    $ oc delete backup <backup_CR_name> -n <velero_namespace>
    ```

  - **Backup** CR を削除し、関連するオブジェクトストレージデータを削除する場合は、次のコマンドを実行します。

    ```
    $ velero backup delete <backup_CR_name> -n <velero_namespace>
    ```

    ここでは、以下のようになります。

    <backup_CR_name>
    Backup カスタムリソースの名前を指定します。
    <velero_namespace>

Backup カスタムリソースを含む namespace を指定します。

関連情報

- Velero CLI ツールをダウンロードする

4.4.2. アプリケーションの復元

アプリケーションのバックアップを復元するには、**Restore** カスタムリソース (CR) を作成します。復元 CR の作成 を参照してください。

Restore (CR) を編集することでアプリケーションを復元する際に、Pod 内のコンテナーでコマンドを実行するための復元フックを作成できます。復元フックの作成 を参照してください。

4.4.2.1. 復元 CR の作成

Restore CR を作成して、**Backup** カスタムリソース (CR) を復元します。

前提条件

- OpenShift API for Data Protection (OADP) Operator をインストールする必要があります。
- **DataProtectionApplication** CR は **Ready** 状態である必要があります。
- Velero **Backup** CR が必要です。
- バックアップ時に永続ボリューム (PV) の容量が要求されたサイズと一致するよう、要求されたサイズを調整します。

手順

1. 次の例のように、**Restore** CR を作成します。

```
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. **Backup** CR の名前
2. オプション: 復元プロセスに含めるリソースの配列を指定します。リソースは、ショートカット (**Pods** は **po** など) または完全修飾の場合があります。指定しない場合、すべてのリソースが含まれます。

3 オプション: **RestorePVs** パラメーターを **false** に設定すると、コンテナーストレージインターフェイス (CSI) スナップショットの **VolumeSnapshot** から、または

2. 次のコマンドを入力して、**Restore** CR のステータスが **Completed** であることを確認します。

```
$ oc get restore -n openshift-adp <restore> -o jsonpath='{.status.phase}'
```

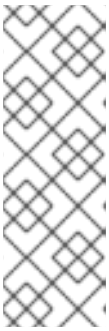
3. 次のコマンドを入力して、バックアップリソースが復元されたことを確認します。

```
$ oc get all -n <namespace> 1
```

1 バックアップした namespace。

4. Restic を使用して **DeploymentConfig** オブジェクトを復元する場合、または復元後のフックを使用する場合は、次のコマンドを入力して **dc-restic-post-restore.sh** クリーンアップスクリプトを実行します。

```
$ bash dc-restic-post-restore.sh <restore-name>
```



注記

復元プロセスの過程で、OADP Velero プラグインは、**DeploymentConfig** オブジェクトをスケールダウンし、Pod をスタンドアロン Pod として復元して、クラスターが復元された **DeploymentConfig** Pod を復元時にすぐに削除しないようにし、Restic および復元後のフックを完了できるようにします。復元された Pod でのアクション。クリーンアップスクリプトは、これらの切断された Pod を削除し、**DeploymentConfig** オブジェクトを適切な数のレプリカに戻します。

例4.1 dc-restic-post-restore.sh クリーンアップスクリプト

```
#!/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
```

4.4.2.2. 復元フックの作成

Restore カスタムリソース (CR) を編集して、アプリケーションの復元中に Pod 内のコンテナーでコマンドを実行する復元フックを作成します。

2 種類の復元フックを作成できます。

- **init** フックは、init コンテナーを Pod に追加して、アプリケーションコンテナーが起動する前にセットアップタスクを実行します。
  Restic バックアップを復元する場合は、復元フック init コンテナーの前に **restic-wait** init コンテナーが追加されます。

- **exec** フックは、復元された Pod のコンテナーでコマンドまたはスクリプトを実行します。

手順

- 次の例のように、**Restore CR** の **spec.hooks** ブロックにフックを追加します。

```
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. オプション: フックが適用される namespace の配列。この値が指定されていない場合、フックはすべてのネームスペースに適用されます。
2. 現在、Pod は、フックを適用できる唯一のサポート対象リソースです。
3. オプション: このフックは、ラベルセレクターに一致するオブジェクトにのみ適用されます。
4. オプション: Timeout は、**initContainers** が完了するまで Velero が待機する最大時間を指定します。
5. オプション: コンテナーが指定されていない場合、コマンドは Pod の最初のコンテナーで実行されます。
6. これは、追加される init コンテナーのエントリーポイントです。
7.

オプション: コンテナーの準備が整うまでの待機時間。これは、コンテナーが起動して同じコンテナー内の先行するフックが完了するのに十分な長さである必要があります。設定

8 オプション: コマンドの実行を待機する時間。デフォルトは **30s** です。

9 エラー処理に許可される値は、**Fail** および **Continue** です。

- **Continue**: コマンドの失敗のみがログに記録されます。
- **Fail**: Pod 内のコンテナーで復元フックが実行されなくなりました。**Restore** CR のステータスは **PartiallyFailed** になります。

4.5. トラブルシューティング

OpenShift CLI ツール または Velero CLI ツール を使用して、Velero カスタムリソース (CR) をデバッグできます。Velero CLI ツールは、より詳細なログおよび情報を提供します。

インストールの問題、CR のバックアップと復元の問題、および Restic の問題 を確認できます。

must-gather ツール を使用して、ログ、CR 情報、および Prometheus メトリックデータを収集できます。

Velero CLI ツールは、次の方法で入手できます。

- Velero CLI ツールをダウンロードする
- クラスター内の Velero デプロイメントで Velero バイナリーにアクセスする

4.5.1. Velero CLI ツールをダウンロードする

Velero のドキュメントページ の手順に従って、Velero CLI ツールをダウンロードしてインストールできます。

このページには、以下に関する手順が含まれています。

- Homebrew を使用した macOS
- GitHub
- Chocolatey を使用した Windows

前提条件

- DNS とコンテナーネットワークが有効になっている、v1.16 以降の Kubernetes クラスターにアクセスできる。
- **kubectl** をローカルにインストールしている。

手順

1. ブラウザーを開き、"Install the CLI" on the Verleo website に移動します。
2. macOS、GitHub、または Windows の適切な手順に従います。

3. 次の表に従って、OADP および OpenShift Container Platform のバージョンに適した Velero バージョンをダウンロードします。

表4.2 OADP-Velero-OpenShift Container Platform バージョンの関係

OADP のバージョン	Velero のバージョン	OpenShift Container Platform バージョン
1.0.0	1.7	4.6 以降
1.0.1	1.7	4.6 以降
1.0.2	1.7	4.6 以降
1.0.3	1.7	4.6 以降
1.1.0	{velero-version}	4.9 以降
1.1.1	{velero-version}	4.9 以降
1.1.2	{velero-version}	4.9 以降

4.5.2. クラスター内の Velero デプロイメントで Velero バイナリーにアクセスする

shell コマンドを使用して、クラスター内の Velero デプロイメントの Velero バイナリーにアクセスできます。

前提条件

- **DataProtectionApplication** カスタムリソースのステータスが **Reconcile complete** である。

手順

- 次のコマンドを入力して、必要なエイリアスを設定します。

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

4.5.3. OpenShift CLI ツールを使用した Velero リソースのデバッグ

OpenShift CLI ツールを使用して Velero カスタムリソース (CR) と **Velero** Pod ログを確認することで、失敗したバックアップまたは復元をデバッグできます。

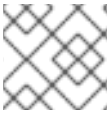
Velero CR
oc describe コマンドを使用して、**Backup** または **Restore** CR に関連する警告とエラーの要約を取得します。

```
$ oc describe <velero_cr> <cr_name>
```

Velero Pod ログ
oc logs コマンドを使用して、**Velero** Pod ログを取得します。

```
$ oc logs pod/<velero>
```

Velero Pod のデバッグログ
次の例に示すとおり、**DataProtectionApplication** リソースで Velero ログレベルを指定できます。



注記

このオプションは、OADP 1.0.3 以降で使用できます。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
spec:
  configuration:
    velero:
      logLevel: warning
```

次の **logLevel** 値を使用できます。

- **trace**
- **debug**
- **info**
- **warning**
- **error**
- **致命的**
- **panic**

ほとんどのログには **debug** を使用することをお勧めします。

4.5.4. Velero CLI ツールを使用した Velero リソースのデバッグ

Velero CLI ツールを使用して、**Backup** および **Restore** カスタムリソース (CR) をデバッグし、ログを取得できます。

Velero CLI ツールは、OpenShift CLI ツールよりも詳細な情報を提供します。

構文
oc exec コマンドを使用して、Velero CLI コマンドを実行します。

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> <command> <cr_name>
```

例

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

ヘルプオプション
velero --help オプションを使用して、すべての Velero CLI コマンドを一覧表示します。

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  --help
```

describe コマンド
velero describe コマンドを使用して、**Backup** または **Restore** CR に関連する警告とエラーの要約を取得します。

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> describe <cr_name>
```

例

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  backup describe 0e44ae00-5dc3-11eb-9ca8-df7e5254778b-2d8ql
```

logs コマンド
velero logs コマンドを使用して、**Backup** または **Restore** CR のログを取得します。

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  <backup_restore_cr> logs <cr_name>
```

例

```
$ oc -n openshift-adp exec deployment/velero -c velero -- ./velero \
  restore logs ccc7c2d0-6017-11eb-afab-85d0007f5a19-x4lbf
```

4.5.5. メモリーまたは CPU の不足により Pod がクラッシュまたは再起動する

メモリーまたは CPU の不足により Velero または Restic Pod がクラッシュした場合、これらのリソースのいずれかに対して特定のリソースリクエストを設定できます。

4.5.5.1. Velero Pod のリソースリクエストの設定

oadp_v1alpha1_dpa.yaml ファイルの **configuration.velero.podConfig.resourceAllocations** 仕様フィールドを使用して、**Velero** Pod に対する特定のリソース要求を設定できます。

手順

- YAML ファイルで **CPU** および **memory** リソースのリクエストを設定します。

  Velero ファイルの例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
configuration:
  velero:
    podConfig:
      resourceAllocations:
```

```
      requests:
        cpu: 500m
        memory: 256Mi
```

4.5.5.2. Restic Pod のリソースリクエストの設定

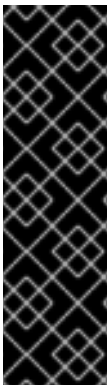
configuration.restic.podConfig.resourceAllocations 仕様フィールドを使用して、**Restic** Pod の特定のリソース要求を設定できます。

手順

- YAML ファイルで **CPU** および **memory** リソースのリクエストを設定します。

  Restic ファイルの例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
configuration:
  restic:
    podConfig:
      resourceAllocations:
        requests:
          cpu: 500m
          memory: 256Mi
```



重要

リソース要求フィールドの値は、Kubernetes リソース要件と同じ形式に従う必要があります。また、**configuration.velero.podConfig.resourceAllocations** または **configuration.restic.podConfig.resourceAllocations** を指定しない場合、Velero Pod または Restic Pod のデフォルトの **resources** 仕様は次のようになります。

```
requests:
  cpu: 500m
  memory: 128Mi
```

4.5.6. Velero と受付 Webhook に関する問題

Velero では、復元中に受付 Webhook の問題を解決する機能が制限されています。受付 Webhook を使用するワークロードがある場合は、追加の Velero プラグインを使用するか、ワークロードの復元方法を変更する必要がある場合があります。

通常、受付 Webhook を使用するワークロードでは、最初に特定の種類のリソースを作成する必要があります。通常、受付 Webhook は子リソースをブロックするため、これは特にワークロードに子リソースがある場合に当てはまります。

たとえば、**service.serving.knative.dev** などの最上位オブジェクトを作成または復元すると、通常、子リソースが自動的に作成されます。最初にこれを行う場合、Velero を使用してこれらのリソースを作成および復元する必要はありません。これにより、Velero が使用する可能性のある受付 Webhook によって子リソースがブロックされるという問題が回避されます。

4.5.6.1. 受付 Webhook を使用する Velero バックアップの回避策の復元

このセクションでは、受付 Webhook を使用するいくつかのタイプの Velero バックアップのリソースを復元するために必要な追加の手順について説明します。

4.5.6.1.1. Knative リソースの復元

Velero を使用して受付 Webhook を使用する Knative リソースをバックアップする際に問題が発生する場合があります。

受付 Webhook を使用する Knative リソースをバックアップおよび復元する場合は、常に最上位の **Service** リソースを最初に復元することで、このような問題を回避できます。

手順

- 最上位の **service.serving.knavtive.dev Service** リソースを復元します。

```
$ velero restore <restore_name> \
  --from-backup=<backup_name> --include-resources \
  service.serving.knavtive.dev
```

4.5.6.1.2. IBM AppConnect リソースの復元

Velero を使用して受付 Webhook を持つ IBM AppConnect リソースを復元するときに問題が発生した場合は、この手順のチェックを実行できます。

手順

1. クラスター内の **kind: MutatingWebhookConfiguration** の受付プラグインの変更があるかチェックします。

```
$ oc get mutatingwebhookconfigurations
```

2. 各 **kind: MutatingWebhookConfiguration** の YAML ファイルを調べて、問題が発生しているオブジェクトの作成をブロックするルールがないことを確認します。詳細は、the official Kuberbetes documentation を参照してください。

3. バックアップ時に使用される **type: Configuration.appconnect.ibm.com/v1beta1** の **spec.version** が、インストールされている Operator のサポート対象であることを確認してください。

関連情報

- 受付プラグイン
- Webhook 受付プラグイン
- Webhook 受付プラグインのタイプ

4.5.7. インストールの問題

Data Protection Application をインストールするときに、無効なディレクトリーまたは誤った認証情報を使用することによって問題が発生する可能性があります。

4.5.7.1. バックアップストレージに無効なディレクトリーが含まれています

Velero Pod ログにエラーメッセージ **Backup storage contains invalid top-level directories** が表示されます。

原因

オブジェクトストレージには、Velero ディレクトリーではないトップレベルのディレクトリーが含まれています。

解決方法

オブジェクトストレージが Velero 専用でない場合は、**DataProtectionApplication** マニフェストで **spec.backupLocations.velero.objectStorage.prefix** パラメーターを設定して、バケットの接頭辞を指定する必要があります。

4.5.7.2. 不正な AWS 認証情報

oadp-aws-registry Pod ログにエラーメッセージ **InvalidAccessKeyId: The AWS Access Key Id you provided does not exist in our records.** が表示されます。

Velero Pod ログには、エラーメッセージ **NoCredentialProviders: no valid providers in chain** が表示されます。

原因

Secret オブジェクトの作成に使用された **credentials-velero** ファイルの形式が正しくありません。

解決方法

次の例のように、**credentials-velero** ファイルが正しくフォーマットされていることを確認します。

サンプル credentials-velero ファイル

```
[default] 1
aws_access_key_id=AKIAIOSFODNN7EXAMPLE 2
aws_secret_access_key=wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
```

1 AWS デフォルトプロファイル。

2 値を引用符 ("、') で囲まないでください。

4.5.8. CR の問題のバックアップおよび復元

Backup および **Restore** カスタムリソース (CR) でこれらの一般的な問題が発生する可能性があります。

4.5.8.1. バックアップ CR はボリュームを取得できません

Backup CR は、エラーメッセージ **InvalidVolume.NotFound: The volume 'vol-xxxx' does not exist** を表示します。

原因

永続ボリューム (PV) とスナップショットの場所は異なるリージョンにあります。

解決方法

1. **DataProtectionApplication** マニフェストの **spec.snapshotLocations.velero.config.region** キーの値を編集して、スナップショットの場所が PV と同じリージョンにあるようにします。

2. 新しい **Backup** CR を作成します。

4.5.8.2. バックアップ CR ステータスは進行中のままです

Backup CR のステータスは **InProgress** のフェーズのままであり、完了しません。

原因

バックアップが中断された場合は、再開することができません。

解決方法

1. **Backup** CR の詳細を取得します。

```
$ oc -n {namespace} exec deployment/velero -c velero -- ./velero \
  backup describe <backup>
```

2. **Backup** CR を削除します。

```
$ oc delete backup <backup> -n openshift-adp
```

進行中の **Backup** CR はファイルをオブジェクトストレージにアップロードしていないため、バックアップの場所をクリーンアップする必要はありません。

3. 新しい **Backup** CR を作成します。

4.5.8.3. バックアップ CR ステータスが PartlyFailed のままになる

Restic が使用されていない **Backup** CR のステータスは、**PartiallyFailed** フェーズのままで、完了しません。関連する PVC のスナップショットは作成されません。

原因

CSI スナップショットクラスに基づいてバックアップが作成されているが、ラベルがない場合、CSI スナップショットプラグインはスナップショットの作成に失敗します。その結果、**Velero** Pod は次のようなエラーをログに記録します。

+

```
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
```

解決方法

1. **Backup** CR を削除します。

```
$ oc delete backup <backup> -n openshift-adp
```

2. 必要に応じて、**BackupStorageLocation** に保存されているデータをクリーンアップして、領域を解放します。
3. ラベル **velero.io/csi-volumesnapshot-class=true** を **VolumeSnapshotClass** オブジェクトに適用します。

```
$ oc label volumesnapshotclass/<snapclass_name> velero.io/csi-volumesnapshot-class=true
```

4. 新しい **Backup** CR を作成します。

4.5.9. Restic の問題

Restic を使用してアプリケーションのバックアップを作成すると、これらの問題が発生する可能性があります。

4.5.9.1. root_squash が有効になっている NFS データボリュームの Restic パーミッションエラー

Restic Pod ログには、エラーメッセージ **controller=pod-volume-backup error="fork/exec/usr/bin/restic: permission denied"** が表示されます。

原因

NFS データボリュームで **root_squash** が有効になっている場合、**Restic** は **nfsnobody** にマッピングされ、バックアップを作成する権限がありません。

解決方法

この問題を解決するには、**Restic** の補足グループを作成し、そのグループ ID を **DataProtectionApplication** マニフェストに追加します。

1. NFS データボリューム上に **Restic** の補足グループを作成します。
2. NFS ディレクトリーに **setgid** ビットを設定して、グループの所有権が継承されるようにします。
3. 次の例のように、**spec.configuration.restic.supplementalGroups** パラメーターおよびグループ ID を **DataProtectionApplication** マニフェストに追加します。

```
spec:
  configuration:
    restic:
      enable: true
      supplementalGroups:
      - <group_id> 1
```

1 補助グループ ID を指定します。

4. **Restic** Pod が再起動し、変更が適用されるまで待機します。

4.5.9.2. バケットが空になった後に、Restic Backup CR を再作成することはできない

namespace の Restic **Backup** CR を作成し、オブジェクトストレージバケットを空にしてから、同じ namespace の **Backup** CR を再作成すると、再作成された **Backup** CR は失敗します。

velero Pod ログにエラーメッセージ**stderr=Fatal: unable to open config file: Stat: The specified key does not exist.\nIs there a repository at the following location?** が表示されます。

原因

オブジェクトストレージから Restic ディレクトリーが削除された場合、Velero は **ResticRepository** マニフェストから Restic リポジトリーを再作成または更新しません。詳細については、Velero issue 4421 を参照してください。

解決方法

- 次のコマンドを実行して、関連する Restic リポジトリーを namespace から削除します。

```
$ oc delete resticrepository openshift-adp <name_of_the_restic_repository>
```

次のエラーログでは、**mysql-persistent** が問題のある Restic リポジトリーです。わかりやすくするために、リポジトリーの名前は斜体で表示されます。

```
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
```

4.5.10. must-gather ツールの使用

must-gather ツールを使用して、OADP カスタムリソースのログ、メトリクス、および情報を収集できます。

must-gather データはすべてのカスタマーケースに割り当てられる必要があります。

前提条件

- **cluster-admin** ロールを持つユーザーとして OpenShift Container Platform クラスターにログインする必要があります。
- OpenShift CLI (**oc**) がインストールされている。

手順

1. **must-gather** データを保存するディレクトリーに移動します。
2. 次のデータ収集オプションのいずれかに対して、**oc adm must-gather** コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.1
```

-

データは **must-gather/must-gather.tar.gz** として保存されます。このファイルを Red Hat カスタマーポータル で作成したサポートケースにアップロードすることができます。

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.1 \
  -- /usr/bin/gather_metrics_dump
```

この操作には長時間かかる場合があります。データは **must-gather/metrics/prom_data.tar.gz** として保存されます。

Prometheus コンソールを使用したメトリクスデータの表示
Prometheus コンソールでメトリックデータを表示できます。

手順

1. **prom_data.tar.gz** ファイルを解凍します。

```
$ tar -xvzf must-gather/metrics/prom_data.tar.gz
```

2. ローカルの Prometheus インスタンスを作成します。

```
$ make prometheus-run
```

このコマンドでは、Prometheus URL が出力されます。

出力

```
Started Prometheus on http://localhost:9090
```

3. Web ブラウザーを起動して URL に移動し、Prometheus Web コンソールを使用してデータを表示します。

4. データを確認した後に、Prometheus インスタンスおよびデータを削除します。

```
$ make prometheus-cleanup
```

4.6. OADP で使用される API

このドキュメントには、OADP で使用できる次の API に関する情報が記載されています。

- Velero API
- OADP API

4.6.1. Velero API

Velero API ドキュメントは、Red Hat ではなく、Velero によって管理されています。これは Velero API types にあります。

4.6.2. OADP API

次の表は、OADP API の構造を示しています。

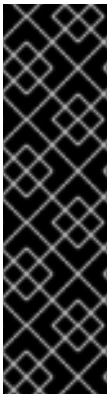
表4.3 DataProtectionApplicationSpec

プロパティー	型	説明
backupLocations	[] **BackupLocation**	**BackupStorageLocations** に使用する設定のリストを定義します。
snapshotLocations	[] **SnapshotLocation**	**VolumeSnapshotLocations** に使用する設定のリストを定義します。
unsupportedOverrides	map [UnsupportedImageKey] string	デプロイされた依存イメージを開発用にオーバーライドするために使用できます。オプションは、**veleroImageFqin**、**awsPluginImageFqin**、**openshiftPluginImageFqin**、**azurePluginImageFqin**、**gcpPluginImageFqin**、**csiPluginImageFqin**、**dataMoverImageFqin**、**resticRestoreImageFqin**、**kubevirtPluginImageFqin**、および **operator-type** です。
podAnnotations	map [string] string	Operator によってデプロイされた Pod にアノテーションを追加するために使用されます。
podDnsPolicy	**DNSPolicy**	Pod の DNS の設定を定義します。
podDnsConfig	**PodDNSConfig**	**DNSPolicy** から生成されたパラメーターに加えて、Pod の DNS パラメーターを定義します。
backupImages	*bool	イメージのバックアップと復元を有効にするためにレジストリーを展開するかどうかを指定するために使用されます。
configuration	***ApplicationConfig**	データ保護アプリケーションのサーバー設定を定義するために使用されます。
features	***Features**	テクノロジープレビュー機能を有効にするための DPA の設定を定義します。

OADP API の完全なスキーマ定義。

表4.4 BackupLocation

プロパティー	型	説明
velero	*velero.BackupStorageLocationSpec	Backup Storage Location で説明されているとおり、ボリュームスナップショットの保存場所。
bucket	*CloudStorageLocation	[テクノロジープレビュー] 一部のクラウドストレージプロバイダーで、バックアップストレージの場所として使用するバケットの作成を自動化します。



重要

bucket パラメーターはテクノロジープレビュー機能としてのみ提供されます。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、テクノロジープレビュー機能のサポート範囲 を参照してください。

BackupLocation タイプの完全なスキーマ定義。

表4.5 SnapshotLocation

プロパティー	型	説明
velero	*VolumeSnapshotLocationSpec	Volume Snapshot Location で説明されているとおり、ボリュームスナップショットの保存場所。

SnapshotLocation タプの完全なスキーマ定義。

表4.6 ApplicationConfig

プロパティー	型	説明
velero	*VeleroConfig	Velero サーバーの設定を定義します。
restic	*ResticConfig	Restic サーバーの設定を定義します。

ApplicationConfig タイプの完全なスキーマ定義。

表4.7 VeleroConfig

プロパティー	型	説明
featureFlags	[] string	Velero インスタンスで有効にする機能のリストを定義します。
defaultPlugins	[] string	次のタイプのデフォルトの Velero プラグインをインストールできます: **aws**、**azure**、**csi**、**gcp**、**kubevirt**、および **openshift**。
customPlugins	[]CustomPlugin	カスタム Velero プラグインのインストールに使用されます。 デフォルトおよびカスタムのプラグインについては、OADP plug-ins で説明しています。
restoreResourcesVersionPriority	string	**EnableAPIGroupVersions** 機能フラグと組み合わせて使用するために定義されている場合に作成される設定マップを表します。このフィールドを定義すると、**EnableAPIGroupVersions** が Velero サーバー機能フラグに自動的に追加されます。
noDefaultBackupLocation	bool	デフォルトのバックアップストレージの場所を設定せずに Velero をインストールするには、インストールを確認するために **noDefaultBackupLocation** フラグを設定する必要があります。
podConfig	***PodConfig**	**Velero** Pod の設定を定義します。
logLevel	string	Velero サーバーのログレベル (最も詳細なログを記録するには **debug** を使用し、Velero のデフォルトは未設定のままにします)。有効なオプションは、**trace**、**debug**、**info**、**warning**、**error**、**fatal**、および **panic** です。

VeleroConfig タイプの完全なスキーマ定義。

表4.8 CustomPlugin

プロパティー	型	説明
name	string	カスタムプラグインの名前。
image	string	カスタムプラグインのイメージ。

CustomPlugin タイプの完全なスキーマ定義。

表4.9 ResticConfig

プロパティー	型	説明
enable	*bool	**true** に設定すると、Restic を使用したバックアップと復元が有効になります。**false** に設定すると、スナップショットが必要になります。
supplementalGroups	[]int64	**Restic** Pod に適用される Linux グループを定義します。
timeout	string	Restic タイムアウトを定義するユーザー指定の期間文字列。デフォルト値は **1hr** (1 時間) です。期間文字列は、符号付きの場合もある 10 進数のシーケンスであり、それぞれに **300ms**、-1.5h` または **2h45m** などのオプションの分数と単位接尾辞が付いています。有効な時間単位は、**ns**、**us** (または **µs**)、**ms**、**s**、**m**、および **h** です。
podConfig	***PodConfig**	**Restic** Pod の設定を定義します。

ResticConfig タイプの完全なスキーマ定義。

表4.10 PodConfig

プロパティー	型	説明
nodeSelector	map [string] string	**Velero podSpec** または **Restic podSpec** に提供される **nodeSelector** を定義します。
tolerations	[]Toleration	Velero デプロイメントまたは Restic **daemonset** に適用される toleration のリストを定義します。

プロパティー	型	説明
resourceAllocations	ResourceRequirements	Setting Velero CPU and memory resource allocations の説明に従って、**Velero** Pod または **Restic** Pod の特定のリソースの **limits** および **requests** を設定します。
labels	map [string] string	Pod に追加するラベル。

PodConfig タイプの完全なスキーマ定義。

表4.11 機能

プロパティー	型	説明
dataMover	***DataMover**	Data Mover の設定を定義します。

Features タイプの完全なスキーマ定義。

表4.12 DataMover

プロパティー	型	説明
enable	bool	**true** に設定すると、ボリュームスナップショットムーバーコントローラーと変更された CSI Data Mover プラグインがデプロイされます。**false** に設定すると、これらはデプロイされません。
credentialName	string	Data Mover のユーザー指定の Restic **Secret** 名。
timeout	string	**VolumeSnapshotBackup** と **VolumeSnapshotRestore** が完了するまでのユーザー指定の期間文字列。デフォルトは **10m** (10分) です。期間文字列は、符号付きの場合もある 10 進数のシーケンスであり、それぞれに **300ms**、-1.5h` または **2h45m** などのオプションの分数と単位接尾辞が付いています。有効な時間単位は、**ns**、**us** (または **µs**)、**ms**、**s**、**m**、および **h** です。

OADP API の詳細については、OADP Operator を参照してください。

4.7. OADP の高度な特徴と機能

このドキュメントでは、OpenShift API for Data Protection (OADP) の高度な特徴と機能に関する情報を提供します。

4.7.1. 同一クラスター上での異なる Kubernetes API バージョンの操作

4.7.1.1. クラスター上の Kubernetes API グループバージョンの一覧表示

ソースクラスターは複数のバージョンの API を提供する場合があり、これらのバージョンの 1 つが優先 API バージョンになります。たとえば、**Example** という名前の API を持つソースクラスターは、**example.com/v1** および **example.com/v1beta2** API グループで使用できる場合があります。

Velero を使用してそのようなソースクラスターをバックアップおよび復元する場合、Velero は、Kubernetes API の優先バージョンを使用するリソースのバージョンのみをバックアップします。

上記の例では、**example.com/v1** が優先 API である場合、Velero は **example.com/v1** を使用するリソースのバージョンのみをバックアップします。さらに、Velero がターゲットクラスターでリソースを復元するには、ターゲットクラスターで使用可能な API リソースのセットに **example.com/v1** が登録されている必要があります。

したがって、ターゲットクラスター上で Kubernetes API グループバージョンのリストを生成して、優先 API バージョンが使用可能な API リソースのセットに登録されていることを確認する必要があります。

手順

- 以下のコマンドを入力します。

```
$ oc api-resources
```

4.7.1.2. API グループバージョンの有効化について

デフォルトでは、Velero は Kubernetes API の優先バージョンを使用するリソースのみをバックアップします。ただし、Velero には、この制限を克服する機能 (Enable API Group Versions) も含まれています。ソースクラスターでこの機能を有効にすると、Velero は優先バージョンだけでなく、クラスターでサポートされている **すべての** Kubernetes API グループバージョンをバックアップします。バージョンがバックアップ .tar ファイルに保存されると、目的のクラスターで復元できるようになります。

たとえば、**Example** という名前の API を持つソースクラスターが、**example.com/v1** および **example.com/v1beta2** API グループで使用でき、**example.com/v1** が優先 API だとします。

Enable API Group Versions 機能を有効にしないと、Velero は **Example** の優先 API グループバージョン (**example.com/v1**) のみをバックアップします。この機能を有効にすると、Velero は **example.com/v1beta2** もバックアップします。

宛先クラスターで Enable API Group Versions 機能が有効になっている場合、Velero は、API グループバージョンの優先順位に基づいて、復元するバージョンを選択します。

注記

Enable API Group Versions はまだベータ版です。

Velero は次のアルゴリズムを使用して API バージョンに優先順位を割り当てます。この場合、**1** は優先順位が最も高くなります。

1. **宛先** クラスターの優先バージョン
2. source_ クラスターの優先バージョン
3. Kubernetes バージョンの優先順位が最も高い共通の非優先サポート対象バージョン

関連情報

- Enable API Group Versions Feature

4.7.1.3. Enable API Group Versions の使用

Velero の Enable API Group Versions 機能を使用して、優先バージョンだけでなく、クラスターでサポートされている **すべての** Kubernetes API グループバージョンをバックアップできます。

注記

Enable API Group Versions はまだベータ版です。

手順

- **EnableAPIGroupVersions** 機能フラグを設定します。

```
apiVersion: oadp.openshift.io/vialpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      featureFlags:
      - EnableAPIGroupVersions
```

関連情報

- Enable API Group Versions Feature

4.7.2. 1 つのクラスターからデータをバックアップし、別のクラスターに復元する

4.7.2.1. あるクラスターからのデータのバックアップと別のクラスターへの復元について

{oadp-first} は、同じ OpenShift Container Platform クラスター内のアプリケーションデータをバックアップおよび復元するように設計されています。Migration Toolkit for Containers (MTC) は、アプリケーションデータを含むコンテナーを 1 つの OpenShift Container Platform クラスターから別のクラスターに移行するように設計されています。

OADP を使用して、1 つの OpenShift Container Platform クラスターからアプリケーションデータをバックアップし、それを別のクラスターに復元できます。ただし、これを行うことは、MTC または OADP を使用して同じクラスター上でバックアップと復元を行うよりも複雑です。

OADP を使用して 1 つのクラスターからデータをバックアップし、それを別のクラスターに復元するには、OADP を使用して同じクラスター上でデータをバックアップおよび復元する場合に適用される前提条件と手順に加えて、次の要素を考慮する必要があります。

- Operator
- Velero の使用
- UID と GID の範囲

4.7.2.1.1. Operator

バックアップと復元を成功させるには、アプリケーションのバックアップから Operator を除外する必要があります。

4.7.2.1.2. Velero の使用

OADP が構築されている Velero は、クラウドプロバイダー間での永続ボリュームスナップショットの移行をネイティブにサポートしていません。クラウドプラットフォーム間でボリュームスナップショットデータを移行するには、ファイルシステムレベルでボリュームの内容をバックアップする Velero Restic ファイルシステムバックアップオプションを有効にする **か、または** CSI スナップショットに OADP Data Mover を使用する必要があります。

注記

OADP 1.1 以前では、Velero Restic ファイルシステムのバックアップオプションは **restic** と呼ばれます。OADP 1.2 以降では、Velero Restic ファイルシステムのバックアップオプションは **file-system-backup** と呼ばれます。

注記

Velero のファイルシステムバックアップ機能は Kopia と Restic の両方をサポートしていますが、現在 OADP は Restic のみをサポートしています。

- AWS リージョン間または Microsoft Azure リージョン間でデータを移行するには、Velero の File System Backup も使用する必要があります。
- Velero は、ソースクラスターより **前の** Kubernetes バージョンを使用したクラスターへのデータの復元をサポートしていません。
- 理論的には、移行元よりも **新しい** Kubernetes バージョンを備えた移行先にワークロードを移行することは可能ですが、カスタムリソースごとにクラスター間の API グループの互換性を考慮する必要があります。Kubernetes バージョンのアップグレードによりコアまたはネイティブ API グループの互換性が失われる場合は、まず影響を受けるカスタムリソースを更新する必要があります。

4.7.2.1.3. UID と GID の範囲

あるクラスターからデータをバックアップし、それを別のクラスターに復元する場合、UID (ユーザー ID) および GID (グループ ID) の範囲に関して潜在的な問題が発生する可能性があります。次のセクションでは、これらの潜在的な問題と軽減策について説明します。

問題点のまとめ

namespace の UID 範囲および GID 範囲は、宛先クラスターで変更される可能性があります。OADP は、OpenShift UID 範囲のメタデータをバックアップおよび復元しません。バックアップされたアプ

リケーションに特定の UID が必要な場合は、復元時にその範囲が使用可能であることを確認してください。OpenShift の UID 範囲および GID 範囲の詳細は、A Guide to OpenShift and UIDs を参照してください。

問題の詳細な説明

シェルコマンド **oc create namespace** を使用して OpenShift Container Platform でネームスペースを作成すると、OpenShift Container Platform は、使用可能な UID プールからの一意のユーザー ID (UID) 範囲、補足グループ (GID) 範囲、および一意の SELinux MCS ラベルを namespace に割り当てます。この情報は、クラスターの **metadata.annotations** フィールドに保存されます。この情報はセキュリティーコンテキスト制約 (SCC) アノテーションの一部であり、次のコンポーネントで設定されます。

- **openshift.io/sa.scc.mcs**
- **openshift.io/sa.scc.supplemental-groups**
- **openshift.io/sa.scc.uid-range**

OADP を使用して namespace を復元すると、宛先クラスターの情報をリセットせずに、**metadata.annotations** 内の情報が自動的に使用されます。その結果、次のいずれかに該当する場合、ワークロードはバックアップデータにアクセスできない可能性があります。

- たとえば、別のクラスター上に、異なる SCC アノテーションを持つ既存の namespace があります。この場合、OADP はバックアップ時に、復元しようとしている namespace ではなく、既存の namespace を再利用します。
- バックアップではラベルセレクターが使用されましたが、ワークロードが実行される namespace にはラベルがありません。この場合、OADP は namespace をバックアップしませんが、代わりに、バックアップした namespace のアノテーションを含まない新しい namespace を復元中に作成します。これにより、新しい UID 範囲が namespace に割り当てられます。
  OpenShift Container Platform が永続ボリュームデータのバックアップ時から変更された namespace アノテーションに基づいて Pod に **securityContext** UID を割り当てる場合、これはお客様 のワークロードにとって問題になる可能性があります。
- コンテナー UID はファイル所有者の UID と一致しなくなりました。
- OpenShift Container Platform がバックアップクラスターのデータと一致するように宛先クラスターの UID 範囲を変更しなかったため、エラーが発生します。その結果、バックアップクラスターは宛先クラスターとは異なる UID を持つことになり、アプリケーションは宛先クラスターに対してデータの読み取りまたは書き込みを行うことができなくなります。

軽減策

次の 1 つ以上の緩和策を使用して、UID 範囲および GID 範囲の問題を解決できます。

- 簡単な緩和策:
  - **Backup** CR のラベルセレクターを使用して、バックアップに含めるオブジェクトをフィルター処理する場合は、必ずこのラベルセレクターをワークスペースを含む namespace に追加してください。
  - 同じ名前の namespace を復元する前に、宛先クラスター上の namespace の既存のバージョンを削除してください。
- 高度な緩和策:

- 移行後の UID 範囲の修正の手順 1~4 を実行して、移行後に UID 範囲を修正します。ステップ 1 はオプションです。

あるクラスターでのデータのバックアップと別のクラスターでのリストアの問題の解決に重点を置いた、OpenShift Container Platform の UID 範囲および GID 範囲の詳細な説明は、A Guide to OpenShift and UIDs を 参照してください。

4.7.2.2. 1つのクラスターからデータをバックアップし、別のクラスターに復元する

一般に、同じクラスターにデータをバックアップおよび復元するのと同じ方法で、1 つの OpenShift Container Platform クラスターからデータをバックアップし、別の OpenShift Container Platform クラスターに復元します。ただし、ある OpenShift Container Platform クラスターからデータをバックアップし、それを別のクラスターにリストアする場合は、追加の前提条件と手順の違いがいくつかあります。

前提条件

- プラットフォーム (AWS、Microsoft Azure、GCP など) でのバックアップと復元に関連するすべての前提条件、特にデータ保護アプリケーション (DPA) の前提条件については、このガイドの関連セクションで説明されています。

手順

- ご使用のプラットフォームに指定されている手順に次の追加を加えます。
  - リソースを別のクラスターに復元するには、バックアップストアの場所 (BSL) とボリュームスナップショットの場所が同じ名前とパスを持つようにしてください。
  - 同じオブジェクトストレージの場所の認証情報をクラスター全体で共有します。
  - 最良の結果を得るには、OADP を使用して宛先クラスターに namespace を作成します。
  - Velero **file-system-backup** オプションを使用する場合は、次のコマンドを実行して、バックアップ中に使用する **--default-volumes-to-fs-backup** フラグを有効にします。

```
$ velero backup create <backup_name> --default-volumes-to-fs-backup
<any_other_options>
```

注記

OADP 1.2 以降では、Velero Restic オプションは **file-system-backup** と呼ばれます。

4.7.3. 関連情報

API グループのバージョンの詳細は、同一クラスター上での異なる Kubernetes API バージョンの操作を参照してください。

OADP Data Mover の詳細は、CSI スナップショットに Data Mover を使用する を参照してください。

OADP での Restic の使用の詳細は、Restic を使用したアプリケーションのバックアップ を参照してください。

第5章 コントロールプレーンのバックアップおよび復元

5.1. ETCD のバックアップ

etcd は OpenShift Container Platform のキーと値のストアであり、すべてのリソースオブジェクトの状態を保存します。

クラスターの etcd データを定期的にバックアップし、OpenShift Container Platform 環境外の安全な場所に保存するのが理想的です。インストールの 24 時間後に行われる最初の証明書のローテーションが完了するまで etcd のバックアップを実行することはできません。ローテーションの完了前に実行すると、バックアップに期限切れの証明書が含まれることになります。etcd スナップショットは I/O コストが高いため、ピーク使用時間以外に etcd バックアップを取得することもお勧めします。

クラスターのアップグレード後に必ず etcd バックアップを作成してください。これは、クラスターを復元する際に、同じ z-stream リリースから取得した etcd バックアップを使用する必要があるために重要になります。たとえば、OpenShift Container Platform 4.y.z クラスターは、4.y.z から取得した etcd バックアップを使用する必要があります。

重要

コントロールプレーンホストでバックアップスクリプトの単一の呼び出しを実行して、クラスターの etcd データをバックアップします。各コントロールプレーンホストのバックアップを取得しないでください。

etcd のバックアップを作成した後に、クラスターの直前の状態への復元を実行できます。

5.1.1. etcd データのバックアップ

以下の手順に従って、etcd スナップショットを作成し、静的 Pod のリソースをバックアップして etcd データをバックアップします。このバックアップは保存でき、etcd を復元する必要がある場合に後で使用することができます。

重要

単一のコントロールプレーンホストからのバックアップのみを保存します。クラスター内の各コントロールプレーンホストからのバックアップは取得しないでください。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- クラスター全体のプロキシーが有効になっているかどうかを確認している。

ヒント

oc get proxy cluster -o yaml の出力を確認して、プロキシーが有効にされているかどうかを確認できます。プロキシーは、**httpProxy**、**httpsProxy**、および **noProxy** フィールドに値が設定されている場合に有効にされます。

手順

1. コントロールプレーンノードのデバッグセッションを開始します。

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

2. ルートディレクトリーを **/host** に変更します。

```
sh-4.2# chroot /host
```

3. クラスター全体のプロキシーが有効になっている場合は、 **NO_PROXY**、**HTTP_PROXY**、および **HTTPS_PROXY** 環境変数をエクスポートしていることを確認します。

4. **etcd-snapshot-backup.sh** スクリプトを実行し、バックアップの保存先となる場所を渡します。

ヒント

cluster-backup.sh スクリプトは etcd Cluster Operator のコンポーネントとして維持され、**etcdctl snapshot save** コマンドに関連するラッパーです。

```
sh-4.4# /usr/local/bin/cluster-backup.sh /home/core/assets/backup
```

スクリプトの出力例

```
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
```

この例では、コントロールプレーンホストの **/home/core/assets/backup/** ディレクトリーにファイルが 2 つ作成されます。

- **snapshot_<datetimestamp>.db**: このファイルは etcd スナップショットです。**cluster-backup.sh** スクリプトで、その有効性を確認します。

- **static_kuberesources_<datetimestamp>.tar.gz**: このファイルには、静的 Pod のリソースが含まれます。etcd 暗号化が有効にされている場合、etcd スナップショットの暗号化キーも含まれます。

注記

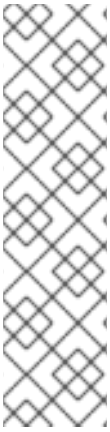
etcd 暗号化が有効にされている場合、セキュリティー上の理由から、この 2 つ目のファイルを etcd スナップショットとは別に保存することが推奨されます。ただし、このファイルは etcd スナップショットから復元するために必要になります。

etcd 暗号化はキーではなく値のみを暗号化することに注意してください。つまり、リソースタイプ、namespace、およびオブジェクト名は暗号化されません。

5.2. 正常でない ETCD メンバーの置き換え

本書では、単一の正常でない etcd メンバーを置き換えるプロセスについて説明します。

このプロセスは、マシンが実行されていないか、ノードが準備状態にないことによって etcd メンバーが正常な状態にないか、etcd Pod がクラッシュループしているためにこれが正常な状態にないかによって異なります。



注記

コントロールプレーンホストの大部分を損失した場合は、この手順ではなく、ディザスターリカバリー手順に従って、以前のクラスター状態への復元 を行います。

コントロールプレーンの証明書が置き換えているメンバーで有効でない場合は、この手順ではなく、期限切れのコントロールプレーン証明書からの回復手順を実行する必要があります。

コントロールプレーンノードが失われ、新規ノードが作成される場合、etcd クラスター Operator は新規 TLS 証明書の生成と、ノードの etcd メンバーとしての追加を処理します。

5.2.1. 前提条件

- 正常でない etcd メンバーを置き換える前に、etcd バックアップを作成します。

5.2.2. 正常でない etcd メンバーの特定

クラスターに正常でない etcd メンバーがあるかどうかを特定することができます。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。

手順

1. 以下のコマンドを使用して **EtcdMembersAvailable** ステータス条件のステータスを確認します。

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="EtcdMembersAvailable")]}{.message}{"\n"}'
```

2. 出力を確認します。

```
2 of 3 members are available, ip-10-0-131-183.ec2.internal is unhealthy
```

この出力例は、**ip-10-0-131-183.ec2.internal** etcd メンバーが正常ではないことを示しています。

5.2.3. 正常でない etcd メンバーの状態の判別

正常でない etcd メンバーを置き換える手順は、etcd メンバーが以下のどの状態にあるかによって異なります。

- マシンが実行されていないか、ノードが準備状態にない
- etcd Pod がクラッシュループしている。

以下の手順では、etcd メンバーがどの状態にあるかを判別します。これにより、正常でない etcd メンバーを置き換えるために実行する必要のある手順を確認できます。

注記

マシンが実行されていないか、ノードが準備状態にないものの、すぐに正常な状態に戻ることが予想される場合は、etcd メンバーを置き換える手順を実行する必要はありません。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- 正常でない etcd メンバーを特定している。

手順

1. **マシンが実行されていない**かどうかを判別します。

```
$ oc get machines -A -ojsonpath='{range .items[*]}{@.status.nodeRef.name}{"\t"}
{@.status.providerStatus.instanceState}{"\n"}' | grep -v running
```

出力例

```
ip-10-0-131-183.ec2.internal  stopped (1)
```

(1) この出力には、ノードおよびノードのマシンのステータスを一覧表示されます。ステータスが **running** 以外の場合は、**マシンは実行されていません。**

マシンが実行されていない 場合は、**マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え**の手順を実行します。

2. **ノードが準備状態にない**かどうかを判別します。

以下のシナリオのいずれかが true の場合、**ノードは準備状態にありません。**

- マシンが実行されている場合は、ノードに到達できないかどうかを確認します。

```
$ oc get nodes -o jsonpath='{range .items[*]}{"\n"}{.metadata.name}{"\t"}{range
.spec.taints[*]}{.key}{" "}' | grep unreachable
```

出力例

```
ip-10-0-131-183.ec2.internal node-role.kubernetes.io/master
node.kubernetes.io/unreachable node.kubernetes.io/unreachable 1
```

1 ノードが **unreachable** テイントと共に一覧表示される場合、**ノードの準備はできていません。**

- ノードが以前として到達可能である場合は、そのノードが **NotReady** として一覧表示されているかどうかを確認します。

```
$ oc get nodes -l node-role.kubernetes.io/master | grep "NotReady"
```

出力例

```
ip-10-0-131-183.ec2.internal   NotReady   master   122m   v1.23.0 1
```

1 ノードが **NotReady** として一覧表示されている場合、**ノードの準備はできていません。**

ノードの準備ができていない 場合は、**マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え**の手順を実行します。

3. **etcd Pod がクラッシュループしている**かどうかを判別します。
マシンが実行され、ノードが準備できている場合は、etcd Pod がクラッシュループしているかどうかを確認します。

   a. すべてのコントロールプレーンノードが **Ready** として一覧表示されていることを確認します。

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

出力例

```
NAME                           STATUS   ROLES    AGE     VERSION
ip-10-0-131-183.ec2.internal   Ready    master   6h13m   v1.23.0
ip-10-0-164-97.ec2.internal    Ready    master   6h13m   v1.23.0
ip-10-0-154-204.ec2.internal   Ready    master   6h13m   v1.23.0
```

   b. etcd Pod のステータスが **Error** または **CrashloopBackoff** のいずれかであるかどうかを確認します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
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 この Pod のこのステータスは **Error** であるため、**etcd Pod はクラッシュループしています。**

etcd Pod がクラッシュループしている場合、**etcd Pod がクラッシュループしている場合の正常でない etcd メンバーの置き換え**についての手順を実行します。

5.2.4. 正常でない etcd メンバーの置き換え

正常でない etcd メンバーの状態に応じて、以下のいずれかの手順を使用します。

- マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え
- etcd Pod がクラッシュループしている場合の正常でない etcd メンバーの置き換え
- 異常停止したベアメタル etcd メンバーの置き換え

5.2.4.1. マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え

マシンが実行されていない、またはノードの準備ができていない、正常ではない etcd メンバーを置き換える手順を説明します。

前提条件

- 正常でない etcd メンバーを特定している。
- マシンが実行されていないか、またはノードが準備状態にないことを確認している。
- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- etcd のバックアップを取得している。

重要

問題が発生した場合にクラスターを復元できるように、この手順を実行する前に etcd バックアップを作成しておくことは重要です。

手順

1. 正常でないメンバーを削除します。
   a. 影響を受けるノード上に **ない** Pod を選択します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
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. 実行中の etcd コンテナーに接続し、影響を受けるノードにない Pod の名前を渡します。クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

c. メンバーの一覧を確認します。

```
sh-4.2# etcdctl member list -w table
```

出力例

```
+------------------+---------+----------------------------+---------------------------+---------------
-----------+
|        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 |
+------------------+---------+----------------------------+---------------------------+---------------
-----------+
```

これらの値はこの手順で後ほど必要となるため、ID および正常でない etcd メンバーの名前を書き留めておきます。**$ etcdctl endpoint health** コマンドは、補充手順が完了し、新しいメンバーが追加されるまで、削除されたメンバーを一覧表示します。

d. ID を **etcdctl member remove** コマンドに指定して、正常でない etcd メンバーを削除します。

```
sh-4.2# etcdctl member remove 6fc1e7c9db35841d
```

出力例

```
Member 6fc1e7c9db35841d removed from cluster ead669ce1fbfb346
```

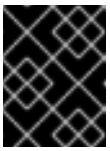
e. メンバーの一覧を再度表示し、メンバーが削除されたことを確認します。

```
sh-4.2# etcdctl member list -w table
```

出力例

```
+-----------------+---------+----------------------------+--------------------------+---------------
-----------+
|        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 |
+-----------------+---------+----------------------------+--------------------------+---------------
-----------+
```

これでノードシェルを終了できます。



重要

メンバーを削除した後、残りの etcd インスタンスが再起動している間、クラスターに短時間アクセスできない場合があります。

2. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

3. 削除された正常でない etcd メンバーの古いシークレットを削除します。

   a. 削除された正常でない etcd メンバーのシークレットを一覧表示します。

```
$ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
```

1 この手順で先ほど書き留めた正常でない etcd メンバーの名前を渡します。

以下の出力に示されるように、ピア、サービング、およびメトリクスシークレットがあります。

出力例

```
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. 削除された正常でない etcd メンバーのシークレットを削除します。

      i. ピアシークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
```

      ii. 提供シークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
```

iii. メトリクスシークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-183.ec2.internal
```

4. コントロールプレーンマシンを削除して再度作成します。このマシンが再作成されると、新規リビジョンが強制的に実行され、etcd は自動的にスケールアップします。
インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。それ以外の場合は、最初に作成する際に使用した方法と同じ方法を使用して新規マスターを作成する必要があります。

a. 正常でないメンバーのマシンを取得します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
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 これは正常でないノードのコントロールプレーンマシンです (**ip-10-0-131-183.ec2.internal**)。

b. マシン設定をファイルシステムのファイルに保存します。

```
$ oc get machine clustername-8qw5l-master-0 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1 正常でないノードのコントロールプレーンマシンの名前を指定します。

c. 直前の手順で作成された **new-master-machine.yaml** ファイルを編集し、新しい名前を割り当て、不要なフィールドを削除します。

i. **status** セクション全体を削除します。

```
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. **metadata.name** フィールドを新規の名前に変更します。
古いマシンと同じベース名を維持し、最後の番号を次に利用可能な番号に変更することが推奨されます。この例では、**clustername-8qw5l-master-0** は **clustername-8qw5l-master-3** に変更されています。

以下に例を示します。

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. **spec.providerID** フィールドを削除します。

```
  providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

d. 正常でないメンバーのマシンを削除します。

```
$ oc delete machine -n openshift-machine-api clustername-8qw5l-master-0 1
```

1 正常でないノードのコントロールプレーンマシンの名前を指定します。

e. マシンが削除されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
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. **new-master-machine.yaml** ファイルを使用して新規マシンを作成します。

```
$ oc apply -f new-master-machine.yaml
```

g. 新規マシンが作成されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
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 新規マシン **clustername-8qw5l-master-3** が作成され、**Provisioning** から **Running** にフェーズが変更されると準備状態になります。

新規マシンが作成されるまでに数分の時間がかかる場合があります。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

5. 次のコマンドを入力して、クォーラムガードをオンに戻します。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

6. 次のコマンドを入力して、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

7. 単一ノードの OpenShift を使用している場合は、ノードを再起動します。そうしないと、etcd クラスター Operator で次のエラーが発生する可能性があります。

出力例

```
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]
```

検証

1. すべての etcd Pod が適切に実行されていることを確認します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
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
```

直前のコマンドの出力に 2 つの Pod のみが一覧表示される場合、etcd の再デプロイメントを手動で強制できます。クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-"$( date --rfc-3339=ns )""}}' --type=merge 1
```

1 **forceRedeploymentReason** 値は一意である必要があります。そのため、タイムスタンプが付加されます。

2. 3 つの etcd メンバーがあることを確認します。

   a. 実行中の etcd コンテナーに接続し、影響を受けるノードになかった Pod の名前を渡します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

   ```
   $ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
   ```

   b. メンバーの一覧を確認します。

   ```
   sh-4.2# etcdctl member list -w table
   ```

   出力例

   ```
   +------------------+---------+-----------------------------+--------------------------+----------------
   -----------+
   |        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 |
   +------------------+---------+-----------------------------+--------------------------+----------------
   -----------+
   ```

   直前のコマンドの出力に 4 つ以上の etcd メンバーが表示される場合、不要なメンバーを慎重に削除する必要があります。

> **警告**
>
> 必ず適切な etcd メンバーを削除します。適切な etcd メンバーを削除すると、クォーラム (定足数) が失われる可能性があります。

5.2.4.2. etcd Pod がクラッシュループしている場合の正常でない etcd メンバーの置き換え

この手順では、etcd Pod がクラッシュループしている場合の正常でない etcd メンバーを置き換える手順を説明します。

前提条件

- 正常でない etcd メンバーを特定している。
- etcd Pod がクラッシュループしていることを確認している。

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- etcd のバックアップを取得している。

重要

問題が発生した場合にクラスターを復元できるように、この手順を実行する前に etcd バックアップを作成しておくことは重要です。

手順

1. クラッシュループしている etcd Pod を停止します。
   a. クラッシュループしているノードをデバッグします。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc debug node/ip-10-0-131-183.ec2.internal 1
```

   1 これを正常でないノードの名前に置き換えます。

   b. ルートディレクトリーを **/host** に変更します。

```
sh-4.2# chroot /host
```

   c. 既存の etcd Pod ファイルを kubelet マニフェストディレクトリーから移動します。

```
sh-4.2# mkdir /var/lib/etcd-backup
```

```
sh-4.2# mv /etc/kubernetes/manifests/etcd-pod.yaml /var/lib/etcd-backup/
```

   d. etcd データディレクトリーを別の場所に移動します。

```
sh-4.2# mv /var/lib/etcd/ /tmp
```

   これでノードシェルを終了できます。

2. 正常でないメンバーを削除します。
   a. 影響を受けるノード上に **ない** Pod を選択します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

   出力例

```
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. 実行中の etcd コンテナーに接続し、影響を受けるノードにない Pod の名前を渡します。

クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

c. メンバーの一覧を確認します。

```
sh-4.2# etcdctl member list -w table
```

出力例

```
+------------------+---------+----------------------------+--------------------------+----------------
-----------+
|        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 |
+------------------+---------+----------------------------+--------------------------+----------------
-----------+
```

これらの値はこの手順で後ほど必要となるため、ID および正常でない etcd メンバーの名前を書き留めておきます。

d. ID を **etcdctl member remove** コマンドに指定して、正常でない etcd メンバーを削除します。

```
sh-4.2# etcdctl member remove 62bcf33650a7170a
```

出力例

```
Member 62bcf33650a7170a removed from cluster ead669ce1fbfb346
```

e. メンバーの一覧を再度表示し、メンバーが削除されたことを確認します。

```
sh-4.2# etcdctl member list -w table
```

出力例

```
+------------------+---------+----------------------------+--------------------------+----------------
-----------+
|        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 |
+-----------------+---------+----------------------------+-------------------------+---------------
-----------+
```

これでノードシェルを終了できます。

3. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

4. 削除された正常でない etcd メンバーの古いシークレットを削除します。

   a. 削除された正常でない etcd メンバーのシークレットを一覧表示します。

```
$ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
```

   1 この手順で先ほど書き留めた正常でない etcd メンバーの名前を渡します。

   以下の出力に示されるように、ピア、サービング、およびメトリクスシークレットがあります。

   出力例

```
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. 削除された正常でない etcd メンバーのシークレットを削除します。

      i. ピアシークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
```

      ii. 提供シークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
```

      iii. メトリクスシークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-
183.ec2.internal
```

5. etcd の再デプロイメントを強制的に実行します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "single-master-recovery-
'"$( date --rfc-3339=ns )"'"}}' --type=merge 1
```

-

1 **forceRedeploymentReason** 値は一意である必要があります。そのため、タイムスタンプが付加されます。

etcd クラスター Operator が再デプロイを実行する場合、すべてのコントロールプレーンノードで etcd Pod が機能していることを確認します。

6. 次のコマンドを入力して、クォーラムガードをオンに戻します。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

7. 次のコマンドを入力して、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

8. 単一ノードの OpenShift を使用している場合は、ノードを再起動します。そうしないと、etcd クラスター Operator で次のエラーが発生する可能性があります。

出力例

```
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]
```

検証

- 新しいメンバーが利用可能で、正常な状態にあることを確認します。

  a. 再度実行中の etcd コンテナーに接続します。
  クラスターにアクセスできるターミナルで、cluster-admin ユーザーとして以下のコマンドを実行します。

  ```
  $ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
  ```

  b. すべてのメンバーが正常であることを確認します。

  ```
  sh-4.2# etcdctl endpoint health
  ```

  出力例

  ```
  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
  ```

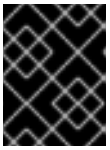
5.2.4.3. マシンが実行されていないか、ノードが準備状態にない場合の正常でないベアメタル etcd メンバーの置き換え

以下の手順では、マシンが実行されていないか、ノードが準備状態にない場合の正常でない ベアメタル etcd メンバーを置き換える手順を説明します。

インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。それ以外の場合は、最初に作成したときと同じ方法で、新しいコントロールプレーンノードを作成する必要があります。

前提条件

- 正常でないベアメタル etcd メンバーを特定している。
- マシンが実行されていないか、ノードが準備状態にないことを確認している。
- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- etcd のバックアップを取得している。



重要

問題が発生した場合にクラスターを復元できるように、この手順を実行する前に etcd バックアップを作成しておく。

手順

1. 正常でないメンバーを確認し、削除します。
   a. 影響を受けるノード上に **ない** Pod を選択します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd -o wide
```

   出力例

```
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. 実行中の etcd コンテナーに接続し、影響を受けるノードにない Pod の名前を渡します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

   c. メンバーの一覧を確認します。

```
sh-4.2# etcdctl member list -w table
```

出力例

```
+-----------------+---------+-------------------+-------------------------+-------------------------
-+--------------------+
| 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 |
+-----------------+---------+-------------------+-------------------------+-------------------------
-+--------------------+
```

これらの値はこの手順で後ほど必要となるため、ID および正常でない etcd メンバーの名前を書き留めておきます。**etcdctl endpoint health** コマンドは、置き換えの手順が完了し、新規メンバーが追加されるまで、削除されたメンバーを一覧表示します。

d. ID を **etcdctl member remove** コマンドに指定して、正常でない etcd メンバーを削除します。

警告

必ず適切な etcd メンバーを削除します。適切な etcd メンバーを削除すると、クォーラム (定足数) が失われる可能性があります。

```
sh-4.2# etcdctl member remove 7a8197040a5126c8
```

出力例

```
Member 7a8197040a5126c8 removed from cluster b23536c33f2cdd1b
```

e. メンバーの一覧を再度表示し、メンバーが削除されたことを確認します。

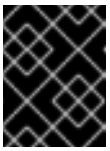
```
sh-4.2# etcdctl member list -w table
```

出力例

```
+-----------------+---------+-------------------+-------------------------+-------------------------
-+------------------------+
| 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 |
+-----------------+---------+-------------------+-------------------------+-------------------------
-+-------------------------+
```

これでノードシェルを終了できます。



重要

メンバーを削除した後、残りの etcd インスタンスが再起動している間、クラスターに短時間アクセスできない場合があります。

2. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

3. 以下のコマンドを実行して、削除された正常でない etcd メンバーの古いシークレットを削除します。

   a. 削除された正常でない etcd メンバーのシークレットを一覧表示します。

   ```
   $ oc get secrets -n openshift-etcd | grep openshift-control-plane-2
   ```

   この手順で先ほど書き留めた正常でない etcd メンバーの名前を渡します。

   以下の出力に示されるように、ピア、サービング、およびメトリクスシークレットがあります。

   ```
   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. 削除された正常でない etcd メンバーのシークレットを削除します。

      i. ピアシークレットを削除します。

      ```
      $ oc delete secret etcd-peer-openshift-control-plane-2 -n openshift-etcd

      secret "etcd-peer-openshift-control-plane-2" deleted
      ```

      ii. 提供シークレットを削除します。

      ```
      $ oc delete secret etcd-serving-metrics-openshift-control-plane-2 -n openshift-etcd

      secret "etcd-serving-metrics-openshift-control-plane-2" deleted
      ```

      iii. メトリクスシークレットを削除します。

```
$ oc delete secret etcd-serving-openshift-control-plane-2 -n openshift-etcd

secret "etcd-serving-openshift-control-plane-2" deleted
```

4. コントロールプレーンマシンを削除します。
インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。それ以外の場合は、最初に作成したときと同じ方法で、新しいコントロールプレーンノードを作成する必要があります。

   a. 正常でないメンバーのマシンを取得します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
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) これは正常でないノードのコントロールプレーンマシンです (**examplecluster-control-plane-2**)。

   b. マシン設定をファイルシステムのファイルに保存します。

```
$ oc get machine examplecluster-control-plane-2 \ (1)
  -n openshift-machine-api \
  -o yaml \
  > new-master-machine.yaml
```

(1) 正常でないノードのコントロールプレーンマシンの名前を指定します。

   c. 直前の手順で作成された **new-master-machine.yaml** ファイルを編集し、新しい名前を割り当て、不要なフィールドを削除します。

      i. **status** セクション全体を削除します。

```
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. **metadata.name** フィールドを新規の名前に変更します。
古いマシンと同じベース名を維持し、最後の番号を次に利用可能な番号に変更することが推奨されます。この例では、**examplecluster-control-plane-2** が **examplecluster-control-plane-3** に変更されています。

   以下に例を示します。

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: examplecluster-control-plane-3
  ...
```

   a. **spec.providerID** フィールドを削除します。

```
    providerID: baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-61d8-410f-be5b-6a395b056135
```

   b. **metadata.annotations** および **metadata.generation** フィールドを削除します。

```
    annotations:
      machine.openshift.io/instance-state: externally provisioned
    ...
    generation: 2
```

c. **spec.conditions**、**spec.lastUpdated**、**spec.nodeRef**、および **spec.phase** フィールドを削除します。

```
  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. 以下のコマンドを実行して、Bare Metal Operator が利用可能であることを確認します。

```
$ oc get clusteroperator baremetal
```

出力例

```
NAME      VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
baremetal   4.10.x    True        False         False      3d15h
```

7. 次のコマンドを実行して、古い **BareMetalHost** オブジェクトを削除します。

```
$ oc delete bmh openshift-control-plane-2 -n openshift-machine-api
```

出力例

```
baremetalhost.metal3.io "openshift-control-plane-2" deleted
```

8. 次のコマンドを実行して、異常なメンバーのマシンを削除します。

```
$ oc delete machine -n openshift-machine-api examplecluster-control-plane-2
```

BareMetalHost および **Machine** オブジェクトを削除すると、**Machine** コントローラーにより **Node** オブジェクトが自動的に削除されます。

何らかの理由でマシンの削除が遅れたり、コマンドが妨げられて遅れたりする場合は、マシンオブジェクトのファイナライザーフィールドを削除することで強制的に削除できます。

重要

Ctrl+c を押してマシンの削除を中断しないでください。コマンドが完了するまで続行できるようにする必要があります。新しいターミナルウィンドウを開き、ファイナライザーフィールドを編集して削除します。

a. 次のコマンドを実行して、マシン設定を編集します。

```
$ oc edit machine -n openshift-machine-api examplecluster-control-plane-2
```

b. **Machine** カスタムリソースの次のフィールドを削除し、更新されたファイルを保存します。

```
finalizers:
- machine.machine.openshift.io
```

出力例

```
machine.machine.openshift.io/examplecluster-control-plane-2 edited
```

9. 以下のコマンドを実行して、マシンが削除されていることを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
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. 次のコマンドを実行して、ノードが削除されたことを確認します。

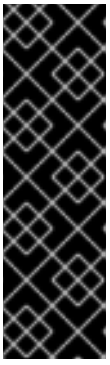
```
$ oc get nodes

NAME                 STATUS ROLES    AGE    VERSION
openshift-control-plane-0 Ready master 3h24m v1.24.0+9546431
openshift-control-plane-1 Ready master 3h24m v1.24.0+9546431
openshift-compute-0       Ready worker 176m v1.24.0+9546431
openshift-compute-1       Ready worker 176m v1.24.0+9546431
```

11. 新しい **BareMetalHost** オブジェクトとシークレットを作成して BMC 認証情報を保存します。

```
$ 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/sda
  userData:
    name: master-user-data-managed
    namespace: openshift-machine-api
EOF
```

注記

ユーザー名とパスワードは、他のベアメタルホストのシークレットで確認できます。**bmc:address** で使用するプロトコルは、他の bmh オブジェクトから取得できます。



重要

既存のコントロールプレーンホストから **BareMetalHost** オブジェクト定義を再利用する場合は、**externallyProvisioned** フィールドを **true** に設定したままにしないでください。

既存のコントロールプレーン **BareMetalHost** オブジェクトが、OpenShift Container Platform インストールプログラムによってプロビジョニングされた場合には、**externallyProvisioned** フラグが **true** に設定されている可能性があります。

検査が完了すると、**BareMetalHost** オブジェクトが作成され、プロビジョニングできるようになります。

12. 利用可能な **BareMetalHost** オブジェクトを使用して作成プロセスを確認します。

```
$ 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. **new-master-machine.yaml** ファイルを使用して新規コントロールプレーンマシンを作成します。

```
$ oc apply -f new-master-machine.yaml
```

b. 新規マシンが作成されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
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) 新規マシン **clustername-8qw5l-master-3** が作成され、**Provisioning** から **Running** にフェーズが変更されると準備状態になります。

新規マシンが作成されるまでに数分の時間がかかる場合があります。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

c. 以下のコマンドを実行して、ベアメタルホストがプロビジョニングされ、エラーが報告されていないことを確認します。

```
$ oc get bmh -n openshift-machine-api
```

出力例

```
$ 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. 以下のコマンドを実行して、新規ノードが追加され、Ready の状態であることを確認します。

```
$ oc get nodes
```

出力例

```
$ oc get nodes
NAME                 STATUS ROLES   AGE   VERSION
openshift-control-plane-0 Ready master 4h26m v1.24.0+9546431
openshift-control-plane-1 Ready master 4h26m v1.24.0+9546431
openshift-control-plane-2 Ready master 12m   v1.24.0+9546431
openshift-compute-0       Ready worker 3h58m v1.24.0+9546431
openshift-compute-1       Ready worker 3h58m v1.24.0+9546431
```

13. 次のコマンドを入力して、クォーラムガードをオンに戻します。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

14. 次のコマンドを入力して、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

15. 単一ノードの OpenShift を使用している場合は、ノードを再起動します。そうしないと、etcd クラスター Operator で次のエラーが発生する可能性があります。

出力例

```
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]
```

検証

1. すべての etcd Pod が適切に実行されていることを確認します。

クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd -o wide
```

出力例

```
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
```

直前のコマンドの出力に 2 つの Pod のみが一覧表示される場合、etcd の再デプロイメントを手動で強制できます。クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-"$( date --rfc-3339=ns )""}}' --type=merge 1
```

1 **forceRedeploymentReason** 値は一意である必要があります。そのため、タイムスタンプが付加されます。

etcd メンバーがちょうど 3 つあることを確認するには、実行中の etcd コンテナーに接続し、影響を受けたノード上になかった Pod の名前を渡します。クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

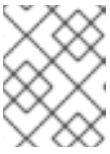
```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

2. メンバーの一覧を確認します。

```
sh-4.2# etcdctl member list -w table
```

出力例

```
+------------------+---------+------------------+--------------------------+--------------------------+---
--------------+
|        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 |
+------------------+---------+------------------+--------------------------+--------------------------+---
--------------+
```



注記

直前のコマンドの出力に 4 つ以上の etcd メンバーが表示される場合、不要なメンバーを慎重に削除する必要があります。

3. 以下のコマンドを実行して、すべての etcd メンバーが正常であることを確認します。

```
# etcdctl endpoint health --cluster
```

出力例

```
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. 以下のコマンドを実行して、すべてのノードが最新のリビジョンであることを確認します。

```
$ oc get etcd -o=jsonpath='{range.items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

```
AllNodesAtLatestRevision
```

5.3. 障害復旧

5.3.1. 障害復旧について

この障害復旧ドキュメントでは、OpenShift Container Platform クラスターで発生する可能性のある複数の障害のある状態からの復旧方法についての管理者向けの情報を提供しています。管理者は、クラスターの状態を機能する状態に戻すために、以下の 1 つまたは複数の手順を実行する必要がある場合があります。

重要

障害復旧には、少なくとも 1 つの正常なコントロールプレーンホストが必要です。

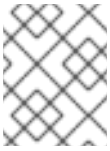
クラスターの直前の状態への復元

このソリューションは、管理者が重要なものを削除した場合など、クラスターを直前の状態に復元する必要がある状態に対応します。これには、大多数のコントロールプレーンホストが失われたために etcd クォーラム (定足数) が失われ、クラスターがオフラインになる状態も含まれます。etcd バックアップを取得している限り、以下の手順に従ってクラスターを直前の状態に復元できます。該当する場合は、コントロールプレーン証明書の期限切れの状態からのリカバリーが必要になる場合もあります。

警告

クラスターの直前の状態への復元は、実行中のクラスターで行う破壊的で、不安定なアクションです。この手順は、最後の手段としてのみ使用してください。

復元の実行前に、クラスターへの影響の詳細についてクラスターの復元を参照してください。



注記

大多数のマスターが依然として利用可能であり、etcd のクォーラムがある場合は、手順に従って単一の正常でない etcd メンバーの置き換えを実行します。

コントロールプレーン証明書の期限切れの状態からのリカバリー

このソリューションは、コントロールプレーン証明書の期限が切れた状態に対応します。たとえば、インストールの 24 時間後に行われる最初の証明書のローテーション前にクラスターをシャットダウンする場合、証明書はローテーションされず、期限切れになります。以下の手順に従って、コントロールプレーン証明書の期限切れの状態からのリカバリーを実行できます。

5.3.2. クラスターの直前の状態への復元

クラスターを直前の状態に復元するには、スナップショットを作成して、事前に etcd データのバックアップを行っている必要があります。このスナップショットを使用して、クラスターの状態を復元します。

5.3.2.1. クラスターの状態の復元について

etcd バックアップを使用して、クラスターを直前の状態に復元できます。これは、以下の状況から回復するために使用できます。

- クラスターは、大多数のコントロールプレーンホストを失いました (クォーラムの喪失)。
- 管理者が重要なものを削除し、クラスターを復旧するために復元する必要があります。

警告

クラスターの直前の状態への復元は、実行中のクラスターで行う破壊的で、不安定なアクションです。これは、最後の手段としてのみ使用してください。

Kubernetes API サーバーを使用してデータを取得できる場合は、etcd が利用できるため、etcd バックアップを使用して復元することはできません。

etcd を効果的に復元すると、クラスターが時間内に元に戻され、すべてのクライアントは競合する並列履歴が発生します。これは、kubelet、Kubernetes コントローラーマネージャー、SDN コントローラー、永続ボリュームコントローラーなどのコンポーネントを監視する動作に影響を与える可能性があります。

etcd のコンテンツがディスク上の実際のコンテンツと一致しないと、Operator チャーンが発生し、ディスク上のファイルが etcd のコンテンツと競合すると、Kubernetes API サーバー、Kubernetes コントローラーマネージャー、Kubernetes スケジューラーなどの Operator が停止する場合があります。この場合は、問題の解決に手動のアクションが必要になる場合があります。

極端な場合、クラスターは永続ボリュームを追跡できなくなり、存在しなくなった重要なワークロードを削除し、マシンのイメージを再作成し、期限切れの証明書を使用して CA バンドルを書き換えることができます。

5.3.2.2. クラスターの直前の状態への復元

保存された etcd のバックアップを使用して、クラスターの以前の状態を復元したり、大多数のコントロールプレーンホストが失われたクラスターを復元したりできます。

重要

クラスターを復元する際に、同じ z-stream リリースから取得した etcd バックアップを使用する必要があります。たとえば、OpenShift Container Platform 4.7.2 クラスターは、4.7.2 から取得した etcd バックアップを使用する必要があります。

前提条件

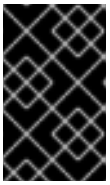
- インストール時に使用したものと同様、証明書ベースの **kubeconfig** ファイルを介して、**cluster-admin** ロールを持つユーザーとしてクラスターにアクセスします。
- リカバリーホストとして使用する正常なコントロールプレーンホストがあること。
- コントロールプレーンホストへの SSH アクセス。
- etcd スナップショットと静的 Pod のリソースの両方を含むバックアップディレクトリー (同じバックアップから取られるもの)。ディレクトリー内のファイル名は、**snapshot_<datetimestamp>.db** および **static_kuberesources_<datetimestamp>.tar.gz** の形式にする必要があります。

重要

非復元コントロールプレーンノードの場合は、SSH 接続を確立したり、静的 Pod を停止したりする必要はありません。他のリカバリー以外のコントロールプレーンマシンを 1 つずつ削除し、再作成します。

手順

1. リカバリーホストとして使用するコントロールプレーンホストを選択します。これは、復元操作を実行するホストです。
2. リカバリーホストを含む、各コントロールプレーンノードへの SSH 接続を確立します。Kubernetes API サーバーは復元プロセスの開始後にアクセスできなくなるため、コントロールプレーンノードにはアクセスできません。このため、別のターミナルで各コントロールプレーンホストに SSH 接続を確立することが推奨されます。



重要

この手順を完了しないと、復元手順を完了するためにコントロールプレーンホストにアクセスすることができなくなり、この状態からクラスターを回復できなくなります。

3. etcd バックアップディレクトリーをリカバリーコントロールプレーンホストにコピーします。この手順では、etcd スナップショットおよび静的 Pod のリソースを含む **backup** ディレクトリーを、リカバリーコントロールプレーンホストの /**home**/**core**/ ディレクトリーにコピーしていることを前提としています。
4. 他のすべてのコントロールプレーンノードで静的 Pod を停止します。

注記

リカバリーホストで静的 Pod を停止する必要はありません。

a. リカバリーホストではないコントロールプレーンホストにアクセスします。

b. 既存の etcd Pod ファイルを kubelet マニフェストディレクトリーから移動します。

```
$ sudo mv /etc/kubernetes/manifests/etcd-pod.yaml /tmp
```

c. etcd Pod が停止していることを確認します。

```
$ sudo crictl ps | grep etcd | grep -v operator
```

コマンドの出力は空であるはずです。空でない場合は、数分待機してから再度確認します。

d. 既存の Kubernetes API サーバー Pod ファイルを kubelet マニフェストディレクトリーから移動します。

```
$ sudo mv /etc/kubernetes/manifests/kube-apiserver-pod.yaml /tmp
```

e. Kubernetes API サーバー Pod が停止していることを確認します。

```
$ sudo crictl ps | grep kube-apiserver | grep -v operator
```

コマンドの出力は空であるはずです。空でない場合は、数分待機してから再度確認します。

f. etcd データディレクトリーを別の場所に移動します。

```
$ sudo mv /var/lib/etcd/ /tmp
```

g. リカバリーホストではない他のコントロールプレーンホストでこの手順を繰り返します。

5. リカバリーコントロールプレーンホストにアクセスします。

6. クラスター全体のプロキシーが有効になっている場合は、 **NO_PROXY**、**HTTP_PROXY**、および **HTTPS_PROXY** 環境変数をエクスポートしていることを確認します。

ヒント

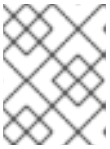
oc get proxy cluster -o yaml の出力を確認して、プロキシーが有効にされているかどうかを確認できます。プロキシーは、**httpProxy**、**httpsProxy**、および **noProxy** フィールドに値が設定されている場合に有効にされます。

7. リカバリーコントロールプレーンホストで復元スクリプトを実行し、パスを etcd バックアップディレクトリーに渡します。

```
$ sudo -E /usr/local/bin/cluster-restore.sh /home/core/backup
```

スクリプトの出力例

```
...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
```



注記

最後の etcd バックアップの後にノード証明書が更新された場合、復元プロセスによってノードが **NotReady** 状態になる可能性があります。

8. ノードをチェックして、**Ready** 状態であることを確認します。

   a. 以下のコマンドを実行します。

   ```
   $ oc get nodes -w
   ```

   出力例

   ```
   NAME              STATUS  ROLES         AGE    VERSION
   host-172-25-75-28   Ready   master        3d20h   v1.23.3+e419edf
   host-172-25-75-38   Ready   infra,worker   3d20h   v1.23.3+e419edf
   host-172-25-75-40   Ready   master        3d20h   v1.23.3+e419edf
   host-172-25-75-65   Ready   master        3d20h   v1.23.3+e419edf
   host-172-25-75-74   Ready   infra,worker   3d20h   v1.23.3+e419edf
   host-172-25-75-79   Ready   worker        3d20h   v1.23.3+e419edf
   host-172-25-75-86   Ready   worker        3d20h   v1.23.3+e419edf
   host-172-25-75-98   Ready   infra,worker   3d20h   v1.23.3+e419edf
   ```

   すべてのノードが状態を報告するのに数分かかる場合があります。

   b. **NotReady** 状態のノードがある場合は、ノードにログインし、各ノードの **/var/lib/kubelet/pki** ディレクトリーからすべての PEM ファイルを削除します。ノードに SSH 接続するか、Web コンソールのターミナルウィンドウを使用できます。

```
$  ssh -i <ssh-key-path> core@<master-hostname>
```

サンプル pki ディレクトリー

```
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
```

9. すべてのコントロールプレーンホストで kubelet サービスを再起動します。

   a. リカバリーホストから以下のコマンドを実行します。

   ```
   $ sudo systemctl restart kubelet.service
   ```

   b. 他のすべてのコントロールプレーンホストでこの手順を繰り返します。

10. 保留中の CSR を承認します。

    a. 現在の CSR の一覧を取得します。

    ```
    $ oc get csr
    ```

    出力例

    ```
    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 保留中の kubelet サービス CSR (ユーザーがプロビジョニングしたインストール用)。

    3 4 保留中の **node-bootstrapper** CSR。

    b. CSR の詳細をレビューし、これが有効であることを確認します。

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

    1 **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

    c. それぞれの有効な **node-bootstrapper** CSR を承認します。

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

d. ユーザーによってプロビジョニングされるインストールの場合は、それぞれの有効な kubelet 提供の CSR を承認します。

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

11. 単一メンバーのコントロールプレーンが正常に起動していることを確認します。

a. リカバリーホストから etcd コンテナーが実行中であることを確認します。

```
$ sudo crictl ps | grep etcd | egrep -v "operator|etcd-guard"
```

出力例

```
3ad41b7908e32
36f86e2eeaaffe662df0d21041eb22b8198e0e58abeeae8c743c3e6e977e8009
About a minute ago   Running             etcd                                0
7c05f8af362f0
```

b. リカバリーホストから、etcd Pod が実行されていることを確認します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
NAME                                READY   STATUS      RESTARTS   AGE
etcd-ip-10-0-143-125.ec2.internal               1/1     Running     1          2m47s
```

ステータスが **Pending** の場合や出力に複数の実行中の etcd Pod が一覧表示される場合、数分待機してから再度チェックを行います。

注記

次の手順は、**OVNKubernetes** Container Network Interface (CNI) プラグインを使用している場合にのみ実行してください。

12. すべてのホストで Open Virtual Network (OVN) Kubernetes Pod を再起動します。

a. ノースバウンドデータベース (nbdb) とサウスバウンドデータベース (sbdb) を削除します。Secure Shell (SSH) を使用してリカバリーホストと残りのコントロールプレーンノードにアクセスし、次のコマンドを実行します。

```
$ sudo rm -f /var/lib/ovn/etc/*.db
```

b. 次のコマンドを実行して、すべての OVN-Kubernetes コントロールプレーン Pod を削除します。

```
$ oc delete pods -l app=ovnkube-master -n openshift-ovn-kubernetes
```

c. 次のコマンドを実行して、OVN-Kubernetes コントロールプレーン Pod が再度デプロイされ、**Running** 状態になっていることを確認します。

```
$ oc get pods -l app=ovnkube-master -n openshift-ovn-kubernetes
```

出力例

```
NAME                   READY   STATUS    RESTARTS   AGE
ovnkube-master-nb24h   4/4     Running   0          48s
```

d. 次のコマンドを実行して、すべての **ovnkube-node** Pod を削除します。

```
$ oc get pods -n openshift-ovn-kubernetes -o name | grep ovnkube-node | while read p ;
do oc delete $p -n openshift-ovn-kubernetes ; done
```

e. 次のコマンドを実行して、すべての **ovnkube-node** Pod が再度デプロイされ、**Running** 状態になっていることを確認します。

```
$ oc get  pods -n openshift-ovn-kubernetes | grep ovnkube-node
```

13. 他の非復旧のコントロールプレーンマシンを 1 つずつ削除して再作成します。マシンが再作成された後、新しいリビジョンが強制され、etcd が自動的にスケールアップします。

- ユーザーがプロビジョニングしたベアメタルインストールを使用する場合は、最初に作成したときと同じ方法を使用して、コントロールプレーンマシンを再作成できます。詳細については、ユーザーがプロビジョニングしたクラスターをベアメタルにインストールするを参照してください。

> **警告**
>
> リカバリーホストのマシンを削除し、再作成しないでください。

- インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。

> **警告**
>
> リカバリーホストのマシンを削除し、再作成しないでください。
>
> インストーラーによってプロビジョニングされたインフラストラクチャーでのベアメタルインストールの場合、コントロールプレーンマシンは再作成されません。詳細については、ベアメタルコントロールプレーンノードの交換を参照してください。

a. 失われたコントロールプレーンホストのいずれかのマシンを取得します。
クラスターにアクセスできるターミナルで、cluster-admin ユーザーとして以下のコマンドを実行します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例:

```
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) これは、失われたコントロールプレーンホストのコントロールプレーンマシンです (**ip-10-0-131-183.ec2.internal**)。

b. マシン設定をファイルシステムのファイルに保存します。

```
$ oc get machine clustername-8qw5l-master-0 \ (1)
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

(1) 失われたコントロールプレーンホストのコントロールプレーンマシンの名前を指定します。

c. 直前の手順で作成された **new-master-machine.yaml** ファイルを編集し、新しい名前を割り当て、不要なフィールドを削除します。

i. **status** セクション全体を削除します。

```
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. **metadata.name** フィールドを新規の名前に変更します。
古いマシンと同じベース名を維持し、最後の番号を次に利用可能な番号に変更することが推奨されます。この例では、**clustername-8qw5l-master-0** は **clustername-8qw5l-master-3** に変更されています。

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. **spec.providerID** フィールドを削除します。

```
providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

iv. **metadata.annotations** および **metadata.generation** フィールドを削除します。

```
annotations:
  machine.openshift.io/instance-state: running
...
generation: 2
```

v. **metadata.resourceVersion** および **metadata.uid** フィールドを削除します。

```
resourceVersion: "13291"
uid: a282eb70-40a2-4e89-8009-d05dd420d31a
```

d. 失われたコントロールプレーンホストのマシンを削除します。

```
$ oc delete machine -n openshift-machine-api clustername-8qw5l-master-0 1
```

1 失われたコントロールプレーンホストのコントロールプレーンマシンの名前を指定します。

e. マシンが削除されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例:

```
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. **new-master-machine.yaml** ファイルを使用してマシンを作成します。

```
$ oc apply -f new-master-machine.yaml
```

g. 新規マシンが作成されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例:

```
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 新規マシン **clustername-8qw5l-master-3** が作成され、**Provisioning** から **Running** にフェーズが変更されると準備状態になります。

新規マシンが作成されるまでに数分の時間がかかる場合があります。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

h. リカバリーホストではない喪失したコントロールプレーンホストで、これらのステップを繰り返します。

14. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

15. リカバリーホスト内の別のターミナルウィンドウで、次のコマンドを実行してリカバリー **kubeconfig** ファイルをエクスポートします。

```
$ export KUBECONFIG=/etc/kubernetes/static-pod-resources/kube-apiserver-
certs/secrets/node-kubeconfigs/localhost-recovery.kubeconfig
```

16. etcd の再デプロイメントを強制的に実行します。
リカバリー **kubeconfig** ファイルをエクスポートしたのと同じターミナルウィンドウで、次のコマンドを実行します。

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-
3339=ns )""}}' --type=merge 1
```

1 **forceRedeploymentReason** 値は一意である必要があります。そのため、タイムスタンプが付加されます。

etcd クラスター Operator が再デプロイメントを実行すると、初期ブートストラップのスケールアップと同様に、既存のノードが新規 Pod と共に起動します。

17. すべてのノードが最新のリビジョンに更新されていることを確認します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

etcd の **NodeInstallerProgressing** 状況条件を確認し、すべてのノードが最新のリビジョンであることを確認します。更新が正常に実行されると、この出力には **AllNodesAtLatestRevision** が表示されます。

```
AllNodesAtLatestRevision
3 nodes are at revision 7 1
```

1 この例では、最新のリビジョン番号は **7** です。

出力に **2 nodes are at revision 6; 1 nodes are at revision 7** などの複数のリビジョン番号が含まれる場合、これは更新が依然として進行中であることを意味します。数分待機した後に再試行します。

18. etcd の再デプロイ後に、コントロールプレーンの新規ロールアウトを強制的に実行します。kubelet が内部ロードバランサーを使用して API サーバーに接続されているため、Kubernetes API サーバーは他のノードに再インストールされます。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

    a. Kubernetes API サーバーの新規ロールアウトを強制的に実行します。

    ```
    $ oc patch kubeapiserver cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
    ```

    すべてのノードが最新のリビジョンに更新されていることを確認します。

    ```
    $ oc get kubeapiserver -o=jsonpath='{range .items[0].status.conditions[?(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
    ```

    NodeInstallerProgressing 状況条件を確認し、すべてのノードが最新のリビジョンであることを確認します。更新が正常に実行されると、この出力には **AllNodesAtLatestRevision** が表示されます。

    ```
    AllNodesAtLatestRevision
    3 nodes are at revision 7 (1)
    ```

    (1) この例では、最新のリビジョン番号は **7** です。

    出力に **2 nodes are at revision 6; 1 nodes are at revision 7** などの複数のリビジョン番号が含まれる場合、これは更新が依然として進行中であることを意味します。数分待機した後に再試行します。

    b. Kubernetes コントローラーマネージャーの新規ロールアウトを強制的に実行します。

    ```
    $ oc patch kubecontrollermanager cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
    ```

    すべてのノードが最新のリビジョンに更新されていることを確認します。

    ```
    $ oc get kubecontrollermanager -o=jsonpath='{range .items[0].status.conditions[?(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
    ```

    NodeInstallerProgressing 状況条件を確認し、すべてのノードが最新のリビジョンであることを確認します。更新が正常に実行されると、この出力には **AllNodesAtLatestRevision** が表示されます。

    ```
    AllNodesAtLatestRevision
    3 nodes are at revision 7 (1)
    ```

    (1) この例では、最新のリビジョン番号は **7** です。

出力に **2 nodes are at revision 6; 1 nodes are at revision 7** などの複数のリビジョン番号が含まれる場合、これは更新が依然として進行中であることを意味します。数分待機した後に再試行します。

c. Kubernetes スケジューラーの新規ロールアウトを強制的に実行します。

```
$ oc patch kubescheduler cluster -p='{"spec": {"forceRedeploymentReason": "recovery-
"'"$( date --rfc-3339=ns )"'"}}' --type=merge
```

すべてのノードが最新のリビジョンに更新されていることを確認します。

```
$ oc get kubescheduler -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

NodeInstallerProgressing 状況条件を確認し、すべてのノードが最新のリビジョンであることを確認します。更新が正常に実行されると、この出力には **AllNodesAtLatestRevision** が表示されます。

```
AllNodesAtLatestRevision
3 nodes are at revision 7 (1)
```

(1) この例では、最新のリビジョン番号は **7** です。

出力に **2 nodes are at revision 6; 1 nodes are at revision 7** などの複数のリビジョン番号が含まれる場合、これは更新が依然として進行中であることを意味します。数分待機した後に再試行します。

19. すべてのコントロールプレーンホストが起動しており、クラスターに参加していることを確認します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
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
```

復元手順の後にすべてのワークロードが通常の動作に戻るようにするには、Kubernetes API 情報を保存している各 Pod を再起動します。これには、ルーター、Operator、サードパーティーコンポーネントなどの OpenShift Container Platform コンポーネントが含まれます。

注記

前の手順が完了したら、すべてのサービスが復元された状態に戻るまで数分間待つ必要がある場合があります。たとえば、**oc login** を使用した認証は、OAuth サーバー Pod が再起動するまですぐに機能しない可能性があります。

即時認証に **system:admin kubeconfig** ファイルを使用することを検討してください。この方法は、OAuth トークンではなく SSL/TLS クライアント証明書に基づいて認証を行います。以下のコマンドを実行し、このファイルを使用して認証できます。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig
```

以下のコマンドを実行て、認証済みユーザー名を表示します。

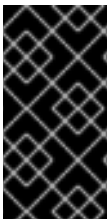
```
$ oc whoami
```

5.3.2.3. 関連情報

- ユーザーによってプロビジョニングされるクラスターのベアメタルへのインストール
- SSH を使用して OpenShift Container Platform インスタンスおよびコントロールプレーンノードにアクセスするための bastion ホストを作成する方法
- ベアメタルコントロールプレーンノードの交換

5.3.2.4. 永続ストレージの状態復元に関する問題および回避策

OpenShift Container Platform クラスターがいずれかの形式の永続ストレージを使用する場合に、クラスターの状態は通常 etcd 外に保存されます。たとえば、Pod で実行されている Elasticsearch クラスター、または **StatefulSet** オブジェクトで実行されているデータベースなどである可能性があります。etcd バックアップから復元する場合には、OpenShift Container Platform のワークロードのステータスも復元されます。ただし、etcd スナップショットが古い場合には、ステータスは無効または期限切れの可能性があります。



重要

永続ボリューム (PV) の内容は etcd スナップショットには含まれません。etcd スナップショットから OpenShift Container Platform クラスターを復元する時に、重要ではないワークロードから重要なデータにアクセスしたり、その逆ができたりする場合があります。

以下は、古いステータスを生成するシナリオ例です。

- MySQL データベースが PV オブジェクトでバックアップされる Pod で実行されている。etcd スナップショットから OpenShift Container Platform を復元すると、Pod の起動を繰り返し試行しても、ボリュームをストレージプロバイダーに戻したり、実行中の MySQL Pod が生成したりされるわけではありません。この Pod は、ストレージプロバイダーでボリュームを復元し、次に PV を編集して新規ボリュームを参照するように手動で復元する必要があります。
- Pod P1 は、ノード X に割り当てられているボリューム A を使用している。別の Pod がノード Y にある同じボリュームを使用している場合に etcd スナップショットが作成された場合に、etcd の復元が実行されると、ボリュームがノード Y に割り当てられていることが原因で Pod P1 が正常に起動できなくなる可能性があります。OpenShift Container Platform はこの割り当

てを認識せず、ボリュームが自動的に切り離されるわけではありません。これが生じる場合には、ボリュームをノード Y から手動で切り離し、ノード X に割り当ててることで Pod P1 を起動できるようにします。

- クラウドプロバイダーまたはストレージプロバイダーの認証情報が etcd スナップショットの作成後に更新された。これが原因で、プロバイダーの認証情報に依存する CSI ドライバーまたは Operator が機能しなくなります。これらのドライバーまたは Operator で必要な認証情報を手動で更新する必要がある場合があります。

- デバイスが etcd スナップショットの作成後に OpenShift Container Platform ノードから削除されたか、名前が変更された。ローカルストレージ Operator で、**/dev/disk/by-id** または **/dev** ディレクトリーから管理する各 PV のシンボリックリンクが作成されます。この状況では、ローカル PV が存在しないデバイスを参照してしまう可能性があります。
この問題を修正するには、管理者は以下を行う必要があります。

  1. デバイスが無効な PV を手動で削除します。

  2. 各ノードからシンボリックリンクを削除します。

  3. **LocalVolume** または **LocalVolumeSet** オブジェクトを削除します (**ストレージ → 永続ストレージの設定 → ローカルボリュームを使用した永続ストレージ → ローカルストレージ Operator のリソースの削除** を参照)。

5.3.3. コントロールプレーン証明書の期限切れの状態からのリカバリー

5.3.3.1. コントロールプレーン証明書の期限切れの状態からのリカバリー

クラスターはコントロールプレーン証明書の期限切れの状態から自動的に回復できます。

ただし、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。ユーザーによってプロビジョニングされるインストールの場合は、保留中の kubelet 提供の CSR を承認しないといけない場合があります。

保留中の CSR を承認するには、以下の手順に従います。

手順

1. 現在の CSR の一覧を取得します。

```
$ oc get csr
```

出力例

```
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 保留中の kubelet サービス CSR (ユーザーがプロビジョニングしたインストール用)。

3 4 保留中の **node-bootstrapper** CSR。

2. CSR の詳細をレビューし、これが有効であることを確認します。

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

1 **<csr_name>** は、現行の CSR の一覧からの CSR の名前です。

3. それぞれの有効な **node-bootstrapper** CSR を承認します。

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

4. ユーザーによってプロビジョニングされるインストールの場合は、それぞれの有効な kubelet 提供の CSR を承認します。

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