OpenShift Container Platform 4.7

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: 2023-02-24

OpenShift Container Platform 4.7 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

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. OPENSHIFT CONTAINER PLATFORM におけるバックアップおよび復元操作の概要

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 1.7 を使用して、名前空間の粒度で 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。
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - Multicloud Object Gateway
  - Noobaa や Minio などの S3 互換の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の機能は、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

バックアップ 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。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

復元 CR を作成して、アプリケーションを復元します。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。

手順

1. クラスターをシャットダウンする場合は、証明書が期限切れになる日付を決定します。

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

出力例

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

1 クラスターが正常に再起動できるようにするために、指定の日付または指定の日付の前に再起動するように計画します。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。

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
```

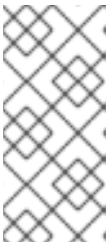
1 **-h 1** は、コントロールプレーンノードがシャットダウンされるまで、このプロセスが継続する時間 (分単位) を示します。10 ノード以上の大規模なクラスターでは、まず始めにすべてのコンピュートノードをシャットダウンする時間を確保するために、10 分以上に設定します。

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.
```

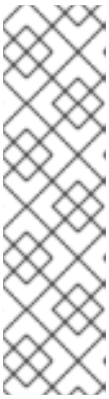
これらの方法のいずれかを使用してノードをシャットダウンすると、Pod は正常に終了するため、データが破損する可能性が低減します。



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 (別名マスターノード) 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。

関連情報

- クラスターの正常な再起動

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 分待機してから、コントロールプレーンノード (別名マスターノード) のステータスの確認を続行します。
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   v1.20.0
   ip-10-0-170-223.ec2.internal   Ready    master   75m   v1.20.0
   ip-10-0-211-16.ec2.internal    Ready    master   75m   v1.20.0
   ```
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   v1.20.0
ip-10-0-182-134.ec2.internal   Ready    worker   64m   v1.20.0
ip-10-0-250-100.ec2.internal   Ready    worker   64m   v1.20.0
```

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                             4.7.0     True        False         False      59m
cloud-credential                           4.7.0     True        False         False      85m
cluster-autoscaler                         4.7.0     True        False         False      73m
config-operator                            4.7.0     True        False         False      73m
console                                    4.7.0     True        False         False      62m
csi-snapshot-controller                    4.7.0     True        False         False      66m
dns                                        4.7.0     True        False         False      76m
etcd                                       4.7.0     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   v1.20.0
ip-10-0-170-223.ec2.internal   Ready    master   82m   v1.20.0
ip-10-0-179-95.ec2.internal    Ready    worker   70m   v1.20.0
ip-10-0-182-134.ec2.internal   Ready    worker   70m   v1.20.0
ip-10-0-211-16.ec2.internal    Ready    master   82m   v1.20.0
ip-10-0-250-100.ec2.internal   Ready    worker   69m   v1.20.0
```

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 の機能とプラグイン

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。

4.1.1. 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

クラスター内のすべてのリソースをバックアップすることも、タイプ、名前空間、またはラベルでリソースをフィルターリングすることもできます。
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。

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名前空間、PV、またはラベルでフィルターリングすることもできます。

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。

4.1.2. 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

OADP プラグイン	機能	ストレージの場所
aws	オブジェクトストアを使用して、Kubernetes オブジェクトをバックアップおよび復元します。	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
azure	オブジェクトストアを使用して、Kubernetes オブジェクトをバックアップおよび復元します。	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	オブジェクトストアを使用して、Kubernetes オブジェクトをバックアップおよび復元します。	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	オブジェクトストアを使用して、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
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。[2]	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. **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。

4.1.3. 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。

4.1.3.1. 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
```

4.1.3.2. 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
```

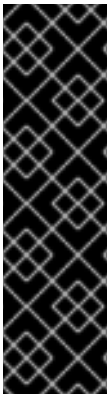
4.2. OADP のインストールおよび設定

4.2.1. 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。

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 S3 互換の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の機能は、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 Container Storage などの CSI スナップショット対応の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 を使用してアプリケーションをバックアップできます。

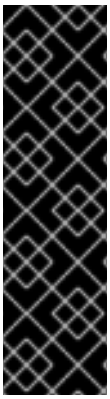
ストレージプロバイダー認証情報用の **Secret** オブジェクトを作成してから、Data Protection Application をインストールします。

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。

4.2.2. 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

OpenShift API for Data Protection (OADP) を Amazon Web Services (AWS) とともにインストールするには、OADP Operator をインストールし、AWS for Velero を設定してから、Data Protection Application をインストールします。



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の機能は、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。

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。

4.2.2.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.7 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。

4.2.2.2. 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** ユーザーに必要な権限を付与します。

```
$ 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。

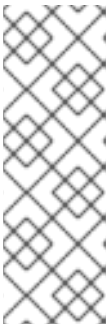
4.2.2.3. バックアップとスナップショットの場所のシークレットの作成

同じ認証情報を使用する場合は、バックアップとスナップショットの場所に **Secret** オブジェクトを作成します。

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

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。



注記

DataProtectionApplication カスタムリソース (CR) をインストールするには、**Secret** が必要です。**spec.backupLocations.credential.name** 値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。

バックアップの場所またはスナップショットの場所を指定しない場合は、空の **credentials-velero** ファイルを使用して、デフォルト名で **Secret** を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。

4.2.2.3.1. さまざまなバックアップおよびスナップショットの場所の認証情報のシークレットを設定

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"
```

4.2.2.4. Data Protection Application の設定

Velero リソース割り当てを設定し、自己署名 CA 証明書を有効にすることができます。

4.2.2.4.1. 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:
        resourceAllocations:
          limits:
            cpu: "1" 1
            memory: 512Mi 2
          requests:
            cpu: 500m 3
            memory: 256Mi 4
```

1 1 値はミリパスまたは CPU 単位で指定してください。デフォルト値は **500m** または **1** CPU 単位です。

2 デフォルト値は **512Mi** です。

3 デフォルト値は **500m** または **1** CPU 単位です。

4 デフォルト値は **256Mi** です。

4.2.2.4.2. 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. SSL/TLS セキュリティーを無効にするには、**false** にする必要があります。

4.2.2.5. 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
    restic:
      enable: true (2)
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket_name> (3)
          prefix: <prefix> (4)
        config:
          region: <region>
          profile: "default"
        credential:
          key: cloud
          name: cloud-credentials (5)
  snapshotLocations: (6)
    - name: default
      velero:
        provider: aws
        config:
          region: <region> (7)
          profile: "default"
```

(1) OpenShift Container Platform クラスターでネームスペースをバックアップおよび復元するには、**openshift** プラグインが必須です。

(2) 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。バックアップ用に Restic を設定するには、**Backup** CR に **spec.defaultVolumesToRestic: true** を追加します。

(3) 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。

4 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**)。

5 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。

6 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。

7 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/oadp-velero-sample-1-aws-registry-5d6968cbdd-d5w9k   1/1     Running   0          95s
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
service/oadp-velero-sample-1-aws-registry-svc              ClusterIP   172.30.130.230   <none>
5000/TCP   95s

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/oadp-velero-sample-1-aws-registry   1/1     1            1           96s
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/oadp-velero-sample-1-aws-registry-5d6968cbdd   1         1         1       96s
replicaset.apps/velero-588db7f655                             1         1         1       96s
```

4.2.2.5.1. 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。

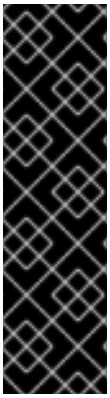
```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - csi (1)
    featureFlags:
    - EnableCSI (2)
```

(1) **csi** のデフォルトプラグインを追加します。

(2) **EnableCSI** 機能フラグを追加します。

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

OpenShift API for Data Protection (OADP) を Microsoft Azure にインストールするには、OADP Operator をインストールし、Azure for Velero を設定してから、Data Protection Application をインストールします。



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の機能は、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。

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。

4.2.3.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.7 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。

4.2.3.2. 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. **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_ACCESS_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。

4.2.3.3. バックアップとスナップショットの場所のシークレットの作成

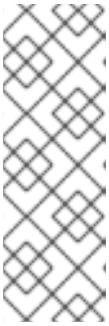
同じ認証情報を使用する場合は、バックアップとスナップショットの場所に **Secret** オブジェクトを作成します。

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

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。



注記

DataProtectionApplication カスタムリソース (CR) をインストールするには、**Secret** が必要です。**spec.backupLocations.credential.name** 値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。

バックアップの場所またはスナップショットの場所を指定しない場合は、空の **credentials-velero** ファイルを使用して、デフォルト名で **Secret** を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。

4.2.3.3.1. さまざまなバックアップおよびスナップショットの場所の認証情報のシークレットを設定

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。

- カスタム名のバックアップ場所 **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) カスタム名のバックアップ場所 **Secret**。

4.2.3.4. Data Protection Application の設定

Velero リソース割り当てを設定し、自己署名 CA 証明書を有効にすることができます。

4.2.3.4.1. 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:
        resourceAllocations:
          limits:
            cpu: "1" (1)
            memory: 512Mi (2)
          requests:
            cpu: 500m (3)
            memory: 256Mi (4)
```

1. 値はミリパスまたは CPU 単位で指定してください。デフォルト値は **500m** または **1** CPU 単位です。
2. デフォルト値は **512Mi** です。
3. デフォルト値は **500m** または **1** CPU 単位です。
4. デフォルト値は **256Mi** です。

4.2.3.4.2. 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
...
```

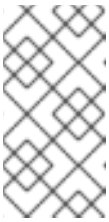
1. Base46 でエンコードされた CA 証明書文字列を指定します。
2. SSL/TLS セキュリティーを無効にするには、**false** にする必要があります。

4.2.3.5. 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)
    restic:
      enable: true (2)
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group> (3)
          storageAccount: <azure_storage_account_id> (4)
          subscriptionId: <azure_subscription_id> (5)
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: cloud-credentials-azure (6)
        provider: azure
        default: true
        objectStorage:
          bucket: <bucket_name> (7)
          prefix: <prefix> (8)
  snapshotLocations: (9)
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          subscriptionId: <azure_subscription_id>
          incremental: "true"
        name: default
        provider: azure
```

(1) OpenShift Container Platform クラスターでネームスペースをバックアップおよび復元するには、**openshift** プラグインが必須です。

(2) 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。バックアップ用に Restic を設定するには、**Backup** CR に **spec.defaultVolumesToRestic: true** を追加します。

(3) Azure リソースグループを指定します。

(4) Azure ストレージアカウント ID を指定します。

(5) Azure サブスクリプション ID 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-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。

(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

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**)。

9 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/oadp-velero-sample-1-aws-registry-5d6968cbdd-d5w9k   1/1     Running   0          95s
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
service/oadp-velero-sample-1-aws-registry-svc              ClusterIP   172.30.130.230   <none>
5000/TCP   95s

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/oadp-velero-sample-1-aws-registry   1/1     1            1           96s
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/oadp-velero-sample-1-aws-registry-5d6968cbdd   1         1         1       96s
replicaset.apps/velero-588db7f655                             1         1         1       96s
```

4.2.3.5.1. 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)
    featureFlags:
    - EnableCSI (2)
```

(1) **csi** のデフォルトプラグインを追加します。

(2) **EnableCSI** 機能フラグを追加します。

4.2.4. 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

OpenShift API for Data Protection (OADP) を Google Cloud Platform (GCP) とともにインストールするには、OADP Operator をインストールし、Velero 用に GCP を設定してから、Data Protection Application をインストールします。

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の機能は、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。

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。

4.2.4.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.7 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。

4.2.4.2. 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** ユーザーに必要な権限を付与します。

```
$ 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
)
```

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。

4.2.4.3. バックアップとスナップショットの場所のシークレットの作成

同じ認証情報を使用する場合は、バックアップとスナップショットの場所に **Secret** オブジェクトを作成します。

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

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。

4.2.4.3.1. さまざまなバックアップおよびスナップショットの場所の認証情報のシークレットを設定

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。

- カスタム名のバックアップ場所 **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 カスタム名のバックアップ場所 **Secret**。

4.2.4.4. Data Protection Application の設定

Velero リソース割り当てを設定し、自己署名 CA 証明書を有効にすることができます。

4.2.4.4.1. 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:
        resourceAllocations:
          limits:
            cpu: "1" 1
            memory: 512Mi 2
```

```
      requests:
        cpu: 500m (3)
        memory: 256Mi (4)
```

(1) 値はミリパスまたは CPU 単位で指定してください。デフォルト値は **500m** または **1** CPU 単位です。

(2) デフォルト値は **512Mi** です。

(3) デフォルト値は **500m** または **1** CPU 単位です。

(4) デフォルト値は **256Mi** です。

4.2.4.4.2. 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。

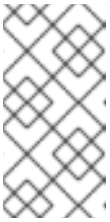
(2) SSL/TLS セキュリティーを無効にするには、**false** にする必要があります。

4.2.4.5. 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
    restic:
```

```
    enable: true 2
backupLocations:
  - velero:
      provider: gcp
      default: true
      credential:
        key: cloud
        name: cloud-credentials-gcp 3
      objectStorage:
        bucket: <bucket_name> 4
        prefix: <prefix> 5
snapshotLocations: 6
  - velero:
      provider: gcp
      default: true
      config:
        project: <project>
        snapshotLocation: us-west1 7
```

1. OpenShift Container Platform クラスターでネームスペースをバックアップおよび復元するには、**openshift** プラグインが必須です。
2. 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。バックアップ用に Restic を設定するには、**Backup** CR に **spec.defaultVolumesToRestic: true** を追加します。
3. 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。
4. 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。
5. 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**)。
6. 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。
7. 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/oadp-velero-sample-1-aws-registry-5d6968cbdd-d5w9k   1/1     Running   0          95s
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
service/oadp-velero-sample-1-aws-registry-svc              ClusterIP   172.30.130.230   <none>
5000/TCP   95s

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/oadp-velero-sample-1-aws-registry   1/1     1            1           96s
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/oadp-velero-sample-1-aws-registry-5d6968cbdd   1         1         1       96s
replicaset.apps/velero-588db7f655                              1         1         1       96s
```

4.2.4.5.1. 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)
    featureFlags:
    - EnableCSI (2)
```

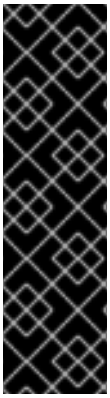
(1) **csi** のデフォルトプラグインを追加します。

(2) **EnableCSI** 機能フラグを追加します。

4.2.5. 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

OpenShift API for Data Protection (OADP) を Multicloud Object Gateway (MCG) とともにインストールするには、OADP Operator をインストールし、**Secret** オブジェクトを作成してから、Data Protection Application をインストールします。

MCG は OpenShift Container Storage (OCS) 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

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の機能は、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にネイティブスナップショット API がある場合は、スナップショットの場所を設定します。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 を使用してバックアップを作成できます。

Restic または Container Storage Interface (CSI) スナップショットの **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。

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。

4.2.5.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.7 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。

4.2.5.2. 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 Container Storage のコンポーネントです。

前提条件

- 適切な OpenShift Container Storage デプロイメントガイド を使用して OpenShift Container Storage 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。

4.2.5.3. バックアップとスナップショットの場所のシークレットの作成

同じ認証情報を使用する場合は、バックアップとスナップショットの場所に **Secret** オブジェクトを作成します。

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

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。

4.2.5.3.1. さまざまなバックアップおよびスナップショットの場所の認証情報のシークレットを設定

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。

- カスタム名のバックアップ場所 **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:
  configuration:
    velero:
      defaultPlugins:
        - aws
        - openshift
    restic:
      enable: true
  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) カスタム名のバックアップ場所 **Secret**。

4.2.5.4. Data Protection Application の設定

Velero リソース割り当てを設定し、自己署名 CA 証明書を有効にすることができます。

4.2.5.4.1. 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:
        resourceAllocations:
          limits:
            cpu: "1" (1)
            memory: 512Mi (2)
          requests:
            cpu: 500m (3)
            memory: 256Mi (4)
```

(1) 値はミリパスまたは CPU 単位で指定してください。デフォルト値は **500m** または **1** CPU 単位です。

(2) デフォルト値は **512Mi** です。

(3) デフォルト値は **500m** または **1** CPU 単位です。

4 デフォルト値は **256Mi** です。

4.2.5.4.2. 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 SSL/TLS セキュリティーを無効にするには、**false** にする必要があります。

4.2.5.5. 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)
    restic:
      enable: true (2)
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url> (3)
          insecureSkipTLSVerify: "true"
          s3ForcePathStyle: "true"
        provider: aws
        default: true
```

```
    credential:
      key: cloud
      name: cloud-credentials (4)
    objectStorage:
      bucket: <bucket_name> (5)
      prefix: <prefix> (6)
```

(1) OpenShift Container Platform クラスターでネームスペースをバックアップおよび復元するには、**openshift** プラグインが必須です。

(2) 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。バックアップ用に Restic を設定するには、**Backup** CR に **spec.defaultVolumesToRestic: true** を追加します。

(3) S3 エンドポイントの URL を指定します。

(4) 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。

(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**)。

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/oadp-velero-sample-1-aws-registry-5d6968cbdd-d5w9k   1/1     Running   0          95s
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
service/oadp-velero-sample-1-aws-registry-svc            ClusterIP   172.30.130.230   <none>
5000/TCP   95s

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/oadp-velero-sample-1-aws-registry   1/1     1            1           96s
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/oadp-velero-sample-1-aws-registry-5d6968cbdd  1         1         1       96s
replicaset.apps/velero-588db7f655                             1         1         1       96s
```

4.2.5.5.1. 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)
    featureFlags:
    - EnableCSI (2)
```

(1) **csi** のデフォルトプラグインを追加します。

(2) **EnableCSI** 機能フラグを追加します。

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

OpenShift Container Storage (OCS) 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。

Multicloud Object Gateway または S3 互換のオブジェクトストレージを、**DataProtectionApplication** カスタムリソース (CR) のバックアップの場所として設定できます。

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の機能は、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にネイティブスナップショット API がある場合は、**DataProtectionApplication** CR でスナップショットの場所としてクラウドストレージを設定できます。Restic または Container Storage Interface (CSI) スナップショットで、スナップショットの場所を指定する必要はありません。

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。

4.2.6.1. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.7 に OpenShift API for Data Protection (OADP) オペレーターをインストールします。

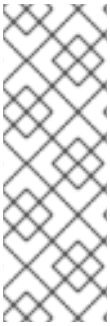
OADP オペレーターは Velero 1.7 をインストールします。

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。



注記

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 をサポートしている場合は、OADP Operator をインストールした後、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し、クラウドストレージをスナップショットの場所として設定します。

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 を使用してバックアップを作成できます。Restic はスナップショットの場所を必要としません。

4.2.6.2. バックアップとスナップショットの場所のシークレットの作成

同じ認証情報を使用する場合は、バックアップとスナップショットの場所に **Secret** オブジェクトを作成します。

バックアップストレージプロバイダーのデフォルトのプラグインを指定しない限り、**Secret** のデフォルト名は **cloud-credentials** です。

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。

4.2.6.2.1. さまざまなバックアップおよびスナップショットの場所の認証情報のシークレットを設定

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。

- カスタム名のバックアップ場所 **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:
  configuration:
    velero:
      defaultPlugins:
        - csi
        - openshift
    featureFlags:
    - EnableCSI
    restic:
      enable: true
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud
          name: <custom_secret> (1)
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
```

(1) カスタム名のバックアップ場所 **Secret**。

4.2.6.3. Data Protection Application の設定

Velero リソース割り当てを設定し、自己署名 CA 証明書を有効にすることができます。

4.2.6.3.1. 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:
        resourceAllocations:
          limits:
            cpu: "1" 1
            memory: 512Mi 2
          requests:
            cpu: 500m 3
            memory: 256Mi 4
```

1. 値はミリパスまたは CPU 単位で指定してください。デフォルト値は **500m** または **1** CPU 単位です。
2. デフォルト値は **512Mi** です。
3. デフォルト値は **500m** または **1** CPU 単位です。
4. デフォルト値は **256Mi** です。

4.2.6.3.2. 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. SSL/TLS セキュリティーを無効にするには、**false** にする必要があります。

4.2.6.4. 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:
        - gcp (1)
        - csi (2)
        - openshift (3)
    restic:
      enable: true (4)
  backupLocations:
    - velero:
        provider: gcp (5)
        default: true
        credential:
          key: cloud
          name: <default_secret> (6)
        objectStorage:
          bucket: <bucket_name> (7)
          prefix: <prefix> (8)
```

(1) 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。

(2) 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。

(3) OpenShift Container Platform クラスターでネームスペースをバックアップおよび復元するには、**openshift** プラグインが必須です。

(4) 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。バックアップ用に Restic を設定するには、**Backup** CR に **spec.defaultVolumesToRestic: true** を追加します。

(5) バックアッププロバイダーを指定します。

(6) 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** に正しいデフォルトの名前を指定する必要があります (例: **cloud-credentials-gcp**)。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。**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。

(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/oadp-velero-sample-1-aws-registry-5d6968cbdd-d5w9k   1/1     Running   0          95s
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
service/oadp-velero-sample-1-aws-registry-svc              ClusterIP   172.30.130.230   <none>
5000/TCP   95s

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/oadp-velero-sample-1-aws-registry   1/1     1            1           96s
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/oadp-velero-sample-1-aws-registry-5d6968cbdd   1         1         1       96s
replicaset.apps/velero-588db7f655                              1         1         1       96s
```

4.2.6.4.1. 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)
  featureFlags:
  - EnableCSI (2)
```

(1) **csi** のデフォルトプラグインを追加します。

(2) **EnableCSI** 機能フラグを追加します。

4.2.7. 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。

4.3. バックアップおよび復元

4.3.1. アプリケーションのバックアップ

バックアップ 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。

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 Container Storage 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。詳細は、CSI volume snapshots 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の機能は、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にネイティブスナップショット API がある場合、または Container Storage Interface (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して永続ボリュームをバックアップします。詳細については、OpenShift Container Platform のドキュメントの Overview of CSI volume snapshots 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。

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。

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。

4.3.1.1. 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. **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
  storageLocation: <velero-sample-1> 2
  ttl: 720h0m0s
```

1. バックアップする名前空間の配列を指定します。
2. **backupStorageLocations** CR の名前を指定します。

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}'
```

4.3.1.2. 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、**VolumeSnapshotClass** カスタムリソース (CR) を作成して CSI ドライバーを登録することにより、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

- 次の例のように、**VolumeSnapshotClass** CR を作成します。

  Ceph RBD

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
deletionPolicy: Retain
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true"
    snapshotter: openshift-storage.rbd.csi.ceph.com
driver: openshift-storage.rbd.csi.ceph.com
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/snapshotter-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
```

  Ceph FS

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true"
driver: openshift-storage.cephfs.csi.ceph.com
deletionPolicy: Retain
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/snapshotter-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
```

  他のクラウドプロバイダー

```
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。

4.3.1.3. 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。

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。

4.3.1.4. 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:
        - <namespace>
        includedResources:
        - pods 2
        excludedResources: []
        labelSelector: 3
          matchLabels:
            app: velero
            component: server
        pre: 4
          - exec:
              container: <container> 5
              command:
              - /bin/uname 6
              - -a
              onError: Fail 7
              timeout: 30s 8
        post: 9
...
```

1. フックが適用される名前空間の配列。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、サポートされているリソースは Pod のみです。
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. バックアップの前に実行するフックの配列。
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. フックが実行するコマンドの配列。
7. 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。
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. 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。

4.3.1.5. 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、名前空間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名前空間の配列。
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}'
```

4.3.2. 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。

復元フック を作成して、init コンテナー、アプリケーションコンテナーの起動前、またはアプリケーションコンテナー自体でコマンドを実行できます。

4.3.2.1. 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
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
```

```
    - restores.velero.io
    - resticrepositories.velero.io
    restorePVs: true
```

1 **Backup** CR の名前

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

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

3. 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 バックアップした名前空間。

4.3.2.2. 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
  - exec:
      container: <container> 4
      command:
      - /bin/bash 5
      - -c
      - "psql < /backup/backup.sql"
      waitTimeout: 5m 6
      execTimeout: 1m 7
      onError: Continue 8
```

1. オプション: フックが適用される名前空間の配列。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、サポートされているリソースは Pod のみです。
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: 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。
5. フックが実行するコマンドの配列。
6. オプション: **waitTimeout** が指定されていない場合、復元は無期限に待機します。コンテナーが開始するのを待つ時間と、コンテナー内の先行するフックが完了するのを待つ時間を指定できます。待機タイムアウトは、コンテナーが復元されたときに開始し、コンテナーがイメージをプルしてボリュームをマウントするのに時間がかかる場合があります。
7. 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。
8. 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。

4.4. 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

4.4.1. 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 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 のバージョン関係

OADP のバージョン	Velero のバージョン
0.2.6	1.6.0
0.5.5	1.7.1
1.0.0	1.7.1
1.0.1	1.7.1
1.0.2	1.7.1

OADP のバージョン	Velero のバージョン
1.0.3	1.7.1

4.4.2. 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'
```

4.4.3. 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。

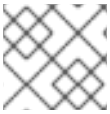
Velero CR
oc describe コマンドを使用して、**Backup** または **Restore** CR に関連する警告とエラーの要約を取得します。

```
$ 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。

4.4.4. 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
```

4.4.5. 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。

4.4.5.1. 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。

4.4.5.2. 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。

4.4.6. 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。

4.4.6.1. 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。

4.4.6.2. 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。

4.4.7. 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。

4.4.7.1. 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。

4.4.7.2. Restic バックアップの復元 CR が "PartiallyFailed"、"Failed"、または "InProgress" のままである

Restic バックアップの **Restore** CR は、**PartiallyFailed** または **Failed** ステータスで完了するか、**InProgress** のままで完了しません。

ステータスが **PartiallyFailed** または **Failed** の場合、**Velero** Pod ログにエラーメッセージ **level=error msg="unable to successfully complete restic restores of pod's volumes"** が表示されます。

ステータスが **InProgress** の場合、**Restore** CR ログは使用できず、**Restic** Pod ログにエラーは表示されません。

原因

DeploymentConfig オブジェクト が **Restore** Pod を再デプロイするため、**Restore** CR が失敗します。

解決方法

1. **ReplicationController**、**DeploymentConfig**、および **TemplateInstances** リソースを除外する **Restore** CR を作成します。

```
$ velero restore create --from-backup=<backup> -n openshift-adp \ 1
  --include-namespaces <namespace> \ 2
  --exclude-resources
replicationcontroller,deploymentconfig,templateinstances.template.openshift.io \
  --restore-volumes=true
```

1. **Backup** CR の名前を指定します。
2. **Backup** CR で **include-namespaces** を指定します。

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

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

3. **ReplicationController** および **DeploymentConfig** リソースを含む **Restore** CR を作成します。

```
$ velero restore create --from-backup=<backup> -n openshift-adp \
  --include-namespaces <namespace> \
  --include-resources replicationcontroller,deploymentconfig \
  --restore-volumes=true
```

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

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

5. 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>
```

4.4.7.3. 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

名前空間の Restic **Backup** CR を作成し、S3 バケットを空にしてから、同じ名前空間の **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 ログには、エラーメッセージ **msg="Error checking repository for stale locks"** が表示されます。

原因

オブジェクトストレージで 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 リポジトリーを作成しません。詳細については、(Velero issue 4421) を参照してください。

4.4.8. 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。

次のデータ収集オプションを使用して、**must-gather** ツールを実行できます。

- 完全な **must-gather** データ収集では、OADP Operator がインストールされているすべての名前空間について、Prometheus メトリック、Pod ログ、および Velero CR 情報が収集されます。
- 重要な **must-gather** データ収集では、Pod ログと Velero CR 情報を特定の期間 (たとえば、1 時間または 24 時間) 収集します。Prometheus メトリックと重複ログは含まれていません。
- タイムアウト付きの **must-gather** データ収集。失敗した **Backup** CR が多数ある場合は、データ収集に長い時間がかかる可能性があります。タイムアウト値を設定することでパフォーマンスを向上させることができます。
- Prometheus メトリクスデータダンプは、Prometheus によって収集されたメトリクスデータを含むアーカイブファイルをダウンロードします。

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 がインストールされている必要があります。

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。
   - Prometheus メトリックを含む、完全な **must-gather** データ収集:

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.0
     ```

     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。
   - Prometheus メトリックを使用しない、特定の期間の必須の **must-gather** データ収集:

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.0 \
       -- /usr/bin/gather_<time>_essential 1
     ```

     1 期間を時間単位で指定します。許可される値は、**1h**、**6h**、**24h**、**72h**、または **all** です。たとえば、**gather_1h_essential** または **gather_all_essential** です。
   - タイムアウト付きの **must-gather** データ収集:

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.0 \
       -- /usr/bin/gather_with_timeout <timeout> 1
     ```

     1 タイムアウト値を秒単位で指定します。
   - Prometheus メトリクスデータダンプ:

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.0 \
  -- /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
```

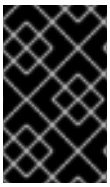
第5章 コントロールプレーンのバックアップおよび復元

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 4.7.2 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。



重要

コントロールプレーンホスト (別名マスターホスト) 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

単一コントロールプレーンホスト (別名マスターホスト) 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ホストに切り替えます。

```
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

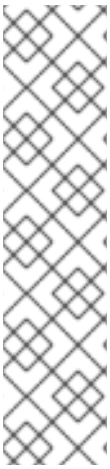
etcd 暗号化が有効にされている場合、セキュリティー上の理由から、この 2 つ目のファイルを etcd スナップショットとは別に保存することが推奨されます。ただし、このファイルは etcd スナップショットから復元するために必要になります。

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、または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、または 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

大多数のコントロールプレーンホスト (別名マスターホスト) が失われ、etcd のクォーラム (定足数) の損失が発生した場合は、この手順ではなく、直前のクラスター状態への復元 に向けた障害復旧手順を実行する必要があります。

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のクラスターへのアクセスがあること。

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
```

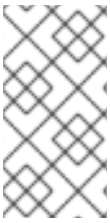
この出力例は、**ip-10-0-131-183.ec2.internal** etcd メンバーが正常ではないことを示しています。

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、または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、または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   v1.20.0 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、または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 (別名マスターノード) が **Ready** と記載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   v1.20.0
ip-10-0-164-97.ec2.internal    Ready    master   6h13m   v1.20.0
ip-10-0-154-204.ec2.internal   Ready    master   6h13m   v1.20.0
```

   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 get pods -n openshift-etcd | grep -v etcd-quorum-guard | grep 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、または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

5.2.4.1. 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場合の正常でない 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、またはノードが準備状態にない場合の正常でない 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。
- 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 get pods -n openshift-etcd | grep -v etcd-quorum-guard | grep 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。

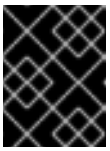
```
sh-4.2# etcdctl member list -w table
```

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. 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
```

3. コントロールプレーンマシン (別名マスターマシン) を削除し、再作成します。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
```

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. 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。

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 get pods -n openshift-etcd | grep -v etcd-quorum-guard | grep 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

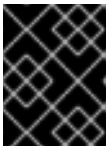
必ず適切な etcd メンバーを削除します。適切な etcd メンバーを削除すると、クォーラム (定足数) が失われる可能性があります。

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ホストに切り替えます。

```
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 get pods -n openshift-etcd | grep -v etcd-quorum-guard | grep 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. 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. 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 (別名マスターノード) に機能する etcd Pod があることを確認します。

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 --cluster
```

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
```

5.3. 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 (別名マスターホスト) 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 (別名マスターホスト) 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

- **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。

- 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 Pod を手動で停止する必要はありません。リカバリースクリプトは、リカバリーホストの Pod を停止します。

   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。

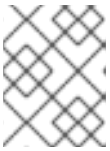
7. リカバリーコントロールプレーンホストで復元スクリプトを実行し、パスを etcd バックアップディレクトリーに渡します。

```
$ 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 | grep -v operator
```

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 get pods -n openshift-etcd | grep -v etcd-quorum-guard | grep etcd
```

注記

このコマンドを実行する前に **oc login** の実行を試行し、以下のエラーを受信すると、認証コントローラーが起動し、再試行するまでしばらく待機します。

```
Unable to connect to the server: EOF
```

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。

c. 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、このステップを繰り返します。

12. 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。これら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。

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。

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-10-0-143-125.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。

13. 別のターミナルウィンドウで、以下のコマンドを使用して **cluster-admin** ロールが割り当てられたユーザーとしてクラスターにログインします。

```
$ oc login -u <cluster_admin> 1
```

1 **<cluster_admin>** については、**cluster-admin** ロールでユーザー名を指定します。

14. 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": "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。

15. 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。

16. 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。

17. 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 get pods -n openshift-etcd | grep -v etcd-quorum-guard | grep etcd
```

出力例

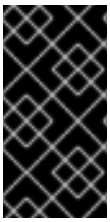
```
etcd-ip-10-0-143-125.ec2.internal        2/2     Running     0      9h
etcd-ip-10-0-154-194.ec2.internal        2/2     Running     0      9h
etcd-ip-10-0-173-171.ec2.internal        2/2     Running     0      9h
```

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。

この手順の完了後、すべてのサービスを復元するまでに数分かかる場合があります。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。

5.3.2.3. 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、または名前が変更された。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)。

関連情報

- 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については、ホストへのアクセス を参照してください。

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>
```