OpenShift Container Platform 4.16

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: 2024-07-03

OpenShift Container Platform 4.16 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 **3**
1.1. コントロールプレーンのバックアップおよび復元の操作 3
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 3

第2章 クラスターの正常なシャットダウン **6**
2.1. 前提条件 6
2.2. クラスターのシャットダウン 6
2.3. 関連情報 8

第3章 クラスターの正常な再起動 **9**
3.1. 前提条件 9
3.2. クラスターの再起動 9

第4章 OADP アプリケーションのバックアップと復元 **12**
4.1. OPENSHIFT API FOR DATA PROTECTION の概要 12
4.2. OADP リリースノート 12
4.3. OADP FEATURES AND PLUGINS 37
4.4. OADP のインストールおよび設定 43
4.5. OADP のアンインストール 130
4.6. OADP のバックアップ 130
4.7. OADP の復元 140
4.8. OADP と ROSA 144
4.9. OADP と AWS STS 156
4.10. OADP 1.2 DATA MOVER 169
4.11. OADP 1.3 DATA MOVER 187
4.12. トラブルシューティング 192
4.13. OADP で使用される API 222
4.14. OADP の高度な特徴と機能 228

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 **236**
5.1. ETCD のバックアップ 236
5.2. 正常でない ETCD メンバーの置き換え 247
5.3. 障害復旧 274

第1章 バックアップおよび復元

1.1. コントロールプレーンのバックアップおよび復元の操作

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

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

クラスターを再度実行する場合は、クラスターを正常に再起動します。

注記

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

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

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

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

関連情報

- マシンライフサイクルフックによるクォーラム保護

1.2. アプリケーションのバックアップおよび復元の操作

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

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

1.2.1. OADP 要件

OADP には以下の要件があります。

- **cluster-admin** ロールを持つユーザーとしてログインする必要があります。
- 次のストレージタイプのいずれかなど、バックアップを保存するためのオブジェクトストレージが必要です。
  - OpenShift Data Foundation
  - Amazon Web Services
  - Microsoft Azure
  - Google Cloud Platform
  - S3 と互換性のあるオブジェクトストレージ
  - IBM Cloud® Object Storage S3

注記

OCP 4.11 以降で CSI バックアップを使用する場合は、OADP 1.1.**x** をインストールします。

OADP 1.0.**x** は、OCP 4.11 以降での CSI バックアップをサポートしていません。OADP 1.0.**x** には Velero 1.7.**x** が含まれており、OCP 4.11 以降には存在しない API グループ **snapshot.storage.k8s.io/v1beta1** が必要です。

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 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

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

注記

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

1.2.2. アプリケーションのバックアップおよび復元

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

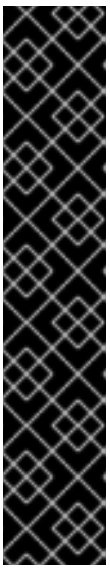
- バックアップ操作の前後にコマンドを実行するための バックアップフックの作成
- バックアップのスケジュール
- ファイルシステムバックアップを使用してアプリケーションをバックアップする: Kopia または Restic
- アプリケーションのバックアップを復元するには、**Restore** (CR) を作成します。復元 CR の作成 を参照してください。
- 復元操作中に init コンテナーまたはアプリケーションコンテナーでコマンドを実行するように 復元フック を設定できます。

第2章 クラスターの正常なシャットダウン

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

2.1. 前提条件

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



重要

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

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

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

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

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。

手順

1. クラスターを長期間シャットダウンする場合は、証明書の有効期限が切れる日付を確認し、次のコマンドを実行します。

```
$ 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. クラスター内のすべてのノードをスケジュール不可としてマークします。クラウドプロバイダーの Web コンソールから、または次のループを実行することでマークできます。

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

出力例

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

3. 次の方法を使用して Pod を退避させます。

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

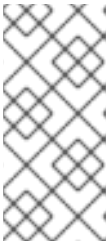
4. クラスターのすべてのノードをシャットダウンします。クラウドプロバイダーの Web コンソールから、または次のループを実行することでマークできます。

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

出力例

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

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



注記

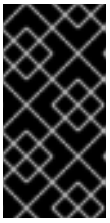
大規模なクラスターでは、シャットダウン時間が長くなるように調整します。

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

注記

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

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



重要

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

2.3. 関連情報

- クラスターの正常な再起動

第3章 クラスターの正常な再起動

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

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

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

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

3.1. 前提条件

- クラスターを正常にシャットダウンしている。

3.2. クラスターの再起動

クラスターの正常なシャットダウン後にクラスターを再起動できます。

前提条件

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

手順

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

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

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

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

   ```
   NAME                          STATUS   ROLES    AGE   VERSION
   ip-10-0-168-251.ec2.internal   Ready    master   75m   v1.29.4
   ip-10-0-170-223.ec2.internal   Ready    master   75m   v1.29.4
   ip-10-0-211-16.ec2.internal    Ready    master   75m   v1.29.4
   ```
4. 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 のリストからの 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   v1.29.4
ip-10-0-182-134.ec2.internal Ready    worker   64m   v1.29.4
ip-10-0-250-100.ec2.internal Ready    worker   64m   v1.29.4
```

6. 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 のリストからの 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                             4.16.0    True        False         False      59m
cloud-credential                           4.16.0    True        False         False      85m
cluster-autoscaler                         4.16.0    True        False         False      73m
config-operator                            4.16.0    True        False         False      73m
console                                    4.16.0    True        False         False      62m
csi-snapshot-controller                    4.16.0    True        False         False      66m
dns                                        4.16.0    True        False         False      76m
etcd                                       4.16.0    True        False         False      76m
...
```

b. 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   v1.29.4
ip-10-0-170-223.ec2.internal   Ready    master   82m   v1.29.4
ip-10-0-179-95.ec2.internal    Ready    worker   70m   v1.29.4
ip-10-0-182-134.ec2.internal   Ready    worker   70m   v1.29.4
ip-10-0-211-16.ec2.internal    Ready    master   82m   v1.29.4
ip-10-0-250-100.ec2.internal   Ready    worker   69m   v1.29.4
```

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。

8. コントロールプレーンとワーカーノードの準備ができたら、クラスター内のすべてのノードをスケジュール可能としてマークします。以下のコマンドを実行します。

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

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 OADP アプリケーションのバックアップと復元

4.1. OPENSHIFT API FOR DATA PROTECTION の概要

OpenShift API for Data Protection (OADP) 製品は、OpenShift Container Platform 上のお客様のアプリケーションを保護します。この製品は、OpenShift Container Platform のアプリケーション、アプリケーション関連のクラスターリソース、永続ボリューム、内部イメージをカバーする包括的な障害復旧保護を提供します。OADP は、コンテナー化されたアプリケーションと仮想マシン (VM) の両方をバックアップすることもできます。

ただし、OADP は etcd または OpenShift Operator の障害復旧ソリューションとしては機能しません。

4.1.1. OpenShift API for Data Protection API

OpenShift API for Data Protection (OADP) は、バックアップをカスタマイズし、不要または不適切なリソースの組み込みを防止するための複数のアプローチを可能にする API を提供します。

OADP は次の API を提供します。

- バックアップ
- 復元
- スケジュール
- BackupStorageLocation
- VolumeSnapshotLocation

関連情報

- etcd のバックアップ

4.2. OADP リリースノート

4.2.1. OADP 1.3 リリースノート

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

4.2.1.1. OADP 1.3.2 リリースノート

OpenShift API for Data Protection (OADP) 1.3.2 リリースノートには、解決された問題と既知の問題が記載されています。

4.2.1.1.1. 解決した問題

BSL に有効なカスタムシークレットが使用されている場合、DPA は調整に失敗します。

Backup Storage Location (BSL) に有効なカスタムシークレットが使用されているが、デフォルトのシークレットが見つからない場合、DPA は調整に失敗します。回避策は、最初に必要なデフォルトの **cloud-credentials** を作成することです。カスタムシークレットが再作成されると、それを使用してその存在を確認できます。

OADP-3193

CVE-2023-45290: oadp-velero-container: Golang net/http: Request.ParseMultipartForm でのメモリー不足

net/http Golang 標準ライブラリーパッケージに不具合が見つかり、OADP の以前のバージョンに影響します。**multipart** フォームを解析する場合、明示的に **Request.ParseMultipartForm** を使用するか、または暗黙的に **Request.FormValue**、**Request.PostFormValue**、または **Request.FormFile** を使用するかにかかわらず、解析されたフォームの合計サイズの制限は、単一のフォーム行の読み取り中に消費されるメモリーには適用されません。これにより、長い行を含む悪意のある入力により、任意の大量のメモリーが割り当てられ、メモリー不足につながる可能性があります。この不具合は OADP 1.3.2 で解決されました。

詳細は、CVE-2023-45290 を参照してください。

CVE-2023-45289: oadp-velero-container: Golang net/http/cookiejar: HTTP リダイレクト時の機密ヘッダーと Cookie の不適切な転送

net/http/cookiejar Golang 標準ライブラリーパッケージに不具合が見つかり、OADP の以前のバージョンに影響します。 サブドメインが一致しない、または初期ドメインと完全に一致しないドメインへの HTTP リダイレクトに従う場合、**http.Client** は **Authorization** や **Cookie** などの機密ヘッダーを転送しません。悪意を持って作成された HTTP リダイレクトにより、機密ヘッダーが予期せず転送される可能性があります。この不具合は OADP 1.3.2 で解決されました。

詳細は、CVE-2023-45289 を参照してください。

CVE-2024-24783: oadp-velero-container: Golang crypto/x509: 不明な公開鍵アルゴリズムを持つ証明書の検証パニック

crypto/x509 Golang 標準ライブラリーパッケージに不具合が見つかり、OADP の以前のバージョンに影響します。不明な公開鍵アルゴリズムを持つ証明書を含む証明書チェーンを検証すると、**Certificate.Verify** がパニックになります。これは、**Config.ClientAuth** を **VerifyClientCertIfGiven** または **RequireAndVerifyClientCert** に設定するすべての **crypto/tls** クライアントおよびサーバーに影響します。デフォルトの動作では、TLS サーバーはクライアント証明書を検証しません。この不具合は OADP 1.3.2 で解決されました。

詳細は、CVE-2024-24783 を参照してください。

CVE-2024-24784: oadp-velero-plugin-container: Golang net/mail: 表示名内のコメントが正しく処理されない

net/mail Golang 標準ライブラリーパッケージに不具合が見つかり、OADP の以前のバージョンに影響します。**ParseAddressList** 関数は、コメント、括弧内のテキスト、および表示名を正しく処理しません。これは準拠するアドレスパーサーとの不整合であるため、異なるパーサーを使用するプログラムによって異なる信頼の決定が行われる可能性があります。この不具合は OADP 1.3.2 で解決されました。

詳細は、CVE-2024-24784 を参照してください。

CVE-2024-24785: oadp-velero-container: Golang: html/template: MarshalJSON メソッドから返されるエラーにより、テンプレートのエスケープが壊れる可能性がある

html/template Golang 標準ライブラリーパッケージに不具合が見つかり、OADP の以前のバージョンに影響します。 **MarshalJSON** メソッドから返されるエラーにユーザーが制御するデータが含まれている場合、そのエラーは HTML/テンプレートパッケージのコンテキスト自動エスケープ動作の中断に使用される可能性があり、後続のアクションにより、予期しないコンテンツがテンプレートに挿入される可能性があります。 この不具合は OADP 1.3.2 で解決されました。

詳細は、CVE-2024-24785 を参照してください。

このリリースで解決されたすべての問題のリストは、Jira の OADP 1.3.2 の解決済みの問題 のリストを参照してください。

4.2.1.1.2. 既知の問題

OADP を復元した後に Cassandra アプリケーション Pod が CrashLoopBackoff ステータスになる

OADP が復元されると、Cassandra アプリケーション Pod が **CrashLoopBackoff** ステータスになる可能性があります。この問題を回避するには、OADP を復元した後、エラーまたは **CrashLoopBackOff** 状態を返す **StatefulSet** Pod を削除します。**StatefulSet** コントローラーがこれらの Pod を再作成し、正常に動作するようになります。

OADP-3767

4.2.1.2. OADP 1.3.1 リリースノート

OpenShift API for Data Protection (OADP) 1.3.1 リリースノートには、新機能と解決された問題が記載されています。

4.2.1.2.1. 新機能

OADP 1.3.0 Data Mover が完全にサポートされるようになりました

OADP 1.3.0 でテクノロジープレビューとして導入された OADP 組み込みの Data Mover が、コンテナー化されたワークロードと仮想マシンのワークロードの両方で完全にサポートされるようになりました。

4.2.1.2.2. 解決した問題

IBM Cloud (R) Object Storage がバックアップストレージプロバイダーとしてサポートされるようになりました

IBM Cloud® Object Storage は、これまでサポートされていなかった AWS S3 互換のバックアップストレージプロバイダーの 1 つです。この更新により、IBM Cloud® Object Storage が AWS S3 互換のバックアップストレージプロバイダーとしてサポートされるようになりました。

OADP-3788

OADP Operator が missing region エラーを正しく報告するようになりました

以前は、AWS Backup Storage Location (BSL) 設定で **region** を指定せずに **profile:default** を指定すると、OADP Operator が Data Protection Application (DPA) カスタムリソース (CR) での **missing region** エラーを報告できませんでした。この更新により、AWS の DPA BSL 仕様の検証が修正されました。その結果、OADP Operator が **missing region** エラーを報告するようになりました。

OADP-3044

カスタムラベルが openshift-adp namespace から削除されなくなりました

以前は、**openshift-adp-controller-manager** Pod が **openshift-adp** namespace に割り当てられたラベルをリセットしていました。これにより、Argo CD などのカスタムラベルを必要とするアプリケーションで同期の問題が発生し、機能が適切に動作しませんでした。この更新により、この問題が修正され、カスタムラベルが **openshift-adp** namespace から削除されなくなりました。

OADP-3189

OADP must-gather イメージが CRD を収集するようになりました

以前は、OADP **must-gather** イメージが、OADP によって提供されるカスタムリソース定義 (CRD) を収集しませんでした。その結果、**omg** ツールを使用してサポートシェルでデータを抽出することができませんでした。この修正により、**must-gather** イメージが OADP によって提供される CRD を収集し、**omg** ツールを使用してデータを抽出できるようになりました。

OADP-3229

ガベージコレクションのデフォルト頻度値の記述が修正されました

以前は、**garbage-collection-frequency** フィールドのデフォルト頻度値の記述が間違っていました。この更新により、**garbage-collection-frequency** の **gc-controller** 調整のデフォルト頻度値が 1 時間という正しい値になりました。

OADP-3486

FIPS モードフラグが OperatorHub で利用可能になりました

fips-compatible フラグを **true** に設定すると、OperatorHub の OADP Operator リストに FIPS モードフラグが追加されます。この機能は OADP 1.3.0 で有効になりましたが、Red Hat Container Catalog には FIPS 対応と表示されていませんでした。

OADP-3495

csiSnapshotTimeout を短い期間に設定した場合に nil ポインターによる CSI プラグインのパニックが発生しなくなりました

以前は、**csiSnapshotTimeout** パラメーターを短い期間に設定すると、CSI プラグインで **plugin panicked: runtime error: invalid memory address or nil pointer dereference** というエラーが発生していました。

この修正により、バックアップが **Timed out awaiting reconciliation of volumesnapshot** エラーで失敗するようになりました。

OADP-3069

このリリースで解決されたすべての問題のリストは、Jira の OADP 1.3.1 の解決済みの問題 のリストを参照してください。

4.2.1.2.3. 既知の問題

IBM Power (R) および IBM Z(R) プラットフォームにデプロイされたシングルノード OpenShift クラスターのバックアップおよびストレージの制限

IBM Power® および IBM Z® プラットフォームにデプロイされたシングルノード OpenShift クラスターのバックアップおよびストレージ関連の次の制限を確認してください。

Storage

現在、IBM Power® および IBM Z® プラットフォームにデプロイされたシングルノードの OpenShift クラスターと互換性があるのは、NFS ストレージのみです。

バックアップ

バックアップおよび復元操作では、**kopia** や **restic** などのファイルシステムバックアップを使用したアプリケーションのバックアップのみがサポートされます。

OADP-3787

OADP を復元した後に Cassandra アプリケーション Pod が CrashLoopBackoff ステータスになる

OADP が復元されると、Cassandra アプリケーション Pod が **CrashLoopBackoff** ステータスになる可能性があります。この問題を回避するには、OADP を復元した後、エラーまたは **CrashLoopBackoff** 状態の **StatefulSet** Pod を削除します。**StatefulSet** コントローラーがこれらの Pod を再作成し、正常に動作するようになります。

OADP-3767

4.2.1.3. OADP 1.3.0 リリースノート

OpenShift API for Data Protection (OADP) 1.3.0 のリリースノートには、新機能、解決された問題とバグ、既知の問題がリストされています。

4.2.1.3.1. 新機能

Velero ビルトイン DataMover

OADP 1.3 には、Container Storage Interface (CSI) ボリュームのスナップショットをリモートオブジェクトストアに移動するために使用できる、ビルトイン Data Mover が含まれています。ビルトイン Data Mover を使用すると、クラスターの障害、誤削除、または破損が発生した場合に、リモートオブジェクトストアからステートフルアプリケーションを復元できます。スナップショットデータを読み取り、統合リポジトリーに書き込むためのアップローダーメカニズムとして Kopia を使用します。

Velero ビルトイン DataMover は、テクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

ファイルシステムバックアップを使用してアプリケーションをバックアップする: Kopia または Restic

Velero のファイルシステムバックアップ (FSB) は、Restic パスと Kopia パスという 2 つのバックアップライブラリーをサポートしています。

Velero を使用すると、ユーザーは 2 つのパスから選択できます。

バックアップの場合は、インストール中に **uploader-type** フラグを使用してパスを指定します。有効な値は、**restic** または **kopia** です。値が指定されていない場合、このフィールドのデフォルトは **kopia** です。インストール後に選択を変更することはできません。

GCP クラウド認証

Google Cloud Platform (GCP) 認証を使用すると、有効期間の短い Google 認証情報を使用できます。

GCP と Workload Identity Federation を使用すると、Identity and Access Management (IAM) を使用して、サービスアカウントに成り代わる機能などの IAM ロールを外部アイデンティティーに付与できます。これにより、サービスアカウントキーに関連するメンテナンスとセキュリティーのリスクが排除されます。

AWS ROSA STS 認証

Red Hat OpenShift Service on AWS (ROSA) クラスターで OpenShift API for Data Protection (OADP) を使用して、アプリケーションデータをバックアップおよび復元できます。

ROSA は、幅広い AWS コンピュート、データベース、分析、機械学習、ネットワーク、モバイル、お

ROSA は、幅広い AWS コンピュート、データベース、分析、機械学習、ネットワーク、モバイル、およびその他のサービスとのシームレスな統合を提供し、差別化されたエクスペリエンスの構築とお客様への提供をさらに高速化します。

AWS アカウントから直接サービスをサブスクライブできます。

クラスターの作成後、OpenShift Web コンソールを使用してクラスターを操作できます。ROSA サービスは、OpenShift API およびコマンドラインインターフェイス (CLI) ツールも使用します。

4.2.1.3.2. 解決した問題

ACM アプリケーションが削減され、復元後にマネージドクラスター上で再作成される

マネージドクラスター上のアプリケーションは削除され、復元をアクティブ化すると再作成されていました。OpenShift API for Data Protection (OADP 1.2) のバックアップおよび復元のプロセスは、古いバージョンよりも高速です。OADP のパフォーマンスが変わったことで、ACM リソースの復元時にこのような動作が発生するようになりました。一部のリソースは他のリソースより前に復元され、その結果、マネージドクラスターからアプリケーションが削除されていました。OADP-2686

Pod セキュリティー標準が原因で Restic の復元が部分的に失敗する

相互運用性のテスト中に、OpenShift Container Platform 4.14 の Pod セキュリティーモードが **enforce** に設定されていたため、Pod が拒否されました。これは、復元順序が原因で発生します。Pod が security context constraints (SCC) リソースの前に作成されることで **podSecurity** 標準に違反していたため、Pod が拒否されました。Velero サーバーで復元優先度フィールドを設定すると、復元は成功します。OADP-2688

Velero が複数の namespace にインストールされている場合、Pod ボリュームのバックアップが失敗する可能性がある

Velero が複数の namespace にインストールされている場合、Pod Volume Backup (PVB) 機能でリグレッションが発生していました。PVB コントローラーは、その namespace 内の PVB に適切に制限されませんでした。OADP-2308

OADP Velero プラグインが "received EOF, stopping recv loop" のメッセージを返す

OADP では、Velero プラグインは別のプロセスとして開始されました。Velero 操作が完了すると、成功しても失敗しても終了します。そのため、デバッグログに **received EOF, stopping recv loop** メッセージが表示された場合、それはエラーの発生ではなく、プラグイン操作の完了を意味しました。OADP-2176

CVE-2023-39325 複数の HTTP/2 対応 Web サーバーが DDoS 攻撃 (Rapid Reset Attack) に対して脆弱である

OADP の以前のリリースでは、リクエストのキャンセルにより複数のストリームがすぐにリセットされる可能性があるため、HTTP/2 プロトコルはサービス拒否攻撃の影響を受けやすかった。サーバーは、接続ごとのアクティブなストリームの最大数に関するサーバー側の制限に達しないようにしながら、ストリームをセットアップおよび破棄する必要がありました。これにより、サーバーリソースの消費によりサービス拒否が発生しました。

詳細は、CVE-2023-39325 (Rapid Reset Attack) を参照してください。

このリリースで解決された問題の一覧は、Jira の OADP 1.3.0 解決済みの問題 に記載されているリストを参照してください。

4.2.1.3.3. 既知の問題

csiSnapshotTimeout が短く設定されている場合、nil ポインターで CSI プラグインエラーが発生する

csiSnapshotTimeout の期間が短く設定されている場合、CSI プラグインが nil ポインターでエラーを引き起こします。短時間でスナップショットを完了できることもありますが、多くの場合、バックアップ **PartiallyFailed** でパニックが発生し、エラー **plugin panicked: runtime error: invalid memory address or nil pointer dereference** が表示されます。

volumeSnapshotContent CR にエラーがある場合、バックアップは PartiallyFailed としてマークされる

いずれかの **VolumeSnapshotContent** CR に **VolumeSnapshotBeingCreated** アノテーションの削除に関連するエラーがある場合、バックアップは **WaitingForPluginOperationsPartiallyFailed** フェーズに遷移します。OADP-2871

初めて 30,000 個のリソースの復元する際に、パフォーマンスの問題が発生する

既存のリソースポリシーを使用せずに 30,000 個のリソースを初めて復元する際に、既存のリソースポリシーを **update** に設定して 2 回目と 3 回目の復元を実行する場合と比べて、2 倍の時間がかかります。OADP-3071

Datadownload 操作が関連する PV を解放する前に、復元後のフックが実行を開始する可能性がある

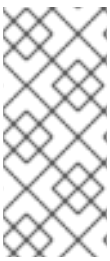
Data Mover 操作は非同期のため、関連する Pod の永続ボリューム (PV) が Data Mover の永続ボリューム要求 (PVC) によって解放される前に、ポストフックが試行される可能性があります。

GCP-Workload Identity Federation VSL バックアップで PartiallyFailed が発生する

GCP Workload Identity が GCP 上で設定されている場合、VSL バックアップで **PartiallyFailed** が発生します。

このリリースにおける既知の問題の完全なリストについては、Jira の OADP 1.3.0 の既知の問題 のリストを参照してください。

4.2.1.3.4. アップグレードの注意事項



注記

必ず次のマイナーバージョンにアップグレードしてください。バージョンは絶対に **スキップしないでください**。新しいバージョンに更新するには、一度に 1 つのチャネルのみアップグレードします。たとえば、OpenShift API for Data Protection (OADP) 1.1 から 1.3 にアップグレードする場合、まず 1.2 にアップグレードし、次に 1.3 にアップグレードします。

4.2.1.3.4.1. OADP 1.2 から 1.3 への変更点

Velero サーバーが、バージョン 1.11 から 1.12 に更新されました。

OpenShift API for Data Protection (OADP) 1.3 は、VolumeSnapshotMover (VSM) や Volsync Data Mover の代わりに Velero のビルトイン Data Mover を使用します。

これにより、以下の変更が発生します。

- **spec.features.dataMover** フィールドと VSM プラグインは OADP 1.3 と互換性がないため、それらの設定を **DataProtectionApplication** (DPA) 設定から削除する必要があります。
- Volsync Operator は Data Mover の機能には不要になったため、削除できます。

- カスタムリソース定義の **volumenapshotbackups.datamover.oadp.openshift.io** および **volumenapshotrestores.datamover.oadp.openshift.io** は不要になったため、削除できます。
- OADP-1.2 Data Mover に使用されるシークレットは不要になったため、削除できます。

OADP 1.3 は、Restic の代替ファイルシステムバックアップツールである Kopia をサポートしています。

- Kopia を使用するには、次の例に示すように、新しい **spec.configuration.nodeAgent** フィールドを使用します。

  例

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

- **spec.configuration.restic** フィールドは OADP 1.3 で非推奨となり、今後の OADP バージョンで削除される予定です。非推奨の警告が表示されないようにするには、**restic** キーとその値を削除し、次の新しい構文を使用します。

  例

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

注記

今後の OADP リリースでは、**kopia** ツールが **uploaderType** のデフォルト値になる予定です。

4.2.1.3.4.2. アップグレード手順

4.2.1.3.4.3. OADP 1.2 の Data Mover (テクノロジープレビュー) からアップグレードする

OpenShift API for Data Protection (OADP) 1.2 Data Mover のバックアップは、OADP 1.3 では **復元できません**。アプリケーションのデータ保護にギャップが生じることを防ぐには、OADP 1.3 にアップグレードする前に次の手順を実行します。

手順

1. クラスターのバックアップが十分で、Container Storage Interface (CSI) ストレージが利用可能な場合は、CSI バックアップを使用してアプリケーションをバックアップします。
2. クラスター外のバックアップが必要な場合:

a. **--default-volumes-to-fs-backup=true or backup.spec.defaultVolumesToFsBackup** オプションを使用するファイルシステムバックアップで、アプリケーションをバックアップします。

b. オブジェクトストレージプラグイン (**velero-plugin-for-aws** など) を使用してアプリケーションをバックアップします。

注記

Restic ファイルシステムバックアップのデフォルトのタイムアウト値は 1 時間です。OADP 1.3.1 以降では、Restic および Kopia のデフォルトのタイムアウト値は 4 時間です。

重要

OADP 1.2 Data Mover バックアップを復元するには、OADP をアンインストールし、OADP 1.2 をインストールして設定する必要があります。

4.2.1.3.4.4. DPA 設定をバックアップする

現在の **DataProtectionApplication** (DPA) 設定をバックアップする必要があります。

手順

- 次のコマンドを実行して、現在の DPA 設定を保存します。

  例

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

4.2.1.3.4.5. OADP Operator をアップグレードする

OpenShift API for Data Protection (OADP) Operator をアップグレードする場合は、次の手順を使用します。

手順

1. OADP Operator のサブスクリプションチャネルを、**steady-1.2** から **stable-1.3** に変更します。
2. Operator とコンテナーが更新され、再起動されるまで待機します。

関連情報

- インストール済み Operator の更新

4.2.1.3.4.6. DPA を新しいバージョンに変換する

Data Mover を使用してバックアップをクラスター外に移動する必要がある場合は、次のように **DataProtectionApplication** (DPA) マニフェストを再設定します。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。

2. **Provided APIs** セクションで、**View more** をクリックします。
3. **DataProtectionApplication** ボックスの **Create instance** をクリックします。
4. **YAML View** をクリックして、現在の DPA パラメーターを表示します。

   現在の DPA の例

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

5. DPA パラメーターを更新します。
   - DPA から、**features.dataMover** キーと値を削除します。
   - VolumeSnapshotMover (VSM) プラグインを削除します。
   - **nodeAgent** キーと値を追加します。

     更新された DPA の例

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

6. DPA が正常に調整するまで待機します。

4.2.1.3.4.7. アップグレードの検証

アップグレードを検証するには、次の手順を使用します。

手順

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

   ```
   $ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

OADP 1.3 では、**DataProtectionApplication** (DPA) 設定を作成するのではなく、バックアップごとにクラスター外へのデータ移動を開始できます。

例

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

例

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: example-backup
  namespace: openshift-adp
spec:
  snapshotMoveData: true
  includedNamespaces:
  - mysql-persistent
  storageLocation: dpa-sample-1
  ttl: 720h0m0s
# ...
```

4.2.2. OADP 1.2 リリースノート

OpenShift API for Data Protection (OADP) 1.2 のリリースノートでは、新機能と拡張機能、非推奨の機能、製品の推奨事項、既知の問題、および解決された問題について説明します。

4.2.2.1. OADP 1.2.5 リリースノート

OpenShift API for Data Protection (OADP) 1.2.5 は、Container Grade Only (CGO) のリリースであり、コンテナーのヘルスグレードを更新するためにリリースされました。OADP 1.2.4 と比較して、製品自体のコードに変更はありません。

4.2.2.1.1. 解決した問題

CVE-2023-2431: oadp-velero-plugin-for-microsoft-azure-container: seccomp プロファイル適用のバイパス

Kubernetes に不具合が見つかり、OADP の以前のバージョンに影響します。この不具合は、**seccomp** プロファイルに localhost タイプを使用し、空のプロファイルフィールドを指定する場合の不具合によって、Kubernetes がローカル認証された攻撃者にセキュリティー制限の回避を許可した場合に発生します。攻撃者は、悪用目的で作成されたリクエストを送信することで、**seccomp** プロファイルの適用を回避できます。 この不具合は OADP 1.2.5 で解決されました。

詳細は、(CVE-2023-2431) を参照してください。

CSI 復元が 'PartiallyFailed' ステータスで終了し、PVC が作成されない

CSI 復元が **PartiallyFailed** ステータスで終了しました。PVC は作成されず、Pod のステータスは **Pending** です。この問題は OADP 1.2.5 で解決されました。

(OADP-1956)

完了した Pod ボリュームで PodVolumeBackup が失敗する

OADP 1.2 の以前のバージョンでは、Restic **podvolumebackup** または Velero バックアップで使用される namespace にボリュームをマウントした完了した Pod がある場合、バックアップは正常に完了しま

せん。これは、**defaultVolumesToFsBackup** が **true** に設定されている場合に発生します。この問題は OADP 1.2.5 で解決されました。

(OADP-1870)

4.2.2.1.2. 既知の問題

データ保護アプリケーション (DPA) は、認証情報のシークレットが更新されても調整を行いません。

現在、OADP Operator は **cloud-credentials** シークレットを更新しても調整を行いません。これは、**cloud-credentials** シークレットに OADP 固有のラベルまたは所有者参照がないことが原因です。空のデータなどの誤った認証情報を使用して **cloud-credentials** シークレットを作成すると、Operator はその空のデータを使用して調整し、Backup Storage Location (BSL) とレジストリーデプロイメントを作成します。その結果、正しい認証情報で **cloud-credentials** シークレットを更新しても、OADP Operator はすぐに調整して新しい認証情報を取得することはしません。

回避策: OADP 1.3 に更新します。

(OADP-3327)

4.2.2.2. OADP 1.2.4 リリースノート

OpenShift API for Data Protection (OADP) 1.2.4 は、コンテナーグレードのみ (CGO) のリリースであり、コンテナーのヘルスグレードを更新するためにリリースされました。OADP 1.2.3 と比較して、製品自体のコードに変更はありません。

4.2.2.2.1. 解決した問題

OADP 1.2.4 には解決された問題はありません。

4.2.2.2.2. 既知の問題

OADP 1.2.4 には次の既知の問題があります。

データ保護アプリケーション (DPA) は、認証情報のシークレットが更新されても調整を行いません。

現在、OADP Operator は **cloud-credentials** シークレットを更新しても調整を行いません。これは、**cloud-credentials** シークレットに OADP 固有のラベルまたは所有者参照がないことが原因です。空のデータなどの誤った認証情報を使用して **cloud-credentials** シークレットを作成すると、Operator はその空のデータを使用して調整し、Backup Storage Location (BSL) とレジストリーデプロイメントを作成します。その結果、正しい認証情報で **cloud-credentials** シークレットを更新しても、Operator はすぐに調整して新しい認証情報を取得することはしません。

回避策: OADP 1.3 に更新します。

(OADP-3327)

4.2.2.3. OADP 1.2.3 リリースノート

4.2.2.3.1. 新機能

OpenShift API for Data Protection (OADP) 1.2.3 のリリースに新機能はありません。

4.2.2.3.2. 解決した問題

以下で強調表示された問題は、OADP 1.2.3 で解決されています。

複数の HTTP/2 対応 Web サーバーが DDoS 攻撃 (Rapid Reset Attack) に対して脆弱です

OADP 1.2 の以前のリリースでは、リクエストのキャンセルにより複数のストリームがすぐにリセットされる可能性があるため、HTTP/2 プロトコルはサービス拒否攻撃の影響を受けやすかった。サーバーは、接続ごとのアクティブなストリームの最大数に関するサーバー側の制限に達しないようにしながら、ストリームをセットアップおよび破棄する必要がありました。これにより、サーバーリソースの消費によりサービス拒否が発生しました。この CVE に関連するすべての OADP 問題のリストは、次の Jira リスト を参照してください。

詳細は、CVE-2023-39325 (Rapid Reset Attack) を参照してください。

OADP 1.2.3 のリリースで解決されたすべての問題の完全なリストについては、Jira の OADP 1.2.3 で解決された問題 のリストを参照してください。

4.2.2.3.3. 既知の問題

OADP 1.2.3 には次の既知の問題があります。

データ保護アプリケーション (DPA) は、認証情報のシークレットが更新されても調整を行いません。

現在、OADP Operator は **cloud-credentials** シークレットを更新しても調整を行いません。これは、**cloud-credentials** シークレットに OADP 固有のラベルまたは所有者参照がないことが原因です。空のデータなどの誤った認証情報を使用して **cloud-credentials** シークレットを作成すると、Operator はその空のデータを使用して調整し、Backup Storage Location (BSL) とレジストリーデプロイメントを作成します。その結果、正しい認証情報で **cloud-credentials** シークレットを更新しても、Operator はすぐに調整して新しい認証情報を取得することはしません。

回避策: OADP 1.3 に更新します。

(OADP-3327)

4.2.2.4. OADP 1.2.2 リリースノート

4.2.2.4.1. 新機能

OpenShift API for Data Protection (OADP) 1.2.2 のリリースに新機能はありません。

4.2.2.4.2. 解決した問題

以下で強調表示された問題は、OADP 1.2.2 で解決されています。

Pod セキュリティー標準が原因で Restic の復元が部分的に失敗します

OADP 1.2 の以前のリリースでは、OpenShift Container Platform 4.14 は、Restic 復元プロセス中に Pod の readiness を妨げる可能性がある Pod Security Admission (PSA) ポリシーを強制していました。

この問題は、OADP 1.2.2 と OADP 1.1.6 のリリースで解決されました。ユーザーは、これらのリリースにアップグレードすることが推奨されます。

詳細は、PSA ポリシーの変更により、OCP 4.14 で部分的に Restic 復元が失敗する を参照してください。(OADP-2094)

内部イメージを使用したアプリケーションのバックアップが、プラグインパニックエラーにより部分的に失敗します

OADP 1.2 の以前のリリースでは、内部イメージを使用したアプリケーションのバックアップが部分的に失敗し、プラグインパニックエラーが返されました。バックアップが部分的に失敗し、Velero ログに次のエラーが記録されます。

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

この問題は、OADP 1.2.2 で解決されました。(OADP-1057)

復元順序が原因で、ACM クラスターの復元が期待どおりに機能しません

OADP 1.2 の以前のリリースでは、復元順序が原因で、ACM クラスターの復元が期待どおりに機能しませんでした。ACM アプリケーションは、復元を有効にした後に削除され、マネージドクラスター上で再作成されました。(OADP-2505)

ボリュームサイズの不一致により、filesystemOverhead を使用している仮想マシンが、バックアップおよび復元時に失敗します

OADP 1.2 の以前のリリースでは、選択したストレージプロバイダー実装が原因で、アプリケーションの永続ボリューム要求 (PVC) のストレージ要求と同じ PVC のスナップショットサイズが異なると、バックアップおよび復元時に filesystemOverhead を使用する仮想マシンが失敗していました。 この問題は、OADP 1.2.2 の Data Mover で解決されました。(OADP-2144)

OADP には、VolSync レプリケーションソースのプルーニング間隔を設定するオプションが含まれていませんでした

OADP 1.2 の以前のリリースでは、VolSync レプリケーションソースの **pruneInterval** を設定するオプションがありませんでした。(OADP-2052)

Velero が複数の namespace にインストールされている場合、Pod ボリュームのバックアップが失敗する可能性がありました

OADP 1.2 の以前のリリースでは、Velero が複数の namespace にインストールされている場合、Pod ボリュームのバックアップが失敗する可能性がありました。(OADP-2409)

VSL がカスタムシークレットを使用する場合、Backup Storage Locations が使用不可フェーズに遷移しました

OADP 1.2 の以前のリリースでは、Volume Snapshot Location がカスタムシークレットを使用した場合、Backup Storage Locations が使用不可フェーズに遷移しました。(OADP-1737)

OADP 1.2.2 のリリースで解決されたすべての問題の完全なリストについては、Jira の OADP 1.2.2 で解決された問題 のリストを参照してください。

4.2.2.4.3. 既知の問題

以下で強調表示されている問題は、OADP 1.2.2 のリリースにおける既知の問題です。

Must-gather コマンドが ClusterRoleBinding リソースの削除に失敗します

oc adm must-gather コマンドは、アドミッション Webhook が原因で、クラスター上に残っている **ClusterRoleBinding** リソースの削除に失敗します。したがって、**ClusterRoleBinding** リソースの削除要求は拒否されます。(OADP-27730)

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

このリリースにおける既知の問題の完全なリストについては、Jira の OADP 1.2.2 の既知の問題 のリストを参照してください。

4.2.2.5. OADP 1.2.1 リリースノート

4.2.2.5.1. 新機能

OpenShift API for Data Protection (OADP) 1.2.1 のリリースには新機能はありません。

4.2.2.5.2. 解決した問題

OADP 1.2.1 のリリースで解決されたすべての問題の完全なリストについては、Jira の OADP 1.2.1 で解決された問題 のリストを参照してください。

4.2.2.5.3. 既知の問題

OADP 1.2.1 のリリースでは、次の問題が既知の問題として強調表示されています。

DataMover Restic の保持ポリシーとプルーニングポリシーが期待どおりに機能しない

VolSync と Restic によって提供される保持機能とプルーニング機能が期待どおりに動作しません。VolSync レプリケーションにはプルーニング間隔を設定する有効なオプションがないため、OADP の外部の S3 ストレージにリモートで保存されたバックアップを管理し、プルーニングする必要があります。詳細は、以下を参照してください。

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

重要

OADP Data Mover はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

このリリースのすべての既知の問題の完全なリストについては、Jira の OADP 1.2.1 の既知の問題 のリストを参照してください。

4.2.2.6. OADP 1.2.0 リリースノート

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

4.2.2.6.1. 新機能

リソースタイムアウト

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

AWS S3 互換のバックアップストレージプロバイダー

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

4.2.2.6.1.1. テクニカルプレビュー機能

Data Mover

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

重要

OADP Data Mover はテクノロジープレビューのみの機能です。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

4.2.2.6.2. 解決した問題

このリリースで解決された問題の一覧は、Jira の OADP 1.2.0 解決済みの問題 に記載されているリストを参照してください。

4.2.2.6.3. 既知の問題

以下で強調表示されている問題は、OADP 1.2.0 のリリースにおける既知の問題です。

複数の HTTP/2 対応 Web サーバーが DDoS 攻撃 (Rapid Reset Attack) に対して脆弱です

HTTP/2 プロトコルは、リクエストのキャンセルによって複数のストリームがすぐにリセットされる可能性があるため、サービス拒否攻撃の影響を受けやすくなっています。サーバーは、接続ごとのアクティブなストリームの最大数に関するサーバー側の制限に達しないようにしながら、ストリームをセットアップおよび破棄する必要があります。これにより、サーバーのリソースが消費され、サービス拒否が発生します。

この問題を解決する OADP 1.2.3 にアップグレードすることを推奨します。

詳細は、CVE-2023-39325 (Rapid Reset Attack) を参照してください。

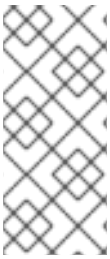
生成されたルートのホスト名を変更すると、不正なホスト名が作成される場合があります。

デフォルトでは、OpenShift Container Platform クラスターは、**openshift.io/host.generated: true** アノテーションがオンになっていることを確認し、生成されたルートと生成されていないルートの両方のフィールドに値を入力します。

.spec.host フィールドの値を、生成されたルートと生成されていないルートのクラスターのベースドメイン名に基づいて変更することはできません。

.spec.host フィールドの値を変更する場合、OpenShift Container Platform クラスターによって生成されたデフォルト値を復元することはできません。OpenShift Container Platform クラスターを復元した後、Operator がそのフィールドの値をリセットします。

4.2.2.6.4. アップグレードの注意事項



注記

必ず次のマイナーバージョンにアップグレードしてください。バージョンは絶対に **スキップしないでください**。新しいバージョンに更新するには、一度に 1 つのチャネルのみアップグレードします。たとえば、OpenShift API for Data Protection (OADP) 1.1 から 1.3 にアップグレードする場合、まず 1.2 にアップグレードし、次に 1.3 にアップグレードします。

4.2.2.6.4.1. OADP 1.1 から 1.2 への変更点

Velero サーバーが、バージョン 1.9 から 1.11 に更新されました。

OADP 1.2 では、**DataProtectionApplication** (DPA) 設定の **dpa.spec.configuration.velero.args** に、次の変更が加えられました。

- **default-volumes-to-restic** フィールドの名前が **default-volumes-to-fs-backup** に変更されました。**dpa.spec.configuration.velero.args** を使用する場合は、OADP のアップグレード後に、新しい名前で DPA に再度追加する必要があります。
- **restic-timeout** フィールドの名前が **fs-backup-timeout** に変更されました。**dpa.spec.configuration.velero.args** を使用する場合は、OADP のアップグレード後に、新しい名前で DPA に再度追加する必要があります。
- **restic** デーモンセットの名前が **node-agent** に変更されました。OADP は、デーモンセットの名前を自動的に更新します。
- カスタムリソース定義 **resticrepositories.velero.io** の名前が **backuprepositories.velero.io** に変更されました。
- カスタムリソース定義 **resticrepositories.velero.io** は、クラスターから削除できます。

4.2.2.6.5. アップグレード手順

4.2.2.6.5.1. DPA 設定をバックアップする

現在の **DataProtectionApplication** (DPA) 設定をバックアップする必要があります。

手順

- 次のコマンドを実行して、現在の DPA 設定を保存します。

例

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

4.2.2.6.5.2. OADP Operator をアップグレードする

OpenShift API for Data Protection (OADP) Operator をアップグレードする場合は、次の手順を使用します。

手順

1. OADP Operator のサブスクリプションチャネルを、**stable-1.1** から **stable-1.2** に変更します。
2. Operator とコンテナーが更新され、再起動されるまで待機します。

関連情報

- Amazon Web Services の設定
- CSI スナップショットに Data Mover を使用する
- インストール済み Operator の更新

4.2.2.6.5.3. DPA を新しいバージョンに変換する

spec.configuration.velero.args スタンザで更新されたフィールドを使用する場合は、新しいパラメーター名を使用するように **DataProtectionApplication** (DPA) マニフェストを設定する必要があります。

手順

1. **Operators → Installed Operators** をクリックして、OADP Operator を選択します。
2. **Provided APIs** を選択し、**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、現在の DPA パラメーターを表示します。

   現在の DPA の例

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

4. DPA パラメーターを更新します。
5. DPA パラメーターの値は変更せずに、名前を更新します。
   a. **default-volumes-to-restic** キーを、**default-volumes-to-fs-backup** に変更します。

b. **default-restic-prune-frequency** キーを、**default-repo-maintain-frequency** に変更します。

c. **restic-timeout** キーを、**fs-backup-timeout** に変更します。

更新された DPA の例

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

6. DPA が正常に調整するまで待機します。

注記

Restic ファイルシステムバックアップのデフォルトのタイムアウト値は 1 時間です。OADP 1.3.1 以降では、Restic および Kopia のデフォルトのタイムアウト値は 4 時間です。

4.2.2.6.5.4. アップグレードの検証

アップグレードを検証するには、次の手順を使用します。

手順

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

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

出力例

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

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

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

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
```

```
deployment.apps/oadp-operator-controller-manager    1/1     1          1          2m9s
deployment.apps/velero                              1/1     1          1          96s

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.2.3. OADP 1.1 リリースノート

OpenShift API for Data Protection (OADP) 1.1 のリリースノートでは、新機能と拡張機能、非推奨の機能、製品の推奨事項、既知の問題、および解決された問題について説明します。

4.2.3.1. OADP 1.1.8 リリースノート

OpenShift API for Data Protection (OADP) 1.1.8 リリースノートには、既知の問題がすべて記載されています。本リリースで解決された問題はありません。

4.2.3.1.1. 既知の問題

OADP 1.1.8 における既知の問題の完全なリストについては、Jira の OADP 1.1.8 の既知の問題 のリストを参照してください。

4.2.3.2. OADP 1.1.7 リリースノート

OADP 1.1.7 リリースノートには、解決された問題と既知の問題がリストされています。

4.2.3.2.1. 解決した問題

以下で強調表示された問題は、OADP 1.1.7 で解決されています。

複数の HTTP/2 対応 Web サーバーが DDoS 攻撃 (Rapid Reset Attack) に対して脆弱です

OADP 1.1 の以前のリリースでは、リクエストのキャンセルにより複数のストリームがすぐにリセットされる可能性があるため、HTTP/2 プロトコルはサービス拒否攻撃の影響を受けやすかった。サーバーは、接続ごとのアクティブなストリームの最大数に関するサーバー側の制限に達しないようにしながら、ストリームをセットアップおよび破棄する必要がありました。これにより、サーバーリソースの消費によりサービス拒否が発生しました。この CVE に関連するすべての OADP 問題のリストは、次の Jira リスト を参照してください。

詳細は、CVE-2023-39325 (Rapid Reset Attack) を参照してください。

OADP 1.1.7 のリリースで解決されたすべての問題の完全なリストについては、Jira の OADP 1.1.7 で解決された問題 のリストを参照してください。

4.2.3.2.2. 既知の問題

OADP 1.1.7 のリリースには既知の問題はありません。

4.2.3.3. OADP 1.1.6 リリースノート

OADP 1.1.6 リリースノートには、新機能、解決された問題とバグ、既知の問題がリストされています。

4.2.3.3.1. 解決した問題

Pod セキュリティー標準が原因で Restic の復元が部分的に失敗します

OCP 4.14 では、**privileged** プロファイルが **enforced** になる Pod セキュリティー標準が導入されました。以前の OADP リリースでは、このプロファイルが原因で Pod が **permission denied** エラーを受信していました。この問題は、復元順序が原因で発生していました。Pod は security context constraints (SCC) リソースの前に作成されました。この Pod が Pod のセキュリティー標準に違反するため、Pod は拒否され、その後失敗しました。OADP-2420

ジョブリソースの復元が部分的に失敗します

以前の OADP リリースでは、OCP 4.14 でジョブリソースの復元が部分的に失敗していました。この問題は古い OCP バージョンでは確認されませんでした。この問題は、ジョブリソースに追加のラベルが存在するために発生しましたが、これは古い OCP バージョンには存在していませんでした。OADP-2530

このリリースで解決された問題の一覧は、Jira の OADP 1.1.6 解決済みの問題 に記載されているリストを参照してください。

4.2.3.3.2. 既知の問題

このリリースにおける既知の問題の完全なリストについては、Jira の OADP 1.1.6 の既知の問題 のリストを参照してください。

4.2.3.4. OADP 1.1.5 リリースノート

OADP 1.1.5 リリースノートには、新機能、解決された問題とバグ、既知の問題がリストされています。

4.2.3.4.1. 新機能

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

4.2.3.4.2. 解決した問題

このリリースで解決された問題の一覧は、Jira の OADP 1.1.5 解決済みの問題 に記載されているリストを参照してください。

4.2.3.4.3. 既知の問題

このリリースにおける既知の問題の完全なリストについては、Jira の OADP 1.1.5 の既知の問題 のリストを参照してください。

4.2.3.5. OADP 1.1.4 リリースノート

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

4.2.3.5.1. 新機能

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

4.2.3.5.2. 解決した問題

すべての velero デプロイメントサーバー引数のサポートを追加しました

以前の OADP リリースでは、OADP はアップストリームのすべての Velero サーバー引数のサポートを促進しませんでした。この問題は OADP 1.1.4 で解決され、アップストリームのすべての Velero サーバー引数がサポートされるようになりました。OADP-1557

復元名と PVC 名に複数の VSR がある場合、Data Mover は誤ったスナップショットから復元できます

以前のリリースの OADP では、クラスター内に同じ Velero **restore** 名と PersistentVolumeClaim (pvc) 名の Volume Snapshot Restore (VSR) リソースが複数ある場合、OADP Data Mover が間違ったスナップショットから復元できる可能性がありました。OADP-1822

Cloud Storage API BSL には OwnerReference が必要です

OADP の以前のリリースでは、**dpa.spec.backupLocations.bucket** で作成された Backup Storage Location (BSL) で **OwnerReference** が欠落しているため、ACM BackupSchedules が検証に失敗しました。OADP-1511

このリリースで解決された問題の一覧は、Jira の OADP 1.1.4 解決済みの問題 に記載されているリストを参照してください。

4.2.3.5.3. 既知の問題

本リリースには、以下の既知の問題があります。

UID/GID 範囲がクラスター上で変更された可能性があるため、OADP バックアップが失敗する可能性があります

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

ArgoCD で使用されるラベルが原因で ArgoCD がプロセス中に使用されると、復元が失敗する場合があります。

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

OADP Velero プラグインが "received EOF, stopping recv loop" のメッセージを返す

Velero プラグインは、別のプロセスとして開始されます。Velero 操作が完了すると、成功したかどうかにかかわらず終了します。つまり、デバッグログに **received EOF, stopping recv loop** メッセージが表示されても、エラーが発生したことを意味するものではありません。このメッセージは、プラグイン操作が完了したことを示します。OADP-2176

このリリースにおける既知の問題の完全なリストについては、Jira の OADP 1.1.4 の既知の問題 のリストを参照してください。

4.2.3.6. OADP 1.1.3 リリースノート

OADP 1.1.3 リリースノートには、新機能、解決された問題とバグ、既知の問題がリストされています。

4.2.3.6.1. 新機能

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

4.2.3.6.2. 解決した問題

このリリースで解決された問題の一覧は、Jira の OADP 1.1.3 解決済みの問題 に記載されているリストを参照してください。

4.2.3.6.3. 既知の問題

このリリースにおける既知の問題の完全なリストについては、Jira の OADP 1.1.3 の既知の問題 のリストを参照してください。

4.2.3.7. OADP 1.1.2 リリースノート

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

4.2.3.7.1. 製品の推奨事項

VolSync

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

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

Velero

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

Restic

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

4.2.3.7.2. 解決した問題

このリリースでは、次の問題が解決されました。

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

4.2.3.7.3. 既知の問題

本リリースには、以下の既知の問題があります。

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

  詳細は、ボリュームのスナップショット を参照してください。

4.2.3.8. OADP 1.1.1 リリースノート

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

4.2.3.8.1. 製品の推奨事項

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

4.2.3.8.2. 既知の問題

本リリースには、以下の既知の問題があります。

- 複数の HTTP/2 対応 Web サーバーが DDoS 攻撃 (Rapid Reset Attack) に対して脆弱です
  HTTP/2 プロトコルは、リクエストのキャンセルによって複数のストリームがすぐにリセットされる可能性があるため、サービス拒否攻撃の影響を受けやすくなっています。サーバーは、接続ごとのアクティブなストリームの最大数に関するサーバー側の制限に達しないようにしながら、ストリームをセットアップおよび破棄する必要があります。これにより、サーバーのリソースが消費され、サービス拒否が発生します。この CVE に関連するすべての OADP 問題のリストは、次の Jira リスト を参照してください。

この問題を解決するには、OADP 1.1.7 または 1.2.3 にアップグレードすることを推奨します。

詳細は、CVE-2023-39325 (Rapid Reset Attack) を参照してください。

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

4.3. OADP FEATURES AND PLUGINS

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

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

4.3.1. OADP の機能

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

バックアップ

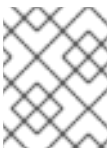
OADP を使用して OpenShift Platform 上のすべてのアプリケーションをバックアップしたり、タイプ、namespace、またはラベルでリソースをフィルターしたりできます。
OADP は、Kubernetes オブジェクトと内部イメージをアーカイブファイルとしてオブジェクトストレージに保存することにより、それらをバックアップします。OADP は、ネイティブクラウドスナップショット API または Container Storage Interface (CSI) を使用してスナップショットを作成することにより、永続ボリューム (PV) をバックアップします。スナップショットをサポートしないクラウドプロバイダーの場合、OADP は Restic を使用してリソースと PV データをバックアップします。

注記

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

復元

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



注記

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

スケジュール

指定した間隔でバックアップをスケジュールできます。

フック

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

4.3.2. 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 プラグインに基づき、カスタムプラグイン を作成できます。

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

表4.1 OADP プラグイン

OADP プラグイン	機能	ストレージの場所
aws	Kubernetes オブジェクトをバックアップし、復元します。	AWS S3
	スナップショットを使用してボリュームをバックアップおよび復元します。	AWS EBS
azure	Kubernetes オブジェクトをバックアップし、復元します。	Microsoft Azure Blob ストレージ
	スナップショットを使用してボリュームをバックアップおよび復元します。	Microsoft Azure マネージドディスク
gcp	Kubernetes オブジェクトをバックアップし、復元します。	Google Cloud Storage
	スナップショットを使用してボリュームをバックアップおよび復元します。	Google Compute Engine ディスク
openshift	OpenShift Container Platform リソースをバックアップおよび復元します。[1]	オブジェクトストア
kubevirt	OpenShift Virtualization リソースをバックアップおよび復元します。[2]	オブジェクトストア

OADP プラグイン	機能	ストレージの場所
csi	CSI スナップショットを使用して、ボリュームをバックアップおよび復元します。[3]	CSI スナップショットをサポートするクラウドストレージ
vsm	VolumeSnapshotMover は、クラスターの削除などの状況で、ステートフルアプリケーションを回復するための復元プロセス中に使用されるスナップショットをクラスターからオブジェクトストアに再配置します。[4]	オブジェクトストア

1. 必須。
2. 仮想マシンディスクは CSI スナップショットまたは Restic でバックアップされます。
3. **csi** プラグインは、Kubernetes CSI スナップショット API を使用します。
   - OADP 1.1 以降は **snapshot.storage.k8s.io/v1** を使用します。
   - OADP 1.0 は **snapshot.storage.k8s.io/v1beta1** を使用します。
4. OADP 1.2 のみ。

4.3.3. 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。

4.3.3.1. 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
```

4.3.3.2. カスタム Velero プラグイン

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

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

ファイルの例:

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

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

4.3.3.3. Velero プラグインがメッセージ "received EOF, stopping recv loop" を返す

注記

Velero プラグインは、別のプロセスとして開始されます。Velero 操作が完了すると、成功したかどうかにかかわらず終了します。デバッグログの **received EOF, stopping recv loop** メッセージは、プラグイン操作が完了したことを示します。エラーが発生したわけではありません。

4.3.4. OADP でサポートされるアーキテクチャー

OpenShift API for Data Protection (OADP) は、次のアーキテクチャーをサポートします。

- AMD64
- ARM64
- PPC64le
- s390x

注記

OADP 1.2.0 以降のバージョンは、ARM64 アーキテクチャーをサポートします。

4.3.5. IBM Power および IBM Z の OADP サポート

OpenShift API for Data Protection (OADP) はプラットフォームに依存しません。以下は、IBM Power® および IBM Z® のみに関連する情報です。

- OADP 1.1.7 は、IBM Power® および IBM Z® 上の OpenShift Container Platform 4.11 に対して正常にテストされました。以下のセクションでは、これらのシステムのバックアップ場所に関する OADP 1.1.7 のテストおよびサポート情報を提供します。
- OADP 1.2.3 は、IBM Power® および IBM Z® 上の OpenShift Container Platform 4.12、4.13、4.14、4.15 に対して正常にテストされました。以下のセクションでは、これらのシステムのバックアップ場所に関する OADP 1.2.3 のテストおよびサポート情報を提供します。
- OADP 1.3.1 は、IBM Power® および IBM Z® 上の OpenShift Container Platform 4.13、4.14、4.15 に対して正常にテストされました。以下のセクションでは、これらのシステムのバックアップ場所に関する OADP 1.3.1 のテストおよびサポート情報を提供します。

4.3.5.1. IBM Power を使用したターゲットバックアップロケーションの OADP サポート

- OpenShift Container Platform 4.11、4.12、および OpenShift API for Data Protection (OADP) 1.1.7 で実行される IBM Power® は、AWS S3 バックアップロケーションターゲットに対して正常にテストされました。テストには AWS S3 ターゲットのみが含まれていましたが、Red Hat は、AWS ではないすべての S3 バックアップロケーションターゲットに対しても、OpenShift Container Platform 4.11 と 4.12、および OADP 1.1.7 に対応する IBM Power の実行をサポートしています。
- OpenShift Container Platform 4.12、4.13、4.14、4.15、および OADP 1.2.3 で実行される IBM Power® は、AWS S3 バックアップロケーションターゲットに対して正常にテストされました。テストには AWS S3 ターゲットのみが含まれていましたが、Red Hat は、AWS ではないすべての S3 バックアップロケーションターゲットに対しても、OpenShift Container Platform 4.12、4.13、4.14、4.15、および OADP 1.2.3 に対応する IBM Power の実行をサポートしています。

- OpenShift Container Platform 4.13、4.14、4.15、および OADP 1.3.1 で実行されている IBM Power® は、AWS S3 バックアップロケーションターゲットに対して正常にテストされました。テストには AWS S3 ターゲットのみが含まれていましたが、Red Hat は、AWS ではないすべての S3 バックアップロケーションターゲットに対しても、OpenShift Container Platform 4.13、4.14、4.15、および OADP 1.3.1 に対応する IBM Power の実行をサポートしています。

4.3.5.2. IBM Z を使用したターゲットバックアップロケーションの OADP テストとサポート

- OpenShift Container Platform 4.11、4.12、および OpenShift API for Data Protection (OADP) 1.1.7 で実行される IBM Z® は、AWS S3 バックアップロケーションターゲットに対して正常にテストされました。テストには AWS S3 ターゲットのみが含まれていましたが、Red Hat は、AWS ではないすべての S3 バックアップロケーションターゲットに対しても、OpenShift Container Platform 4.11 と 4.12、および OADP 1.1.7 に対応する IBM Z® の実行をサポートしています。

- OpenShift Container Platform 4.12、4.13、4.14、4.15、および OADP 1.2.3 で実行されている IBM Z® は、AWS S3 バックアップロケーションターゲットに対して正常にテストされました。テストには AWS S3 ターゲットのみが含まれていましたが、Red Hat は、AWS ではないすべての S3 バックアップロケーションターゲットに対しても、OpenShift Container Platform 4.11、4.12、4.13、4.14、4.15、および OADP 1.2.3 に対応する IBM Z® の実行をサポートしています。

- OpenShift Container Platform 4.13、4.14、4.15、および OADP 1.3.1 で実行されている IBM Z® は、AWS S3 バックアップロケーションターゲットに対して正常にテストされました。テストには AWS S3 ターゲットのみが含まれていましたが、Red Hat は、AWS ではないすべての S3 バックアップロケーションターゲットに対しても、OpenShift Container Platform 4.13、4.14、4.15、および OADP 1.3.1 に対応する IBM Z® の実行をサポートしています。

4.3.5.2.1. IBM Power (R) および IBM Z(R) プラットフォームを使用した OADP の既知の問題

- 現在、IBM Power® および IBM Z® プラットフォームにデプロイされたシングルノードの OpenShift クラスターには、バックアップ方法の制限があります。現在、これらのプラットフォーム上のシングルノード OpenShift クラスターと互換性があるのは、NFS ストレージのみです。さらに、バックアップおよび復元操作では、Kopia や Restic などのファイルシステムバックアップ (FSB) 方式のみがサポートされます。現在、この問題に対する回避策はありません。

4.3.6. OADP プラグインの既知の問題

次のセクションでは、OpenShift API for Data Protection (OADP) プラグインの既知の問題について説明します。

4.3.6.1. シークレットがないことで、イメージストリームのバックアップ中に Velero プラグインでパニックが発生する

バックアップとバックアップ保存場所 (BSL) が Data Protection Application (DPA) の範囲外で管理されている場合、OADP コントローラー (つまり DPA の調整) によって関連する **oadp-<bsl_name>-<bsl_provider>-registry-secret** が作成されません。

バックアップを実行すると、OpenShift Velero プラグインがイメージストリームバックアップでパニックになり、次のパニックエラーが表示されます。

```
024-02-27T10:46:50.028951744Z time="2024-02-27T10:46:50Z" level=error msg="Error backing up item"
backup=openshift-adp/<backup name> error="error executing custom action
```

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

4.3.6.1.1. パニックエラーを回避するための回避策

Velero プラグインのパニックエラーを回避するには、次の手順を実行します。

1. カスタム BSL に適切なラベルを付けます。

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

2. BSL にラベルを付けた後、DPA の調整を待ちます。

注記

DPA 自体に軽微な変更を加えることで、強制的に調整を行うことができます。

3. DPA の調整では、適切な **oadp-<bsl_name>-<bsl_provider>-registry-secret** が作成されていること、正しいレジストリーデータがそこに設定されていることを確認します。

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

4.3.6.2. OpenShift ADP Controller のセグメンテーション違反

cloudstorage と **restic** の両方を有効にして DPA を設定すると、**openshift-adp-controller-manager** Pod がクラッシュし、Pod がクラッシュループのセグメンテーション違反で失敗するまで無期限に再起動します。

velero または **cloudstorage** は相互に排他的なフィールドであるため、どちらか一方だけ定義できます。

- **velero** と **cloudstorage** の両方が定義されている場合、**openshift-adp-controller-manager** は失敗します。
- **velero** と **cloudstorage** のいずれも定義されていない場合、**openshift-adp-controller-manager** は失敗します。

この問題の詳細は、OADP-1054 を参照してください。

4.3.6.2.1. OpenShift ADP Controller のセグメンテーション違反の回避策

DPA の設定時に、**velero** または **cloudstorage** のいずれかを定義する必要があります。DPA で両方の API を定義すると、**openshift-adp-controller-manager** Pod がクラッシュループのセグメンテーション違反で失敗します。

4.4. OADP のインストールおよび設定

4.4.1. 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 Operator は Velero 1.12 をインストールします。

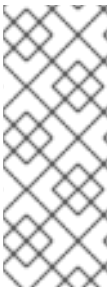
注記

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

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

- Amazon Web Services
- Microsoft Azure
- Google Cloud Platform
- Multicloud Object Gateway
- IBM Cloud® Object Storage S3
- AWS S3 互換オブジェクトストレージ (Multicloud Object Gateway、MinIO など)

個々の OADP デプロイメントごとに、同じ namespace 内に複数のバックアップストレージの場所を設定できます。



注記

特に指定のない限り、"NooBaa" は軽量オブジェクトストレージを提供するオープンソースプロジェクトを指し、"Multicloud Object Gateway (MCG)" は NooBaa の Red Hat ディストリビューションを指します。

MCG の詳細は、アプリケーションを使用して Multicloud Object Gateway にアクセスする を参照してください。

重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

注記

CloudStorage API は、**CloudStorage** オブジェクトを使用しており、OADP で **CloudStorage** API を使用して **BackupStorageLocation** として使用する S3 バケットを自動的に作成するためのテクノロジープレビュー機能です。

CloudStorage API は、既存の S3 バケットを指定して **BackupStorageLocation** オブジェクトを手動作成することをサポートしています。現在、S3 バケットを自動的に作成する **CloudStorage** API は、AWS S3 ストレージに対してのみ有効です。

スナップショットまたはファイルシステムバックアップ (FSB) を使用して、永続ボリューム (PV) をバックアップできます。

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

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

注記

OCP 4.11 以降で CSI バックアップを使用する場合は、OADP 1.1.**x** をインストールします。

OADP 1.0.**x** は、OCP 4.11 以降での CSI バックアップをサポートしていません。OADP 1.0.**x** には Velero 1.7.**x** が含まれており、OCP 4.11 以降には存在しない API グループ **snapshot.storage.k8s.io/v1beta1** が必要です。

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 である場合は、オブジェクトストレージ上の ファイルシステムバックアップによるアプリケーションのバックアップ: Kopia または 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。

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

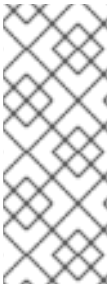
4.4.1.1. AWS S3 互換のバックアップストレージプロバイダー

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

4.4.1.1.1. サポートされているバックアップストレージプロバイダー

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

- MinIO
- Multicloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3
- IBM Cloud® Object Storage S3



注記

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

- Google Cloud Platform (GCP)
- Microsoft Azure

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

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

- Oracle Cloud
- DigitalOcean
- NooBaa (Multicloud Object Gateway (MCG) を使用してインストールされていない場合)
- Tencent Cloud
- Ceph RADOS v12.2.7
- Quobyte
- Cloudian HyperStore

注記

特に指定のない限り、"NooBaa" は軽量オブジェクトストレージを提供するオープンソースプロジェクトを指し、"Multicloud Object Gateway (MCG)" は NooBaa の Red Hat ディストリビューションを指します。

MCG の詳細は、アプリケーションを使用して Multicloud Object Gateway にアクセスする を参照してください。

4.4.1.1.3. 既知の制限があるバックアップストレージプロバイダー

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

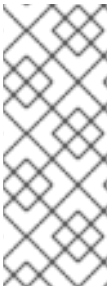
- Swift - バックアップストレージのバックアップストレージ場所として使用できますが、ファイルシステムベースのボリュームバックアップおよび復元については Restic と互換性がありません。

4.4.1.2. OpenShift Data Foundation で障害復旧を行うための Multicloud Object Gateway (MCG) 設定

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

警告

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



注記

特に指定のない限り、"NooBaa" は軽量オブジェクトストレージを提供するオープンソースプロジェクトを指し、"Multicloud Object Gateway (MCG)" は NooBaa の Red Hat ディストリビューションを指します。

MCG の詳細は、アプリケーションを使用して Multicloud Object Gateway にアクセスする を参照してください。

手順

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

関連情報

- Velero ドキュメントのバックアップとスナップショットロケーションの概要

4.4.1.3. OADP 更新チャネルについて

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

次の更新チャネルを利用できます。

- stable チャネルは非推奨になりました。stable チャネルには、**oadp.v1.1.z** および **oadp.v1.0.z** の古いバージョン用の OADP **ClusterServiceVersion** のパッチ (z-stream 更新) が含まれています。
- stable-1.0 チャネルには **oadp.v1.0.z**、OADP 1.0 **ClusterServiceVersion** が含まれています。
- stable-1.1 チャネルには **oadp.v1.1.z**、最新の OADP 1.1 **ClusterServiceVersion** が含まれています。
- stable-1.2 チャネルには、最新の OADP 1.2 **ClusterServiceVersion** の **oadp.v1.2.z** が含まれています。
- stable-1.3 チャネルには **oadp.v1.3.z**、OADP 1.3 **ClusterServiceVersion** が含まれています。

適切な更新チャネルはどれですか?

- stable チャネルは非推奨になりました。すでに安定版チャネルを使用している場合は、引き続き、**oadp.v1.1.z** から更新を取得します。

- OADP 1.**y** をインストールする **stable-1.y** 更新チャネルを選択し、そのパッチを引き続き受け取ります。このチャネルを選択すると、バージョン 1.**y.z** のすべての z-stream パッチを受け取ります。

いつ更新チャネルを切り替える必要がありますか?

- OADP 1.**y** がインストールされていて、その y-stream のパッチのみを受け取りたい場合は、**stable** 更新チャネルから **stable-1.y** 更新チャネルに切り替える必要があります。その後、バージョン 1.**y.z** のすべての z-stream パッチを受け取ります。

- OADP 1.0 がインストールされていて、OADP 1.1 にアップグレードしたい場合、OADP 1.1 のみのパッチを受け取るには、**stable-1.0** 更新チャネルから **stable-1.1** 更新チャネルに切り替える必要があります。その後、バージョン 1.1.**z** のすべての z-stream パッチを受け取ります。

- OADP 1.**y** がインストールされていて、**y** が 0 より大きく、OADP 1.0 に切り替える場合は、OADP Operator を **アンインストール** してから、**stable-1.0** 更新チャネルを使用して再インストールする必要があります。その後、バージョン 1.0.**z** のすべての z-stream パッチを受け取ります。

注記

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

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

OpenShift API for Data Protection (OADP) を同じクラスター上の複数の namespace にインストールすると、複数のプロジェクト所有者が独自の OADP インスタンスを管理できるようになります。このユースケースは、ファイルシステムバックアップ (FSB) と Container Storage Interface (CSI) を使用して検証されています。

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

- 同じクラスター上のすべての OADP デプロイメントは、同じバージョン (1.1.4 など) である必要があります。同じクラスターに異なるバージョンの OADP をインストールすることはサポート**されていません**。

- OADP の個々のデプロイメントには、一意の認証情報セットと一意の **BackupStorageLocation** 設定が必要です。同じ namespace 内で、複数の **BackupStorageLocation** 設定を使用することもできます。

- デフォルトでは、各 OADP デプロイメントには namespace 全体でクラスターレベルのアクセス権があります。OpenShift Container Platform 管理者は、セキュリティーおよび RBAC 設定を注意深く確認し、必要な変更を加えて、各 OADP インスタンスに正しい権限があることを確認する必要があります。

関連情報

- クラスターサービスバージョン

4.4.1.5. 収集したデータに基づく Velero CPU およびメモリーの要件

以下の推奨事項は、スケールおよびパフォーマンスのラボで観察したパフォーマンスに基づいています。バックアップおよび復元リソースは、プラグインのタイプ、そのバックアップまたは復元に必要なリソースの量、そのリソースに関連する永続ボリューム (PV) に含まれるデータの影響を受けます。

4.4.1.5.1. 設定に必要な CPU とメモリー

設定タイプ	[1] 平均使用量	[2] 大量使用時	resourceTimeouts
CSI	Velero: CPU - リクエスト 200m、制限 1000m メモリー - リクエスト 256 Mi、制限 1024 Mi	Velero: CPU - リクエスト 200m、制限 2000m メモリー - リクエスト 256 Mi、制限 2048 Mi	該当なし
Restic	[3] Restic: CPU - リクエスト 1000m、制限 2000m メモリー - リクエスト 16 Gi、制限 32 Gi	[4] Restic: CPU - リクエスト 2000m、制限 8000m メモリー - リクエスト 16 Gi、制限 40 Gi	900 m
[5] Data Mover	該当なし	該当なし	10m - 平均使用量 60m - 大量使用時

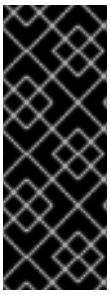
1. 平均使用量 - ほとんどの状況下でこの設定を使用します。
2. 大量使用時 - 大規模な PV (使用量 500 GB)、複数の namespace (100 以上)、または 1 つの namespace に多数の Pod (2000 Pod 以上) があるなどして使用量が大きくなる状況下では、大規模なデータセットを含む場合のバックアップと復元で最適なパフォーマンスを実現するために、この設定を使用します。
3. Restic リソースの使用量は、データの量とデータタイプに対応します。たとえば、多数の小さなファイルや大量のデータがある場合は、Restic が大量のリソースを使用する可能性があります。Velero のドキュメントでは、指定されたデフォルト値である 500 m を参照していますが、ほとんどのテストではリクエスト 200 m、制限 1000 m が適切でした。Velero のドキュメントに記載されているとおり、正確な CPU とメモリー使用量は、環境の制限に加えて、ファイルとディレクトリーの規模に依存します。
4. CPU を増やすと、バックアップと復元の時間を大幅に短縮できます。
5. Data Mover - Data Mover のデフォルトの resourceTimeout は 10 m です。テストでは、大規模な PV (使用量 500 GB) を復元するには、resourceTimeout を 60m に増やす必要があることがわかりました。

注記

このガイド全体に記載されているリソース要件は、平均的な使用量に限定されています。大量に使用する場合は、上の表の説明に従って設定を調整してください。

4.4.1.5.2. 大量使用のための NodeAgent CPU

テストの結果、**NodeAgent** CPU を増やすと、OpenShift API for Data Protection (OADP) を使用する際のバックアップと復元の時間が大幅に短縮されることがわかりました。



重要

Kopia はリソースを積極的に消費するため、実稼働環境で実稼働ワークロードを実行しているノードで Kopia を制限なく使用することは推奨されません。ただし、Kopia を実行する際の制限が低すぎると、CPU が制限され、バックアップと復元が遅くなります。テストの結果、20 コアと 32 Gi メモリーで Kopia を実行した場合、100 GB 超のデータ、複数の namespace、または単一 namespace 内の 2000 超の Pod のバックアップと復元操作がサポートされることが判明しました。

テストでは、これらのリソース仕様では CPU の制限やメモリーの飽和は検出されませんでした。

これらの制限を Ceph MDS Pod に設定するには、rook-ceph Pod の CPU およびメモリーリソースの変更 に記載された手順に従ってください。

制限を設定するには、ストレージクラスターのカスタムリソース (CR) に次の行を追加する必要があります。

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

4.4.2. OADP Operator のインストール

Operator Lifecycle Manager (OLM) を使用して、OpenShift Container Platform 4.16 に OpenShift API for Data Protection (OADP) Operator をインストールできます。

OADP Operator は Velero 1.12 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。

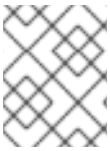
4.4.2.1. OADP-Velero-OpenShift Container Platform バージョンの関係

OADP のバージョン	Velero のバージョン	OpenShift Container Platform バージョン
1.1.0	1.9	4.9 以降
1.1.1	1.9	4.9 以降
1.1.2	1.9	4.9 以降
1.1.3	1.9	4.9 以降
1.1.4	1.9	4.9 以降
1.1.5	1.9	4.9 以降
1.1.6	1.9	4.11 以降
1.1.7	1.9	4.11 以降
1.2.0	1.11	4.11 以降
1.2.1	1.11	4.11 以降
1.2.2	1.11	4.11 以降
1.2.3	1.11	4.11 以降
1.3.0	1.12	4.12 以降

4.4.3. AWS S3 互換ストレージを使用した OpenShift API for Data Protection の設定

OADP Operator をインストールすることで、Amazon Web Services (AWS) S3 互換ストレージを使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.12 をインストールします。

IBM Cloud® S3 が AWS S3 互換のバックアップストレージプロバイダーとしてサポートされています。



注記

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

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

4.4.3.1. Amazon Web Services の設定

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

前提条件

- AWS CLI がインストールされていること。

手順

1. **BUCKET** 変数を設定します。

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

2. **REGION** 変数を設定します。

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

3. AWS S3 バケットを作成します。

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

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

4. IAM ユーザーを作成します。

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

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

5. **velero-policy.json** ファイルを作成します。

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

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

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

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

7. **velero** ユーザーのアクセスキーを作成します。

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

出力例

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

8. **credentials-velero** ファイルを作成します。

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

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

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

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

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

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

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

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

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

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

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

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

重要

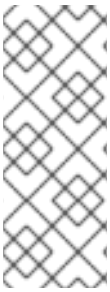
Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

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

4.4.3.2.1. デフォルト Secret の作成

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

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



注記

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

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

前提条件

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

手順

- デフォルト名で **Secret** を作成します。

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

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

4.4.3.2.2. 異なる認証情報のプロファイルの作成

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

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

手順

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

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

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

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

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

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

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
        config:
          region: us-east-1
          profile: "backupStorage"
        credential:
          key: cloud
          name: cloud-credentials
  snapshotLocations:
    - velero:
        provider: aws
        config:
          region: us-west-2
          profile: "volumeSnapshot"
```

4.4.3.3. Data Protection Application の設定

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

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

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

前提条件

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

手順

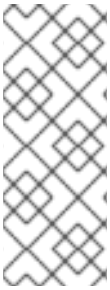
- 次の例のように、**DataProtectionApplication** CR マニフェストの **spec.configuration.velero.podConfig.ResourceAllocations** ブロックの値を編集します。

```
apiVersion: oadp.openshift.io/v1alpha1
```

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

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

2 リストされている **resourceAllocations** は、平均使用量です。



注記

Kopia は OADP 1.3 以降のリリースで選択できます。Kopia はファイルシステムのバックアップに使用できます。組み込みの Data Mover を使用する Data Mover の場合は、Kopia が唯一の選択肢になります。

Kopia は Restic よりも多くのリソースを消費するため、それに応じて CPU とメモリーの要件を調整しなければならない場合があります。

4.4.3.3.2. 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. Base64 でエンコードされた CA 証明書文字列を指定します。
2. **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.4.3.3.2.1. Velero デプロイメント用のエイリアス化した velero コマンドで CA 証明書を使用する

Velero CLI のエイリアスを作成することで、システムにローカルにインストールせずに Velero CLI を使用できます。

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している。
- OpenShift CLI (**oc**) がインストールされている。

1. エイリアス化した Velero コマンドを使用するには、次のコマンドを実行します。

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

2. 次のコマンドを実行して、エイリアスが機能していることを確認します。

例

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

3. このコマンドで CA 証明書を使用するには、次のコマンドを実行して証明書を Velero デプロイメントに追加できます。

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

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

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

4. バックアップログを取得するために、次のコマンドを実行します。

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

このログを使用して、バックアップできないリソースの障害と警告を表示できます。

5. Velero Pod が再起動すると、**/tmp/your-cacert.txt** ファイルが消去されます。そのため、前の手順のコマンドを再実行して **/tmp/your-cacert.txt** ファイルを再作成する必要があります。

6. 次のコマンドを実行すると、**/tmp/your-cacert.txt** ファイルを保存した場所にファイルがまだ存在するかどうかを確認できます。

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

OpenShift API for Data Protection (OADP) の今後のリリースでは、この手順が不要になるように証明書を Velero Pod にマウントする予定です。

4.4.3.4. Data Protection Application 1.2 以前のインストール

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る。
- 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る。
- 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る。
- 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る。
- 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。

注記

Velero は、OADP namespace に **velero-repo-credentials** という名前のシークレットを作成します。これには、デフォルトのバックアップリポジトリーパスワードが含まれます。バックアップリポジトリーを対象とした最初のバックアップを実行する **前** に、base64 としてエンコードされた独自のパスワードを使用してシークレットを更新できます。更新するキーの値は **Data[repository-password]** です。

DPA を作成した後、バックアップリポジトリーを対象としたバックアップを初めて実行するときに、Velero はシークレットが **velero-repo-credentials** のバックアップリポジトリーを作成します。これには、デフォルトのパスワードまたは置き換えたパスワードが含まれます。最初のバックアップの **後** にシークレットパスワードを更新すると、新しいパスワードが **velero-repo-credentials** のパスワードと一致しなくなり、Velero は古いバックアップに接続できなくなります。

手順

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

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

```
      s3Url: <s3_url> 8
    credential:
      key: cloud
      name: cloud-credentials 9
snapshotLocations: 10
 - velero:
    provider: aws
    config:
      region: <region> 11
      profile: "default"
```

1. **openshift** プラグインは必須です。
2. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
3. Restic インストールを無効にする場合は、この値を **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、Restic Pod が各動作ノードで実行していることを意味します。OADP バージョン 1.2 以降では、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。OADP バージョン 1.1 では、**spec.defaultVolumesToRestic: true** を **Backup** CR に追加します。
4. Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。
5. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
6. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。
7. S3 オブジェクトにパススタイルの URL を強制するかどうかを指定します (ブール値)。AWS S3 では必要ありません。S3 互換ストレージにのみ必要です。
8. バックアップを保存するために使用しているオブジェクトストアの URL を指定します。AWS S3 では必要ありません。S3 互換ストレージにのみ必要です。
9. 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。
10. CSI スナップショットまたは Restic を使用して PV をバックアップする場合を除き、スナップショットの場所を指定します。
11. 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。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

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

出力例

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

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.3.5. Data Protection Application 1.3 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る。
- 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る。
- 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る。
- 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る。
- 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 (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - openshift (2)
        - aws
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
      podConfig:
        nodeSelector: <node_selector> (7)
  backupLocations:
    - name: default
      velero:
        provider: aws
        default: true
```

```
      objectStorage:
        bucket: <bucket_name> 8
        prefix: <prefix> 9
      config:
        region: <region>
        profile: "default"
        s3ForcePathStyle: "true" 10
        s3Url: <s3_url> 11
      credential:
        key: cloud
        name: cloud-credentials 12
  snapshotLocations: 13
   - name: default
     velero:
       provider: aws
       config:
         region: <region> 14
         profile: "default"
```

1. OADP のデフォルトの namespace は **openshift-adp** です。namespace は変数であり、設定可能です。
2. **openshift** プラグインは必須です。
3. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
4. 管理要求をサーバーにルーティングする管理エージェント。
5. **nodeAgent** を有効にしてファイルシステムバックアップを実行する場合は、この値を **true** に設定します。
6. アップローダーとして **kopia** または **restic** と入力します。インストール後に選択を変更することはできません。組み込み DataMover の場合は、Kopia を使用する必要があります。**nodeAgent** はデーモンセットをデプロイします。これは、**nodeAgent** Pod が各ワーキングノード上で実行されることを意味します。ファイルシステムバックアップを設定するには、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加します。
7. Kopia または Restic が使用可能なノードを指定します。デフォルトでは、Kopia または Restic はすべてのノードで実行されます。
8. 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。
9. 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**)。
10. S3 オブジェクトにパススタイルの URL を強制するかどうかを指定します (ブール値)。AWS S3 では必要ありません。S3 互換ストレージにのみ必要です。
11. バックアップを保存するために使用しているオブジェクトストアの URL を指定します。AWS S3 では必要ありません。S3 互換ストレージにのみ必要です。
12. 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。

13 CSI スナップショットまたはファイルシステムバックアップ (FSB) を使用して PV をバックアップする場合を除き、スナップショットの場所を指定します。

14 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。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

```
$ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.3.5.1. DataProtectionApplication CR で CSI を有効にする

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

前提条件

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

手順

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

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

(1) **csi** デフォルトプラグインを追加します。

関連情報

- **kubevirt** および **openshift** プラグインを使用した Data Protection Application のインストール

4.4.4. Microsoft Azure を使用した OpenShift API for Data Protection の設定

OADP Operator をインストールすることで、Microsoft Azure を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.12 をインストールします。

注記

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

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

4.4.4.1. 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. **velero** のサービスプリンシパルおよび認証情報を作成します。

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

9. サービスプリンシパルの認証情報を **credentials-velero** ファイルに保存します。

```
$ 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_CLOUD_NAME=AzurePublicCloud
EOF
```

credentials-velero ファイルを使用して、Azure をレプリケーションリポジトリーとして追加します。

10. 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`
```

11. 必要最小限のパーミッションを持つカスタムロールを作成します。

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

12. **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。

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

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

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

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

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

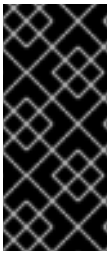
Container Storage Interface (CSI) スナップショットを使用する場合、CSI ドライバーを登録するために **VolumeSnapshotClass** CR を作成するため、スナップショットの場所を指定する必要はありません。

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

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

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

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



重要

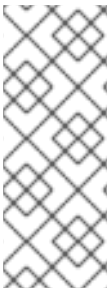
Data Protection Application には、デフォルトの **Secret** が必要です。作成しないと、インストールは失敗します。

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

4.4.4.2.1. デフォルト Secret の作成

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

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



注記

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

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

前提条件

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

手順

- デフォルト名で **Secret** を作成します。

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

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

4.4.4.2.2. 異なる認証情報のシークレットの作成

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

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

(1) カスタム名を持つバックアップ場所の **Secret**。

4.4.4.3. Data Protection Application の設定

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

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

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

前提条件

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

手順

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

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

1. Velero podSpec に提供されるノードセレクターを指定します。
2. リストされている **resourceAllocations** は、平均使用量です。

注記

Kopia は OADP 1.3 以降のリリースで選択できます。Kopia はファイルシステムのバックアップに使用できます。組み込みの Data Mover を使用する Data Mover の場合は、Kopia が唯一の選択肢になります。

Kopia は Restic よりも多くのリソースを消費するため、それに応じて CPU とメモリーの要件を調整しなければならない場合があります。

4.4.4.3.2. 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. Base64 でエンコードされた CA 証明書文字列を指定します。
2. **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.4.4.3.2.1. Velero デプロイメント用のエイリアス化した velero コマンドで CA 証明書を使用する

Velero CLI のエイリアスを作成することで、システムにローカルにインストールせずに Velero CLI を使用できます。

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している。
- OpenShift CLI (**oc**) がインストールされている。

1. エイリアス化した Velero コマンドを使用するには、次のコマンドを実行します。

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

2. 次のコマンドを実行して、エイリアスが機能していることを確認します。

   例

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

3. このコマンドで CA 証明書を使用するには、次のコマンドを実行して証明書を Velero デプロイメントに追加できます。

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

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

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

4. バックアップログを取得するために、次のコマンドを実行します。

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

   このログを使用して、バックアップできないリソースの障害と警告を表示できます。

5. Velero Pod が再起動すると、**/tmp/your-cacert.txt** ファイルが消去されます。そのため、前の手順のコマンドを再実行して **/tmp/your-cacert.txt** ファイルを再作成する必要があります。

6. 次のコマンドを実行すると、**/tmp/your-cacert.txt** ファイルを保存した場所にファイルがまだ存在するかどうかを確認できます。

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

OpenShift API for Data Protection (OADP) の今後のリリースでは、この手順が不要になるように証明書を Velero Pod にマウントする予定です。

4.4.4.4. Data Protection Application 1.2 以前のインストール

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る。

- 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る。
- 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る。
- 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る。
- 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、以下のように 2 つの **Secrets** を作成する必要がある。
  - バックアップの場所用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用の別のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。

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。

注記

Velero は、OADP namespace に **velero-repo-credentials** という名前のシークレットを作成します。これには、デフォルトのバックアップリポジトリーパスワードが含まれます。バックアップリポジトリーを対象とした最初のバックアップを実行する **前** に、base64 としてエンコードされた独自のパスワードを使用してシークレットを更新できます。更新するキーの値は **Data[repository-password]** です。

DPA を作成した後、バックアップリポジトリーを対象としたバックアップを初めて実行するときに、Velero はシークレットが **velero-repo-credentials** のバックアップリポジトリーを作成します。これには、デフォルトのパスワードまたは置き換えたパスワードが含まれます。最初のバックアップの **後** にシークレットパスワードを更新すると、新しいパスワードが **velero-repo-credentials** のパスワードと一致しなくなり、Velero は古いバックアップに接続できなくなります。

手順

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

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

```
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - azure
        - openshift (1)
      resourceTimeout: 10m (2)
    restic:
      enable: true (3)
      podConfig:
        nodeSelector: <node_selector> (4)
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group> (5)
          storageAccount: <azure_storage_account_id> (6)
          subscriptionId: <azure_subscription_id> (7)
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: cloud-credentials-azure (8)
        provider: azure
        default: true
        objectStorage:
          bucket: <bucket_name> (9)
          prefix: <prefix> (10)
  snapshotLocations: (11)
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          subscriptionId: <azure_subscription_id>
          incremental: "true"
        name: default
        provider: azure
```

(1) **openshift** プラグインは必須です。

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

(3) Restic インストールを無効にする場合は、この値を **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、Restic Pod が各動作ノードで実行していることを意味します。OADP バージョン 1.2 以降では、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。OADP バージョン 1.1 では、**spec.defaultVolumesToRestic: true** を **Backup** CR に追加します。

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

(5) Azure リソースグループを指定します。

(6) Azure ストレージアカウント ID を指定します。

7 Azure サブスクリプション ID を指定します。

8 この値を指定しない場合は、デフォルト名の **cloud-credentials-azure** が使用されます。カスタム名を指定すると、バックアップの場所にカスタム名が使用されます。

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

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

11 CSI スナップショットまたは Restic を使用して PV をバックアップする場合は、スナップショットの場所を指定する必要はありません。

4. **Create** をクリックします。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

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

出力例

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

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.4.5. Data Protection Application 1.3 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る。
- 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る。
- 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る。
- 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る。

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 (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - azure
        - openshift (2)
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
      podConfig:
        nodeSelector: <node_selector> (7)
  backupLocations:
    - velero:
        config:
          resourceGroup: <azure_resource_group> (8)
          storageAccount: <azure_storage_account_id> (9)
          subscriptionId: <azure_subscription_id> (10)
          storageAccountKeyEnvVar: AZURE_STORAGE_ACCOUNT_ACCESS_KEY
        credential:
          key: cloud
          name: cloud-credentials-azure (11)
        provider: azure
        default: true
        objectStorage:
          bucket: <bucket_name> (12)
          prefix: <prefix> (13)
  snapshotLocations: (14)
    - velero:
        config:
          resourceGroup: <azure_resource_group>
          subscriptionId: <azure_subscription_id>
          incremental: "true"
        name: default
        provider: azure
```

(1) OADP のデフォルトの namespace は **openshift-adp** です。namespace は変数であり、設定可能です。

(2) **openshift** プラグインは必須です。

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

(4) 管理要求をサーバーにルーティングする管理エージェント。

(5) **nodeAgent** を有効にしてファイルシステムバックアップを実行する場合は、この値を **true** に設定します。

(6) アップローダーとして **kopia** または **restic** と入力します。インストール後に選択を変更す

7 Kopia または Restic が使用可能なノードを指定します。デフォルトでは、Kopia または Restic はすべてのノードで実行されます。

8 Azure リソースグループを指定します。

9 Azure ストレージアカウント ID を指定します。

10 Azure サブスクリプション ID を指定します。

11 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。

12 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。

13 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**)。

14 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。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

```
$ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.4.5.1. DataProtectionApplication CR で CSI を有効にする

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

前提条件

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

手順

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

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

(1) **csi** デフォルトプラグインを追加します。

関連情報

- **kubevirt** および **openshift** プラグインを使用した Data Protection Application のインストール

4.4.5. Google Cloud Platform を使用した OpenShift API for Data Protection の設定

OADP Operator をインストールすることで、Google Cloud Platform (GCP) を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.12 をインストールします。

注記

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

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

4.4.5.1. 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リスト表示します。

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

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

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

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

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

9. **velero.server** カスタムロールを作成します。

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

10. IAM ポリシーバインディングをプロジェクトに追加します。

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

11. IAM サービスアカウントを更新します。

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

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

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

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

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

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

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

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

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

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

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

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

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

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

重要

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

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

4.4.5.2.1. デフォルト Secret の作成

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

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

注記

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

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

前提条件

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

手順

- デフォルト名で **Secret** を作成します。

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

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

4.4.5.2.2. 異なる認証情報のシークレットの作成

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

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

手順

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

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

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

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

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

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

(1) カスタム名を持つバックアップ場所の **Secret**。

4.4.5.3. Data Protection Application の設定

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

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

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

前提条件

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

手順

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

```
apiVersion: oadp.openshift.io/v1alpha1
```

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

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

2 リストされている **resourceAllocations** は、平均使用量です。

注記

Kopia は OADP 1.3 以降のリリースで選択できます。Kopia はファイルシステムのバックアップに使用できます。組み込みの Data Mover を使用する Data Mover の場合は、Kopia が唯一の選択肢になります。

Kopia は Restic よりも多くのリソースを消費するため、それに応じて CPU とメモリーの要件を調整しなければならない場合があります。

4.4.5.3.2. 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. Base64 でエンコードされた CA 証明書文字列を指定します。
2. **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.4.5.3.2.1. Velero デプロイメント用のエイリアス化した velero コマンドで CA 証明書を使用する

Velero CLI のエイリアスを作成することで、システムにローカルにインストールせずに Velero CLI を使用できます。

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している。
- OpenShift CLI (**oc**) がインストールされている。

1. エイリアス化した Velero コマンドを使用するには、次のコマンドを実行します。

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

2. 次のコマンドを実行して、エイリアスが機能していることを確認します。

例

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

3. このコマンドで CA 証明書を使用するには、次のコマンドを実行して証明書を Velero デプロイメントに追加できます。

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

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

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

4. バックアップログを取得するために、次のコマンドを実行します。

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

このログを使用して、バックアップできないリソースの障害と警告を表示できます。

5. Velero Pod が再起動すると、**/tmp/your-cacert.txt** ファイルが消去されます。そのため、前の手順のコマンドを再実行して **/tmp/your-cacert.txt** ファイルを再作成する必要があります。

6. 次のコマンドを実行すると、**/tmp/your-cacert.txt** ファイルを保存した場所にファイルがまだ存在するかどうかを確認できます。

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

OpenShift API for Data Protection (OADP) の今後のリリースでは、この手順が不要になるように証明書を Velero Pod にマウントする予定です。

4.4.5.4. Data Protection Application 1.2 以前のインストール

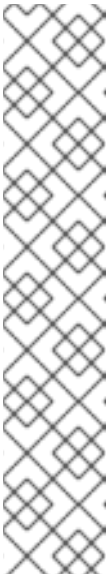
DataProtectionApplication API のインスタンスを作成して、Data Protection Application (DPA) をインストールします。

前提条件

- OADP Operator をインストールする必要がある。
- 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る。
- 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る。
- 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る。
- 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、以下のように 2 つの **Secrets** を作成する必要がある。
  - バックアップの場所用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用の別のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。

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。



注記

Velero は、OADP namespace に **velero-repo-credentials** という名前のシークレットを作成します。これには、デフォルトのバックアップリポジトリーパスワードが含まれます。バックアップリポジトリーを対象とした最初のバックアップを実行する **前** に、base64 としてエンコードされた独自のパスワードを使用してシークレットを更新できます。更新するキーの値は **Data[repository-password]** です。

DPA を作成した後、バックアップリポジトリーを対象としたバックアップを初めて実行するときに、Velero はシークレットが **velero-repo-credentials** のバックアップリポジトリーを作成します。これには、デフォルトのパスワードまたは置き換えたパスワードが含まれます。最初のバックアップの **後** にシークレットパスワードを更新すると、新しいパスワードが **velero-repo-credentials** のパスワードと一致しなくなり、Velero は古いバックアップに接続できなくなります。

手順

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

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

```
    provider: gcp
    default: true
    config:
      project: <project>
      snapshotLocation: us-west1 10
```

1. **openshift** プラグインは必須です。
2. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
3. Restic インストールを無効にする場合は、この値を **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、Restic Pod が各動作ノードで実行していることを意味します。OADP バージョン 1.2 以降では、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。OADP バージョン 1.1 では、**spec.defaultVolumesToRestic: true** を **Backup** CR に追加します。
4. Restic を使用できるノードを指定します。デフォルトでは、Restic はすべてのノードで実行されます。
5. 認証情報を含む秘密鍵。Google Workload Identity 連携クラウド認証の場合は、**service_account.json** を使用します。
6. 認証情報を含むシークレットの名前。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。
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**)。
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を除き、スナップショットの場所を指定します。
10. 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。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

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

出力例

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

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

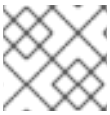
4.4.5.5. Google Workload Identity 連携のクラウド認証

Google Cloud の外で実行されているアプリケーションは、ユーザー名やパスワードなどのサービスアカウントキーを使用して、Google Cloud リソースにアクセスします。これらのサービスアカウントキーは、適切に管理されていない場合、セキュリティーリスクになる可能性があります。

Google Workload Identity 連携を使用すると、Identity and Access Management (IAM) を使用して、サービスアカウントに成り代わる機能などの IAM ロールを外部アイデンティティーに付与できます。これにより、サービスアカウントキーに関連するメンテナンスとセキュリティーのリスクが排除されます。

Workload Identity 連携は、証明書の暗号化と復号化、ユーザー属性の抽出、および検証を処理します。Identity 連携は認証を外部化し、それをセキュリティートークンサービス (STS) に渡すことで、個々の開発者の負担を軽減します。リソースへのアクセスの認可と制御は、引き続きアプリケーションが処理

します。



注記

Google Workload Identity 連携は、OADP 1.3.x 以降で利用できます。

ボリュームをバックアップする場合、Google Workload Identity 連携認証を使用した GCP 上の OADP は、CSI スナップショットのみをサポートします。

Google Workload Identity 連携認証を使用した GCP 上の OADP は、Volume Snapshot Locations (VSL) バックアップをサポートしません。詳細は、Google Workload Identity 連携の既知の問題 を参照してください。

Google Workload Identity 連携クラウド認証を使用しない場合は、**Data Protection Application のインストール** に進みます。

前提条件

- GCP Workload Identity を設定 して、クラスターを主導モードでインストールしている。
- Cloud Credential Operator ユーティリティー (**ccoctl**) と、関連する Workload Identity プールにアクセスできる。

手順

1. 次のコマンドを実行して、**oadp-credrequest** ディレクトリーを作成します。

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

2. 次のように、**CredentialsRequest.yaml** ファイルを作成します。

```
echo 'apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: oadp-operator-credentials
  namespace: openshift-cloud-credential-operator
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: GCPProviderSpec
    permissions:
    - compute.disks.get
    - compute.disks.create
    - compute.disks.createSnapshot
    - compute.snapshots.get
    - compute.snapshots.create
    - compute.snapshots.useReadOnly
    - compute.snapshots.delete
    - compute.zones.get
    - storage.objects.create
    - storage.objects.delete
    - storage.objects.get
    - storage.objects.list
    - iam.serviceAccounts.signBlob
    skipServiceCheck: true
```

```
    secretRef:
      name: cloud-credentials-gcp
      namespace: <OPERATOR_INSTALL_NS>
    serviceAccountNames:
    - velero
  ' > oadp-credrequest/credrequest.yaml
```

3. 次のコマンドを実行し、**ccoctl** ユーティリティーを使用して、**oadp-credrequest** ディレクトリー内の **CredentialsRequest** オブジェクトを処理します。

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

これで、次のステップで **manifests/openshift-adp-cloud-credentials-gcp-credentials.yaml** ファイルを使用できるようになりました。

4. 次のコマンドを実行して、namespace を作成します。

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

5. 次のコマンドを実行して、認証情報を namespace に適用します。

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

4.4.5.5.1. Google Workload Identity 連携の既知の問題

- GCP Workload Identity 連携が設定されている場合、Volume Snapshot Location (VSL) バックアップは **PartiallyFailed** フェーズで終了します。Google Workload Identity 連携認証は、VSL バックアップをサポートしません。

4.4.5.6. Data Protection Application 1.3 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る。
- 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る。
- 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る。
- 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る。

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: <OPERATOR_INSTALL_NS> 1
spec:
  configuration:
    velero:
      defaultPlugins:
        - gcp
        - openshift 2
      resourceTimeout: 10m 3
    nodeAgent: 4
      enable: true 5
      uploaderType: kopia 6
      podConfig:
        nodeSelector: <node_selector> 7
  backupLocations:
    - velero:
        provider: gcp
        default: true
        credential:
          key: cloud 8
          name: cloud-credentials-gcp 9
        objectStorage:
          bucket: <bucket_name> 10
          prefix: <prefix> 11
  snapshotLocations: 12
    - velero:
        provider: gcp
        default: true
        config:
          project: <project>
          snapshotLocation: us-west1 13
  backupImages: true 14
```

1. OADP のデフォルトの namespace は **openshift-adp** です。namespace は変数であり、設定可能です。

2. **openshift** プラグインは必須です。
3. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
4. 管理要求をサーバーにルーティングする管理エージェント。
5. **nodeAgent** を有効にしてファイルシステムバックアップを実行する場合は、この値を **true** に設定します。
6. アップローダーとして **kopia** または **restic** と入力します。インストール後に選択を変更することはできません。組み込み DataMover の場合は、Kopia を使用する必要があります。**nodeAgent** はデーモンセットをデプロイします。これは、**nodeAgent** Pod が各ワーキングノード上で実行されることを意味します。ファイルシステムバックアップを設定するには、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加します。
7. Kopia または Restic が使用可能なノードを指定します。デフォルトでは、Kopia または Restic はすべてのノードで実行されます。
8. 認証情報を含む秘密鍵。Google Workload Identity 連携クラウド認証の場合は、**service_account.json** を使用します。
9. 認証情報を含むシークレットの名前。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。
10. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
11. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。
12. CSI スナップショットまたは Restic を使用して PV をバックアップする場合を除き、スナップショットの場所を指定します。
13. 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。
14. Google Workload Identity 連携は、内部イメージのバックアップをサポートしています。イメージのバックアップを使用しない場合は、このフィールドを **false** に設定します。

4. **Create** をクリックします。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

```
$ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
```

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

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.5.6.1. DataProtectionApplication CR で CSI を有効にする

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

前提条件

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

手順

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

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

(1) **csi** デフォルトプラグインを追加します。

関連情報

- **kubevirt** および **openshift** プラグインを使用した Data Protection Application のインストール

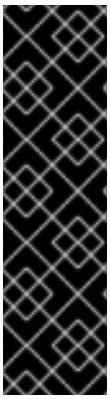
4.4.6. Multicloud Object Gateway を使用した OpenShift API for Data Protection の設定

OADP Operator をインストールすることで、Multicloud Object Gateway (MCG) を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.12 をインストールします。

注記

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

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



重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

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

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

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

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

前提条件

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

手順

1. **NooBaa** カスタムリソースで **describe** コマンドを実行して、S3 エンドポイントである **AWS_ACCESS_KEY_ID** および **AWS_SECRET_ACCESS_KEY** を取得します。

2. **credentials-velero** ファイルを作成します。

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

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

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

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

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

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

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

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

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

シークレット

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

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

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

重要

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

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

4.4.6.2.1. デフォルト Secret の作成

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

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

注記

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

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

前提条件

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

手順

- デフォルト名で **Secret** を作成します。

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

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

4.4.6.2.2. 異なる認証情報のシークレットの作成

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

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

手順

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

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

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

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

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

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

1 カスタム名を持つバックアップ場所の **Secret**。

4.4.6.3. Data Protection Application の設定

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

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

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

前提条件

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

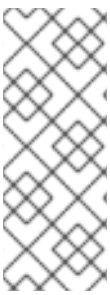
手順

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

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

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

2 リストされている **resourceAllocations** は、平均使用量です。



注記

Kopia は OADP 1.3 以降のリリースで選択できます。Kopia はファイルシステムのバックアップに使用できます。組み込みの Data Mover を使用する Data Mover の場合は、Kopia が唯一の選択肢になります。

Kopia は Restic よりも多くのリソースを消費するため、それに応じて CPU とメモリーの要件を調整しなければならない場合があります。

4.4.6.3.2. 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. Base64 でエンコードされた CA 証明書文字列を指定します。
2. **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.4.6.3.2.1. Velero デプロイメント用のエイリアス化した velero コマンドで CA 証明書を使用する

Velero CLI のエイリアスを作成することで、システムにローカルにインストールせずに Velero CLI を使用できます。

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している。
- OpenShift CLI (**oc**) がインストールされている。

1. エイリアス化した Velero コマンドを使用するには、次のコマンドを実行します。

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

2. 次のコマンドを実行して、エイリアスが機能していることを確認します。

例

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

3. このコマンドで CA 証明書を使用するには、次のコマンドを実行して証明書を Velero デプロイメントに追加できます。

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

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

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

4. バックアップログを取得するために、次のコマンドを実行します。

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

このログを使用して、バックアップできないリソースの障害と警告を表示できます。

5. Velero Pod が再起動すると、**/tmp/your-cacert.txt** ファイルが消去されます。そのため、前の手順のコマンドを再実行して **/tmp/your-cacert.txt** ファイルを再作成する必要があります。

6. 次のコマンドを実行すると、**/tmp/your-cacert.txt** ファイルを保存した場所にファイルがまだ存在するかどうかを確認できます。

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

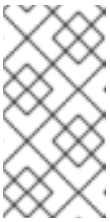
OpenShift API for Data Protection (OADP) の今後のリリースでは、この手順が不要になるように証明書を Velero Pod にマウントする予定です。

4.4.6.4. Data Protection Application 1.2 以前のインストール

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る。
- 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る。
- 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る。
- 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る。
- 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、以下のように 2 つの **Secrets** を作成する必要がある。
  - バックアップの場所用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用の別のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。



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。

注記

Velero は、OADP namespace に **velero-repo-credentials** という名前のシークレットを作成します。これには、デフォルトのバックアップリポジトリーパスワードが含まれます。バックアップリポジトリーを対象とした最初のバックアップを実行する **前** に、base64 としてエンコードされた独自のパスワードを使用してシークレットを更新できます。更新するキーの値は **Data[repository-password]** です。

DPA を作成した後、バックアップリポジトリーを対象としたバックアップを初めて実行するときに、Velero はシークレットが **velero-repo-credentials** のバックアップリポジトリーを作成します。これには、デフォルトのパスワードまたは置き換えたパスワードが含まれます。最初のバックアップの **後** にシークレットパスワードを更新すると、新しいパスワードが **velero-repo-credentials** のパスワードと一致しなくなり、Velero は古いバックアップに接続できなくなります。

手順

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

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

```
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
  configuration:
    velero:
      defaultPlugins:
        - aws
        - openshift (1)
      resourceTimeout: 10m (2)
    restic:
      enable: true (3)
      podConfig:
        nodeSelector: <node_selector> (4)
  backupLocations:
    - velero:
        config:
          profile: "default"
          region: minio
          s3Url: <url> (5)
          insecureSkipTLSVerify: "true"
          s3ForcePathStyle: "true"
        provider: aws
        default: true
        credential:
          key: cloud
          name: cloud-credentials (6)
        objectStorage:
          bucket: <bucket_name> (7)
          prefix: <prefix> (8)
```

(1) **openshift** プラグインは必須です。

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

(3) Restic インストールを無効にする場合は、この値を **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、Restic Pod が各動作ノードで実行していることを意味します。OADP バージョン 1.2 以降では、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加することで、バックアップ用に Restic を設定できます。OADP バージョン 1.1 では、**spec.defaultVolumesToRestic: true** を **Backup** CR に追加します。

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

(5) S3 エンドポイントの URL を指定します。

(6) この値を指定しない場合は、デフォルト名の **cloud-credentials** が使用されます。カスタム名を指定すると、バックアップの場所にカスタム名が使用されます。

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

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

4. **Create** をクリックします。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

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

出力例

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

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.6.5. Data Protection Application 1.3 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る。
- 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る。
- 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る。
- 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る。

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 (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - aws
        - openshift (2)
      resourceTimeout: 10m (3)
    nodeAgent: (4)
      enable: true (5)
      uploaderType: kopia (6)
```

```
    podConfig:
      nodeSelector: <node_selector> 7
backupLocations:
  - velero:
      config:
        profile: "default"
        region: minio
        s3Url: <url> 8
        insecureSkipTLSVerify: "true"
        s3ForcePathStyle: "true"
      provider: aws
      default: true
      credential:
        key: cloud
        name: cloud-credentials 9
      objectStorage:
        bucket: <bucket_name> 10
        prefix: <prefix> 11
```

1. OADP のデフォルトの namespace は **openshift-adp** です。namespace は変数であり、設定可能です。
2. **openshift** プラグインは必須です。
3. Velero CRD の可用性、volumeSnapshot の削除、バックアップリポジトリーの可用性など、タイムアウトが発生するまでに複数の Velero リソースを待機する時間を分単位で指定します。デフォルトは 10m です。
4. 管理要求をサーバーにルーティングする管理エージェント。
5. **nodeAgent** を有効にしてファイルシステムバックアップを実行する場合は、この値を **true** に設定します。
6. アップローダーとして **kopia** または **restic** と入力します。インストール後に選択を変更することはできません。組み込み DataMover の場合は、Kopia を使用する必要があります。**nodeAgent** はデーモンセットをデプロイします。これは、**nodeAgent** Pod が各ワーキングノード上で実行されることを意味します。ファイルシステムバックアップを設定するには、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加します。
7. Kopia または Restic が使用可能なノードを指定します。デフォルトでは、Kopia または Restic はすべてのノードで実行されます。
8. S3 エンドポイントの URL を指定します。
9. 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。
10. バックアップの保存場所としてバケットを指定します。バケットが Velero バックアップ専用のバケットでない場合は、接頭辞を指定する必要があります。
11. バケットが複数の目的で使用される場合は、Velero バックアップの接頭辞を指定します (例: **velero**)。

4. **Create** をクリックします。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

```
$ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.6.5.1. DataProtectionApplication CR で CSI を有効にする

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

前提条件

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

手順

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

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

(1) **csi** デフォルトプラグインを追加します。

関連情報

- Performance tuning guide for Multicloud Object Gateway
- **kubevirt** および **openshift** プラグインを使用した Data Protection Application のインストール

4.4.7. OpenShift Data Foundation を使用した OpenShift API for Data Protection の設定

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

注記

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

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

重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

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

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

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

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

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

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

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

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

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

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

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

重要

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

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

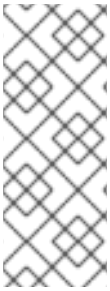
関連情報

- OpenShift Web コンソールを使用した Object Bucket Claim の作成 。

4.4.7.1.1. デフォルト Secret の作成

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

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



注記

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

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

前提条件

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

手順

- デフォルト名で **Secret** を作成します。

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

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

4.4.7.1.2. 異なる認証情報のシークレットの作成

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

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

手順

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

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

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

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

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

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: <dpa_sample>
  namespace: openshift-adp
spec:
...
  backupLocations:
    - velero:
        provider: <provider>
        default: true
        credential:
          key: cloud
          name: <custom_secret> (1)
        objectStorage:
          bucket: <bucket_name>
          prefix: <prefix>
```

(1) カスタム名を持つバックアップ場所の **Secret**。

4.4.7.2. Data Protection Application の設定

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

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

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

前提条件

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

手順

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

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

1. Velero podSpec に提供されるノードセレクターを指定します。
2. リストされている **resourceAllocations** は、平均使用量です。

注記

Kopia は OADP 1.3 以降のリリースで選択できます。Kopia はファイルシステムのバックアップに使用できます。組み込みの Data Mover を使用する Data Mover の場合は、Kopia が唯一の選択肢になります。

Kopia は Restic よりも多くのリソースを消費するため、それに応じて CPU とメモリーの要件を調整しなければならない場合があります。

4.4.7.2.1.1. 収集したデータに基づき Ceph の CPU およびメモリー要件を調整する

以下の推奨事項は、スケールおよびパフォーマンスのラボで観察したパフォーマンスに基づいています。この変更は、特に Red Hat OpenShift Data Foundation (ODF) に関連しています。ODF を使用する場合は、適切なチューニングガイドで公式の推奨事項を確認してください。

4.4.7.2.1.1.1. 設定に必要な CPU とメモリー

バックアップおよび復元操作には、大量の CephFS **PersistentVolumes** (PV) が必要です。**out-of-**

バックアップおよび復元操作には、大量の CephFS **PersistentVolumes** (PV) が必要です。**out-of-memory** (OOM) エラーによる Ceph MDS Pod の再起動を回避するためには、次の設定が推奨されます。

設定タイプ	要求	上限
CPU	要求が 3 に変更されました	上限は 3
メモリー	要求が 8 Gi に変更されました	上限は 128 Gi

4.4.7.2.2. 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. Base64 でエンコードされた CA 証明書文字列を指定します。
2. **insecureSkipTLSVerify** 設定は、**"true"** または **"false"** のいずれかに設定できます。**"true"** に設定すると、SSL/TLS セキュリティーが無効になります。**"false"** に設定すると、SSL/TLS セキュリティーが有効になります。

4.4.7.2.2.1. Velero デプロイメント用のエイリアス化した velero コマンドで CA 証明書を使用する

Velero CLI のエイリアスを作成することで、システムにローカルにインストールせずに Velero CLI を使用できます。

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している。
- OpenShift CLI (**oc**) がインストールされている。

1. エイリアス化した Velero コマンドを使用するには、次のコマンドを実行します。

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

2. 次のコマンドを実行して、エイリアスが機能していることを確認します。

   例

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

3. このコマンドで CA 証明書を使用するには、次のコマンドを実行して証明書を Velero デプロイメントに追加できます。

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

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

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

4. バックアップログを取得するために、次のコマンドを実行します。

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

   このログを使用して、バックアップできないリソースの障害と警告を表示できます。

5. Velero Pod が再起動すると、**/tmp/your-cacert.txt** ファイルが消去されます。そのため、前の手順のコマンドを再実行して **/tmp/your-cacert.txt** ファイルを再作成する必要があります。

6. 次のコマンドを実行すると、**/tmp/your-cacert.txt** ファイルを保存した場所にファイルがまだ存在するかどうかを確認できます。

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

OpenShift API for Data Protection (OADP) の今後のリリースでは、この手順が不要になるように証明書を Velero Pod にマウントする予定です。

4.4.7.3. Data Protection Application 1.2 以前のインストール

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る。
- 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る。
- 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る。
- 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る。
- 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、以下のように 2 つの **Secrets** を作成する必要がある。
  - バックアップの場所用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用の別のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。

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。

注記

Velero は、OADP namespace に **velero-repo-credentials** という名前のシークレットを作成します。これには、デフォルトのバックアップリポジトリーパスワードが含まれます。バックアップリポジトリーを対象とした最初のバックアップを実行する **前** に、base64 としてエンコードされた独自のパスワードを使用してシークレットを更新できます。更新するキーの値は **Data[repository-password]** です。

DPA を作成した後、バックアップリポジトリーを対象としたバックアップを初めて実行するときに、Velero はシークレットが **velero-repo-credentials** のバックアップリポジトリーを作成します。これには、デフォルトのパスワードまたは置き換えたパスワードが含まれます。最初のバックアップの **後** にシークレットパスワードを更新すると、新しいパスワードが **velero-repo-credentials** のパスワードと一致しなくなり、Velero は古いバックアップに接続できなくなります。

手順

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

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

(1) オプション: **kubevirt** プラグインは OpenShift Virtualization で使用されます。

(2) 必要に応じて、バックアッププロバイダーのデフォルトのプラグイン (**gcp** など) を指定します。

(3) CSI スナップショットを使用して PV をバックアップする場合は、**csi** のデフォルトプラグインを指定します。**csi** プラグインは、Velero CSI ベータスナップショット API を使用します。スナップショットの場所を設定する必要はありません。

(4) **openshift** プラグインは必須です。

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

(6) Restic インストールを無効にする場合は、この値を **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、Restic Pod が各動作ノードで実行していることを意味します。OADP バージョン 1.2 以降では、**spec.defaultVolumesToFsBackup: true** を

Backup CR に追加することで、バックアップ用に Restic を設定できます。OADP バージョン 1.1 では、**spec.defaultVolumesToRestic: true** を **Backup** CR に追加します。

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

8 バックアッププロバイダーを指定します。

9 バックアッププロバイダーにデフォルトのプラグインを使用する場合は、**Secret** の正しいデフォルト名を指定します (例: **cloud-credentials-gcp**)。カスタム名を指定すると、そのカスタム名がバックアップの場所に使用されます。**Secret** 名を指定しない場合は、デフォルトの名前が使用されます。

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

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

4. **Create** をクリックします。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

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

出力例

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

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.7.4. Data Protection Application 1.3 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る。
- 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る。
- 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る。
- 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る。

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 (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - kubevirt (2)
        - gcp (3)
        - csi (4)
        - openshift (5)
      resourceTimeout: 10m (6)
    nodeAgent: (7)
      enable: true (8)
      uploaderType: kopia (9)
      podConfig:
        nodeSelector: <node_selector> (10)
  backupLocations:
    - velero:
        provider: gcp (11)
        default: true
        credential:
          key: cloud
          name: <default_secret> (12)
        objectStorage:
          bucket: <bucket_name> (13)
          prefix: <prefix> (14)
```

(1) OADP のデフォルトの namespace は **openshift-adp** です。namespace は変数であり、設定可能です。

(2) オプション: **kubevirt** プラグインは OpenShift Virtualization で使用されます。

(3) 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。

(4) 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。

(5) **openshift** プラグインは必須です。

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

(7) 管理要求をサーバーにルーティングする管理エージェント。

(8)

nodeAgent を有効にしてファイルシステムバックアップを実行する場合は、この値を **true** に設定します。

9 アップローダーとして **kopia** または **restic** と入力します。インストール後に選択を変更することはできません。組み込み DataMover の場合は、Kopia を使用する必要があります。**nodeAgent** はデーモンセットをデプロイします。これは、**nodeAgent** Pod が各ワーキングノード上で実行されることを意味します。ファイルシステムバックアップを設定するには、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加します。

10 Kopia または Restic が使用可能なノードを指定します。デフォルトでは、Kopia または Restic はすべてのノードで実行されます。

11 バックアッププロバイダーを指定します。

12 バックアッププロバイダーにデフォルトのプラグインを使用する場合は、**Secret** の正しいデフォルト名を指定します (例: **cloud-credentials-gcp**)。カスタム名を指定すると、そのカスタム名がバックアップの場所に使用されます。**Secret** 名を指定しない場合は、デフォルトの名前が使用されます。

13 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。

14 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。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

```
$ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

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

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

NAME                                            READY   UP-TO-DATE   AVAILABLE   AGE
```

```
deployment.apps/oadp-operator-controller-manager    1/1     1          1          2m9s
deployment.apps/velero                              1/1     1          1          96s

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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

4.4.7.4.1. OpenShift Data Foundation での障害不復旧用のオブジェクトバケット要求の作成

OpenShift Data Foundation の Multicloud Object Gateway (MCG) バケット **backupStorageLocation** にクラスターストレージを使用する場合は、OpenShift Web コンソールを使用して Object Bucket Claim (OBC) を作成します。

警告

Object Bucket Claim (OBC) の設定に失敗すると、バックアップが利用できなくなる可能性があります。

注記

特に指定のない限り、"NooBaa" は軽量オブジェクトストレージを提供するオープンソースプロジェクトを指し、"Multicloud Object Gateway (MCG)" は NooBaa の Red Hat ディストリビューションを指します。

MCG の詳細は、アプリケーションを使用して Multicloud Object Gateway にアクセスする を参照してください。

手順

- OpenShift Web コンソールを使用した Object Bucket Claim の作成 に記載されているとおり、OpenShift Web コンソールを使用して Object Bucket Claim (OBC) を作成します。

4.4.7.4.2. DataProtectionApplication CR で CSI を有効にする

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

前提条件

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

手順

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

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

(1) **csi** デフォルトプラグインを追加します。

関連情報

- **kubevirt** および **openshift** プラグインを使用した Data Protection Application のインストール

4.4.8. OpenShift Virtualization を使用した OpenShift API for Data Protection の設定

OADP Operator をインストールし、バックアップの場所を設定することで、OpenShift Virtualization を使用した OpenShift API for Data Protection (OADP) をインストールできます。その後、Data Protection Application をインストールできます。

OpenShift API for Data Protection を使用して仮想マシンをバックアップおよび復元します。

注記

OpenShift Virtualization を使用した OpenShift API for Data Protection は、バックアップおよび復元のストレージオプションとして次のものをサポートしています。

- Container Storage Interface (CSI) バックアップ
- DataMover による Container Storage Interface (CSI) バックアップ

次のストレージオプションは対象外です。

- ファイルシステムのバックアップと復元
- ボリュームスナップショットのバックアップと復元

詳細は、ファイルシステムバックアップを使用してアプリケーションをバックアップする: Kopia または Restic 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 OperatorHub 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。

4.4.8.1. OpenShift Virtualization を使用した OADP のインストールと設定

クラスター管理者は、OADP Operator をインストールして OADP をインストールします。

OADP Operator の最新バージョンは、Velero 1.12 をインストールします。

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。

手順

1. ストレージプロバイダーの指示に従って、OADP Operator をインストールします。
2. **kubevirt** および **openshift** OADP プラグインを使用して Data Protection Application (DPA) をインストールします。
3. **Backup** カスタムリソース (CR) を作成して、仮想マシンをバックアップします。

警告

Red Hat のサポート対象は、次のオプションに限られています。

- CSI バックアップ
- DataMover による CSI バックアップ

Restore CR を作成して **Backup** CR を復元します。

関連情報

- OADP プラグイン
- **Backup** カスタムリソース (CR)
- **Restore** CR
- ネットワークが制限された環境での Operator Lifecycle Manager の使用

4.4.8.2. Data Protection Application 1.3 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る。
- 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る。
- 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る。
- 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る。
- 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、以下のように 2 つの **Secrets** を作成する必要がある。
  - バックアップの場所用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用の別のカスタム名を持つ **Secret**。この **Secret** を **DataProtectionApplication** CR に追加します。

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 (1)
spec:
  configuration:
    velero:
      defaultPlugins:
        - kubevirt (2)
        - gcp (3)
        - csi (4)
        - openshift (5)
      resourceTimeout: 10m (6)
    nodeAgent: (7)
      enable: true (8)
      uploaderType: kopia (9)
      podConfig:
        nodeSelector: <node_selector> (10)
  backupLocations:
    - velero:
        provider: gcp (11)
        default: true
        credential:
          key: cloud
          name: <default_secret> (12)
        objectStorage:
          bucket: <bucket_name> (13)
          prefix: <prefix> (14)
```

(1) OADP のデフォルトの namespace は **openshift-adp** です。namespace は変数であり、設定可能です。

(2) **kubevirt** プラグインは OpenShift Virtualization に必須です。

(3) バックアッププロバイダーのプラグインがある場合には、それを指定します (例: **gcp**)。

(4) CSI 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、**csi** プラグインが必須です。**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。

(5) **openshift** プラグインは必須です。

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

(7) 管理要求をサーバーにルーティングする管理エージェント。

(8) **nodeAgent** を有効にしてファイルシステムバックアップを実行する場合は、この値を **true** に設定します。

(9) 組み込み DataMover を使用するには、アップローダーとして **kopia** と入力します。**nodeAgent** はデーモンセットをデプロイします。これは、**nodeAgent** Pod が各ワーキングノード上で実行されることを意味します。ファイルシステムバックアップを設定するには、**spec.defaultVolumesToFsBackup: true** を **Backup** CR に追加します。

10 Kopia が利用可能なノードを指定します。デフォルトでは、Kopia はすべてのノードで実行されます。

11 バックアッププロバイダーを指定します。

12 バックアッププロバイダーにデフォルトのプラグインを使用する場合は、**Secret** の正しいデフォルト名を指定します (例: **cloud-credentials-gcp**)。カスタム名を指定すると、そのカスタム名がバックアップの場所に使用されます。**Secret** 名を指定しない場合は、デフォルトの名前が使用されます。

13 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。

14 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。

検証

1. 次のコマンドを実行して OpenShift API for Data Protection (OADP) リソースを表示し、インストールを検証します。

```
$ 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/node-agent-9cq4q                                 1/1     Running   0          94s
pod/node-agent-m4lts                                 1/1     Running   0          94s
pod/node-agent-pv4kr                                 1/1     Running   0          95s
pod/velero-588db7f655-n842v                          1/1     Running   0          95s

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

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

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

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

2. 次のコマンドを実行して、**DataProtectionApplication** (DPA) が調整されていることを確認します。

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

出力例

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

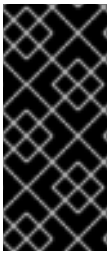
3. **type** が **Reconciled** に設定されていることを確認します。

4. 次のコマンドを実行して、バックアップの保存場所を確認し、**PHASE** が **Available** であることを確認します。

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

出力例

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



重要

Red Hat は、OADP バージョン 1.3.0 以降と OpenShift Virtualization バージョン 4.14 以降の組み合わせのみをサポートします。

バージョン 1.3.0 より前の OADP は、OpenShift Virtualization のバックアップと復元ではサポートされていません。

4.5. OADP のアンインストール

4.5.1. 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 Operator の削除 を参照してください。

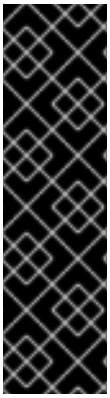
4.6. OADP のバックアップ

4.6.1. アプリケーションのバックアップ

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

- **Backup** CR は、Kubernetes リソースと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に作成します。
- クラウドプロバイダーがネイティブスナップショット API を備えている場合、または CSI スナップショットをサポートしている場合、**Backup** CR はスナップショットを作成することによって永続ボリューム (PV) をバックアップします。CSI スナップショットの操作の詳細は、CSI スナップショットを使用した永続ボリュームのバックアップ を参照してください。

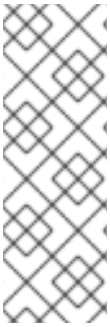
CSI ボリュームスナップショットの詳細は、CSI ボリュームスナップショット を参照してください。https://docs.redhat.com/en/documentation/openshift_container_platform/4.16/html-single/storage/#persistent-storage-csi-snapshots



重要

オブジェクトストレージのバケット作成を自動化する **CloudStorage** API は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

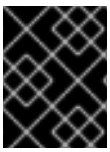


注記

CloudStorage API は、**CloudStorage** オブジェクトを使用しており、OADP で **CloudStorage** API を使用して **BackupStorageLocation** として使用する S3 バケットを自動的に作成するためのテクノロジープレビュー機能です。

CloudStorage API は、既存の S3 バケットを指定して **BackupStorageLocation** オブジェクトを手動作成することをサポートしています。現在、S3 バケットを自動的に作成する **CloudStorage** API は、AWS S3 ストレージに対してのみ有効です。

- 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、Kopia または Restic を使用してバックアップを作成できます。ファイルシステムバックアップを使用してアプリケーションをバックアップする: Kopia または Restic を参照してください。



重要

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

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

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

4.6.1.1. 既知の問題

OpenShift Container Platform 4.16 は、Restic 復元プロセス中に Pod の readiness を妨げる可能性がある Pod Security Admission (PSA) ポリシーを強制します。

この問題は OADP 1.1.6 および OADP 1.2.2 リリースで解決されており、これらのリリースにアップグレードすることが推奨されます。

詳細は、PSA ポリシーの変更により、OCP 4.15 で部分的に Restic 復元が失敗する を参照してください。

関連情報

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

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

手順

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

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

出力例

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

2. 次の例のように、**Backup** CR を作成します。

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  labels:
    velero.io/storage-location: default
  namespace: openshift-adp
spec:
  hooks: {}
  includedNamespaces:
  - <namespace> 1
```

```
includedResources: [] 2
excludedResources: [] 3
storageLocation: <velero-sample-1> 4
ttl: 720h0m0s
labelSelector: 5
  matchLabels:
    app: <label_1>
    app: <label_2>
    app: <label_3>
orLabelSelectors: 6
- matchLabels:
    app: <label_1>
    app: <label_2>
    app: <label_3>
```

1. バックアップする namespace の配列を指定します。
2. オプション: バックアップに含めるリソースの配列を指定します。リソースは、ショートカット (Pods は po など) または完全修飾の場合があります。指定しない場合、すべてのリソースが含まれます。
3. オプション: バックアップから除外するリソースの配列を指定します。リソースは、ショートカット (Pods は po など) または完全修飾の場合があります。
4. **backupStorageLocations** CR の名前を指定します。
5. 指定したラベルを **すべて** 持つバックアップリソースの {key,value} ペアのマップ。
6. 指定したラベルを **1つ以上** 持つバックアップリソースの {key,value} ペアのマップ。

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

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

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

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

詳細は、バックアップ CR の作成 を参照してください。

前提条件

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

手順

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

  設定ファイルのサンプル

```
apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshotClass
metadata:
  name: <volume_snapshot_class_name>
  labels:
    velero.io/csi-volumesnapshot-class: "true" (1)
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: true (2)
driver: <csi_driver>
deletionPolicy: <deletion_policy_type> (3)
```

1. **true** に設定する必要があります。
2. **true** に設定する必要があります。
3. OADP は、CSI および Data Mover のバックアップと復元に対して、**Retain** および **Delete** 削除ポリシータイプをサポートしています。OADP 1.2 Data Mover の場合、削除ポリシータイプを **Retain** に設定します。

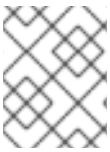
次のステップ

- これで、**Backup** CR を作成できます。

4.6.4. ファイルシステムバックアップを使用してアプリケーションをバックアップする: Kopia または Restic

OADP を使用して、Pod にアタッチされている Kubernetes ボリュームを、そのボリュームのファイルシステムからバックアップおよび復元できます。このプロセスは、File System Backup (FSB) または Pod Volume Backup (PVB) と呼ばれます。これは、オープンソースのバックアップツール Restic または Kopia のモジュールを使用して実行できます。

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、FSB を使用してバックアップを作成できます。



注記

Restic は、デフォルトで OADP Operator によってインストールされます。必要に応じて、代わりに Kopia をインストールすることもできます。

FSB と OADP の統合により、ほぼすべてのタイプの Kubernetes ボリュームをバックアップおよび復元するためのソリューションが提供されます。この統合は OADP の追加機能であり、既存の機能を置き換えるものではありません。

Backup カスタムリソース (CR) を編集して、Kopia または Restic で 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。

注記

OADP バージョン 1.3 以降では、アプリケーションのバックアップに Kopia または Restic を使用できます。

ビルトイン DataMover の場合は、Kopia を使用する必要があります。

OADP バージョン 1.2 以前の場合、アプリケーションのバックアップには Restic のみ使用できます。

重要

FSB は、**hostPath** ボリュームのバックアップをサポートしません。詳細は、FSB の制限事項 を参照してください。

前提条件

- OpenShift API for Data Protection (OADP) Operator をインストールしている。
- **DataProtectionApplication** CR で **spec.configuration.nodeAgent.enable** を **false** に設定して、デフォルトの **nodeAgent** インストールを無効にしていない。
- **DataProtectionApplication** CR で **spec.configuration.nodeAgent.uploaderType** を **kopia** または **restic** に設定して、Kopia または Restic をアップローダーとして選択している。
- **DataProtectionApplication** CR が **Ready** 状態である。

手順

- 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:
  defaultVolumesToFsBackup: true 1
...
```

1 OADP バージョン 1.2 以降では、**defaultVolumesToFsBackup: true** 設定を **spec** ブロック内に追加します。OADP バージョン 1.1 では、**defaultVolumesToRestic: true** を追加します。

4.6.5. バックアップフックの作成

バックアップを実行する際に、バックアップされる Pod に基づいて、Pod 内のコンテナーで実行するコマンドを 1 つ以上指定できます。

コマンドは、カスタムアクション処理の前 (**プリ** フック)、またはすべてのカスタムアクションが完了し、カスタムアクションで指定された追加アイテムがバックアップされた後 (**ポスト** フック) に実行するように設定できます。

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。

手順

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

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  name: <backup>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces: 2
        - <namespace>
        includedResources: []
        - pods 3
        excludedResources: [] 4
        labelSelector: 5
          matchLabels:
            app: velero
            component: server
        pre: 6
          - exec:
              container: <container> 7
              command:
              - /bin/uname 8
              - -a
              onError: Fail 9
              timeout: 30s 10
        post: 11
...
```

1. オプション: フックが適用される namespace を指定できます。この値が指定されていない場合、フックはすべてのネームスペースに適用されます。
2. オプション: フックが適用されない namespace を指定できます。
3. 現在、Pod は、フックを適用できる唯一のサポート対象リソースです。
4. オプション: フックが適用されないリソースを指定できます。
5. オプション: このフックは、ラベルに一致するオブジェクトにのみ適用されます。この値が指定されていない場合、フックはすべてのオブジェクトに適用されます。
6. バックアップの前に実行するフックの配列。
7. オプション: コンテナーが指定されていない場合、コマンドは Pod の最初のコンテナーで実行されます。

8 これは、追加される **init** コンテナーのエントリーポイントです。

9 エラー処理に許可される値は、**Fail** と **Continue** です。デフォルトは **Fail** です。

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

11 このブロックでは、バックアップ後に実行するフックの配列を、バックアップ前のフックと同じパラメーターで定義します。

4.6.6. スケジュール CR を使用したバックアップのスケジュール設定

スケジュール操作を使用すると、Cron 式で指定された特定の時間にデータのバックアップを作成できます。

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

警告

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

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

前提条件

- OpenShift API for Data Protection (OADP) Operator をインストールしている。
- **DataProtectionApplication** CR が **Ready** 状態である。

手順

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

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

出力例

```
NAMESPACE       NAME              PHASE       LAST VALIDATED   AGE   DEFAULT
openshift-adp   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
    defaultVolumesToFsBackup: 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。

注記

特定の間隔でバックアップをスケジュールするには、次の形式で **<duration_in_minutes>** を入力します。

```
schedule: "*/10 * * * *"
```

引用符 (**" "**) の間に分の値を入力します。

2. バックアップを作成する namespace の配列。
3. **backupStorageLocations** CR の名前。
4. オプション: OADP バージョン 1.2 以降では、Restic を使用してボリュームのバックアップを実行するときに、**defaultVolumesToFsBackup: true** キーと値のペアを設定に追加します。OADP バージョン 1.1 では、Restic でボリュームをバックアップするときに、**defaultVolumesToRestic: true** のキーと値のペアを追加します。

1. 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}'
```

4.6.7. バックアップの削除

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

警告

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

前提条件

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

手順

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

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

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

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

    ここでは、以下のようになります。

    <backup_CR_name>
    Backup カスタムリソースの名前。

    <velero_namespace>
    Backup カスタムリソースを含む namespace。

4.6.8. Kopia について

Kopia は、高速かつセキュアなオープンソースのバックアップおよび復元ツールです。これを使用して、データの暗号化されたスナップショットを作成し、そのスナップショットを選択したリモートストレージまたはクラウドストレージに保存できます。

Kopia は、ネットワークおよびローカルストレージの場所、および多くのクラウドまたはリモートストレージの場所をサポートしています。以下はその一部です。

- Amazon S3 および S3 と互換性のあるクラウドストレージ
- Azure Blob Storage
- Google Cloud Storage プラットフォーム

Kopia は、スナップショットにコンテンツアドレスを指定できるストレージを使用します。

- スナップショットは常に増分されます。すでに以前のスナップショットに含まれているデータは、リポジトリーに再アップロードされません。リポジトリーに再度アップロードされるのは、ファイルが変更されたときだけです。
- 保存されたデータは重複排除されます。同じファイルのコピーが複数存在する場合、そのうちの 1 つだけが保存されます。

- ファイルが移動された場合、またはファイルの名前が変更された場合、Kopia はそれらが同じコンテンツであることを認識し、それらを再度アップロードしません。

4.6.8.1. OADP と Kopia の統合

OADP 1.3 は、Pod ボリュームバックアップのバックアップメカニズムとして、Restic に加えて Kopia をサポートします。インストール時に、**DataProtectionApplication** カスタムリソース (CR) の **uploaderType** フィールドを設定して、どちらかを選択する必要があります。使用できる値は、**restic** または **kopia** です。**uploaderType** を指定しない場合、OADP 1.3 はデフォルトで Kopia をバックアップメカニズムとして使用します。データは統合リポジトリーに書き込まれ、統合リポジトリーから読み取られます。

次の例は、Kopia を使用するように設定された **DataProtectionApplication** CR を示しています。

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    nodeAgent:
      enable: true
      uploaderType: kopia
# ...
```

4.7. OADP の復元

4.7.1. アプリケーションの復元

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

Restore CR を編集することで、Pod 内のコンテナーでコマンドを実行するための復元フックを作成できます。復元フックの作成 を参照してください。

4.7.1.1. 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 をインストールしている。
- **DataProtectionApplication** CR が **Ready** 状態である。
- Velero **Backup** CR がある。
- 永続ボリューム (PV) の容量は、バックアップ時に要求されたサイズと一致する必要があります。必要に応じて、要求されたサイズを調整します。

手順

1. 次の例のように、**Restore** CR を作成します。

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  backupName: <backup> 1
  includedResources: [] 2
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  restorePVs: true 3
```

1. **Backup** CR の名前
2. オプション: 復元プロセスに含めるリソースの配列を指定します。リソースは、ショートカット (**Pods** は **po** など) または完全修飾の場合があります。指定しない場合、すべてのリソースが含まれます。
3. オプション: **restorePVs** パラメーターを **false** に設定すると、コンテナーストレージインターフェイス (CSI) スナップショットの **VolumeSnapshot** から、または **VolumeSnaphshotLocation** が設定されている場合はネイティブスナップショットからの **PersistentVolumes** の復元をオフにすることができます。

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

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

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

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

1. バックアップした namespace。

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

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

注記

復元プロセス中に、OADP Velero プラグインは **DeploymentConfig** オブジェクトをスケールダウンし、Pod をスタンドアロン Pod として復元します。これは、クラスターが復元された **DeploymentConfig** Pod を復元時にすぐに削除することを防ぎ、復元フックと復元後のフックが復元された Pod 上でアクションを完了できるようにするために行われます。以下に示すクリーンアップスクリプトは、これらの切断された Pod を削除し、**DeploymentConfig** オブジェクトを適切な数のレプリカにスケールアップします。

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

```
#!/bin/bash
set -e

# if sha256sum exists, use it to check the integrity of the file
if command -v sha256sum >/dev/null 2>&1; then
  CHECKSUM_CMD="sha256sum"
else
  CHECKSUM_CMD="shasum -a 256"
fi

label_name () {
    if [ "${#1}" -le "63" ]; then
 echo $1
 return
    fi
    sha=$(echo -n $1|$CHECKSUM_CMD)
    echo "${1:0:57}${sha:0:6}"
}

OADP_NAMESPACE=${OADP_NAMESPACE:=openshift-adp}

if [[ $# -ne 1 ]]; then
    echo "usage: ${BASH_SOURCE} restore-name"
    exit 1
fi

echo using OADP Namespace $OADP_NAMESPACE
echo restore: $1

label=$(label_name $1)
echo label: $label

echo Deleting disconnected restore pods
oc delete pods -l oadp.openshift.io/disconnected-from-dc=$label

for dc in $(oc get dc --all-namespaces -l oadp.openshift.io/replicas-modified=$label -o
jsonpath='{range .items[*]}{.metadata.namespace}{","}{.metadata.name}{","}
{.metadata.annotations.oadp\.openshift\.io/original-replicas}{","}
{.metadata.annotations.oadp\.openshift\.io/original-paused}{"\n"}')
do
    IFS=',' read -ra dc_arr <<< "$dc"
    if [ ${#dc_arr[0]} -gt 0 ]; then
 echo Found deployment ${dc_arr[0]}/${dc_arr[1]}, setting replicas: ${dc_arr[2]}, paused:
```

```
${dc_arr[3]}
 cat <<EOF | oc patch dc  -n ${dc_arr[0]} ${dc_arr[1]} --patch-file /dev/stdin
spec:
  replicas: ${dc_arr[2]}
  paused: ${dc_arr[3]}
EOF
    fi
done
```

4.7.1.2. 復元フックの作成

Restore 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復元フックを作成します。

2 種類の復元フックを作成できます。

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

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

手順

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

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore>
  namespace: openshift-adp
spec:
  hooks:
    resources:
      - name: <hook_name>
        includedNamespaces:
        - <namespace> 1
        excludedNamespaces:
        - <namespace>
        includedResources:
        - pods 2
        excludedResources: []
        labelSelector: 3
          matchLabels:
            app: velero
            component: server
        postHooks:
        - init:
            initContainers:
            - name: restore-hook-init
              image: alpine:latest
              volumeMounts:
              - mountPath: /restores/pvc1-vm
```

```
        name: pvc1-vm
      command:
      - /bin/ash
      - -c
    timeout: 4
  - exec:
    container: <container> 5
    command:
    - /bin/bash 6
    - -c
    - "psql < /backup/backup.sql"
    waitTimeout: 5m 7
    execTimeout: 1m 8
    onError: Continue 9
```

1. オプション: フックが適用される namespace の配列。この値が指定されていない場合、フックはすべてのネームスペースに適用されます。
2. 現在、Pod は、フックを適用できる唯一のサポート対象リソースです。
3. オプション: このフックは、ラベルセレクターに一致するオブジェクトにのみ適用されます。
4. オプション: Timeout は、**initContainers** が完了するまで Velero が待機する最大時間を指定します。
5. オプション: コンテナーが指定されていない場合、コマンドは Pod の最初のコンテナーで実行されます。
6. これは、追加される init コンテナーのエントリーポイントです。
7. オプション: コンテナーの準備が整うまでの待機時間。これは、コンテナーが起動して同じコンテナー内の先行するフックが完了するのに十分な長さである必要があります。設定されていない場合、復元プロセスの待機時間は無期限になります。
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. 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。

4.8. OADP と ROSA

4.8.1. OADP を使用して ROSA クラスター上のアプリケーションをバックアップする

Red Hat OpenShift Service on AWS (ROSA) クラスターで OpenShift API for Data Protection (OADP) を使用して、アプリケーションデータをバックアップおよび復元できます。

ROSA は、フルマネージドのターンキーアプリケーションプラットフォームであり、アプリケーションを構築してデプロイすることにより、お客様に価値を提供することに集中できます。

ROSA は、幅広い Amazon Web Services (AWS) コンピュート、データベース、分析、機械学習、ネットワーク、モバイル、およびその他のサービスとのシームレスな統合を提供し、差別化されたエクスペリエンスの構築とお客様への提供をさらに高速化します。

AWS アカウントから直接サービスをサブスクライブできます。

クラスターを作成した後、OpenShift Container Platform Web コンソールを使用して、または Red Hat OpenShift Cluster Manager を介してクラスターを操作できます。ROSA では、OpenShift API やコマンドラインインターフェイス (CLI) ツールも使用できます。

ROSA のインストールの詳細は、Red Hat OpenShift Service on AWS (ROSA) のインストールのインタラクティブな説明 を参照してください。

OpenShift API for Data Protection (OADP) をインストールする前に、OADP が Amazon Web Services API を使用できるように、OADP のロールとポリシーの認証情報を設定する必要があります。

このプロセスは次の 2 段階で実行されます。

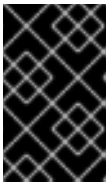
1. AWS 認証情報を準備します。
2. OADP Operator をインストールし、IAM ロールを付与します。

4.8.1.1. OADP 用の AWS 認証情報を準備する

Amazon Web Services アカウントは、OpenShift API for Data Protection (OADP) インストールを受け入れるように準備および設定する必要があります。

手順

1. 次のコマンドを実行して、以下の環境変数を作成します。



重要

ROSA クラスターに一致するようにクラスター名を変更し、管理者としてクラスターにログインしていることを確認します。続行する前に、すべてのフィールドが正しく出力されていることを確認します。

```
$ export CLUSTER_NAME=my-cluster (1)
  export ROSA_CLUSTER_ID=$(rosa describe cluster -c ${CLUSTER_NAME} --output json |
jq -r .id)
  export REGION=$(rosa describe cluster -c ${CLUSTER_NAME} --output json | jq -r
.region.id)
  export OIDC_ENDPOINT=$(oc get authentication.config.openshift.io cluster -o
jsonpath='{.spec.serviceAccountIssuer}' | sed 's|^https://||')
  export AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)
  export CLUSTER_VERSION=$(rosa describe cluster -c ${CLUSTER_NAME} -o json | jq -r
.version.raw_id | cut -f -2 -d '.')
  export ROLE_NAME="${CLUSTER_NAME}-openshift-oadp-aws-cloud-credentials"
  export SCRATCH="/tmp/${CLUSTER_NAME}/oadp"
  mkdir -p ${SCRATCH}
  echo "Cluster ID: ${ROSA_CLUSTER_ID}, Region: ${REGION}, OIDC Endpoint:
  ${OIDC_ENDPOINT}, AWS Account ID: ${AWS_ACCOUNT_ID}"
```

(1) **my-cluster** は、ROSA クラスター名に置き換えます。

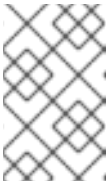
2. AWS アカウントで、AWS S3 へのアクセスを許可する IAM ポリシーを作成します。

   a. 以下のコマンドを実行して、ポリシーが存在するかどうかを確認します。

```
$ POLICY_ARN=$(aws iam list-policies --query "Policies[?
PolicyName=='RosaOadpVer1'].{ARN:Arn}" --output text) 1
```

   1 **RosaOadp** は、実際のポリシー名に置き換えます。

   b. 以下のコマンドを入力してポリシー JSON ファイルを作成し、ROSA でポリシーを作成します。



注記

ポリシー ARN が見つからない場合、コマンドはポリシーを作成します。ポリシー ARN がすでに存在する場合、**if** ステートメントはポリシーの作成を意図的にスキップします。

```
$ if [[ -z "${POLICY_ARN}" ]]; then
  cat << EOF > ${SCRATCH}/policy.json 1
  {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:CreateBucket",
        "s3:DeleteBucket",
        "s3:PutBucketTagging",
        "s3:GetBucketTagging",
        "s3:PutEncryptionConfiguration",
        "s3:GetEncryptionConfiguration",
        "s3:PutLifecycleConfiguration",
        "s3:GetLifecycleConfiguration",
        "s3:GetBucketLocation",
        "s3:ListBucket",
        "s3:GetObject",
        "s3:PutObject",
        "s3:DeleteObject",
        "s3:ListBucketMultipartUploads",
        "s3:AbortMultipartUploads",
        "s3:ListMultipartUploadParts",
        "s3:DescribeSnapshots",
        "ec2:DescribeVolumes",
        "ec2:DescribeVolumeAttribute",
        "ec2:DescribeVolumesModifications",
        "ec2:DescribeVolumeStatus",
        "ec2:CreateTags",
        "ec2:CreateVolume",
        "ec2:CreateSnapshot",
        "ec2:DeleteSnapshot"
      ],
      "Resource": "*"
    }
```

```
  ]}
EOF

  POLICY_ARN=$(aws iam create-policy --policy-name "RosaOadpVer1" \
  --policy-document file:///${SCRATCH}/policy.json --query Policy.Arn \
  --tags Key=rosa_openshift_version,Value=${CLUSTER_VERSION}
Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-oadp Key=operator_name,Value=openshift-
oadp \
  --output text)
  fi
```

1 **SCRATCH** は、環境変数用に作成された一時ディレクトリーの名前です。

c. 以下のコマンドを実行してポリシー ARN を表示します。

```
$ echo ${POLICY_ARN}
```

3. クラスターの IAM ロール信頼ポリシーを作成します。

a. 次のコマンドを実行して、信頼ポリシーファイルを作成します。

```
$ cat <<EOF > ${SCRATCH}/trust-policy.json
 {
    "Version":2012-10-17",
    "Statement": [{
      "Effect": "Allow",
      "Principal": {
        "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-
provider/${OIDC_ENDPOINT}"
      },
      "Action": "sts:AssumeRoleWithWebIdentity",
      "Condition": {
        "StringEquals": {
          "${OIDC_ENDPOINT}:sub": [
            "system:serviceaccount:openshift-adp:openshift-adp-controller-manager",
            "system:serviceaccount:openshift-adp:velero"]
        }
      }
    }]
 }
EOF
```

b. 以下のコマンドを実行してロールを作成します。

```
$ ROLE_ARN=$(aws iam create-role --role-name \
  "${ROLE_NAME}" \
  --assume-role-policy-document file://${SCRATCH}/trust-policy.json \
--tags Key=rosa_cluster_id,Value=${ROSA_CLUSTER_ID}
Key=rosa_openshift_version,Value=${CLUSTER_VERSION}
Key=rosa_role_prefix,Value=ManagedOpenShift
Key=operator_namespace,Value=openshift-adp Key=operator_name,Value=openshift-
oadp \
   --query Role.Arn --output text)
```

c. 次のコマンドを実行して、ロール ARN を表示します。

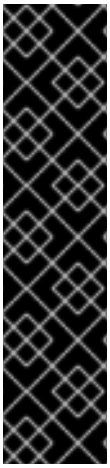
```
$ echo ${ROLE_ARN}
```

4. 次のコマンドを実行して、IAM ポリシーを IAM ロールにアタッチします。

```
$ aws iam attach-role-policy --role-name "${ROLE_NAME}" \
  --policy-arn ${POLICY_ARN}
```

4.8.1.2. OADP Operator のインストールおよび IAM ロールの指定

AWS Security Token Service (AWS STS) は、IAM またはフェデレーションされたユーザーの短期認証情報を提供するグローバル Web サービスです。STS を使用した OpenShift Container Platform (ROSA) は、ROSA クラスターに推奨される認証情報モードです。このドキュメントでは、AWS STS を使用する (ROSA) に OpenShift API for Data Protection (OADP) をインストールする方法を説明します。



重要

Restic と Kopia は、AWS STS を使用する ROSA 環境の OADP ではサポートされていません。Restic および Kopia のノードエージェントが無効になっていることを確認してください。ボリュームをバックアップする場合、AWS STS を使用する ROSA の OADP は、ネイティブスナップショットと Container Storage Interface (CSI) スナップショットのみをサポートします。

STS 認証を使用する Amazon ROSA クラスターでは、別の AWS リージョンでのバックアップデータの復元はサポートされていません。

Data Mover 機能は現在、ROSA クラスターではサポートされていません。データの移動にはネイティブ AWS S3 ツールを使用できます。

前提条件

- 必要なアクセス権とトークンを備えた OpenShift Container Platform ROSA クラスター。詳細は、前の手順である **OADP 用の AWS 認証情報の準備** を参照してください。バックアップと復元に 2 つの異なるクラスターを使用する予定の場合は、**ROLE_ARN** を含む AWS 認証情報をクラスターごとに準備する必要があります。

手順

1. 次のコマンドを入力して、AWS トークンファイルから OpenShift Container Platform シークレットを作成します。

   a. 認証情報ファイルを作成します。

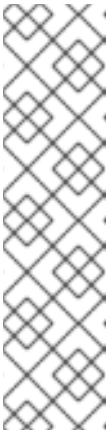
```
$ cat <<EOF > ${SCRATCH}/credentials
  [default]
  role_arn = ${ROLE_ARN}
  web_identity_token_file = /var/run/secrets/openshift/serviceaccount/token
EOF
```

   b. OADP の namespace を作成します。

```
$ oc create namespace openshift-adp
```

c. OpenShift Container Platform シークレットを作成します。

```
$ oc -n openshift-adp create secret generic cloud-credentials \
  --from-file=${SCRATCH}/credentials
```



注記

OpenShift Container Platform バージョン 4.14 以降では、OADP Operator は Operator Lifecycle Manager (OLM) および Cloud Credentials Operator (CCO) を通じて、標準化された新しい STS ワークフローをサポートします。このワークフローでは、上記シークレットの作成は必要ありません。OpenShift Container Platform Web コンソールを使用して、OLM で管理される Operator のインストール中にロール ARN のみ指定する必要があります。詳細は、**Web コンソールを使用して OperatorHub からインストールする** を参照してください。

前述のシークレットは CCO によって自動的に作成されます。

2. OADP Operator をインストールします。

   a. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** ページを表示します。

   b. **OADP Operator** を検索します。

   c. **role_ARN** フィールドに、前に作成した role_arn を貼り付け、**Install** をクリックします。

3. 次のコマンドを入力し、AWS 認証情報を使用して AWS クラウドストレージを作成します。

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: CloudStorage
  metadata:
    name: ${CLUSTER_NAME}-oadp
    namespace: openshift-adp
  spec:
    creationSecret:
      key: credentials
      name: cloud-credentials
    enableSharedConfig: true
    name: ${CLUSTER_NAME}-oadp
    provider: aws
    region: $REGION
EOF
```

4. 次のコマンドを入力して、アプリケーションのストレージのデフォルトストレージクラスを確認します。

```
$ oc get pvc -n <namespace>
```

出力例

```
NAME     STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS   AGE
```

```
applog   Bound    pvc-351791ae-b6ab-4e8b-88a4-30f73caf5ef8   1Gi        RWO            gp3-
csi         4d19h
mysql    Bound    pvc-16b8e009-a20a-4379-accc-bc81fedd0621   1Gi        RWO            gp3-
csi         4d19h
```

5. 次のコマンドを実行してストレージクラスを取得します。

```
$ oc get storageclass
```

出力例

```
NAME                PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
gp2                 kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   true
4d21h
gp2-csi             ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3                 ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3-csi (default)   ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
```

注記

次のストレージクラスが機能します。

- gp3-csi
- gp2-csi
- gp3
- gp2

すべてのアプリケーション、またはバックアップされるアプリケーションが Container Storage Interface (CSI) で永続ボリューム (PV) を使用している場合は、OADP DPA 設定に CSI プラグインを含めることをお勧めします。

6. バックアップとボリュームスナップショットが保存されるストレージへの接続を設定するために、**DataProtectionApplication** リソースを作成します。

   a. CSI ボリュームのみを使用している場合は、次のコマンドを入力して Data Protection Application をデプロイします。

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true (1)
    features:
      dataMover:
```

```
      enable: false
  backupLocations:
  - bucket:
      cloudStorageRef:
        name: ${CLUSTER_NAME}-oadp
      credential:
        key: credentials
        name: cloud-credentials
      prefix: velero
      default: true
      config:
        region: ${REGION}
  configuration:
    velero:
      defaultPlugins:
      - openshift
      - aws
      - csi
    restic:
      enable: false
EOF
```

1 ROSA は内部イメージバックアップをサポートします。イメージのバックアップを使用しない場合は、このフィールドを **false** に設定します。

a. CSI ボリュームまたは非 CSI ボリュームを使用している場合は、次のコマンドを入力して Data Protection Application をデプロイします。

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true 1
    features:
      dataMover:
        enable: false
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
      velero:
        defaultPlugins:
        - openshift
        - aws
```

```
    nodeAgent: 2
      enable: false
      uploaderType: restic
  snapshotLocations:
    - velero:
        config:
          credentialsFile: /tmp/credentials/openshift-adp/cloud-credentials-credentials 3
          enableSharedConfig: "true" 4
          profile: default 5
          region: ${REGION} 6
        provider: aws
EOF
```

1. ROSA は内部イメージバックアップをサポートします。イメージのバックアップを使用しない場合は、このフィールドを false に設定します。
2. **nodeAgent** 属性に関する重要な注記を参照してください。
3. **credentialsFile** フィールドは、Pod のバケット認証情報のマウント先です。
4. **enableSharedConfig** フィールドを使用すると、**snapshotLocations** がバケットに定義された認証情報を共有または再利用できます。
5. AWS 認証情報ファイルに設定されているプロファイル名を使用します。
6. **region** は、お使いの AWS リージョンに指定します。これはクラスターリージョンと同じである必要があります。

これで、**アプリケーションのバックアップ** で説明されているとおり、OpenShift Container Platform アプリケーションをバックアップおよび復元する準備が整いました。

重要

OADP は ROSA 環境で Restic をサポートしていないため、**restic** の **enable** パラメーターは **false** に設定されています。

OADP 1.2 を使用する場合は、次の設定を置き換えます。

```
nodeAgent:
  enable: false
  uploaderType: restic
```

次の設定に置き換えます。

```
restic:
  enable: false
```

バックアップと復元に 2 つの異なるクラスターを使用する場合、cloudstorage CR と OADP **DataProtectionApplication** 設定の両方で、2 つのクラスターの AWS S3 ストレージ名が同じである必要があります。

関連情報

- Web コンソールを使用した OperatorHub からのインストール
- アプリケーションのバックアップ

4.8.1.3. 例: オプションのクリーンアップを使用して OADP ROSA STS 上のワークロードをバックアップする

4.8.1.3.1. OADP と ROSA STS を使用したバックアップの実行

次の **hello-world** アプリケーションの例では、永続ボリューム (PV) が接続されていません。Red Hat OpenShift Service on AWS (ROSA) STS を使用して、OpenShift API for Data Protection (OADP) でバックアップを実行します。

どちらの Data Protection Application (DPA) 設定も機能します。

1. 次のコマンドを実行して、バックアップするワークロードを作成します。

```
$ oc create namespace hello-world
```

```
$ oc new-app -n hello-world --image=docker.io/openshift/hello-openshift
```

2. 次のコマンドを実行してルートを公開します。

```
$ oc expose service/hello-openshift -n hello-world
```

3. 次のコマンドを実行して、アプリケーションが動作していることを確認します。

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

出力例

```
Hello OpenShift!
```

4. 次のコマンドを実行して、ワークロードをバックアップします。

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Backup
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    includedNamespaces:
    - hello-world
    storageLocation: ${CLUSTER_NAME}-dpa-1
    ttl: 720h0m0s
EOF
```

5. バックアップが完了するまで待ってから、次のコマンドを実行します。

```
$ watch "oc -n openshift-adp get backup hello-world -o json | jq .status"
```

出力例

```
{
  "completionTimestamp": "2022-09-07T22:20:44Z",
  "expiration": "2022-10-07T22:20:22Z",
  "formatVersion": "1.1.0",
  "phase": "Completed",
  "progress": {
    "itemsBackedUp": 58,
    "totalItems": 58
  },
  "startTimestamp": "2022-09-07T22:20:22Z",
  "version": 1
}
```

6. 次のコマンドを実行して、デモワークロードを削除します。

```
$ oc delete ns hello-world
```

7. 次のコマンドを実行して、バックアップからワークロードを復元します。

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Restore
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    backupName: hello-world
EOF
```

8. 次のコマンドを実行して、復元が完了するまで待ちます。

```
$ watch "oc -n openshift-adp get restore hello-world -o json | jq .status"
```

出力例

```
{
  "completionTimestamp": "2022-09-07T22:25:47Z",
  "phase": "Completed",
  "progress": {
    "itemsRestored": 38,
    "totalItems": 38
  },
  "startTimestamp": "2022-09-07T22:25:28Z",
  "warnings": 9
}
```

9. 次のコマンドを実行して、ワークロードが復元されていることを確認します。

```
$ oc -n hello-world get pods
```

出力例

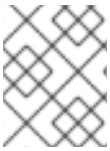
```
NAME                         READY   STATUS    RESTARTS   AGE
hello-openshift-9f885f7c6-kdjpj   1/1     Running   0          90s
```

10. 次のコマンドを実行して JSONPath を確認します。

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

出力例

```
Hello OpenShift!
```



注記

トラブルシューティングのヒントについては、OADP チームの トラブルシューティングドキュメント を参照してください。

4.8.1.3.2. OADP と ROSA STS を使用してバックアップ後のクラスターをクリーンアップする

この例のバックアップおよび S3 バケットと OpenShift API for Data Protection (OADP) Operator をアンインストールする必要がある場合は、次の手順を実行します。

手順

1. 次のコマンドを実行して、ワークロードを削除します。

```
$ oc delete ns hello-world
```

2. 次のコマンドを実行して、Data Protection Application (DPA) を削除します。

```
$ oc -n openshift-adp delete dpa ${CLUSTER_NAME}-dpa
```

3. 次のコマンドを実行して、クラウドストレージを削除します。

```
$ oc -n openshift-adp delete cloudstorage ${CLUSTER_NAME}-oadp
```

警告

このコマンドがハングした場合は、次のコマンドを実行してファイナライザーを削除する必要がある場合があります。

```
$ oc -n openshift-adp patch cloudstorage ${CLUSTER_NAME}-oadp -p
'{"metadata":{"finalizers":null}}' --type=merge
```

4. Operator が不要になった場合は、次のコマンドを実行して削除します。

```
$ oc -n openshift-adp delete subscription oadp-operator
```

5. Operator から namespace を削除します。

```
$ oc delete ns openshift-adp
```

6. バックアップおよび復元リソースが不要になった場合は、次のコマンドを実行してクラスターからリソースを削除します。

```
$ oc delete backup hello-world
```

7. AWS S3 のバックアップ、復元、およびリモートオブジェクトを削除するには、次のコマンドを実行します。

```
$ velero backup delete hello-world
```

8. カスタムリソース定義 (CRD) が不要になった場合は、次のコマンドを実行してクラスターから削除します。

```
$ for CRD in `oc get crds | grep velero | awk '{print $1}'`; do oc delete crd $CRD; done
```

9. 次のコマンドを実行して、AWS S3 バケットを削除します。

```
$ aws s3 rm s3://${CLUSTER_NAME}-oadp --recursive
```

```
$ aws s3api delete-bucket --bucket ${CLUSTER_NAME}-oadp
```

10. 次のコマンドを実行して、ロールからポリシーを切り離します。

```
$ aws iam detach-role-policy --role-name "${ROLE_NAME}"  --policy-arn "${POLICY_ARN}"
```

11. 以下のコマンドを実行してロールを削除します。

```
$ aws iam delete-role --role-name "${ROLE_NAME}"
```

4.9. OADP と AWS STS

4.9.1. OADP を使用して AWS STS クラスター上のアプリケーションをバックアップする

OADP Operator をインストールすることで、Amazon Web Services (AWS) を使用して OpenShift API for Data Protection (OADP) をインストールします。Operator は Velero 1.12 をインストールします。

注記

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

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

OADP は、AWS Security Token Service (STS) (AWS STS) クラスターに手動でインストールできます。Amazon AWS は、ユーザーのために権限が限られた一時的な認証情報を要求できる AWS STS を Web サービスとして提供しています。STS を使用すると、API 呼び出し、AWS コンソール、または AWS コマンドラインインターフェイス (CLI) を介してリソースへの一時的なアクセスを信頼できるユーザーに提供できます。

OpenShift API for Data Protection (OADP) をインストールする前に、OADP が Amazon Web Services API を使用できるように、OADP のロールとポリシーの認証情報を設定する必要があります。

このプロセスは次の 2 段階で実行されます。

1. AWS 認証情報を準備します。
2. OADP Operator をインストールし、IAM ロールを付与します。

4.9.1.1. OADP 用の AWS STS 認証情報を準備する

Amazon Web Services アカウントは、OpenShift API for Data Protection (OADP) インストールを受け入れるように準備および設定する必要があります。次の手順に従って AWS 認証情報を準備します。

手順

1. 次のコマンドを実行して、**cluster_name** 環境変数を定義します。

```
$ export CLUSTER_NAME= <AWS_cluster_name> 1
```

1 変数は任意の値に設定できます。

2. 次のコマンドを実行して、**AWS_ACCOUNT_ID, OIDC_ENDPOINT** などの **cluster** の詳細をすべて取得します。

```
$ export CLUSTER_VERSION=$(oc get clusterversion version -o
jsonpath='{.status.desired.version}{"\n"}')

export AWS_CLUSTER_ID=$(oc get clusterversion version -o jsonpath='{.spec.clusterID}
{"\n"}')

export OIDC_ENDPOINT=$(oc get authentication.config.openshift.io cluster -o
jsonpath='{.spec.serviceAccountIssuer}' | sed 's|^https://||')

export REGION=$(oc get infrastructures cluster -o
jsonpath='{.status.platformStatus.aws.region}' --allow-missing-template-keys=false || echo
us-east-2)

export AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query Account --output text)

export ROLE_NAME="${CLUSTER_NAME}-openshift-oadp-aws-cloud-credentials"
```

3. 次のコマンドを実行して、すべてのファイルを保存するための一時ディレクトリーを作成します。

```
$ export SCRATCH="/tmp/${CLUSTER_NAME}/oadp"
mkdir -p ${SCRATCH}
```

4. 次のコマンドを実行して、収集したすべての詳細を表示します。

```
$ echo "Cluster ID: ${AWS_CLUSTER_ID}, Region: ${REGION}, OIDC Endpoint:
${OIDC_ENDPOINT}, AWS Account ID: ${AWS_ACCOUNT_ID}"
```

5. AWS アカウントで、AWS S3 へのアクセスを許可する IAM ポリシーを作成します。

   a. 次のコマンドを実行して、ポリシーが存在するかどうかを確認します。

```
$ export POLICY_NAME="OadpVer1" 1
```

   1 変数は任意の値に設定できます。

```
$ POLICY_ARN=$(aws iam list-policies --query "Policies[?
PolicyName=='$POLICY_NAME'].{ARN:Arn}" --output text)
```

   b. 次のコマンドを入力してポリシー JSON ファイルを作成し、ポリシーを作成します。

注記

ポリシー ARN が見つからない場合、コマンドはポリシーを作成します。ポリシー ARN がすでに存在する場合、**if** ステートメントはポリシーの作成を意図的にスキップします。

```
$ if [[ -z "${POLICY_ARN}" ]]; then
cat << EOF > ${SCRATCH}/policy.json
{
"Version": "2012-10-17",
"Statement": [
 {
   "Effect": "Allow",
   "Action": [
     "s3:CreateBucket",
     "s3:DeleteBucket",
     "s3:PutBucketTagging",
     "s3:GetBucketTagging",
     "s3:PutEncryptionConfiguration",
     "s3:GetEncryptionConfiguration",
     "s3:PutLifecycleConfiguration",
     "s3:GetLifecycleConfiguration",
     "s3:GetBucketLocation",
     "s3:ListBucket",
     "s3:GetObject",
     "s3:PutObject",
     "s3:DeleteObject",
     "s3:ListBucketMultipartUploads",
     "s3:AbortMultipartUpload",
     "s3:ListMultipartUploadParts",
     "ec2:DescribeSnapshots",
     "ec2:DescribeVolumes",
```

```
      "ec2:DescribeVolumeAttribute",
      "ec2:DescribeVolumesModifications",
      "ec2:DescribeVolumeStatus",
      "ec2:CreateTags",
      "ec2:CreateVolume",
      "ec2:CreateSnapshot",
      "ec2:DeleteSnapshot"
    ],
    "Resource": "*"
  }
]}
EOF

POLICY_ARN=$(aws iam create-policy --policy-name $POLICY_NAME \
--policy-document file:///${SCRATCH}/policy.json --query Policy.Arn \
--tags Key=openshift_version,Value=${CLUSTER_VERSION}
Key=operator_namespace,Value=openshift-adp Key=operator_name,Value=oadp \
--output text) 1
fi
```

1 **SCRATCH** は、ファイルを保存するために作成した一時ディレクトリーの名前です。

c. 以下のコマンドを実行してポリシー ARN を表示します。

```
$ echo ${POLICY_ARN}
```

6. クラスターの IAM ロール信頼ポリシーを作成します。

a. 次のコマンドを実行して、信頼ポリシーファイルを作成します。

```
$ cat <<EOF > ${SCRATCH}/trust-policy.json
{
   "Version": "2012-10-17",
   "Statement": [{
     "Effect": "Allow",
     "Principal": {
       "Federated": "arn:aws:iam::${AWS_ACCOUNT_ID}:oidc-
provider/${OIDC_ENDPOINT}"
     },
     "Action": "sts:AssumeRoleWithWebIdentity",
     "Condition": {
       "StringEquals": {
         "${OIDC_ENDPOINT}:sub": [
           "system:serviceaccount:openshift-adp:openshift-adp-controller-manager",
           "system:serviceaccount:openshift-adp:velero"]
       }
     }
   }]
}
EOF
```

b. 次のコマンドを実行して、クラスターの IAM ロール信頼ポリシーを作成します。

```
$ ROLE_ARN=$(aws iam create-role --role-name \
```

```
    "${ROLE_NAME}" \
    --assume-role-policy-document file://${SCRATCH}/trust-policy.json \
    --tags Key=cluster_id,Value=${AWS_CLUSTER_ID}
  Key=openshift_version,Value=${CLUSTER_VERSION}
  Key=operator_namespace,Value=openshift-adp Key=operator_name,Value=oadp --
  query Role.Arn --output text)
```

c. 次のコマンドを実行して、ロール ARN を表示します。

```
$ echo ${ROLE_ARN}
```

7. 次のコマンドを実行して、IAM ポリシーを IAM ロールにアタッチします。

```
$ aws iam attach-role-policy --role-name "${ROLE_NAME}" --policy-arn ${POLICY_ARN}
```

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

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

前提条件

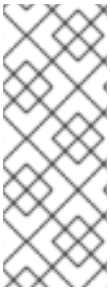
- OpenShift API for Data Protection (OADP) Operator がインストールされている必要があります。

手順

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

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

1. Velero podSpec に提供されるノードセレクターを指定します。
2. リストされている **resourceAllocations** は、平均使用量です。



注記

Kopia は OADP 1.3 以降のリリースで選択できます。Kopia はファイルシステムのバックアップに使用できます。組み込みの Data Mover を使用する Data Mover の場合は、Kopia が唯一の選択肢になります。

Kopia は Restic よりも多くのリソースを消費するため、それに応じて CPU とメモリーの要件を調整しなければならない場合があります。

4.9.1.2. OADP Operator のインストールおよび IAM ロールの指定

AWS Security Token Service (AWS STS) は、IAM またはフェデレーションされたユーザーの短期認証情報を提供するグローバル Web サービスです。このドキュメントでは、OpenShift API for Data Protection (OADP) を AWS STS クラスターに手動でインストールする方法を説明します。

重要

Restic と Kopia は、OADP AWS STS 環境ではサポートされていません。Restic および Kopia のノードエージェントが無効になっていることを確認してください。ボリュームをバックアップする場合、AWS STS 上の OADP は、ネイティブスナップショットと Container Storage Interface (CSI) スナップショットのみをサポートします。

STS 認証を使用する AWS クラスターでは、バックアップデータを別の AWS リージョンに復元することはサポートされていません。

Data Mover 機能は現在、AWS STS クラスターではサポートされていません。データの移動にはネイティブ AWS S3 ツールを使用できます。

前提条件

- 必要なアクセス権とトークンを備えた OpenShift Container Platform AWS STS クラスター。詳細は、前の手順である **OADP 用の AWS 認証情報の準備** を参照してください。バックアップと復元に 2 つの異なるクラスターを使用する予定の場合は、**ROLE_ARN** を含む AWS 認証情報をクラスターごとに準備する必要があります。

手順

1. 次のコマンドを入力して、AWS トークンファイルから OpenShift Container Platform シークレットを作成します。

   a. 認証情報ファイルを作成します。

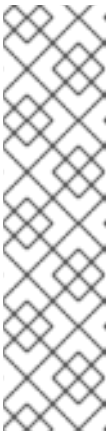
   ```
   $ cat <<EOF > ${SCRATCH}/credentials
     [default]
     role_arn = ${ROLE_ARN}
     web_identity_token_file = /var/run/secrets/openshift/serviceaccount/token
   EOF
   ```

   b. OADP の namespace を作成します。

   ```
   $ oc create namespace openshift-adp
   ```

   c. OpenShift Container Platform シークレットを作成します。

```
$ oc -n openshift-adp create secret generic cloud-credentials \
  --from-file=${SCRATCH}/credentials
```



注記

OpenShift Container Platform バージョン 4.14 以降では、OADP Operator は Operator Lifecycle Manager (OLM) および Cloud Credentials Operator (CCO) を通じて、標準化された新しい STS ワークフローをサポートします。このワークフローでは、上記シークレットの作成は必要ありません。OpenShift Container Platform Web コンソールを使用して、OLM で管理される Operator のインストール中にロール ARN のみ指定する必要があります。詳細は、**Web コンソールを使用して OperatorHub からインストールする** を参照してください。

前述のシークレットは CCO によって自動的に作成されます。

2. OADP Operator をインストールします。

   a. OpenShift Container Platform Web コンソールで、**Operators → OperatorHub** ページを表示します。

   b. **OADP Operator** を検索します。

   c. **role_ARN** フィールドに、前に作成した role_arn を貼り付け、 **Install** をクリックします。

3. 次のコマンドを入力し、AWS 認証情報を使用して AWS クラウドストレージを作成します。

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: CloudStorage
  metadata:
    name: ${CLUSTER_NAME}-oadp
    namespace: openshift-adp
  spec:
    creationSecret:
      key: credentials
      name: cloud-credentials
    enableSharedConfig: true
    name: ${CLUSTER_NAME}-oadp
    provider: aws
    region: $REGION
EOF
```

4. 次のコマンドを入力して、アプリケーションのストレージのデフォルトストレージクラスを確認します。

```
$ oc get pvc -n <namespace>
```

出力例

```
NAME     STATUS   VOLUME                                     CAPACITY   ACCESS MODES
STORAGECLASS   AGE
applog   Bound    pvc-351791ae-b6ab-4e8b-88a4-30f73caf5ef8   1Gi        RWO          gp3-
```

```
csi     4d19h
mysql   Bound    pvc-16b8e009-a20a-4379-accc-bc81fedd0621   1Gi        RWO            gp3-
csi     4d19h
```

5. 次のコマンドを実行してストレージクラスを取得します。

```
$ oc get storageclass
```

出力例

```
NAME                PROVISIONER             RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
gp2                 kubernetes.io/aws-ebs   Delete          WaitForFirstConsumer   true
4d21h
gp2-csi             ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3                 ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
gp3-csi (default)   ebs.csi.aws.com         Delete          WaitForFirstConsumer   true
4d21h
```

注記

次のストレージクラスが機能します。

- gp3-csi
- gp2-csi
- gp3
- gp2

すべてのアプリケーション、またはバックアップされるアプリケーションが Container Storage Interface (CSI) で永続ボリューム (PV) を使用している場合は、OADP DPA 設定に CSI プラグインを含めることをお勧めします。

6. バックアップとボリュームスナップショットが保存されるストレージへの接続を設定するために、**DataProtectionApplication** リソースを作成します。

   a. CSI ボリュームのみを使用している場合は、次のコマンドを入力して Data Protection Application をデプロイします。

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true (1)
    features:
      dataMover:
        enable: false
```

```
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
      velero:
        defaultPlugins:
        - openshift
        - aws
        - csi
      restic:
        enable: false
EOF
```

1. イメージのバックアップを使用しない場合は、このフィールドを **false** に設定します。

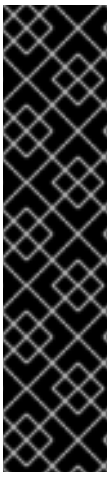
a. CSI ボリュームまたは非 CSI ボリュームを使用している場合は、次のコマンドを入力して Data Protection Application をデプロイします。

```
$ cat << EOF | oc create -f -
  apiVersion: oadp.openshift.io/v1alpha1
  kind: DataProtectionApplication
  metadata:
    name: ${CLUSTER_NAME}-dpa
    namespace: openshift-adp
  spec:
    backupImages: true 1
    features:
      dataMover:
        enable: false
    backupLocations:
    - bucket:
        cloudStorageRef:
          name: ${CLUSTER_NAME}-oadp
        credential:
          key: credentials
          name: cloud-credentials
        prefix: velero
        default: true
        config:
          region: ${REGION}
    configuration:
      velero:
        defaultPlugins:
        - openshift
        - aws
      nodeAgent: 2
```

```
      enable: false
      uploaderType: restic
    snapshotLocations:
      - velero:
          config:
            credentialsFile: /tmp/credentials/openshift-adp/cloud-credentials-credentials 3
            enableSharedConfig: "true" 4
            profile: default 5
            region: ${REGION} 6
          provider: aws
EOF
```

1. イメージのバックアップを使用しない場合は、このフィールドを **false** に設定します。
2. **nodeAgent** 属性に関する重要な注記を参照してください。
3. **credentialsFile** フィールドは、Pod のバケット認証情報のマウント先です。
4. **enableSharedConfig** フィールドを使用すると、**snapshotLocations** がバケットに定義された認証情報を共有または再利用できます。
5. AWS 認証情報ファイルに設定されているプロファイル名を使用します。
6. **region** は、お使いの AWS リージョンに指定します。これはクラスターリージョンと同じである必要があります。

これで、**アプリケーションのバックアップ** で説明されているとおり、OpenShift Container Platform アプリケーションをバックアップおよび復元する準備が整いました。



重要

OADP 1.2 を使用する場合は、次の設定を置き換えます。

```
nodeAgent:
  enable: false
  uploaderType: restic
```

次の設定に置き換えます。

```
restic:
  enable: false
```

バックアップと復元に 2 つの異なるクラスターを使用する場合、cloudstorage CR と OADP **DataProtectionApplication** 設定の両方で、2 つのクラスターの AWS S3 ストレージ名が同じである必要があります。

関連情報

- Web コンソールを使用した OperatorHub からのインストール
- アプリケーションのバックアップ

4.9.1.3. オプションのクリーンアップを使用して OADP AWS STS 上のワークロードをバックアップする

4.9.1.3.1. OADP と AWS STS を使用したバックアップの実行

次の **hello-world** アプリケーションの例では、永続ボリューム (PV) が接続されていません。OpenShift API for Data Protection (OADP) と Amazon Web Services (AWS) (AWS STS) を使用してバックアップを実行します。

どちらの Data Protection Application (DPA) 設定も機能します。

1. 次のコマンドを実行して、バックアップするワークロードを作成します。

```
$ oc create namespace hello-world
```

```
$ oc new-app -n hello-world --image=docker.io/openshift/hello-openshift
```

2. 次のコマンドを実行してルートを公開します。

```
$ oc expose service/hello-openshift -n hello-world
```

3. 次のコマンドを実行して、アプリケーションが動作していることを確認します。

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

出力例

```
Hello OpenShift!
```

4. 次のコマンドを実行して、ワークロードをバックアップします。

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Backup
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    includedNamespaces:
    - hello-world
    storageLocation: ${CLUSTER_NAME}-dpa-1
    ttl: 720h0m0s
EOF
```

5. バックアップが完了するまで待ってから、次のコマンドを実行します。

```
$ watch "oc -n openshift-adp get backup hello-world -o json | jq .status"
```

出力例

```
{
  "completionTimestamp": "2022-09-07T22:20:44Z",
```

```
  "expiration": "2022-10-07T22:20:22Z",
  "formatVersion": "1.1.0",
  "phase": "Completed",
  "progress": {
    "itemsBackedUp": 58,
    "totalItems": 58
  },
  "startTimestamp": "2022-09-07T22:20:22Z",
  "version": 1
}
```


6. 次のコマンドを実行して、デモワークロードを削除します。

```
$ oc delete ns hello-world
```

7. 次のコマンドを実行して、バックアップからワークロードを復元します。

```
$ cat << EOF | oc create -f -
  apiVersion: velero.io/v1
  kind: Restore
  metadata:
    name: hello-world
    namespace: openshift-adp
  spec:
    backupName: hello-world
EOF
```

8. 次のコマンドを実行して、復元が完了するまで待ちます。

```
$ watch "oc -n openshift-adp get restore hello-world -o json | jq .status"
```

出力例


```
{
  "completionTimestamp": "2022-09-07T22:25:47Z",
  "phase": "Completed",
  "progress": {
    "itemsRestored": 38,
    "totalItems": 38
  },
  "startTimestamp": "2022-09-07T22:25:28Z",
  "warnings": 9
}
```


9. 次のコマンドを実行して、ワークロードが復元されていることを確認します。

```
$ oc -n hello-world get pods
```

出力例

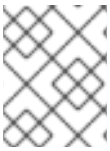
```
NAME                         READY   STATUS    RESTARTS   AGE
hello-openshift-9f885f7c6-kdjpj   1/1     Running   0          90s
```

10. 次のコマンドを実行して JSONPath を確認します。

```
$ curl `oc get route/hello-openshift -n hello-world -o jsonpath='{.spec.host}'`
```

出力例

```
Hello OpenShift!
```



注記

トラブルシューティングのヒントについては、OADP チームの トラブルシューティングドキュメント を参照してください。

4.9.1.3.2. OADP と AWS STS を使用してバックアップ後のクラスターをクリーンアップする

この例のバックアップおよび S3 バケットと OpenShift API for Data Protection (OADP) Operator をアンインストールする必要がある場合は、次の手順を実行します。

手順

1. 次のコマンドを実行して、ワークロードを削除します。

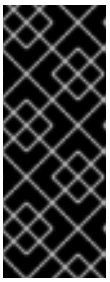
```
$ oc delete ns hello-world
```

2. 次のコマンドを実行して、Data Protection Application (DPA) を削除します。

```
$ oc -n openshift-adp delete dpa ${CLUSTER_NAME}-dpa
```

3. 次のコマンドを実行して、クラウドストレージを削除します。

```
$ oc -n openshift-adp delete cloudstorage ${CLUSTER_NAME}-oadp
```



重要

このコマンドがハングした場合は、次のコマンドを実行してファイナライザーを削除する必要がある場合があります。

```
$ oc -n openshift-adp patch cloudstorage ${CLUSTER_NAME}-oadp -p
'{"metadata":{"finalizers":null}}' --type=merge
```

4. Operator が不要になった場合は、次のコマンドを実行して削除します。

```
$ oc -n openshift-adp delete subscription oadp-operator
```

5. 次のコマンドを実行して、Operator から namespace を削除します。

```
$ oc delete ns openshift-adp
```

6. バックアップおよび復元リソースが不要になった場合は、次のコマンドを実行してクラスターからリソースを削除します。

```
$ oc delete backup hello-world
```

7. AWS S3 のバックアップ、復元、およびリモートオブジェクトを削除するには、次のコマンドを実行します。

```
$ velero backup delete hello-world
```

8. カスタムリソース定義 (CRD) が不要になった場合は、次のコマンドを実行してクラスターから削除します。

```
$ for CRD in `oc get crds | grep velero | awk '{print $1}'`; do oc delete crd $CRD; done
```

9. 次のコマンドを実行して、AWS S3 バケットを削除します。

```
$ aws s3 rm s3://${CLUSTER_NAME}-oadp --recursive
```

```
$ aws s3api delete-bucket --bucket ${CLUSTER_NAME}-oadp
```

10. 次のコマンドを実行して、ロールからポリシーを切り離します。

```
$ aws iam detach-role-policy --role-name "${ROLE_NAME}"  --policy-arn "${POLICY_ARN}"
```

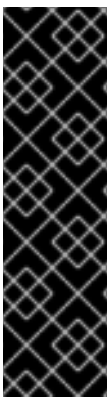
11. 以下のコマンドを実行してロールを削除します。

```
$ aws iam delete-role --role-name "${ROLE_NAME}"
```

4.10. OADP 1.2 DATA MOVER

4.10.1. OADP Data Mover の概要

OADP Data Mover を使用すると、クラスターの障害が発生した場合、誤って削除した場合、または破損した場合に、ストアからステートフルアプリケーションを復元できます。



重要

OADP 1.2 Data Mover はテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

- OADP Data Mover を使用して、Container Storage Interface (CSI) ボリュームのスナップショットをリモートオブジェクトストアにバックアップできます。CSI スナップショットに Data Mover を使用する を参照してください。

- OADP 1.2 Data Mover を使用して、CephFS、CephRBD、またはその両方を使用するクラスターのアプリケーションデータをバックアップおよび復元できます。Ceph Storage での OADP 1.2 Data Mover の使用 を参照してください。

注記

OADP 1.3 Data Mover では、移行後のフックが正常に機能しない可能性があります。

OADP 1.2 Data Mover は、同期プロセスを使用してアプリケーションデータをバックアップおよび復元します。プロセスは同期しているため、復元後のフックが開始されるのは、必ず関連する Pod の永続ボリューム (PV) が Data Mover の永続ボリューム要求 (PVC) により解放された後になります。

しかし、OADP 1.3 Data Mover は非同期プロセスを使用します。このような順序の違いにより、Data Mover の PVC によって関連する PV が解放される前に、復元後のフックが呼び出される可能性があります。これが発生した場合、Pod は **Pending** ステータスのままになり、フックを実行できません。Pod が解放される前にフックの試行がタイムアウトになり、復元操作で **PartiallyFailed** が発生する可能性があります。

4.10.1.1. OADP Data Mover の前提条件

- 別の namespace で実行されているステートフルアプリケーションがある。
- Operator Lifecycle Manager (OLM) を使用して OADP Operator をインストールしている。
- 適切な **VolumeSnapshotClass** と **StorageClass** を作成している。
- OLM を使用して VolSync オペレーターをインストールしている。

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

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

Data Mover ソリューションは、VolSync の Restic オプションを使用します。

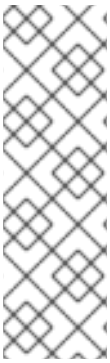
Data Mover は、CSI ボリュームスナップショットのバックアップとリストアのみをサポートします。

OADP 1.2 Data Mover では、**VolumeSnapshotBackups** (VSB) および **VolumeSnapshotRestore** (VSR) は、VolumeSnapshotMover (VSM) を使用してキューに入れられます。VSM のパフォーマンスは、同時に **InProgress** で VSB と VSR の同時数を指定することで向上します。すべての非同期プラグイン操作が完了すると、バックアップは完了としてマークされます。

重要

OADP 1.2 Data Mover はテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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



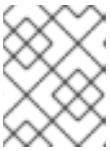
注記

Red Hat では、ODF CephFS ボリュームのバックアップおよびリストアに OADP 1.2 Data Mover を使用しているお客様には、パフォーマンスを向上させるために OpenShift Container Platform バージョン 4.12 以降をアップグレードまたはインストールすることを推奨します。OADP Data Mover は、OpenShift Container Platform バージョン 4.12 以降の CephFS シャローボリュームを利用できます。これは、私たちのテストに基づくと、バックアップ時間のパフォーマンスを向上させることができます。

- CephFS ROX の詳細

前提条件

- **StorageClass** および **VolumeSnapshotClass** カスタムリソース (CR) が CSI をサポートしていることを確認している。
- **snapshot.storage.kubernetes.io/is-default-class: "true"** のアノテーションを持つ **VolumeSnapshotClass** CR が 1 つだけであることを確認している。



注記

OpenShift Container Platform バージョン 4.12 以降では、これが唯一のデフォルト **VolumeSnapshotClass** であることを確認してください。

- **VolumeSnapshotClass** CR の **deletionPolicy** が **Retain** に設定されていることを確認している。
- 注釈 **storageclass.kubernetes.io/is-default-class: "true"** を持つ **StorageClass** CR が 1 つだけであることを確認している。
- **VolumeSnapshotClass** CR にラベル **velero.io/csi-volumesnapshot-class: "true"** を追加している。
- **OADP namespace** に **oc annotate --overwrite namespace/openshift-adp volsync.backube/privileged-movers="true"** のアノテーションが追加されていることを確認している。

注記

OADP 1.2 では、ほとんどのシナリオで **privileged-movers** 設定は必要ありません。復元コンテナーの権限は、Volsync コピーに対して適切である必要があります。一部のユーザーシナリオでは、**privileged-mover** = **true** 設定で解決する必要がある権限エラーが発生する場合があります。

- Operator Lifecycle Manager (OLM) を使用して VolSync Operator をインストールしました。

注記

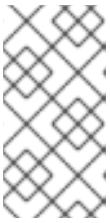
OADP Data Mover を使用するには、VolSync Operator が必要です。

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

手順

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

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



注記

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

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

Data Protection Application (DPA) CR の例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  backupLocations:
    - velero:
        config:
          profile: default
          region: us-east-1
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: <bucket_name>
          prefix: <bucket-prefix>
        provider: aws
  configuration:
    restic:
      enable: <true_or_false>
```

```
    velero:
      itemOperationSyncFrequency: "10s"
      defaultPlugins:
       - openshift
       - aws
       - csi
       - vsm
  features:
    dataMover:
      credentialName: restic-secret
      enable: true
      maxConcurrentBackupVolumes: "3" (1)
      maxConcurrentRestoreVolumes: "3" (2)
      pruneInterval: "14" (3)
      volumeOptions: (4)
      sourceVolumeOptions:
          accessMode: ReadOnlyMany
          cacheAccessMode: ReadWriteOnce
          cacheCapacity: 2Gi
      destinationVolumeOptions:
          storageClass: other-storageclass-name
          cacheAccessMode: ReadWriteMany
  snapshotLocations:
    - velero:
        config:
          profile: default
          region: us-west-2
        provider: aws
```

(1) オプション: バックアップのためにキューに入れることができるスナップショットの数の上限を指定します。デフォルト値は **10** です。

(2) オプション: 復元のためにキューに入れることができるスナップショットの数の上限を指定します。デフォルト値は **10** です。

(3) オプション: リポジトリーで Restic プルーニングを実行する間隔 (日数) を指定します。プルーン操作ではデータを再パックして領域を解放しますが、プロセスの一部として大量の I/O トラフィックが生成される可能性もあります。このオプションを設定すると、参照されなくなったデータによるストレージ消費とアクセスコストとの間のトレードオフが可能になります。

(4) オプション: バックアップと復元の VolumeSync ボリュームオプションを指定します。

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

VolumeSnapshotBackup CRD の例

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotBackup
metadata:
  name: <vsb_name>
  namespace: <namespace_name> (1)
spec:
```

```
  volumeSnapshotContent:
    name: <snapcontent_name>
  protectedNamespace: <adp_namespace> 2
  resticSecretRef:
    name: <restic_secret_name>
```

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

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

VolumeSnapshotRestore CRD の例

```
apiVersion: datamover.oadp.openshift.io/v1alpha1
kind: VolumeSnapshotRestore
metadata:
  name: <vsr_name>
  namespace: <namespace_name> 1
spec:
  protectedNamespace: <protected_ns> 2
  resticSecretRef:
    name: <restic_secret_name>
  volumeSnapshotMoverBackupRef:
    sourcePVCData:
      name: <source_pvc_name>
      size: <source_pvc_size>
    resticrepository: <your_restic_repo>
    volumeSnapshotClassName: <vsclass_name>
```

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

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

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

   a. バックアップ CR を作成します。

```
  apiVersion: velero.io/v1
  kind: Backup
  metadata:
    name: <backup_name>
    namespace: <protected_ns> 1
  spec:
    includedNamespaces:
    - <app_ns> 2
    storageLocation: velero-sample-1
```

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

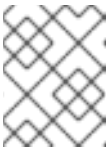
2 バックアップするアプリケーションの namespace を指定します。

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

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

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

DPA で設定されたオブジェクトストアにスナップショットが作成されます。



注記

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

4. 次の手順を実行して、ボリュームスナップショットを復元できます。

a. アプリケーションの namespace と、Velero CSI プラグインによって作成された **VolumeSnapshotContent** を削除します。

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

Restore CR の例

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore_name>
  namespace: <protected_ns>
spec:
  backupName: <previous_backup_name>
  restorePVs: true
```

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

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

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

d. アプリケーションデータとリソースが復元されたかどうかを確認します。

注記

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

4.10.3. Ceph Storage での OADP 1.2 Data Mover の使用

OADP 1.2 Data Mover を使用して、CephFS、CephRBD、またはその両方を使用するクラスターのアプリケーションデータをバックアップおよび復元できます。

OADP 1.2 Data Mover は、大規模環境をサポートする Ceph 機能を活用します。その 1 つはシャローコピー方式で、OpenShift Container Platform 4.12 以降で利用できます。この機能は、ソース Persistent

Volume Claim (PVC) にあるもの以外の **StorageClass** および **AccessMode** リソースのバックアップと復元をサポートします。

重要

CephFS のシャローコピー機能はバックアップ機能です。これは復元操作の一部ではありません。

4.10.3.1. Ceph Storage で OADP 1.2 Data Mover を使用するための前提条件

以下の前提条件は、Ceph Storage を使用するクラスター内で OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用するすべてのデータのバックアップおよびリストア操作に適用されます。

- OpenShift Container Platform 4.12 以降がインストールされている。
- OADP Operator がインストールされている。
- namespace **openshift-adp** にシークレットの **cloud-credentials** が作成されている。
- Red Hat OpenShift Data Foundation がインストールされている。
- Operator Lifecycle Manager を使用して最新の VolSync Operator をインストールしている。

4.10.3.2. OADP 1.2 Data Mover で使用するカスタムリソースの定義

Red Hat OpenShift Data Foundation をインストールすると、デフォルトの CephFS、CephRBD **StorageClass** および **VolumeSnapshotClass** カスタムリソース (CR) が自動的に作成されます。これらの CR は、OpenShift API for Data Protection (OADP) 1.2 Data Mover で使用するために定義する必要があります。

CR を定義した後、バックアップおよび復元操作を実行する前に、環境にその他の変更をいくつか加える必要があります。

4.10.3.2.1. OADP 1.2 Data Mover で使用する CephFS カスタムリソースの定義

Red Hat OpenShift Data Foundation をインストールすると、デフォルトの CephFS **StorageClass** カスタムリソース (CR) とデフォルトの CephFS **VolumeSnapshotClass** CR が自動的に作成されます。これらの CR は、OpenShift API for Data Protection (OADP) 1.2 Data Mover で使用するために定義できます。

手順

1. 以下の例のように **VolumeSnapshotClass** CR を定義します。

VolumeSnapshotClass CR の例

```
apiVersion: snapshot.storage.k8s.io/v1
deletionPolicy: <deletion_policy_type> 1
driver: openshift-storage.cephfs.csi.ceph.com
kind: VolumeSnapshotClass
metadata:
  annotations:
    snapshot.storage.kubernetes.io/is-default-class: true 2
  labels:
```

```
    velero.io/csi-volumesnapshot-class: true 3
  name: ocs-storagecluster-cephfsplugin-snapclass
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/snapshotter-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
```

1 OADP は、CSI および Data Mover のバックアップと復元に対して、**Retain** および **Delete** 削除ポリシータイプをサポートしています。OADP 1.2 Data Mover の場合、削除ポリシータイプを **Retain** に設定します。

2 **true** に設定する必要があります。

3 **true** に設定する必要があります。

2. 以下の例のように **StorageClass** CR を定義します。

StorageClass CR の例

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ocs-storagecluster-cephfs
  annotations:
    description: Provides RWO and RWX Filesystem volumes
    storageclass.kubernetes.io/is-default-class: true 1
provisioner: openshift-storage.cephfs.csi.ceph.com
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-cephfs-node
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  fsName: ocs-storagecluster-cephfilesystem
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

1 **true** に設定する必要があります。

4.10.3.2.2. OADP 1.2 Data Mover で使用する CephRBD カスタムリソースの定義

Red Hat OpenShift Data Foundation をインストールすると、デフォルトの CephRBD **StorageClass** カスタムリソース (CR) とデフォルトの CephRBD **VolumeSnapshotClass** CR が自動的に作成されます。これらの CR は、OpenShift API for Data Protection (OADP) 1.2 Data Mover で使用するために定義できます。

手順

1. 以下の例のように **VolumeSnapshotClass** CR を定義します。

VolumeSnapshotClass CR の例

```
apiVersion: snapshot.storage.k8s.io/v1
deletionPolicy: <deletion_policy_type> 1
driver: openshift-storage.rbd.csi.ceph.com
kind: VolumeSnapshotClass
metadata:
  labels:
    velero.io/csi-volumesnapshot-class: true 2
  name: ocs-storagecluster-rbdplugin-snapclass
parameters:
  clusterID: openshift-storage
  csi.storage.k8s.io/snapshotter-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/snapshotter-secret-namespace: openshift-storage
```

1. OADP は、CSI および Data Mover のバックアップと復元に対して、**Retain** および **Delete** 削除ポリシータイプをサポートしています。OADP 1.2 Data Mover の場合、削除ポリシータイプを **Retain** に設定します。
2. **true** に設定する必要があります。

2. 以下の例のように **StorageClass** CR を定義します。

StorageClass CR の例

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ocs-storagecluster-ceph-rbd
  annotations:
    description: 'Provides RWO Filesystem volumes, and RWO and RWX Block volumes'
provisioner: openshift-storage.rbd.csi.ceph.com
parameters:
  csi.storage.k8s.io/fstype: ext4
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-rbd-provisioner
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-rbd-node
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-rbd-provisioner
  imageFormat: '2'
  clusterID: openshift-storage
  imageFeatures: layering
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  pool: ocs-storagecluster-cephblockpool
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

4.10.3.2.3. OADP 1.2 Data Mover で使用する追加のカスタムリソースの定義

デフォルトの **StorageClass** および CephRBD **VolumeSnapshotClass** カスタムリソース (CR) を再定義した後、次の CR を作成する必要があります。

- シャローコピー機能を使用するように定義された CephFS **StorageClass** CR
- Restic **Secret** CR

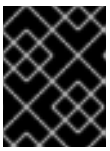
手順

1. 次の例のように CephFS **StorageClass** CR を作成し、**backingSnapshot** パラメーターを **true** に設定します。

   backingSnapshot を true に設定した CephFS StorageClass CR の例

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: ocs-storagecluster-cephfs-shallow
  annotations:
    description: Provides RWO and RWX Filesystem volumes
    storageclass.kubernetes.io/is-default-class: false
provisioner: openshift-storage.cephfs.csi.ceph.com
parameters:
  csi.storage.k8s.io/provisioner-secret-namespace: openshift-storage
  csi.storage.k8s.io/provisioner-secret-name: rook-csi-cephfs-provisioner
  csi.storage.k8s.io/node-stage-secret-name: rook-csi-cephfs-node
  csi.storage.k8s.io/controller-expand-secret-name: rook-csi-cephfs-provisioner
  clusterID: openshift-storage
  fsName: ocs-storagecluster-cephfilesystem
  csi.storage.k8s.io/controller-expand-secret-namespace: openshift-storage
  backingSnapshot: true (1)
  csi.storage.k8s.io/node-stage-secret-namespace: openshift-storage
reclaimPolicy: Delete
allowVolumeExpansion: true
volumeBindingMode: Immediate
```

   (1) **true** に設定する必要があります。

   重要

   CephFS **VolumeSnapshotClass** および **StorageClass** CR の **Provisioner** の値が同じであることを確認してください。

2. 次の例のように Restic **Secret** CR を設定します。

   Restic Secret CR の例

```
apiVersion: v1
kind: Secret
metadata:
  name: <secret_name>
  namespace: <namespace>
type: Opaque
stringData:
  RESTIC_PASSWORD: <restic_password>
```

4.10.3.3. OADP 1.2 Data Mover と CephFS ストレージを使用したデータのバックアップと復元

OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用すると、CephFS のシャローコピー機能を有効にすることで、CephFS ストレージを使用してデータをバックアップおよびリストアできます。

前提条件

- ステートフルアプリケーションが、CephFS をプロビジョナーとして使用し、永続ボリューム要求 (PVC) を持つ別の namespace で実行されている。
- **StorageClass** および **VolumeSnapshotClass** カスタムリソース (CR) が、CephFS および OADP 1.2 Data Mover 用に定義されています。
- **openshift-adp** namespace にシークレットの **cloud-credentials** がある。

4.10.3.3.1. CephFS ストレージで使用する DPA の作成

OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用して、CephFS ストレージでデータをバックアップおよびリストアするには、Data Protection Application (DPA) CR を作成する必要があります。

手順

1. OADP 1.2 Data Mover のために、次のコマンドを実行して、**VolumeSnapshotClass** CR の **deletionPolicy** フィールドが **Retain** に設定されていることを確認する必要があります。

```
$ oc get volumesnapshotclass -A  -o jsonpath='{range .items[*]}{"Name: "}{.metadata.name}{"  "}{"Retention Policy: "}{.deletionPolicy}{"\n"}{end}'
```

2. 次のコマンドを実行して、**VolumeSnapshotClass** CR のラベルが **true** に設定されていることを確認します。

```
$ oc get volumesnapshotclass -A  -o jsonpath='{range .items[*]}{"Name: "}{.metadata.name}{"  "}{"labels: "}{.metadata.labels}{"\n"}{end}'
```

3. 次のコマンドを実行して、**StorageClass** CR の **storageclass.kubernetes.io/is-default-class** アノテーションが **true** に設定されていることを確認します。

```
$ oc get storageClass -A  -o jsonpath='{range .items[*]}{"Name: "}{.metadata.name}{"  "}{"annotations: "}{.metadata.annotations}{"\n"}{end}'
```

4. 次の例のような Data Protection Application (DPA) CR を作成します。

DPA CR の例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  backupLocations:
```

```
 - velero:
     config:
       profile: default
       region: us-east-1
     credential:
       key: cloud
       name: cloud-credentials
     default: true
     objectStorage:
       bucket: <my_bucket>
       prefix: velero
     provider: aws
configuration:
   restic:
     enable: false (1)
   velero:
     defaultPlugins:
       - openshift
       - aws
       - csi
       - vsm
features:
   dataMover:
     credentialName: <restic_secret_name> (2)
     enable: true (3)
     volumeOptionsForStorageClasses: (4)
       ocs-storagecluster-cephfs:
         sourceVolumeOptions:
           accessMode: ReadOnlyMany
           cacheAccessMode: ReadWriteMany
           cacheStorageClassName: ocs-storagecluster-cephfs
           storageClassName: ocs-storagecluster-cephfs-shallow
```

1. **enable** フィールドにはデフォルト値はありません。有効な値は **true** または **false** です。
2. OADP 1.2 Data Mover および Ceph を操作するための環境を準備したときに作成した Restic **Secret** を使用します。Restic **Secret** を使用しない場合、CR はこのパラメーターのデフォルト値 **dm-credential** を使用します。
3. **enable** フィールドにはデフォルト値はありません。有効な値は **true** または **false** です。
4. オプションのパラメーター。各 **storageClass** ボリュームに対して、異なる **VolumeOptionsForStorageClass** ラベルのセットを定義できます。この設定では、異なるプロバイダーのボリュームのバックアップが提供されます。オプションの **VolumeOptionsForStorageClass** パラメーターは通常 CephFS で使用されますが、どのストレージタイプでも使用できます。

4.10.3.3.2. OADP 1.2 Data Mover と CephFS ストレージを使用したデータのバックアップ

OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用すると、CephFS ストレージのシャローコピー機能を有効にすることで、CephFS ストレージを使用してデータをバックアップできます。

手順

1. 次の例のように、**Backup** CR を作成します。

   Backup CR の例

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

2. 次の手順を実行して、**VolumeSnapshotBackup** CR の進行状況を監視します。

   a. すべての **VolumeSnapshotBackup** CR の進行状況を確認するには、次のコマンドを実行します。

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

   b. 特定の **VolumeSnapshotBackup** CR の進行状況を確認するには、次のコマンドを実行します。

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

3. **VolumeSnapshotBackup** CR のステータスが **Completed** になるまで、数分間待ちます。

4. Restic **Secret** で指定されたスナップショットがオブジェクトストアに少なくとも 1 つあることを確認します。**/<OADP-namespace>** という接頭辞を持つ、対象の **BackupStorageLocation** ストレージプロバイダーでこのスナップショットを確認できます。

4.10.3.3.3. OADP 1.2 Data Mover と CephFS ストレージを使用したデータのリストア

CephFS ストレージのシャローコピー機能がバックアップ手順で有効になっている場合、OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用して、CephFS ストレージを使用してデータをリストアできます。シャローコピー機能は復元手順では使用されません。

手順

1. 次のコマンドを実行して、アプリケーションの namespace を削除します。

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

2. 次のコマンドを実行して、バックアップ中に作成された **VolumeSnapshotContent** CR を削除します。

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

3. 次の例のように、**Restore** CR を作成します。

   Restore CR の例

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore_name>
  namespace: <protected_ns>
spec:
  backupName: <previous_backup_name>
```

4. 次の手順を実行して、**VolumeSnapshotRestore** CR の進行状況を監視します。
   a. すべての **VolumeSnapshotRestore** CR の進行状況を確認するには、次のコマンドを実行します。

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

   b. 特定の **VolumeSnapshotRestore** CR の進行状況を確認するには、次のコマンドを実行します。

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

5. 次のコマンドを実行して、アプリケーションデータが復元されたことを確認します。

   ```
   $ oc get route <route_name> -n <app_ns> -ojsonpath="{.spec.host}"
   ```

4.10.3.4. OADP 1.2 Data Mover と分割ボリューム (CephFS および Ceph RBD) を使用したデータのバックアップとリストア

OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用すると、**分割ボリューム** のある環境、つまり CephFS と CephRBD の両方を使用する環境でデータをバックアップおよびリストアできます。

前提条件

- ステートフルアプリケーションが、CephFS をプロビジョナーとして使用し、永続ボリューム要求 (PVC) を持つ別の namespace で実行されている。
- **StorageClass** および **VolumeSnapshotClass** カスタムリソース (CR) が、CephFS および OADP 1.2 Data Mover 用に定義されています。
- **openshift-adp** namespace にシークレットの **cloud-credentials** がある。

4.10.3.4.1. 分割ボリュームで使用する DPA の作成

OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用して、分割ボリュームでデータをバックアップおよびリストアするには、Data Protection Application (DPA) CR を作成する必要があります。

手順

- 次の例のように、Data Protection Application (DPA) CR を作成します。

  分割ボリューム環境の DPA CR の例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: velero-sample
  namespace: openshift-adp
spec:
  backupLocations:
    - velero:
        config:
          profile: default
          region: us-east-1
        credential:
          key: cloud
          name: cloud-credentials
        default: true
        objectStorage:
          bucket: <my-bucket>
          prefix: velero
        provider: aws
  configuration:
    restic:
      enable: false
    velero:
      defaultPlugins:
        - openshift
        - aws
        - csi
        - vsm
  features:
    dataMover:
      credentialName: <restic_secret_name> 1
      enable: true
      volumeOptionsForStorageClasses: 2
        ocs-storagecluster-cephfs:
          sourceVolumeOptions:
            accessMode: ReadOnlyMany
            cacheAccessMode: ReadWriteMany
            cacheStorageClassName: ocs-storagecluster-cephfs
            storageClassName: ocs-storagecluster-cephfs-shallow
        ocs-storagecluster-ceph-rbd:
          sourceVolumeOptions:
            storageClassName: ocs-storagecluster-ceph-rbd
            cacheStorageClassName: ocs-storagecluster-ceph-rbd
          destinationVolumeOptions:
            storageClassName: ocs-storagecluster-ceph-rbd
            cacheStorageClassName: ocs-storagecluster-ceph-rbd
```

1 OADP 1.2 Data Mover および Ceph を操作するための環境を準備したときに作成した Restic **Secret** を使用します。そうしない場合、CR はこのパラメーターのデフォルト値 **dm-credential** を使用します。

2 **storageClass** ボリュームごとに異なる **VolumeOptionsForStorageClass** ラベルのセットを定義できるため、異なるプロバイダーのボリュームへのバックアップが可能になります。**VolumeOptionsForStorageClass** パラメーターは、CephFS で使用するためのものです。ただし、オプションの **VolumeOptionsForStorageClass** パラメーターは、どのス

トレージタイプでも使用できます。

4.10.3.4.2. OADP 1.2 Data Mover と分割ボリュームを使用したデータのバックアップ

OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用して、分割ボリュームのある環境でデータをバックアップできます。

手順

1. 次の例のように、**Backup** CR を作成します。

   Backup CR の例

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

2. 次の手順を実行して、**VolumeSnapshotBackup** CR の進行状況を監視します。

   a. すべての **VolumeSnapshotBackup** CR の進行状況を確認するには、次のコマンドを実行します。

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

   b. 特定の **VolumeSnapshotBackup** CR の進行状況を確認するには、次のコマンドを実行します。

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

3. **VolumeSnapshotBackup** CR のステータスが **Completed** になるまで、数分間待ちます。

4. Restic **Secret** で指定されたスナップショットがオブジェクトストアに少なくとも 1 つあることを確認します。**/<OADP-namespace>** という接頭辞を持つ、対象の **BackupStorageLocation** ストレージプロバイダーでこのスナップショットを確認できます。

4.10.3.4.3. OADP 1.2 Data Mover と分割ボリュームを使用したデータのリストア

CephFS ストレージのシャローコピー機能がバックアップ手順で有効になっている場合、OpenShift API for Data Protection (OADP) 1.2 Data Mover を使用して、分割ボリュームのある環境でデータをリストアできます。シャローコピー機能は復元手順では使用されません。

手順

1. 次のコマンドを実行して、アプリケーションの namespace を削除します。

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

2. 次のコマンドを実行して、バックアップ中に作成された **VolumeSnapshotContent** CR を削除します。

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

3. 次の例のように、**Restore** CR を作成します。

Restore CR の例

```
apiVersion: velero.io/v1
kind: Restore
metadata:
  name: <restore_name>
  namespace: <protected_ns>
spec:
  backupName: <previous_backup_name>
```

4. 次の手順を実行して、**VolumeSnapshotRestore** CR の進行状況を監視します。

   a. すべての **VolumeSnapshotRestore** CR の進行状況を確認するには、次のコマンドを実行します。

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

   b. 特定の **VolumeSnapshotRestore** CR の進行状況を確認するには、次のコマンドを実行します。

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

5. 次のコマンドを実行して、アプリケーションデータが復元されたことを確認します。

```
$ oc get route <route_name> -n <app_ns> -ojsonpath="{.spec.host}"
```

4.10.3.5. OADP 1.2 の削除ポリシー

削除ポリシーは、システムからデータを削除するためのルールを決定します。保存期間、データの機密性、コンプライアンス要件などの要素に基づいて、削除をいつどのように行うかを指定します。規制を遵守し、貴重な情報を保護しながら、データの削除を効果的に管理します。

4.10.3.5.1. OADP 1.2 の削除ポリシーガイドライン

OADP 1.2 の次の削除ポリシーガイドラインを確認してください。

- OADP 1.2.x Data Mover を使用してバックアップとリストアを行うには、**VolumeSnapshotClass** カスタムリソース (CR) で **deletionPolicy** フィールドを **Retain** に設定します。
- OADP 1.2.x で CSI によるバックアップとリストアを使用するには、**VolumeSnapshotClass** CR の **deletionPolicy** フィールドを **Retain** または **Delete** に設定できます。

重要

OADP 1.2.x Data Mover のバックアップとリストアはテクノロジープレビュー機能であり、サポートを受けるにはサポート例外が必要です。

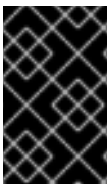
4.11. OADP 1.3 DATA MOVER

4.11.1. OADP 1.3 Data Mover について

OADP 1.3 には、Container Storage Interface (CSI) ボリュームのスナップショットをリモートオブジェクトストアに移動するために使用できる、ビルトイン Data Mover が含まれています。ビルトイン Data Mover を使用すると、クラスターの障害、誤削除、または破損が発生した場合に、リモートオブジェクトストアからステートフルアプリケーションを復元できます。スナップショットデータを読み取り、統合リポジトリーに書き込むためのアップローダーメカニズムとして Kopia を使用します。

OADP は、以下で CSI スナップショットをサポートします。

- Red Hat OpenShift Data Foundation
- Kubernetes Volume Snapshot API をサポートする Container Storage Interface (CSI) ドライバーを使用するその他のクラウドストレージプロバイダー



重要

OADP 1.3 でテクノロジープレビューとして導入された OADP 組み込みの Data Mover が、コンテナー化されたワークロードと仮想マシンのワークロードの両方で完全にサポートされるようになりました。

4.11.1.1. ビルトイン Data Mover の有効化

ビルトイン Data Mover を有効にするには、CSI プラグインを組み込み、**DataProtectionApplication** カスタムリソース (CR) でノードエージェントを有効にする必要があります。ノードエージェントは、データ移動モジュールをホストする Kubernetes デーモンセットです。これには、Data Mover のコントローラー、アップローダー、リポジトリーが含まれます。

DataProtectionApplication マニフェストの例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
  name: dpa-sample
spec:
  configuration:
    nodeAgent:
      enable: true 1
      uploaderType: kopia 2
    velero:
      defaultPlugins:
      - openshift
      - aws
      - csi 3
      defaultSnapshotMoveData: true
      defaultVolumesToFSBackup: 4
```

```
    featureFlags:
    - EnableCSI
# ...
```

1. ノードエージェントを有効にするフラグ。
2. アップローダーの種類。使用できる値は、**restic** または **kopia** です。ビルトイン Data Mover は、**uploaderType** フィールドの値に関係なく、デフォルトのアップローダーメカニズムとして Kopia を使用します。
3. デフォルトプラグインのリストに含まれる CSI プラグイン。
4. OADP 1.3.1 以降では、**fs-backup** をオプトアウトするボリュームにのみ Data Mover を使用する場合、**true** に設定します。ボリュームにデフォルトで Data Mover を使用する場合は **false** に設定します。

4.11.1.2. ビルトイン Data Mover のコントローラーとカスタムリソース定義 (CRD)

ビルトイン Data Mover 機能には、バックアップと復元を管理するための CRD として定義された 3 つの新しい API オブジェクトが導入されています。

- **DataDownload**: ボリュームスナップショットのデータダウンロードを表します。CSI プラグインは、復元するボリュームごとに 1 つの **DataDownload** オブジェクトを作成します。**DataDownload** CR には、ターゲットボリューム、指定された Data Mover、現在のデータダウンロードの進行状況、指定されたバックアップリポジトリー、プロセス完了後の現在のデータダウンロードの結果に関する情報が含まれます。
- **DataUpload**: ボリュームスナップショットのデータアップロードを表します。CSI プラグインは、CSI スナップショットごとに 1 つの **DataUpload** オブジェクトを作成します。**DataUpload** CR には、指定されたスナップショット、指定された Data Mover、指定されたバックアップリポジトリー、現在のデータアップロードの進行状況、およびプロセス完了後の現在のデータアップロードの結果に関する情報が含まれます。
- **BackupRepository**: バックアップリポジトリーのライフサイクルを表し、管理します。OADP は、namespace の最初の CSI スナップショットバックアップまたは復元が要求されると、namespace ごとにバックアップリポジトリーを作成します。

4.11.2. CSI スナップショットのバックアップおよび復元のデータ移動

OADP 1.3 Data Mover を使用して、永続ボリュームのバックアップと復元を実行できます。

4.11.2.1. 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

OADP Data Mover を使用して、Container Storage Interface (CSI) ボリュームのスナップショットをリモートオブジェクトストアにバックアップできます。

前提条件

- **cluster-admin** ロールでクラスターにアクセスできる。
- OADP Operator がインストールされている。
- CSI プラグインを組み込み、**DataProtectionApplication** カスタムリソース (CR) でノードエージェントを有効にしている。

- 別の namespace で実行されている永続ボリュームを持つアプリケーションがある。
- **metadata.labels.velero.io/csi-volumesnapshot-class: "true"** のキー/値ペアを **VolumeSnapshotClass** CR に追加している。

手順

1. 次の例のように、**Backup** オブジェクトの YAML ファイルを作成します。

   Backup CR の例

```
kind: Backup
apiVersion: velero.io/v1
metadata:
  name: backup
  namespace: openshift-adp
spec:
  csiSnapshotTimeout: 10m0s
  defaultVolumesToFsBackup: 1
  includedNamespaces:
  - mysql-persistent
  itemOperationTimeout: 4h0m0s
  snapshotMoveData: true 2
  storageLocation: default
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - dpa-sample-1
# ...
```

   1. **fs-backup** をオプトアウトするボリュームにのみ Data Mover を使用する場合、**true** に設定します。ボリュームにデフォルトで Data Mover を使用する場合は **false** に設定します。
   2. CSI スナップショットのリモートオブジェクトストレージへの移動を有効にするには、**true** に設定します。

2. マニフェストを適用します。

```
$ oc create -f backup.yaml
```

   スナップショットの作成が完了すると、**DataUpload** CR が作成されます。

検証

- **DataUpload** CR の **status.phase** フィールドを監視して、スナップショットデータがリモートオブジェクトストアに正常に転送されたことを確認します。使用される値は、**In Progress**、**Completed**、**Failed**、または **Canceled** です。オブジェクトストアは、**DataProtectionApplication** CR の **backupLocations** スタンザで設定されます。
  - 次のコマンドを実行して、すべての **DataUpload** オブジェクトのリストを取得します。

```
$ oc get datauploads -A
```

出力例

```
NAMESPACE       NAME              STATUS      STARTED   BYTES DONE   TOTAL
BYTES   STORAGE LOCATION   AGE     NODE
openshift-adp   backup-test-1-sw76b   Completed   9m47s    108104082    108104082
dpa-sample-1       9m47s   ip-10-0-150-57.us-west-2.compute.internal
openshift-adp   mongo-block-7dtpf     Completed   14m      1073741824   1073741824
dpa-sample-1       14m     ip-10-0-150-57.us-west-2.compute.internal
```

- 次のコマンドを実行して、**DataUpload** オブジェクトの **status.phase** フィールドの値を確認します。

```
$ oc get datauploads <dataupload_name> -o yaml
```

出力例

```
apiVersion: velero.io/v2alpha1
kind: DataUpload
metadata:
  name: backup-test-1-sw76b
  namespace: openshift-adp
spec:
  backupStorageLocation: dpa-sample-1
  csiSnapshot:
    snapshotClass: ""
    storageClass: gp3-csi
    volumeSnapshot: velero-mysql-fq8sl
  operationTimeout: 10m0s
  snapshotType: CSI
  sourceNamespace: mysql-persistent
  sourcePVC: mysql
status:
  completionTimestamp: "2023-11-02T16:57:02Z"
  node: ip-10-0-150-57.us-west-2.compute.internal
  path: /host_pods/15116bac-cc01-4d9b-8ee7-
609c3bef6bde/volumes/kubernetes.io~csi/pvc-eead8167-556b-461a-b3ec-
441749e291c4/mount
  phase: Completed (1)
  progress:
    bytesDone: 108104082
    totalBytes: 108104082
  snapshotID: 8da1c5febf25225f4577ada2aeb9f899
  startTimestamp: "2023-11-02T16:56:22Z"
```

(1) これは、スナップショットデータがリモートオブジェクトストアに正常に転送されたことを示しています。

4.11.2.2. CSI ボリュームスナップショットの復元

Restore CR を作成することで、ボリュームスナップショットを復元できます。

注記

OAPD 1.3 のビルトイン Data Mover を使用して、OADP 1.2 から Volsync バックアップを復元することはできません。OADP 1.3 にアップグレードする前に、Restic を使用してすべてのワークロードのファイルシステムバックアップを実行することが推奨されます。

前提条件

- **cluster-admin** ロールでクラスターにアクセスできる。
- データの復元元となる OADP **Backup** CR がある。

手順

1. 次の例のように、**Restore** CR の YAML ファイルを作成します。

   Restore CR の例

   ```
   apiVersion: velero.io/v1
   kind: Restore
   metadata:
     name: restore
     namespace: openshift-adp
   spec:
     backupName: <backup>
   # ...
   ```

2. マニフェストを適用します。

   ```
   $ oc create -f restore.yaml
   ```

   復元が開始されると、**DataDownload** が作成されます。

検証

- **DataDownload** CR の **status.phase** フィールドをチェックすることで、復元プロセスのステータスを監視できます。使用される値は、**In Progress**、**Completed**、**Failed**、または **Canceled** です。
  - すべての **DataDownload** オブジェクトのリストを取得するには、次のコマンドを実行します。

    ```
    $ oc get datadownloads -A
    ```

    出力例

    ```
    NAMESPACE       NAME                  STATUS      STARTED   BYTES DONE   TOTAL
    BYTES   STORAGE LOCATION   AGE     NODE
    openshift-adp   restore-test-1-sk7lg   Completed   7m11s     108104082    108104082
    dpa-sample-1       7m11s   ip-10-0-150-57.us-west-2.compute.internal
    ```

  - 次のコマンドを入力して、特定の **DataDownload** オブジェクトの **status.phase** フィールドの値を確認します。

```
$ oc get datadownloads <datadownload_name> -o yaml
```

出力例

```
apiVersion: velero.io/v2alpha1
kind: DataDownload
metadata:
  name: restore-test-1-sk7lg
  namespace: openshift-adp
spec:
  backupStorageLocation: dpa-sample-1
  operationTimeout: 10m0s
  snapshotID: 8da1c5febf25225f4577ada2aeb9f899
  sourceNamespace: mysql-persistent
  targetVolume:
    namespace: mysql-persistent
    pv: ""
    pvc: mysql
status:
  completionTimestamp: "2023-11-02T17:01:24Z"
  node: ip-10-0-150-57.us-west-2.compute.internal
  phase: Completed 1
  progress:
    bytesDone: 108104082
    totalBytes: 108104082
  startTimestamp: "2023-11-02T17:00:52Z"
```

1 CSI スナップショットデータが正常に復元されたことを示します。

4.11.2.3. OADP 1.3 の削除ポリシー

削除ポリシーは、システムからデータを削除するためのルールを決定します。保存期間、データの機密性、コンプライアンス要件などの要素に基づいて、削除をいつどのように行うかを指定します。規制を遵守し、貴重な情報を保護しながら、データの削除を効果的に管理します。

4.11.2.3.1. OADP 1.3 の削除ポリシーガイドライン

OADP 1.3 の次の削除ポリシーガイドラインを確認してください。

- OADP 1.3.x でいずれかのタイプのバックアップおよびリストア方法を使用する場合は、**VolumeSnapshotClass** カスタムリソース (CR) の **deletionPolicy** フィールドを **Retain** または **Delete** に設定できます。

4.12. 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。

ログと CR 情報は、**must-gather** ツール を使用して収集できます。

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

4.12.1. 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、Velero Web サイト上の "Install the CLI" 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. 使用している OADP および OpenShift Container Platform のバージョンに適切な Velero バージョンをダウンロードします。

4.12.1.1. OADP-Velero-OpenShift Container Platform バージョンの関係

OADP のバージョン	Velero のバージョン	OpenShift Container Platform バージョン
1.1.0	1.9	4.9 以降
1.1.1	1.9	4.9 以降
1.1.2	1.9	4.9 以降
1.1.3	1.9	4.9 以降
1.1.4	1.9	4.9 以降
1.1.5	1.9	4.9 以降
1.1.6	1.9	4.11 以降

OADP のバージョン	Velero のバージョン	OpenShift Container Platform バージョン
1.1.7	1.9	4.11 以降
1.2.0	1.11	4.11 以降
1.2.1	1.11	4.11 以降
1.2.2	1.11	4.11 以降
1.2.3	1.11	4.11 以降
1.3.0	1.12	4.12 以降

4.12.2. 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'
```

4.12.3. 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** を使用することを推奨します。

4.12.4. 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 コマンドをリスト表示します。

```
$ 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
```

次の種類の復元エラーと警告が、**velero describe** リクエストの出力に表示されます。

- **Velero**: Velero 自体の操作に関連するメッセージのリスト (クラウドへの接続、バックアップファイルの読み取りなどに関連するメッセージなど)
- **Cluster**: クラスタースコープのリソースのバックアップまたは復元に関連するメッセージのリスト
- **Namespaces**: namespace に保存されているリソースのバックアップまたは復元に関連するメッセージのリスト

これらのカテゴリーのいずれかで 1 つ以上のエラーが発生すると、**Restore** 操作のステータスが **PartiallyFailed** になり、**Completed** ではなくなります。警告によって完了ステータスが変化することはありません。

重要

- リソース固有のエラー、つまり **Cluster** および **Namespaces** エラーの場合、**restore description --details** 出力に、Velero が復元に成功したすべてのリソースのリストが含まれています。このようなエラーが発生したリソースについては、そのリソースが実際にクラスター内に存在するかどうかを確認してください。
- **describe** コマンドの出力に **Velero** エラーがあっても、リソース固有のエラーがない場合は、ワークロードの復元で実際の問題が発生することなく復元が完了した可能性があります。ただし、復元後のアプリケーションを十分に検証してください。
  たとえば、出力に **PodVolumeRestore** またはノードエージェント関連のエラーが含まれている場合は、**PodVolumeRestores** と **DataDownloads** のステータスを確認します。これらのいずれも失敗していないか、まだ実行中である場合は、ボリュームデータが完全に復元されている可能性があります。

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
```

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

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

関連情報

- ディスクおよびメモリーの要件

4.12.5.1. Velero Pod のリソースリクエストの設定

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

手順

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

  Velero ファイルの例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
configuration:
  velero:
    podConfig:
      resourceAllocations: 1
        requests:
          cpu: 200m
          memory: 256Mi
```

1 リストされている **resourceAllocations** は、平均使用量です。

4.12.5.2. Restic Pod のリソースリクエストの設定

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

手順

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

  Restic ファイルの例

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
...
configuration:
  restic:
```

```
podConfig:
  resourceAllocations: 1
    requests:
      cpu: 1000m
      memory: 16Gi
```

1 リストされている **resourceAllocations** は、平均使用量です。

重要

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

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

4.12.6. Velero と受付 Webhook に関する問題

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

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

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

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

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

4.12.6.1.1. Knative リソースの復元

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

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

手順

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

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

4.12.6.1.2. IBM AppConnect リソースの復元

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

手順

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

```
$ oc get mutatingwebhookconfigurations
```

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

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

4.12.6.2. OADP プラグインの既知の問題

次のセクションでは、OpenShift API for Data Protection (OADP) プラグインの既知の問題について説明します。

4.12.6.2.1. シークレットがないことで、イメージストリームのバックアップ中に Velero プラグインでパニックが発生する

バックアップとバックアップ保存場所 (BSL) が Data Protection Application (DPA) の範囲外で管理されている場合、OADP コントローラー (つまり DPA の調整) によって関連する **oadp-<bsl_name>-<bsl_provider>-registry-secret** が作成されません。

バックアップを実行すると、OpenShift Velero プラグインがイメージストリームバックアップでパニックになり、次のパニックエラーが表示されます。

```
024-02-27T10:46:50.028951744Z time="2024-02-27T10:46:50Z" level=error msg="Error backing up
item"
backup=openshift-adp/<backup name> error="error executing custom action
(groupResource=imagestreams.image.openshift.io,
namespace=<BSL Name>, name=postgres): rpc error: code = Aborted desc = plugin panicked:
runtime error: index out of range with length 1, stack trace: goroutine 94…
```

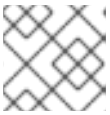
4.12.6.2.1.1. パニックエラーを回避するための回避策

Velero プラグインのパニックエラーを回避するには、次の手順を実行します。

1. カスタム BSL に適切なラベルを付けます。

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

2. BSL にラベルを付けた後、DPA の調整を待ちます。



注記

DPA 自体に軽微な変更を加えることで、強制的に調整を行うことができます。

3. DPA の調整では、適切な **oadp-<bsl_name>-<bsl_provider>-registry-secret** が作成されていること、正しいレジストリーデータがそこに設定されていることを確認します。

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

4.12.6.2.2. OpenShift ADP Controller のセグメンテーション違反

cloudstorage と **restic** の両方を有効にして DPA を設定すると、**openshift-adp-controller-manager** Pod がクラッシュし、Pod がクラッシュループのセグメンテーション違反で失敗するまで無期限に再起動します。

velero または **cloudstorage** は相互に排他的なフィールドであるため、どちらか一方だけ定義できます。

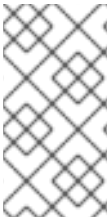
- **velero** と **cloudstorage** の両方が定義されている場合、**openshift-adp-controller-manager** は失敗します。
- **velero** と **cloudstorage** のいずれも定義されていない場合、**openshift-adp-controller-manager** は失敗します。

この問題の詳細は、OADP-1054 を参照してください。

4.12.6.2.2.1. OpenShift ADP Controller のセグメンテーション違反の回避策

DPA の設定時に、**velero** または **cloudstorage** のいずれかを定義する必要があります。DPA で両方の API を定義すると、**openshift-adp-controller-manager** Pod がクラッシュループのセグメンテーション違反で失敗します。

4.12.6.3. Velero プラグインがメッセージ "received EOF, stopping recv loop" を返す



注記

Velero プラグインは、別のプロセスとして開始されます。Velero 操作が完了すると、成功したかどうかにかかわらず終了します。デバッグログの **received EOF, stopping recv loop** メッセージは、プラグイン操作が完了したことを示します。エラーが発生したわけではありません。

関連情報

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

4.12.7. インストールの問題

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

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

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

原因

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

解決方法

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

4.12.7.2. 不正な AWS 認証情報

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

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

原因

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

解決方法

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

サンプル credentials-velero ファイル

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

1 AWS デフォルトプロファイル。

2 値を引用符 ("、') で囲まないでください。

4.12.8. OADP Operator の問題

OpenShift API for Data Protection (OADP) Operator では、解決できない問題が原因で問題が発生する可能性があります。

4.12.8.1. OADP Operator がサイレントに失敗する

OADP Operator の S3 バケットは空である可能性がありますが、**oc get po -n <OADP_Operator_namespace>** コマンドを実行すると、Operator のステータスが **Running** であることがわかります。この場合、Operator は実行中であると誤報告するため、**サイレントに失敗した** と言

われます。

原因

この問題は、クラウド認証情報で提供される権限が不十分な場合に発生します。

解決方法

バックアップ保存場所 (BSL) のリストを取得し、各 BSL のマニフェストで認証情報の問題を確認します。

手順

1. 次のコマンドのいずれかを実行して、BSL のリストを取得します。
   a. OpenShift CLI を使用する場合:

      ```
      $ oc get backupstoragelocation -A
      ```

   b. Velero CLI を使用する場合:

      ```
      $ velero backup-location get -n <OADP_Operator_namespace>
      ```

2. BSL のリストを使用し、次のコマンドを実行して各 BSL のマニフェストを表示し、各マニフェストにエラーがないか調べます。

   ```
   $ oc get backupstoragelocation -n <namespace> -o yaml
   ```

結果の例:

```
apiVersion: v1
items:
- apiVersion: velero.io/v1
  kind: BackupStorageLocation
  metadata:
    creationTimestamp: "2023-11-03T19:49:04Z"
    generation: 9703
    name: example-dpa-1
    namespace: openshift-adp-operator
    ownerReferences:
    - apiVersion: oadp.openshift.io/v1alpha1
      blockOwnerDeletion: true
      controller: true
      kind: DataProtectionApplication
      name: example-dpa
      uid: 0beeeaff-0287-4f32-bcb1-2e3c921b6e82
    resourceVersion: "24273698"
    uid: ba37cd15-cf17-4f7d-bf03-8af8655cea83
  spec:
    config:
      enableSharedConfig: "true"
      region: us-west-2
    credential:
      key: credentials
      name: cloud-credentials
```

```
    default: true
    objectStorage:
      bucket: example-oadp-operator
      prefix: example
    provider: aws
  status:
    lastValidationTime: "2023-11-10T22:06:46Z"
    message: "BackupStorageLocation \"example-dpa-1\" is unavailable: rpc
      error: code = Unknown desc = WebIdentityErr: failed to retrieve credentials\ncaused
      by: AccessDenied: Not authorized to perform sts:AssumeRoleWithWebIdentity\n\tstatus
      code: 403, request id: d3f2e099-70a0-467b-997e-ff62345e3b54"
    phase: Unavailable
kind: List
metadata:
  resourceVersion: ""
```

4.12.9. OADP タイムアウト

タイムアウトを延長すると、複雑なプロセスやリソースを大量に消費するプロセスが途中で終了することなく正常に完了できます。この設定により、エラー、再試行、または失敗の可能性を減らすことができます。

過度に長いタイムアウトを設定しないように、論理的な方法でタイムアウト延長のバランスをとってください。過度に長いと、プロセス内の根本的な問題が隠れる可能性があります。プロセスのニーズとシステム全体のパフォーマンスを満たす適切なタイムアウト値を慎重に検討して、監視してください。

次に、さまざまな OADP タイムアウトと、これらのパラメーターをいつどのように実装するかの手順を示します。

4.12.9.1. Restic タイムアウト

timeout は Restic タイムアウトを定義します。デフォルト値は **1h** です。

以下のシナリオでは Restic **timeout** を使用します。

- PV データの使用量の合計が 500GB を超える Restic バックアップの場合。
- バックアップが次のエラーでタイムアウトになる場合:

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all PodVolumeBackups to complete"
```

手順

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

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

```
    restic:
      timeout: 1h
# ...
```

4.12.9.2. Velero リソースのタイムアウト

resourceTimeout は Velero カスタムリソース定義 (CRD) の可用性、 **volumeSnapshot** の削除、リポジトリーの可用性など、タイムアウトが発生する前に複数の Velero リソースを待機する時間を定義します。デフォルトは **10m** です。

次のシナリオでは **resourceTimeout** を使用します。

- 合計 PV データ使用量が 1TB を超えるバックアップの場合。このパラメーターは、バックアップを完了としてマークする前に、Velero が Container Storage Interface (CSI) スナップショットをクリーンアップまたは削除しようとするときのタイムアウト値として使用されます。
    - このクリーンアップのサブタスクは VSC にパッチを適用しようとします。このタイムアウトはそのタスクに使用できます。
- Restic または Kopia のファイルシステムベースのバックアップのバックアップリポジトリーを作成または準備できるようにするため。
- カスタムリソース (CR) またはバックアップからリソースを復元する前に、クラスター内で Velero CRD が利用可能かどうかを確認します。

手順

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

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: <dpa_name>
spec:
  configuration:
    velero:
      resourceTimeout: 10m
# ...
```

4.12.9.3. Data Mover のタイムアウト

timeout は、**VolumeSnapshotBackup** および **VolumeSnapshotRestore** を完了するためにユーザーが指定したタイムアウトです。デフォルト値は **10m** です。

次のシナリオでは、Data Mover **timeout** を使用します。

- **VolumeSnapshotBackups** (VSB) および **VolumeSnapshotRestores** (VSR) を作成する場合は、10 分後にタイムアウトします。
- 合計 PV データ使用量が 500GB を超える大規模環境向け。**1h** のタイムアウトを設定します。
- **VolumeSnapshotMover** (VSM) プラグインを使用します。
- OADP 1.1.x のみ。

手順

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

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: <dpa_name>
spec:
  features:
    dataMover:
      timeout: 10m
# ...
```

4.12.9.4. CSI スナップショットのタイムアウト

CSISnapshotTimeout は、タイムアウトとしてエラーを返す前に、**CSI VolumeSnapshot** ステータスが **ReadyToUse** になるまで待機する作成時の時間を指定します。デフォルト値は **10m** です。

以下のシナリオでは、**CSISnapshotTimeout** を使用します。

- CSI プラグイン。
- スナップショットの作成に 10 分以上かかる可能性がある非常に大規模なストレージボリュームの場合。ログにタイムアウトが見つかった場合は、このタイムアウトを調整します。

注記

通常、**CSISnapshotTimeout** のデフォルト値は、デフォルト設定で大規模なストレージボリュームに対応できるため、調整する必要はありません。

手順

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

```
apiVersion: velero.io/v1
kind: Backup
metadata:
 name: <backup_name>
spec:
 csiSnapshotTimeout: 10m
# ...
```

4.12.9.5. Velero におけるアイテム操作のデフォルトタイムアウト

defaultItemOperationTimeout では、非同期の **BackupItemActions** および **RestoreItemActions** がタイムアウトになる前に完了するのを待機する時間を定義します。デフォルト値は **1h** です。

以下のシナリオでは、**defaultItemOperationTimeout** を使用します。

- Data Mover 1.2.x のみ。

- 特定のバックアップまたは復元が非同期アクションの完了を待機する時間を指定します。OADP 機能のコンテキストでは、この値は、Container Storage Interface (CSI) Data Mover 機能に関連する非同期アクションに使用されます。

- **defaultItemOperationTimeout** が、**defaultItemOperationTimeout** を使用して Data Protection Application (DPA) に定義されている場合、バックアップおよび復元操作の両方に適用されます。**itemOperationTimeout** では、以下の Item operation timeout - restore セクションおよび Item operation timeout - backup セクションで説明されているように、バックアップのみを定義するか、これらの CR の復元のみを定義できます。

手順

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

```
apiVersion: oadp.openshift.io/v1alpha1
kind: DataProtectionApplication
metadata:
 name: <dpa_name>
spec:
  configuration:
    velero:
      defaultItemOperationTimeout: 1h
# ...
```

4.12.9.6. アイテム操作のタイムアウト - 復元

ItemOperationTimeout は **RestoreItemAction** 操作の待機に使用される時間を指定します。デフォルト値は **1h** です。

以下のシナリオでは、復元 **ItemOperationTimeout** を使用します。

- Data Mover 1.2.x のみ。

- Data Mover の場合は、**BackupStorageLocation** にアップロードおよびダウンロードします。タイムアウトに達しても復元アクションが完了しない場合、失敗としてマークされます。ストレージボリュームのサイズが大きいため、タイムアウトの問題が原因で Data Mover の操作が失敗する場合は、このタイムアウト設定を増やす必要がある場合があります。

手順

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

```
apiVersion: velero.io/v1
kind: Restore
metadata:
 name: <restore_name>
spec:
 itemOperationTimeout: 1h
# ...
```

4.12.9.7. アイテム操作のタイムアウト - バックアップ

ItemOperationTimeout は、非同期 **BackupItemAction** 操作の待機時間を指定します。デフォルト値は **1h** です。

次のシナリオでは、バックアップ **ItemOperationTimeout** を使用します。

- Data Mover 1.2.x のみ。
- Data Mover の場合は、**BackupStorageLocation** にアップロードおよびダウンロードします。タイムアウトに達してもバックアップアクションが完了しない場合は、失敗としてマークされます。ストレージボリュームのサイズが大きいため、タイムアウトの問題が原因で Data Mover の操作が失敗する場合は、このタイムアウト設定を増やす必要がある場合があります。

手順

- 次の例のように、**Backuo** CR マニフェストの **Backup.spec.itemOperationTimeout** ブロックの値を編集します。

```
apiVersion: velero.io/v1
kind: Backup
metadata:
 name: <backup_name>
spec:
 itemOperationTimeout: 1h
# ...
```

4.12.10. 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。

4.12.10.1. 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。

4.12.10.2. 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。

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

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

原因

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

```
time="2023-02-17T16:33:13Z" level=error msg="Error backing up item" backup=openshift-adp/user1-backup-check5 error="error executing custom action (groupResource=persistentvolumeclaims, namespace=busy1, name=pvc1-user1): rpc error: code = Unknown desc = failed to get volumesnapshotclass for storageclass ocs-storagecluster-ceph-rbd: failed to get volumesnapshotclass for provisioner openshift-storage.rbd.csi.ceph.com, ensure that the desired volumesnapshot class has the velero.io/csi-volumesnapshot-class label" logSource="/remote-source/velero/app/pkg/backup/backup.go:417" name=busybox-79799557b5-vprq
```

解決方法

1. **Backup** CR を削除します。

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

2. 必要に応じて、**BackupStorageLocation** に保存されているデータをクリーンアップして、領域を解放します。

3. ラベル **velero.io/csi-volumesnapshot-class=true** を **VolumeSnapshotClass** オブジェクトに適用します。

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

4. 新しい **Backup** CR を作成します。

4.12.11. 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。

4.12.11.1. 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。

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

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

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

原因

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

解決方法

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

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

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

```
time="2021-12-29T18:29:14Z" level=info msg="1 errors
encountered backup up item" backup=velero/backup65
logSource="pkg/backup/backup.go:431" name=mysql-7d99fc949-qbkds
time="2021-12-29T18:29:14Z" level=error msg="Error backing up item"
backup=velero/backup65 error="pod volume backup failed: error running
restic backup, stderr=Fatal: unable to open config file: Stat: The
specified key does not exist.\nIs there a repository at the following
location?\ns3:http://minio-minio.apps.mayap-oadp-
veleo-1234.qe.devcluster.openshift.com/mayapvelerooadp2/velero1/
restic/mysql-persistent\n: exit status 1" error.file="/remote-source/
src/github.com/vmware-tanzu/velero/pkg/restic/backupper.go:184"
error.function="github.com/vmware-tanzu/velero/
pkg/restic.(*backupper).BackupPodVolumes"
logSource="pkg/backup/backup.go:435" name=mysql-7d99fc949-qbkds
```

4.12.11.3. PSA ポリシーの変更により、OCP 4.14 での Restic 復元が部分的に失敗する

OpenShift Container Platform 4.14 は、Restic 復元プロセス中に Pod の readiness を妨げる可能性がある Pod Security Admission (PSA) ポリシーを強制します。

Pod の作成時に **SecurityContextConstraints** (SCC) リソースが見つからず、Pod 上の PSA ポリシーが必要な標準を満たすように設定されていないと、Pod の許可は拒否されます。

この問題は、Velero のリソース復元順序が原因で発生します。

サンプルエラー

```
\"level=error\" in line#2273: time=\"2023-06-12T06:50:04Z\"
level=error msg=\"error restoring mysql-869f9f44f6-tp5lv: pods\\\
"mysql-869f9f44f6-tp5lv\\\" is forbidden: violates PodSecurity\\\
"restricted:v1.24\\\": privil eged (container \\\"mysql\\\
" must not set securityContext.privileged=true),
allowPrivilegeEscalation != false (containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.allowPrivilegeEscalation=false), unrestricted
capabilities (containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.capabilities.drop=[\\\"ALL\\\"]), seccompProfile
(pod or containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.seccompProfile.type to \\\
"RuntimeDefault\\\" or \\\"Localhost\\\")\" logSource=\"/remote-
source/velero/app/pkg/restore/restore.go:1388\" restore=openshift-adp/todolist-backup-0780518c-
08ed-11ee-805c-0a580a80e92c\n
velero container contains \"level=error\" in line#2447: time=\"2023-06-12T06:50:05Z\"
level=error msg=\"Namespace todolist-mariadb,
resource restore error: error restoring pods/todolist-mariadb/mysql-869f9f44f6-tp5lv: pods \\\
"mysql-869f9f44f6-tp5lv\\\" is forbidden: violates PodSecurity \\\"restricted:v1.24\\\": privileged
(container \\\
"mysql\\\" must not set securityContext.privileged=true),
allowPrivilegeEscalation != false (containers \\\
"restic-wait\\\",\\\"mysql\\\" must set securityContext.allowPrivilegeEscalation=false), unrestricted
capabilities (containers \\\
"restic-wait\\\", \\\"mysql\\\" must set securityContext.capabilities.drop=[\\\"ALL\\\"]), seccompProfile
(pod or containers \\\
```

```
"restic-wait\\\", \\\"mysql\\\" must set securityContext.seccompProfile.type to \\\
"RuntimeDefault\\\" or \\\"Localhost\\\")\"
logSource=\"/remote-source/velero/app/pkg/controller/restore_controller.go:510\"
restore=openshift-adp/todolist-backup-0780518c-08ed-11ee-805c-0a580a80e92c\n]",
```

解決方法

1. DPA カスタムリソース (CR) で、Velero サーバーの **restore-resource-priorities** フィールドを確認または設定して、**securitycontextconstraints** がリソースのリストの **pods** の前に順番にリストされていることを確認します。

```
$ oc get dpa -o yaml
```

DPA CR の例

```
# ...
configuration:
  restic:
    enable: true
  velero:
    args:
      restore-resource-priorities:
'securitycontextconstraints,customresourcedefinitions,namespaces,storageclasses,volumesnaps
hotclass.snapshot.storage.k8s.io,volumesnapshotcontents.snapshot.storage.k8s.io,volumesnap
hots.snapshot.storage.k8s.io,datauploads.velero.io,persistentvolumes,persistentvolumeclaims,s
rviceaccounts,secrets,configmaps,limitranges,pods,replicasets.apps,clusterclasses.cluster.x-
k8s.io,endpoints,services,-,clusterbootstraps.run.tanzu.vmware.com,clusters.cluster.x-
k8s.io,clusterresourcesets.addons.cluster.x-k8s.io' (1)
    defaultPlugins:
    - gcp
    - openshift
```

(1) 既存のリストアリソース優先順位リストがある場合は、その既存のリストを完全なリストと組み合わせてください。

2. デプロイメントの警告が発生しないように、Fixing PodSecurity Admission warnings for deployments で説明されているように、アプリケーション Pod のセキュリティー標準が調整されていることを確認します。アプリケーションがセキュリティー標準に準拠していない場合は、SCC に関係なくエラーが発生する可能性があります。

注記

この解決策は一時的なものであり、これに対処するために継続的な議論が進行中です。

関連情報

- Fixing PodSecurity Admission warnings for deployments

4.12.12. 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、メトリック、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 メトリックデータダンプは、Prometheus によって収集されたメトリック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している。
- OpenShift CLI (**oc**) がインストールされている。
- Red Hat Enterprise Linux (RHEL) 8.x と OADP 1.2 を使用している。
- Red Hat Enterprise Linux (RHEL) 9.x と OADP 1.3 を使用している。

手順

1. **must-gather** データを保存するディレクトリーに移動します。
2. 次のデータ収集オプションのいずれかに対して、**oc adm must-gather** コマンドを実行します。
   - Prometheus メトリックを含む、完全な **must-gather** データ収集:
     a. OADP 1.2 の場合は、次のコマンドを実行します。

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.2
     ```

     b. OADP 1.3 の場合は、次のコマンドを実行します。

     ```
     $ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3
     ```

     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:v1.1 \
       -- /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:v1.1 \
  -- /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 メトリックデータダンプ:

a. OADP 1.2 の場合は、次のコマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel8:v1.2 --
/usr/bin/gather_metrics_dump
```

b. OADP 1.3 の場合は、次のコマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 --
/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。

関連情報

- クラスターデータの収集

4.12.12.1. 安全でない TLS 接続で must-gather を使用する

カスタム CA 証明書が使用されている場合、**must-gather** Pod は **velero logs/describe** の出力を取得できません。安全でない TLS 接続で **must-gather** ツールを使用するには、**gather_without_tls** フラグを **must-gather** コマンドに渡します。

手順

- 次のコマンドを使用して、値を **true** に設定した **gather_without_tls** フラグを **must-gather** ツールに渡します。

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 --
/usr/bin/gather_without_tls <true/false>
```

デフォルトでは、フラグの値は **false** に設定されています。安全でない TLS 接続を許可するには、値を **true** に設定します。

4.12.12.2. must-gather ツールを使用する場合のオプションの組み合わせ

現時点では、たとえば安全でない TLS 接続を許可しながらタイムアウトしきい値を指定するなど、must-gather スクリプトを組み合わせることはできません。状況によっては、次の例のように must-gather コマンドラインで内部変数を設定することで、この制限を回避できます。

```
oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 -- skip_tls=true
/usr/bin/gather_with_timeout <timeout_value_in_seconds>
```

この例では、**gather_with_timeout** スクリプトを実行する前に、**skip_tls** 変数を設定します。その結果、**Gather_with_timeout** と **Gather_without_tls** が組み合わされます。

この方法で指定できる他の変数は次のとおりです。

- **logs_since**、デフォルト値は **72h**
- **request_timeout**、デフォルト値は **0s**

DataProtectionApplication カスタムリソース (CR) が **s3Url** および **insecureSkipTLS: true** で設定されている場合、CA 証明書がないため、CR は必要なログを収集しません。これらのログを収集するには、次のオプションを指定して **must-gather** コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/oadp/oadp-mustgather-rhel9:v1.3 --
/usr/bin/gather_without_tls true
```

4.12.13. OADP モニタリング

OpenShift Container Platform は、ユーザーと管理者がクラスターを効果的に監視および管理できるだけでなく、クラスター上で実行されているユーザーアプリケーションやサービスのワークロードパフォーマンスを監視および分析できるようにする監視スタックを提供します (イベント発生時のアラートの受信など)。

関連情報

- モニタリングスタック

4.12.13.1. OADP モニタリングの設定

OADP Operator は、OpenShift モニタリングスタックによって提供される OpenShift ユーザーワークロードモニタリングを利用して、Velero サービスエンドポイントからメトリックを取得します。モニタリングスタックを使用すると、ユーザー定義のアラートルールを作成したり、OpenShift メトリッククエリーフロントエンドを使用してメトリックをクエリーしたりできます。

ユーザーワークロードモニタリングを有効にすると、Grafana などの Prometheus 互換のサードパーティー UI を設定して使用し、Velero メトリックを視覚化することができます。

メトリックをモニタリングするには、ユーザー定義プロジェクトのモニタリングを有効にし、**openshift-adp** namespace に存在するすでに有効な OADP サービスエンドポイントからそれらのメトリックを取得する **ServiceMonitor** リソースを作成する必要があります。

前提条件

- **cluster-admin** パーミッションを持つアカウントを使用して OpenShift Container Platform クラスターにアクセスできる。
- クラスター監視 config map が作成されました。

手順

1. **openshift-monitoring** namespace で **cluster-monitoring-config ConfigMap** オブジェクトを編集します。

```
$ oc edit configmap cluster-monitoring-config -n openshift-monitoring
```

2. **data** セクションの **config.yaml** フィールドで、**enableUserWorkload** オプションを追加または有効にします。

```
apiVersion: v1
data:
  config.yaml: |
    enableUserWorkload: true (1)
kind: ConfigMap
metadata:
# ...
```

(1) このオプションを追加するか、**true** に設定します

3. しばらく待って、**openshift-user-workload-monitoring** namespace で次のコンポーネントが稼働しているかどうかを確認して、ユーザーワークロードモニタリングのセットアップを検証します。

```
$ oc get pods -n openshift-user-workload-monitoring
```

出力例

```
NAME                                   READY   STATUS    RESTARTS   AGE
prometheus-operator-6844b4b99c-b57j9   2/2     Running   0          43s
prometheus-user-workload-0             5/5     Running   0          32s
prometheus-user-workload-1             5/5     Running   0          32s
thanos-ruler-user-workload-0           3/3     Running   0          32s
thanos-ruler-user-workload-1           3/3     Running   0          32s
```

4. **openshift-user-workload-monitoring** に **user-workload-monitoring-config** ConfigMap が存在することを確認します。存在する場合、この手順の残りの手順はスキップしてください。

```
$ oc get configmap user-workload-monitoring-config -n openshift-user-workload-monitoring
```

出力例

```
Error from server (NotFound): configmaps "user-workload-monitoring-config" not found
```

5. ユーザーワークロードモニタリングの **user-workload-monitoring-config ConfigMap** オブジェクトを作成し、**2_configure_user_workload_monitoring.yaml** ファイル名に保存します。

出力例

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: user-workload-monitoring-config
```

```
  namespace: openshift-user-workload-monitoring
data:
  config.yaml: |
```

6. **2_configure_user_workload_monitoring.yaml** ファイルを適用します。

```
$ oc apply -f 2_configure_user_workload_monitoring.yaml
configmap/user-workload-monitoring-config created
```

4.12.13.2. OADP サービスモニターの作成

OADP は、DPA の設定時に作成される **openshift-adp-velero-metrics-svc** サービスを提供します。ユーザーのワークロード監視で使用されるサービスモニターは、定義されたサービスを指す必要があります。

次のコマンドを実行して、サービスの詳細を取得します。

手順

1. **openshift-adp-velero-metrics-svc** サービスが存在することを確認します。これには、**ServiceMonitor** オブジェクトのセレクターとして使用される **app.kubernetes.io/name=velero** ラベルが含まれている必要があります。

```
$ oc get svc -n openshift-adp -l app.kubernetes.io/name=velero
```

出力例

```
NAME                           TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
openshift-adp-velero-metrics-svc   ClusterIP   172.30.38.244   <none>        8085/TCP   1h
```

2. 既存のサービスラベルと一致する **ServiceMonitor** YAML ファイルを作成し、そのファイルを **3_create_oadp_service_monitor.yaml** として保存します。サービスモニターは **openshift-adp-velero-metrics-svc** サービスが存在する **openshift-adp** namespace に作成されます。

ServiceMonitor オブジェクトの例

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    app: oadp-service-monitor
  name: oadp-service-monitor
  namespace: openshift-adp
spec:
  endpoints:
  - interval: 30s
    path: /metrics
    targetPort: 8085
    scheme: http
  selector:
    matchLabels:
      app.kubernetes.io/name: "velero"
```

3. **3_create_oadp_service_monitor.yaml** ファイルを適用します。

```
$ oc apply -f 3_create_oadp_service_monitor.yaml
```

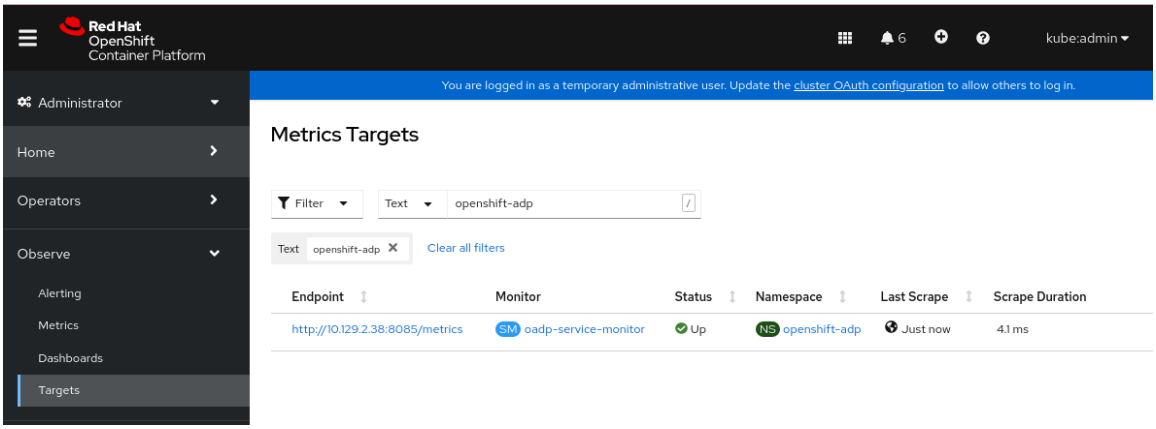
出力例

```
servicemonitor.monitoring.coreos.com/oadp-service-monitor created
```

検証

- OpenShift Container Platform Web コンソールの **Administrator** パースペクティブを使用して、新しいサービスモニターが **Up** 状態であることを確認します。
  a. **Observe → Targets** ページに移動します。
  b. **Filter** が選択されていないこと、または **User** ソースが選択されていることを確認し、**Text** 検索フィールドに **openshift-adp** と入力します。
  c. サービスモニターの **Status** のステータスが **Up** であることを確認します。

図4.1 OADP メトリックのターゲット



4.12.13.3. アラートルールの作成

OpenShift Container Platform モニタリングスタックでは、アラートルールを使用して設定されたアラートを受信できます。OADP プロジェクトのアラートルールを作成するには、ユーザーワークロードの監視で収集されたメトリックの 1 つを使用します。

手順

1. サンプル **OADPBackupFailing** アラートを含む **PrometheusRule** YAML ファイルを作成し、**4_create_oadp_alert_rule.yaml** として保存します。

OADPBackupFailing アラートのサンプル

```
apiVersion: monitoring.coreos.com/v1
kind: PrometheusRule
metadata:
  name: sample-oadp-alert
  namespace: openshift-adp
spec:
  groups:
```

```
- name: sample-oadp-backup-alert
  rules:
  - alert: OADPBackupFailing
    annotations:
      description: 'OADP had {{$value | humanize}} backup failures over the last 2 hours.'
      summary: OADP has issues creating backups
    expr: |
      increase(velero_backup_failure_total{job="openshift-adp-velero-metrics-svc"}[2h]) > 0
    for: 5m
    labels:
      severity: warning
```

このサンプルでは、アラートは次の条件で表示されます。

- 過去 2 時間に失敗した新しいバックアップの数が 0 より大きく増加しており、その状態が少なくとも 5 分間継続します。
- 最初の増加時間が 5 分未満の場合、アラートは **Pending** 状態になり、その後、**Firing** 状態に変わります。

2. **4_create_oadp_alert_rule.yaml** ファイルを適用して、**openshift-adp** namespace に **PrometheusRule** オブジェクトを作成します。

```
$ oc apply -f 4_create_oadp_alert_rule.yaml
```

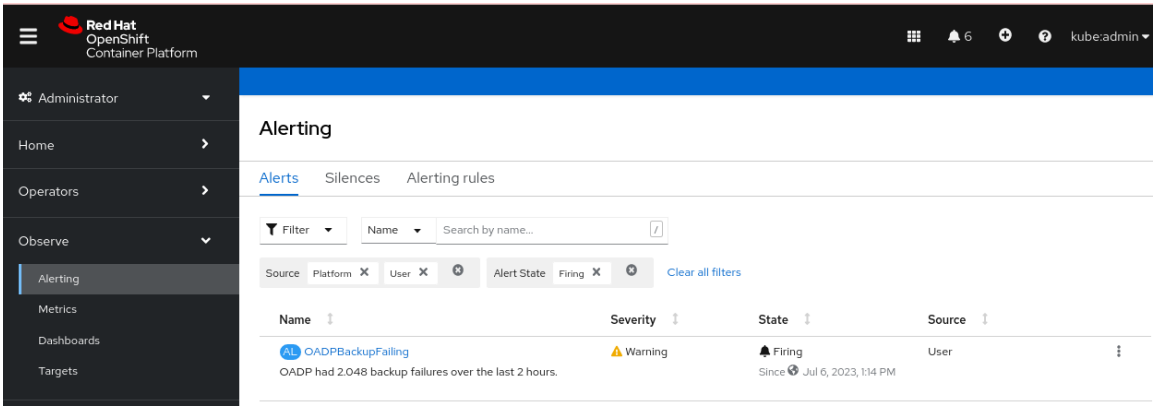
出力例

```
prometheusrule.monitoring.coreos.com/sample-oadp-alert created
```

検証

- アラートがトリガーされた後は、次の方法でアラートを表示できます。
  - **Developer** パースペクティブで、**Observe** メニューを選択します。
  - **Observe → Alerting** メニューの下の **Administrator** パースペクティブで、**Filter** ボックスの **User** を選択します。それ以外の場合、デフォルトでは **Platform** アラートのみが表示されます。

図4.2 OADP バックアップ失敗アラート



関連情報

- アラートの管理

4.12.13.4. 利用可能なメトリックのリスト

これらは、OADP によって提供されるメトリックとその Type のリストです。

メトリクス名	説明	型
kopia_content_cache_hit_bytes	キャッシュから取得したバイト数	カウンター
kopia_content_cache_hit_count	コンテンツがキャッシュから取得された回数	カウンター
kopia_content_cache_malformed	不正なコンテンツがキャッシュから読み取られた回数	カウンター
kopia_content_cache_miss_count	コンテンツがキャッシュ内で見つからずフェッチされた回数	カウンター
kopia_content_cache_missed_bytes	基盤となるストレージから取得したバイト数	カウンター
kopia_content_cache_miss_error_count	基盤となるストレージでコンテンツが見つからなかった回数	カウンター
kopia_content_cache_store_error_count	コンテンツをキャッシュに保存できなかった回数	カウンター
kopia_content_get_bytes	**GetContent()** を使用して取得されたバイト数	カウンター
kopia_content_get_count	**GetContent()** が呼び出された回数	カウンター
kopia_content_get_error_count	**GetContent()** が呼び出され、結果がエラーであった回数	カウンター
kopia_content_get_not_found_count	**GetContent()** が呼び出されて結果が見つからなかった回数	カウンター
kopia_content_write_bytes	**WriteContent()** に渡されるバイト数	カウンター
kopia_content_write_count	**WriteContent()** が呼び出された回数	カウンター
velero_backup_attempt_total	試行されたバックアップの合計数	カウンター

メトリクス名	説明	型
velero_backup_deletion_attempt_total	試行されたバックアップ削除の合計数	カウンター
velero_backup_deletion_failure_total	失敗したバックアップ削除の合計数	カウンター
velero_backup_deletion_success_total	成功したバックアップ削除の合計数	カウンター
velero_backup_duration_seconds	バックアップの完了にかかる時間 (秒単位)	ヒストグラム
velero_backup_failure_total	失敗したバックアップの合計数	カウンター
velero_backup_items_errors	バックアップ中に発生したエラーの合計数	ゲージ
velero_backup_items_total	バックアップされたアイテムの総数	ゲージ
velero_backup_last_status	バックアップの最終ステータス。値 1 は成功、値 0 は成功です。	ゲージ
velero_backup_last_successful_timestamp	最後にバックアップが正常に実行された時刻、秒単位の Unix タイムスタンプ	ゲージ
velero_backup_partial_failure_total	部分的に失敗したバックアップの合計数	カウンター
velero_backup_success_total	成功したバックアップの合計数	カウンター
velero_backup_tarball_size_bytes	バックアップのサイズ (バイト単位)	ゲージ
velero_backup_total	既存のバックアップの現在の数	ゲージ
velero_backup_validation_failure_total	検証に失敗したバックアップの合計数	カウンター
velero_backup_warning_total	警告されたバックアップの総数	カウンター
velero_csi_snapshot_attempt_total	CSI が試行したボリュームスナップショットの合計数	カウンター

メトリクス名	説明	型
velero_csi_snapshot_failure_total	CSI で失敗したボリュームスナップショットの総数	カウンター
velero_csi_snapshot_success_total	CSI が成功したボリュームスナップショットの総数	カウンター
velero_restore_attempt_total	試行された復元の合計数	カウンター
velero_restore_failed_total	失敗したリストアの合計数	カウンター
velero_restore_partial_failure_total	部分的に失敗したリストアの合計数	カウンター
velero_restore_success_total	成功した復元の合計数	カウンター
velero_restore_total	現在の既存のリストアの数	ゲージ
velero_restore_validation_failed_total	検証に失敗したリストアの失敗の合計数	カウンター
velero_volume_snapshot_attempt_total	試行されたボリュームスナップショットの総数	カウンター
velero_volume_snapshot_failure_total	失敗したボリュームスナップショットの総数	カウンター
velero_volume_snapshot_success_total	成功したボリュームスナップショットの総数	カウンター

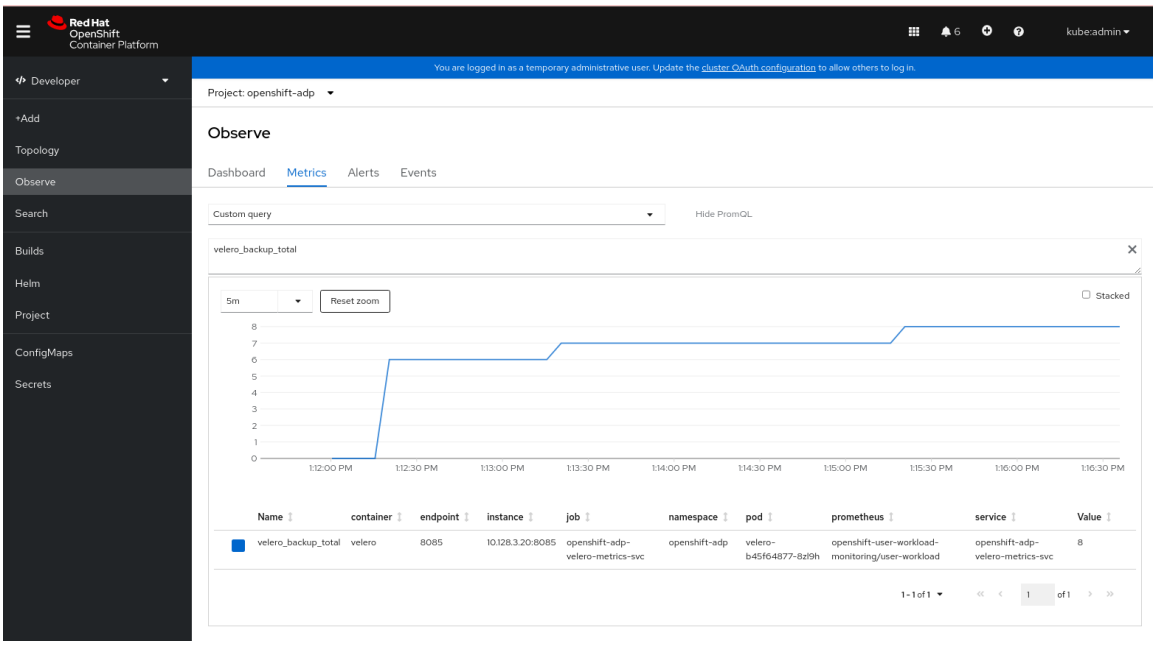
4.12.13.5. Observe UI を使用したメトリックの表示

OpenShift Container Platform Web コンソールのメトリックは、**Administrator** または **Developer** パースペクティブから表示できます。これらのパースペクティブには、**openshift-adp** プロジェクトへのアクセス権が必要です。

手順

- Observe → Metrics ページに移動します。
  - Developer パースペクティブを使用している場合は、次の手順に従います。
    a. **Custom query** を選択するか、**Show PromQL** リンクをクリックします。
    b. クエリーを入力し、**Enter** をクリックします。
  - Administrator パースペクティブを使用している場合は、テキストフィールドに式を入力し、**Run Queries** を選択します。

図4.3 OADP メトリッククエリー



4.13. OADP で使用される API

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

- Velero API
- OADP API

4.13.1. Velero API

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

4.13.2. OADP API

次の表は、OADP API の構造を示しています。

表4.2 DataProtectionApplicationSpec

プロパティー	型	説明
backupLocations	[] **BackupLocation**	**BackupStorageLocations** に使用する設定のリストを定義します。
snapshotLocations	[] **SnapshotLocation**	**VolumeSnapshotLocations** に使用する設定のリストを定義します。

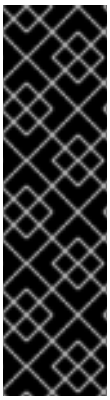
プロパティー	型	説明
unsupportedOverrides	map [UnsupportedImageKey] string	デプロイされた依存イメージを開発用にオーバーライドするために使用できます。オプションは、**veleroImageFqin**、**awsPluginImageFqin**、**openshiftPluginImageFqin**、**azurePluginImageFqin**、**gcpPluginImageFqin**、**csiPluginImageFqin**、**dataMoverImageFqin**、**resticRestoreImageFqin**、**kubevirtPluginImageFqin**、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

OADP API の完全なスキーマ定義。

表4.3 BackupLocation

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

プロパティー	型	説明
bucket	*CloudStorageLocation	[テクノロジープレビュー] 一部のクラウドストレージプロバイダーで、バックアップストレージの場所として使用するバケットの作成を自動化します。



重要

bucket パラメーターはテクノロジープレビュー機能としてのみ提供されます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

BackupLocation タイプの完全なスキーマ定義。

表4.4 SnapshotLocation

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

SnapshotLocation タプの完全なスキーマ定義。

表4.5 ApplicationConfig

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

ApplicationConfig タイプの完全なスキーマ定義。

表4.6 VeleroConfig

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

VeleroConfig タイプの完全なスキーマ定義。

表4.7 CustomPlugin

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

CustomPlugin タイプの完全なスキーマ定義。

表4.8 ResticConfig

プロパティー	型	説明
enable	*bool	**true** に設定すると、Restic を使用したバックアップと復元が有効になります。**false** に設定すると、スナップショットが必要になります。
supplementalGroups	[]int64	**Restic** Pod に適用される Linux グループを定義します。
timeout	string	Restic タイムアウトを定義するユーザー指定の期間文字列。デフォルト値は **1hr** (1 時間) です。期間文字列は、符号付きの場合もある 10 進数のシーケンスであり、それぞれに **300ms**、-1.5h` または **2h45m** などのオプションの分数と単位接尾辞が付いています。有効な時間単位は、**ns**、**us** (または **µs**)、**ms**、**s**、**m**、および **h** です。
podConfig	***PodConfig**	**Restic** Pod の設定を定義します。

ResticConfig タイプの完全なスキーマ定義。

表4.9 PodConfig

プロパティー	型	説明
nodeSelector	map [string] string	**Velero podSpec** または **Restic podSpec** に提供される **nodeSelector** を定義します。
tolerations	[]Toleration	Velero デプロイメントまたは Restic **daemonset** に適用される toleration のリストを定義します。

プロパティー	型	説明
resourceAllocations	ResourceRequirements	Setting Velero CPU and memory resource allocations の説明に従って、**Velero** Pod または **Restic** Pod の特定のリソースの **limits** および **requests** を設定します。
labels	map [string] string	Pod に追加するラベル。

PodConfig タイプの完全なスキーマ定義。

表4.10 機能

プロパティー	型	説明
dataMover	***DataMover**	Data Mover の設定を定義します。

Features タイプの完全なスキーマ定義。

表4.11 DataMover

プロパティー	型	説明
enable	bool	**true** に設定すると、ボリュームスナップショットムーバーコントローラーと変更された CSI Data Mover プラグインがデプロイされます。**false** に設定すると、これらはデプロイされません。
credentialName	string	Data Mover のユーザー指定の Restic **Secret** 名。
timeout	string	**VolumeSnapshotBackup** と **VolumeSnapshotRestore** が完了するまでのユーザー指定の期間文字列。デフォルトは **10m** (10分) です。期間文字列は、符号付きの場合もある 10 進数のシーケンスであり、それぞれに **300ms**、-1.5h` または **2h45m** などのオプションの分数と単位接尾辞が付いています。有効な時間単位は、**ns**、**us** (または **µs**)、**ms**、**s**、**m**、および **h** です。

OADP API の詳細については、OADP Operator を参照してください。

4.14. OADP の高度な特徴と機能

このドキュメントでは、OpenShift API for Data Protection (OADP) の高度な特徴と機能に関する情報を提供します。

4.14.1. 同一クラスター上での異なる Kubernetes API バージョンの操作

4.14.1.1. クラスター上の Kubernetes API グループバージョンのリスト表示

ソースクラスターは複数のバージョンの API を提供する場合があり、これらのバージョンの 1 つが優先 API バージョンになります。たとえば、**Example** という名前の API を持つソースクラスターは、**example.com/v1** および **example.com/v1beta2** API グループで使用できる場合があります。

Velero を使用してそのようなソースクラスターをバックアップおよび復元する場合、Velero は、Kubernetes API の優先バージョンを使用するリソースのバージョンのみをバックアップします。

上記の例では、**example.com/v1** が優先 API である場合、Velero は **example.com/v1** を使用するリソースのバージョンのみをバックアップします。さらに、Velero がターゲットクラスターでリソースを復元するには、ターゲットクラスターで使用可能な API リソースのセットに **example.com/v1** が登録されている必要があります。

したがって、ターゲットクラスター上の Kubernetes API グループバージョンのリストを生成して、優先 API バージョンが使用可能な API リソースのセットに登録されていることを確認する必要があります。

手順

- 以下のコマンドを入力します。

```
$ oc api-resources
```

4.14.1.2. API グループバージョンの有効化について

デフォルトでは、Velero は Kubernetes API の優先バージョンを使用するリソースのみをバックアップします。ただし、Velero には、この制限を克服する機能 (Enable API Group Versions) も含まれています。ソースクラスターでこの機能を有効にすると、Velero は優先バージョンだけでなく、クラスターでサポートされている **すべての** Kubernetes API グループバージョンをバックアップします。バージョンがバックアップ .tar ファイルに保存されると、目的のクラスターで復元できるようになります。

たとえば、**Example** という名前の API を持つソースクラスターが、**example.com/v1** および **example.com/v1beta2** API グループで使用でき、**example.com/v1** が優先 API だとします。

Enable API Group Versions 機能を有効にしないと、Velero は **Example** の優先 API グループバージョン (**example.com/v1**) のみをバックアップします。この機能を有効にすると、Velero は **example.com/v1beta2** もバックアップします。

宛先クラスターで Enable API Group Versions 機能が有効になっている場合、Velero は、API グループバージョンの優先順位に基づいて、復元するバージョンを選択します。

注記

Enable API Group Versions はまだベータ版です。

Velero は次のアルゴリズムを使用して API バージョンに優先順位を割り当てます。この場合、**1** は優先順位が最も高くなります。

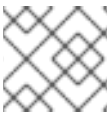
1. **宛先** クラスターの優先バージョン
2. source_ クラスターの優先バージョン
3. Kubernetes バージョンの優先順位が最も高い共通の非優先サポート対象バージョン

関連情報

- Enable API Group Versions Feature

4.14.1.3. Enable API Group Versions の使用

Velero の Enable API Group Versions 機能を使用して、優先バージョンだけでなく、クラスターでサポートされている **すべての** Kubernetes API グループバージョンをバックアップできます。



注記

Enable API Group Versions はまだベータ版です。

手順

- **EnableAPIGroupVersions** 機能フラグを設定します。

```
apiVersion: oadp.openshift.io/vialpha1
kind: DataProtectionApplication
...
spec:
  configuration:
    velero:
      featureFlags:
      - EnableAPIGroupVersions
```

関連情報

- Enable API Group Versions Feature

4.14.2. 1 つのクラスターからデータをバックアップし、別のクラスターに復元する

4.14.2.1. あるクラスターからのデータのバックアップと別のクラスターへの復元について

OpenShift API for Data Protection (OADP) は、同じ OpenShift Container Platform クラスター内のアプリケーションデータをバックアップおよび復元するように設計されています。Migration Toolkit for Containers (MTC) は、アプリケーションデータを含むコンテナーを 1 つの OpenShift Container Platform クラスターから別のクラスターに移行するように設計されています。

OADP を使用して、1 つの OpenShift Container Platform クラスターからアプリケーションデータをバックアップし、それを別のクラスターに復元できます。ただし、これを行うことは、MTC または OADP を使用して同じクラスター上でバックアップと復元を行うよりも複雑です。

OADP を使用して 1 つのクラスターからデータをバックアップし、それを別のクラスターに復元するには、OADP を使用して同じクラスター上でデータをバックアップおよび復元する場合に適用される前提条件と手順に加えて、次の要素を考慮する必要があります。

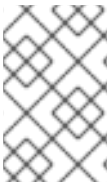
- Operator
- Velero の使用
- UID と GID の範囲

4.14.2.1.1. Operator

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

4.14.2.1.2. Velero の使用

OADP が構築されている Velero は、クラウドプロバイダー間での永続ボリュームスナップショットの移行をネイティブにサポートしていません。クラウドプラットフォーム間でボリュームスナップショットデータを移行するには、ファイルシステムレベルでボリュームの内容をバックアップする Velero Restic ファイルシステムバックアップオプションを有効にする **か、または** CSI スナップショットに OADP Data Mover を使用する必要があります。



注記

OADP 1.1 以前では、Velero Restic ファイルシステムのバックアップオプションは **restic** と呼ばれます。OADP 1.2 以降では、Velero Restic ファイルシステムのバックアップオプションは **file-system-backup** と呼ばれます。

- AWS リージョン間または Microsoft Azure リージョン間でデータを移行するには、Velero の File System Backup も使用する必要があります。
- Velero は、ソースクラスターより **前の** Kubernetes バージョンを使用したクラスターへのデータの復元をサポートしていません。
- 理論的には、移行元よりも **新しい** Kubernetes バージョンを備えた移行先にワークロードを移行することは可能ですが、カスタムリソースごとにクラスター間の API グループの互換性を考慮する必要があります。Kubernetes バージョンのアップグレードによりコアまたはネイティブ API グループの互換性が失われる場合は、まず影響を受けるカスタムリソースを更新する必要があります。

4.14.2.2. バックアップする Pod ボリュームの判断方法

ファイルシステムバックアップ (FSB) を使用してバックアップ操作を開始する前に、バックアップするボリュームが含まれる Pod を指定する必要があります。Velero では、このプロセスを適切な Pod ボリュームの "検出" と呼んでいます。

Velero は、Pod ボリュームの判断方法として 2 つのアプローチをサポートしています。

- **オプトインアプローチ**: オプトインアプローチでは、ボリュームをバックアップに含める (**オプトインする**) ことを示す必要があります。これを行うには、バックアップするボリュームを含む各 Pod に、そのボリュームの名前でラベルを付けます。Velero は、永続ボリューム (PV) を見つけると、そのボリュームをマウントした Pod を確認します。Pod にボリュームの名前が付いている場合、Velero は Pod をバックアップします。

- **オプトアウトアプローチ**: オプトアウトアプローチでは、バックアップから除外するボリュームを指定する必要があります。これを行うには、バックアップしないボリュームを含む各 Pod に、そのボリュームの名前でラベルを付けます。Velero は、PV を見つけると、そのボリュームをマウントした Pod を確認します。Pod にボリューム名のラベルが付いている場合、Velero はその Pod をバックアップしません。

4.14.2.2.1. 制限事項

- FSB は、**hostpath** ボリュームのバックアップと復元をサポートしていません。ただし、FSB はローカルボリュームのバックアップと復元をサポートします。

- Velero は、作成するすべてのバックアップリポジトリーに静的な共通暗号化キーを使用します。**この静的キーは、バックアップストレージにアクセスできれば、誰でもバックアップデータを復号化できることを意味します**。バックアップストレージへのアクセスを制限することが重要です。

- PVC の場合、すべての増分バックアップチェーンは Pod が再スケジュールされても維持されます。
  emptyDir ボリュームなどの **PVC ではない** Pod ボリュームの場合、たとえば **ReplicaSet** やデプロイメントなどによって Pod が削除または再作成されると、そのボリュームの次回のバックアップは増分バックアップではなく完全バックアップになります。Pod ボリュームのライフサイクルは、その Pod によって定義されると想定されます。

- バックアップデータは増分的に保存できますが、データベースなどの大きなファイルのバックアップには時間がかかることがあります。これは、FSB が重複排除を使用して、バックアップが必要な差分を見つけるためです。

- FSB は、Pod が実行されているノードのファイルシステムにアクセスすることで、ボリュームからデータを読み書きします。そのため、FSB は Pod からマウントされたボリュームのみバックアップでき、PVC から直接バックアップすることはできません。一部の Velero ユーザーは、Velero バックアップを実行する前に、無限スリープがある BusyBox や Alpine コンテナーなどのステージング Pod を実行して PVC と PV のペアをマウントすることで、この制限を克服しています。

- FSB では、ボリュームは **<hostPath>/<pod UID>** の下にマウントされ、**<hostPath>** が設定可能であると想定します。vCluster などの一部の Kubernetes システムでは、ボリュームを **<pod UID>** サブディレクトリーにマウントしないため、そのようなシステムでは VFSB は期待どおり機能しません。

4.14.2.2.2. オプトインメソッドを使用して Pod ボリュームをバックアップする

オプトインメソッドを使用して、ファイルシステムバックアップ (FSB) でバックアップする必要のあるボリュームを指定できます。これを行うには、**backup.velero.io/backup-volumes** コマンドを使用します。

手順

- バックアップするボリュームを 1 つ以上含む各 Pod で、次のコマンドを入力します。

```
$ oc -n <your_pod_namespace> annotate pod/<your_pod_name> \
  backup.velero.io/backup-volumes=<your_volume_name_1>, \ <your_volume_name_2>>,...,
<your_volume_name_n>
```

  ここでは、以下のようになります。

  <your_volume_name_x>

Pod 仕様の x 番目のボリュームの名前を指定します。

4.14.2.2.3. オプトアウトメソッドを使用して Pod ボリュームをバックアップする

オプトアウトアプローチを使用する場合、いくつかの例外を除き、すべての Pod ボリュームがファイルシステムバックアップ (FSB) を使用してバックアップされます。

- デフォルトのサービスアカウントトークン、シークレット、設定マップをマウントするボリューム。
- **hostPath** volumes

オプトアウトメソッドを使用して、**バックアップしない** ボリュームを指定できます。これを行うには、**backup.velero.io/backup-volumes-excludes** コマンドを使用します。

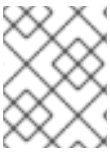
手順

- バックアップしないボリュームを 1 つ以上含む各 Pod で、次のコマンドを実行します。

```
$ oc -n <your_pod_namespace> annotate pod/<your_pod_name> \
  backup.velero.io/backup-volumes-excludes=<your_volume_name_1>, \
<your_volume_name_2>>,...,<your_volume_name_n>
```

ここでは、以下のようになります。

<your_volume_name_x>
Pod 仕様の x 番目のボリュームの名前を指定します。



注記

--default-volumes-to-fs-backup フラグを指定して **velero install** コマンドを実行することで、すべての Velero バックアップに対してこの動作を有効にできます。

4.14.2.3. UID と GID の範囲

あるクラスターからデータをバックアップし、それを別のクラスターに復元する場合、UID (ユーザー ID) および GID (グループ ID) の範囲で問題が発生する可能性があります。次のセクションでは、これらの潜在的な問題と軽減策について説明します。

問題点のまとめ

宛先クラスターによっては、namespace の UID と GID の範囲が変更される場合があります。OADP は、OpenShift UID 範囲のメタデータをバックアップおよび復元しません。バックアップされたアプリケーションに特定の UID が必要な場合は、復元時にその範囲が使用可能であることを確認してください。OpenShift の UID 範囲および GID 範囲の詳細は、A Guide to OpenShift and UIDs を参照してください。

問題の詳細

シェルコマンド **oc create namespace** を使用して OpenShift Container Platform でネームスペースを作成すると、OpenShift Container Platform は、使用可能な UID プールからの一意のユーザー ID (UID) 範囲、補足グループ (GID) 範囲、および一意の SELinux MCS ラベルを namespace に割り当てます。この情報は、クラスターの **metadata.annotations** フィールドに保存されます。この情報は、セキュリティーコンテキスト制約 (SCC) アノテーションの一部であり、次のコンポーネントで構成されています。

- **openshift.io/sa.scc.mcs**

- **openshift.io/sa.scc.supplemental-groups**
- **openshift.io/sa.scc.uid-range**

OADP を使用して namespace を復元すると、宛先クラスターの情報をリセットせずに、**metadata.annotations** 内の情報が自動的に使用されます。その結果、次のいずれかに該当する場合、ワークロードはバックアップデータにアクセスできない可能性があります。

- 他の SCC アノテーションを持つ既存の namespace が (たとえば別のクラスター上に) ある。この場合、OADP はバックアップ中に、復元する namespace ではなく既存の namespace を使用します。
- バックアップ中にラベルセレクターが使用されたが、ワークロードが実行された namespace にそのラベルがない。この場合、OADP はその namespace をバックアップしませんが、バックアップされた namespace のアノテーションを含まない新しい namespace を復元中に作成します。これにより、新しい UID 範囲が namespace に割り当てられます。
  OpenShift Container Platform が、永続ボリュームデータのバックアップ後に変更された namespace アノテーションに基づいて Pod に **securityContext** UID を割り当てる場合、これはお客様のワークロードにとって問題になる可能性があります。
- コンテナーの UID がファイル所有者の UID と一致しなくなった。
- OpenShift Container Platform がバックアップクラスターデータと一致するように宛先クラスターの UID 範囲を変更していないため、エラーが発生する。その結果、バックアップクラスターは宛先クラスターとは異なる UID を持つことになり、アプリケーションは宛先クラスターに対してデータの読み取りまたは書き込みを行うことができなくなります。

  軽減策

  次の 1 つ以上の緩和策を使用して、UID 範囲および GID 範囲の問題を解決できます。

- 簡単な緩和策:
  - **Backup** CR のラベルセレクターを使用して、バックアップに含めるオブジェクトをフィルター処理する場合は、必ずこのラベルセレクターをワークスペースを含む namespace に追加してください。
  - 同じ名前の namespace を復元する前に、宛先クラスター上の namespace の既存のバージョンを削除してください。
- 高度な緩和策:
  - 移行後に OpenShift namespace 内の重複する UID 範囲を解決する ことで、移行後の UID 範囲を修正します。ステップ 1 はオプションです。

あるクラスターでのデータのバックアップと別のクラスターでのリストアの問題の解決に重点を置いた、OpenShift Container Platform の UID 範囲および GID 範囲の詳細な説明は、A Guide to OpenShift and UIDs を 参照してください。

4.14.2.4. 1 つのクラスターからデータをバックアップし、別のクラスターに復元する

一般に、同じクラスターにデータをバックアップおよび復元するのと同じ方法で、1 つの OpenShift Container Platform クラスターからデータをバックアップし、別の OpenShift Container Platform クラスターに復元します。ただし、ある OpenShift Container Platform クラスターからデータをバックアップし、それを別のクラスターにリストアする場合は、追加の前提条件と手順の違いがいくつかあります。

前提条件

- お使いのプラットフォーム (AWS、Microsoft Azure、GCP など) でのバックアップと復元に関連するすべての前提条件、特に Data Protection Application (DPA) の前提条件については、このガイドの関連セクションで説明されています。

手順

- ご使用のプラットフォームに指定されている手順に次の追加を加えます。
  - リソースを別のクラスターに復元するには、バックアップストアの場所 (BSL) とボリュームスナップショットの場所が同じ名前とパスを持つようにしてください。
  - 同じオブジェクトストレージの場所の認証情報をクラスター全体で共有します。
  - 最良の結果を得るには、OADP を使用して宛先クラスターに namespace を作成します。
  - Velero **file-system-backup** オプションを使用する場合は、次のコマンドを実行して、バックアップ中に使用する **--default-volumes-to-fs-backup** フラグを有効にします。

```
$ velero backup create <backup_name> --default-volumes-to-fs-backup
<any_other_options>
```

注記

OADP 1.2 以降では、Velero Restic オプションは **file-system-backup** と呼ばれます。

4.14.3. OADP ストレージクラスマッピング

4.14.3.1. ストレージクラスマッピング

ストレージクラスマッピングを使用すると、さまざまな種類のデータにどのストレージクラスを適用するかを指定するルールまたはポリシーを定義できます。この機能は、アクセス頻度、データの重要性、コストの考慮事項に基づいて、ストレージクラスを決定するプロセスを自動化します。データがその特性と使用パターンに最適なストレージクラスに確実に保存されるようにすることで、ストレージの効率とコスト効率を最適化します。

change-storage-class-config フィールドを使用して、データオブジェクトのストレージクラスを変更できます。これにより、ニーズやアクセスパターンに応じて、標準ストレージからアーカイブストレージへなど、異なるストレージ層間でデータを移動することで、コストとパフォーマンスを最適化できます。

4.14.3.1.1. MTC を使用したストレージクラスマッピング

Migration Toolkit for Containers (MTC) を使用すると、アプリケーションデータを含むコンテナーを 1 つの OpenShift Container Platform クラスターから別のクラスターに移行したり、ストレージクラスのマッピングと変換を行うことができます。永続ボリューム (PV) のストレージクラスは、同じクラスター内で移行することで変換できます。これを行うには、MTC Web コンソールで移行計画を作成して実行する必要があります。

4.14.3.1.2. OADP を使用したストレージクラスマッピング

Velero プラグイン v1.1.0 以降で OpenShift API for Data Protection (OADP) を使用すると、Velero namespace の config map でストレージクラスマッピングを設定することにより、復元中に永続ボリューム (PV) のストレージクラスを変更できます。

OADP を使用して ConfigMap をデプロイするには、**change-storage-class-config** フィールドを使用します。クラウドプロバイダーに基づいて、ストレージクラスマッピングを変更する必要があります。

手順

1. 次のコマンドを実行して、ストレージクラスマッピングを変更します。

```
$ cat change-storageclass.yaml
```

2. 次の例に示すように、Velero namespace に config map を作成します。

例

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: change-storage-class-config
  namespace: openshift-adp
  labels:
    velero.io/plugin-config: ""
    velero.io/change-storage-class: RestoreItemAction
data:
  standard-csi: ssd-csi
```

3. 次のコマンドを実行して、ストレージクラスマッピング設定を保存します。

```
$ oc create -f change-storage-class-config
```

4.14.4. 関連情報

- 同一クラスター上での異なる Kubernetes API バージョンの使用
- CSI スナップショットでの Data Mover の使用
- ファイルシステムバックアップを使用したアプリケーションのバックアップ: Kopia または Restic.
- ストレージクラスの変換

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推奨します。

クラスターのアップグレード後に必ず etcd バックアップを作成してください。これは、クラスターを復元する際に、同じ z-stream リリースから取得した etcd バックアップを使用する必要があるために重要になります。たとえば、OpenShift Container Platform 4.y.z クラスターは、4.y.z から取得した etcd バックアップを使用する必要があります。

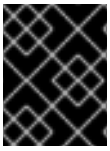
重要

コントロールプレーンホストでバックアップスクリプトの単一の呼び出しを実行して、クラスターの etcd データをバックアップします。各コントロールプレーンホストのバックアップを取得しないでください。

etcd のバックアップを作成した後に、クラスターの直前の状態への復元を実行できます。

5.1.1. etcd データのバックアップ

以下の手順に従って、etcd スナップショットを作成し、静的 Pod のリソースをバックアップして etcd データをバックアップします。このバックアップは保存でき、etcd を復元する必要がある場合に後で使用することができます。



重要

単一のコントロールプレーンホストからのバックアップのみを保存します。クラスター内の各コントロールプレーンホストからのバックアップは取得しないでください。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- クラスター全体のプロキシーが有効になっているかどうかを確認している。

ヒント

oc get proxy cluster -o yaml の出力を確認して、プロキシーが有効にされているかどうかを確認できます。プロキシーは、**httpProxy**、**httpsProxy**、および **noProxy** フィールドに値が設定されている場合に有効にされます。

手順

1. コントロールプレーンノードの root としてデバッグセッションを開始します。

```
$ oc debug --as-root node/<node_name>
```

2. デバッグシェルで root ディレクトリーを **/host** に変更します。

```
sh-4.4# 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. デバッグシェルで **cluster-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
```

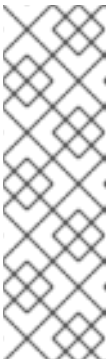
スクリプトの出力例

```
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。

5.1.2. 関連情報

- 不健全な etcd クラスターの回復

5.1.3. 自動 etcd バックアップの作成

etcd の自動バックアップ機能は、繰り返しバックアップとシングルバックアップの両方をサポートします。繰り返しバックアップでは、ジョブがトリガーされるたびにシングルバックアップを開始する cron ジョブが作成されます。

重要

etcd バックアップの自動化はテクノロジープレビュー機能です。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

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

etcd の自動バックアップを有効にするには、次の手順を実行します。

警告

クラスターで **TechPreviewNoUpgrade** 機能セットを有効にすると、マイナーバージョンの更新ができなくなります。**TechPreviewNoUpgrade** 機能セットは無効にできません。実稼働クラスターではこの機能セットを有効にしないでください。

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。

- OpenShift CLI (**oc**) にアクセスできる。

手順

1. 次の内容で、**enable-tech-preview-no-upgrade.yaml** という名前の **FeatureGate** カスタムリソース (CR) ファイルを作成します。

```
apiVersion: config.openshift.io/v1
kind: FeatureGate
metadata:
  name: cluster
spec:
  featureSet: TechPreviewNoUpgrade
```

2. CR を適用し、自動バックアップを有効にします。

```
$ oc apply -f enable-tech-preview-no-upgrade.yaml
```

3. 関連する API を有効にするのに時間がかかります。次のコマンドを実行して、カスタムリソース定義 (CRD) が作成されたことを確認します。

```
$ oc get crd | grep backup
```

出力例

```
backups.config.openshift.io 2023-10-25T13:32:43Z
etcdbackups.operator.openshift.io 2023-10-25T13:32:04Z
```

5.1.3.1. シングル etcd バックアップの作成

次の手順でカスタムリソース (CR) を作成して適用することで、シングル etcd バックアップを作成します。

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。
- OpenShift CLI (**oc**) にアクセスできる。

手順

- 動的にプロビジョニングされたストレージが利用可能な場合は、次の手順を実行して、単一の自動 etcd バックアップを作成します。

  a. 次の例のような内容で、**etcd-backup-pvc.yaml** という名前の永続ボリューム要求 (PVC) を作成します。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
  namespace: openshift-etcd
spec:
```

```
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 200Gi 1
  volumeMode: Filesystem
```

1 PVC に利用できるストレージの量。この値は、要件に合わせて調整します。

b. 以下のコマンドを実行して PVC を適用します。

```
$ oc apply -f etcd-backup-pvc.yaml
```

c. 次のコマンドを実行して、PVC が作成されたことを確認します。

```
$ oc get pvc
```

出力例

```
NAME              STATUS    VOLUME   CAPACITY   ACCESS MODES
STORAGECLASS   AGE
etcd-backup-pvc   Bound                                      51s
```

注記

動的 PVC は、マウントされるまで **Pending** 状態から遷移しません。

d. 次の例のような内容で、**etcd-single-backup.yaml** という名前の CR ファイルを作成します。

```
apiVersion: operator.openshift.io/v1alpha1
kind: EtcdBackup
metadata:
  name: etcd-single-backup
  namespace: openshift-etcd
spec:
  pvcName: etcd-backup-pvc 1
```

1 バックアップを保存する PVC の名前。この値は、使用している環境に応じて調整してください。

e. CR を適用してシングルバックアップを開始します。

```
$ oc apply -f etcd-single-backup.yaml
```

- 動的にプロビジョニングされたストレージが利用できない場合は、次の手順を実行して、単一の自動 etcd バックアップを作成します。

a. 次の内容で、**etcd-backup-local-storage.yaml** という名前の **StorageClass** CR ファイルを作成します。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: etcd-backup-local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: Immediate
```

b. 次のコマンドを実行して、**StorageClass** CR を適用します。

```
$ oc apply -f etcd-backup-local-storage.yaml
```

c. 次の例のような内容の **etcd-backup-pv-fs.yaml** という名前の PV を作成します。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: etcd-backup-pv-fs
spec:
  capacity:
    storage: 100Gi (1)
  volumeMode: Filesystem
  accessModes:
  - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  storageClassName: etcd-backup-local-storage
  local:
    path: /mnt
  nodeAffinity:
    required:
      nodeSelectorTerms:
      - matchExpressions:
      - key: kubernetes.io/hostname
        operator: In
        values:
        - <example_master_node> (2)
```

(1) PV が使用できるストレージの量。この値は、要件に合わせて調整します。

(2) この値は、この PV を割り当てるノードに置き換えます。

d. 次のコマンドを実行して、PV が作成されたことを確認します。

```
$ oc get pv
```

出力例

```
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
CLAIM   STORAGECLASS            REASON   AGE
etcd-backup-pv-fs   100Gi      RWO            Retain           Available           etcd-backup-
local-storage           10s
```

e. 次の例のような内容で、**etcd-backup-pvc.yaml** という名前の PVC を作成します。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
  namespace: openshift-etcd
spec:
  accessModes:
  - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 10Gi 1
```

1 PVC に利用できるストレージの量。この値は、要件に合わせて調整します。

f. 以下のコマンドを実行して PVC を適用します。

```
$ oc apply -f etcd-backup-pvc.yaml
```

g. 次の例のような内容で、**etcd-single-backup.yaml** という名前の CR ファイルを作成します。

```
apiVersion: operator.openshift.io/v1alpha1
kind: EtcdBackup
metadata:
  name: etcd-single-backup
  namespace: openshift-etcd
spec:
  pvcName: etcd-backup-pvc 1
```

1 バックアップを保存する永続ボリューム要求 (PVC) の名前。この値は、使用している環境に応じて調整してください。

h. CR を適用してシングルバックアップを開始します。

```
$ oc apply -f etcd-single-backup.yaml
```

5.1.3.2. 繰り返し etcd バックアップの作成

etcd の自動繰り返しバックアップを作成するには、次の手順に従います。

可能であれば、動的にプロビジョニングされたストレージを使用して、作成された etcd バックアップデータを安全な外部の場所に保存します。動的にプロビジョニングされたストレージが利用できない場合は、バックアップの復元にアクセスしやすくするために、バックアップデータを NFS 共有に保存することを検討してください。

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。
- OpenShift CLI (**oc**) にアクセスできる。

手順

1. 動的にプロビジョニングされたストレージが利用可能な場合は、次の手順を実行して、自動化された繰り返しバックアップを作成します。

   a. 次の例のような内容で、**etcd-backup-pvc.yaml** という名前の永続ボリューム要求 (PVC) を作成します。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
  namespace: openshift-etcd
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 200Gi 1
  volumeMode: Filesystem
  storageClassName: etcd-backup-local-storage
```

   1 PVC に利用できるストレージの量。この値は、要件に合わせて調整します。

注記

次の各プロバイダーでは、**accessModes** キーと **storageClassName** キーを変更する必要があります。

Provider	accessModes 値	storageClassName 値
versioned-installer-efc_operator-ci プロファイルを持つ AWS	**- ReadWriteMany**	**efs-sc**
Google Cloud Platform	**- ReadWriteMany**	**filestore-csi**
Microsoft Azure	**- ReadWriteMany**	**azurefile-csi**

   b. 以下のコマンドを実行して PVC を適用します。

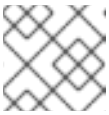
```
$ oc apply -f etcd-backup-pvc.yaml
```

   c. 次のコマンドを実行して、PVC が作成されたことを確認します。

```
$ oc get pvc
```

出力例

```
NAME              STATUS    VOLUME   CAPACITY   ACCESS MODES
STORAGECLASS   AGE
etcd-backup-pvc   Bound                                      51s
```



注記

動的 PVC は、マウントされるまで **Pending** 状態から遷移しません。

2. 動的にプロビジョニングされたストレージが使用できない場合は、次の手順を実行してローカルストレージ PVC を作成します。

警告

保存されているバックアップデータが格納されたノードを削除するか、該当ノードへのアクセスを失うと、データが失われる可能性があります。

a. 次の内容で、**etcd-backup-local-storage.yaml** という名前の **StorageClass** CR ファイルを作成します。

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
  name: etcd-backup-local-storage
provisioner: kubernetes.io/no-provisioner
volumeBindingMode: Immediate
```

b. 次のコマンドを実行して、**StorageClass** CR を適用します。

```
$ oc apply -f etcd-backup-local-storage.yaml
```

c. 適用された **StorageClass** から、次の例のような内容の **etcd-backup-pv-fs.yaml** という名前の PV を作成します。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: etcd-backup-pv-fs
spec:
  capacity:
    storage: 100Gi (1)
  volumeMode: Filesystem
  accessModes:
  - ReadWriteMany
  persistentVolumeReclaimPolicy: Delete
  storageClassName: etcd-backup-local-storage
  local:
    path: /mnt/
  nodeAffinity:
```

```
      required:
        nodeSelectorTerms:
        - matchExpressions:
          - key: kubernetes.io/hostname
            operator: In
            values:
            - <example_master_node> 2
```

1 PV が使用できるストレージの量。この値は、要件に合わせて調整します。

2 この値を、この PV を接続するマスターノードに置き換えます。

ヒント

次のコマンドを実行して、使用可能なノードのリストを表示します。

```
$ oc get nodes
```

d. 次のコマンドを実行して、PV が作成されたことを確認します。

```
$ oc get pv
```

出力例

```
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS
CLAIM   STORAGECLASS            REASON   AGE
etcd-backup-pv-fs   100Gi      RWX            Delete           Available           etcd-backup-
local-storage           10s
```

e. 次の例のような内容で、**etcd-backup-pvc.yaml** という名前の PVC を作成します。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: etcd-backup-pvc
spec:
  accessModes:
  - ReadWriteMany
  volumeMode: Filesystem
  resources:
    requests:
      storage: 10Gi 1
  storageClassName: etcd-backup-local-storage
```

1 PVC に利用できるストレージの量。この値は、要件に合わせて調整します。

f. 以下のコマンドを実行して PVC を適用します。

```
$ oc apply -f etcd-backup-pvc.yaml
```

3. **etcd-snippet-backups.yaml** という名前のカスタムリソース定義 (CRD) ファイルを作成します

3. **etcd-recurring-backups.yaml** という名前のカスタムリソース定義 (CRD) ファイルを作成します。作成された CRD の内容は、自動化されたバックアップのスケジュールと保持タイプを定義します。
15 個のバックアップを保持する **RetentionNumber** のデフォルトの保持タイプでは、次の例のような内容を使用します。

```
apiVersion: config.openshift.io/v1alpha1
kind: Backup
metadata:
  name: etcd-recurring-backup
spec:
  etcd:
    schedule: "20 4 * * *" (1)
    timeZone: "UTC"
    pvcName: etcd-backup-pvc
```

(1) 定期的なバックアップの **CronTab** スケジュール。この値は、必要に応じて調整します。

バックアップの最大数に基づいて保持を使用するには、次のキーと値のペアを **etcd** キーに追加します。

```
spec:
  etcd:
    retentionPolicy:
      retentionType: RetentionNumber (1)
      retentionNumber:
        maxNumberOfBackups: 5 (2)
```

(1) 保持タイプ。指定しない場合、デフォルトは **RetentionNumber** です。

(2) 保持するバックアップの最大数。この値は、必要に応じて調整します。指定しない場合、デフォルトは 15 個のバックアップです。

> **警告**
>
> 既知の問題により、保持されるバックアップの数が設定された値に 1 を加えた数になります。

バックアップのファイルサイズに基いて保持する場合は、以下を使用します。

```
spec:
  etcd:
    retentionPolicy:
      retentionType: RetentionSize
      retentionSize:
        maxSizeOfBackupsGb: 20 (1)
```

(1) 保持するバックアップの最大ファイルサイズ (ギガバイト単位)。この値は、必要に応じて

調整します。指定しない場合、デフォルトは 10 GB になります。

警告

既知の問題により、保持されるバックアップの最大サイズが設定値より最大 10 GB 大きくなります。

4. 次のコマンドを実行して、CRD で定義される cron ジョブを作成します。

```
$ oc create -f etcd-recurring-backup.yaml
```

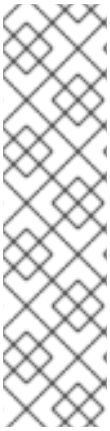
5. 作成された cron ジョブを検索するには、次のコマンドを実行します。

```
$ oc get cronjob -n openshift-etcd
```

5.2. 正常でない ETCD メンバーの置き換え

本書では、単一の正常でない etcd メンバーを置き換えるプロセスについて説明します。

このプロセスは、マシンが実行されていないか、ノードが準備状態にないことによって etcd メンバーが正常な状態にないか、etcd Pod がクラッシュループしているためにこれが正常な状態にないかによって異なります。



注記

コントロールプレーンホストの大部分を損失した場合は、この手順ではなく、ディザスターリカバリー手順に従って、以前のクラスター状態への復元 を行います。

コントロールプレーンの証明書が置き換えているメンバーで有効でない場合は、この手順ではなく、期限切れのコントロールプレーン証明書からの回復手順を実行する必要があります。

コントロールプレーンノードが失われ、新規ノードが作成される場合、etcd クラスター Operator は新規 TLS 証明書の生成と、ノードの etcd メンバーとしての追加を処理します。

5.2.1. 前提条件

- 正常でない etcd メンバーを置き換える前に、etcd バックアップを作成します。

5.2.2. 正常でない etcd メンバーの特定

クラスターに正常でない etcd メンバーがあるかどうかを特定することができます。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。

手順

1. 以下のコマンドを使用して **EtcdMembersAvailable** ステータス条件のステータスを確認します。

```
$ oc get etcd -o=jsonpath='{range .items[0].status.conditions[?
(@.type=="EtcdMembersAvailable")]}{.message}{"\n"}'
```

2. 出力を確認します。

```
2 of 3 members are available, ip-10-0-131-183.ec2.internal is unhealthy
```

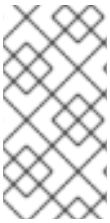
この出力例は、**ip-10-0-131-183.ec2.internal** etcd メンバーが正常ではないことを示しています。

5.2.3. 正常でない etcd メンバーの状態の判別

正常でない etcd メンバーを置き換える手順は、etcd メンバーが以下のどの状態にあるかによって異なります。

- マシンが実行されていないか、ノードが準備状態にない
- etcd Pod がクラッシュループしている。

以下の手順では、etcd メンバーがどの状態にあるかを判別します。これにより、正常でない etcd メンバーを置き換えるために実行する必要のある手順を確認できます。



注記

マシンが実行されていないか、ノードが準備状態にないものの、すぐに正常な状態に戻ることが予想される場合は、etcd メンバーを置き換える手順を実行する必要はありません。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

前提条件

- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- 正常でない etcd メンバーを特定している。

手順

1. **マシンが実行されていない**かどうかを判別します。

```
$ oc get machines -A -ojsonpath='{range .items[*]}{@.status.nodeRef.name}{"\t"}
{@.status.providerStatus.instanceState}{"\n"}' | grep -v running
```

出力例

```
ip-10-0-131-183.ec2.internal  stopped 1
```

1 この出力には、ノードおよびノードのマシンのステータスをリスト表示されます。ステータスが **running** 以外の場合は、**マシンは実行されていません**。

マシンが実行されていない 場合は、**マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え**の手順を実行します。

2. **ノードが準備状態にない**かどうかを判別します。
以下のシナリオのいずれかが true の場合、**ノードは準備状態にありません。**

- マシンが実行されている場合は、ノードに到達できないかどうかを確認します。

```
$ oc get nodes -o jsonpath='{range .items[*]}{"\n"}{.metadata.name}{"\t"}{range
.spec.taints[*]}{.key}{" "}' | grep unreachable
```

出力例

```
ip-10-0-131-183.ec2.internal node-role.kubernetes.io/master
node.kubernetes.io/unreachable node.kubernetes.io/unreachable 1
```

1 ノードが **unreachable** テイントと共にリスト表示される場合、**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** としてリスト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   v1.29.4 1
```

1 ノードが **NotReady** としてリスト表示されている場合、**ノードの準備はできていません。**

ノードの準備ができていない 場合は、**マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え**の手順を実行します。

3. **etcd Pod がクラッシュループしている**かどうかを判別します。
マシンが実行され、ノードが準備できている場合は、etcd Pod がクラッシュループしているかどうかを確認します。

a. すべてのコントロールプレーンノードが **Ready** としてリスト表示されていることを確認します。

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

出力例

```
NAME                          STATUS   ROLES    AGE     VERSION
ip-10-0-131-183.ec2.internal   Ready    master   6h13m   v1.29.4
ip-10-0-164-97.ec2.internal    Ready    master   6h13m   v1.29.4
ip-10-0-154-204.ec2.internal   Ready    master   6h13m   v1.29.4
```

b. etcd Pod のステータスが **Error** または **CrashloopBackoff** のいずれかであるかどうかを確認します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
etcd-ip-10-0-131-183.ec2.internal           2/3     Error       7          6h9m  1
etcd-ip-10-0-164-97.ec2.internal            3/3     Running     0          6h6m
etcd-ip-10-0-154-204.ec2.internal           3/3     Running     0          6h6m
```

1 この Pod のこのステータスは **Error** であるため、**etcd Pod はクラッシュループしています**。

etcd Pod がクラッシュループしている場合、**etcd Pod がクラッシュループしている場合の正常でない etcd メンバーの置き換え**についての手順を実行します。

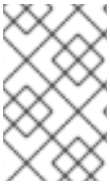
5.2.4. 正常でない etcd メンバーの置き換え

正常でない etcd メンバーの状態に応じて、以下のいずれかの手順を使用します。

- マシンが実行されていないか、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
- 異常停止したベアメタル etcd メンバーの置き換え

5.2.4.1. マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーの置き換え

以下の手順では、マシンが実行されていないか、ノードが準備状態にない場合の正常でない etcd メンバーを置き換える手順を説明します。

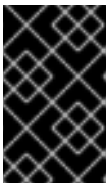


注記

クラスターがコントロールプレーンマシンセットを使用している場合、より簡単な etcd リカバリー手順については、コントロールプレーンマシンセットのトラブルシューティングの劣化した etcd Operator のリカバリーを参照してください。

前提条件

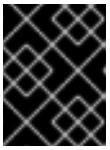
- 正常でない etcd メンバーを特定している。
- マシンが実行されていないか、ノードが準備状態にないことを確認している。



重要

他のコントロールプレーンノードの電源がオフになっている場合は、待機する必要があります。異常な etcd メンバーの交換が完了するまで、コントロールプレーンノードの電源をオフのままにしておく必要があります。

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



重要

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

手順

1. 正常でないメンバーを削除します。
   a. 影響を受けるノード上に **ない** Pod を選択します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
etcd-ip-10-0-131-183.ec2.internal             3/3     Running     0          123m
etcd-ip-10-0-164-97.ec2.internal              3/3     Running     0          123m
etcd-ip-10-0-154-204.ec2.internal             3/3     Running     0          124m
```

   b. 実行中の etcd コンテナーに接続し、影響を受けるノードにない Pod の名前を渡します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

   c. メンバーのリストを確認します。

```
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リスト表示します。

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. メンバーのリストを再度表示し、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

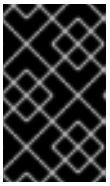
```
+------------------+---------+-----------------------------+---------------------------+----------------
-----------+
|        ID        | STATUS  |             NAME            |        PEER ADDRS         |      CLIENT
ADDRS       |
+------------------+---------+-----------------------------+---------------------------+----------------
-----------+
| 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。

2. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。



重要

クォーラムガードをオフにすると、設定の変更を反映するために残りの etcd インスタンスが再起動するまで、短時間クラスターにアクセスできなくなる可能性があります。

注記

etcd は、2 つのメンバーで実行されている場合、新たなメンバー障害を許容できません。残りのメンバーのいずれかを再起動すると、クォーラムが破棄され、クラスターでダウンタイムが発生します。クォーラムガードによって、ダウンタイムを引き起こす可能性のある設定変更による再起動から etcd が保護されるため、この手順を完了するには、クォーラムガードを無効にする必要があります。

3. 次のコマンドを実行して、影響を受けるノードを削除します。

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

4. 削除された正常でない etcd メンバーの古いシークレットを削除します。

   a. 削除された正常でない etcd メンバーのシークレット一覧を表示します。

```
$ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
```

   1 この手順で先ほど書き留めた正常でない etcd メンバーの名前を渡します。

   以下の出力に示されるように、ピア、サービング、およびメトリクスシークレットがあります。

   出力例

```
etcd-peer-ip-10-0-131-183.ec2.internal             kubernetes.io/tls                     2      47m
etcd-serving-ip-10-0-131-183.ec2.internal          kubernetes.io/tls                     2      47m
etcd-serving-metrics-ip-10-0-131-183.ec2.internal  kubernetes.io/tls                     2
47m
```

   b. 削除された正常でない etcd メンバーのシークレットを削除します。

      i. ピアシークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
```

      ii. 提供シークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
```

      iii. メトリクスシークレットを削除します。

```
$ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-
183.ec2.internal
```

5. コントロールプレーンマシンを削除し、再作成します。このマシンが再作成されると、新しいリビジョンが強制的に適用され、etcd は自動的にスケールアップします。
インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。それ以外の場合は、最初に作成する際に使用した方法と同じ方法を使用して新規マスターを作成する必要があります。

   a. 正常でないメンバーのマシンを取得します。

クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
NAME                                PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-0          Running   m4.xlarge   us-east-1   us-east-1a
3h37m   ip-10-0-131-183.ec2.internal   aws:///us-east-1a/i-0ec2782f8287dfb7e   stopped
1
clustername-8qw5l-master-1          Running   m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running   m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-east-
1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-east-1b
3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-east-
1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-06861c00007751b0a
running
```

1 これは正常でないノードのコントロールプレーンマシンです (**ip-10-0-131-183.ec2.internal**)。

b. マシン設定をファイルシステムのファイルに保存します。

```
$ oc get machine clustername-8qw5l-master-0 \ 1
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

1 正常でないノードのコントロールプレーンマシンの名前を指定します。

c. 直前の手順で作成された **new-master-machine.yaml** ファイルを編集し、新しい名前を割り当て、不要なフィールドを削除します。

i. **status** セクション全体を削除します。

```
status:
  addresses:
  - address: 10.0.131.183
    type: InternalIP
  - address: ip-10-0-131-183.ec2.internal
    type: InternalDNS
  - address: ip-10-0-131-183.ec2.internal
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Node
```

```
    name: ip-10-0-131-183.ec2.internal
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
  providerStatus:
    apiVersion: awsproviderconfig.openshift.io/v1beta1
    conditions:
    - lastProbeTime: "2020-04-20T16:53:50Z"
      lastTransitionTime: "2020-04-20T16:53:50Z"
      message: machine successfully created
      reason: MachineCreationSucceeded
      status: "True"
      type: MachineCreation
    instanceId: i-0fdb85790d76d0c3f
    instanceState: stopped
    kind: AWSMachineProviderStatus
```

ii. **metadata.name** フィールドを新規の名前に変更します。
古いマシンと同じベース名を維持し、最後の番号を次に利用可能な番号に変更することが推奨されます。この例では、**clustername-8qw5l-master-0** は **clustername-8qw5l-master-3** に変更されています。

以下に例を示します。

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: clustername-8qw5l-master-3
  ...
```

iii. **spec.providerID** フィールドを削除します。

```
    providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

d. 正常でないメンバーのマシンを削除します。

```
$ oc delete machine -n openshift-machine-api clustername-8qw5l-master-0 1
```

1 正常でないノードのコントロールプレーンマシンの名前を指定します。

e. マシンが削除されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
NAME                                PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-1          Running   m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running   m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large    us-east-1   us-east-
```

```
1a  3h28m  ip-10-0-129-226.ec2.internal  aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb  Running  m4.large  us-east-1  us-east-1b
3h28m  ip-10-0-144-248.ec2.internal  aws:///us-east-1b/i-0cb45ac45a166173b  running
clustername-8qw5l-worker-us-east-1c-pkg26  Running  m4.large  us-east-1  us-east-
1c  3h28m  ip-10-0-170-181.ec2.internal  aws:///us-east-1c/i-06861c00007751b0a
running
```

f. **new-master-machine.yaml** ファイルを使用して新規マシンを作成します。

```
$ oc apply -f new-master-machine.yaml
```

g. 新規マシンが作成されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
NAME                                PHASE         TYPE        REGION      ZONE         AGE
NODE                          PROVIDERID                          STATE
clustername-8qw5l-master-1          Running       m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-154-204.ec2.internal   aws:///us-east-1b/i-096c349b700a19631   running
clustername-8qw5l-master-2          Running       m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-164-97.ec2.internal    aws:///us-east-1c/i-02626f1dba9ed5bba   running
clustername-8qw5l-master-3          Provisioning  m4.xlarge   us-east-1   us-east-1a
85s     ip-10-0-133-53.ec2.internal    aws:///us-east-1a/i-015b0888fe17bc2c8   running
(1)
clustername-8qw5l-worker-us-east-1a-wbtgd  Running     m4.large    us-east-1   us-
east-1a  3h28m  ip-10-0-129-226.ec2.internal  aws:///us-east-1a/i-010ef6279b4662ced
running
clustername-8qw5l-worker-us-east-1b-lrdxb  Running     m4.large    us-east-1   us-east-
1b  3h28m  ip-10-0-144-248.ec2.internal  aws:///us-east-1b/i-0cb45ac45a166173b
running
clustername-8qw5l-worker-us-east-1c-pkg26  Running     m4.large    us-east-1   us-
east-1c  3h28m  ip-10-0-170-181.ec2.internal  aws:///us-east-1c/i-06861c00007751b0a
running
```

(1) 新規マシン **clustername-8qw5l-master-3** が作成され、**Provisioning** から **Running** にフェーズが変更されると準備状態になります。

新規マシンが作成されるまでに数分の時間がかかる場合があります。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

6. 次のコマンドを入力して、クォーラムガードをオンに戻します。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

7. 次のコマンドを入力して、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

8. 単一ノードの OpenShift を使用している場合は、ノードを再起動します。そうしないと、etcd クラスター Operator で次のエラーが発生する可能性があります。

出力例

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

検証

1. すべての etcd Pod が適切に実行されていることを確認します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
etcd-ip-10-0-133-53.ec2.internal             3/3     Running     0          7m49s
etcd-ip-10-0-164-97.ec2.internal             3/3     Running     0          123m
etcd-ip-10-0-154-204.ec2.internal            3/3     Running     0          124m
```

直前のコマンドの出力に 2 つの Pod のみがリスト表示される場合、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. メンバーのリストを確認します。

```
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。

関連情報

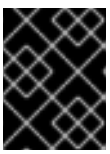
- 劣化した etcd Operator のリカバリー

5.2.4.2. etcd Pod がクラッシュループしている場合の正常でない etcd メンバーの置き換え

この手順では、etcd Pod がクラッシュループしている場合の正常でない etcd メンバーを置き換える手順を説明します。

前提条件

- 正常でない etcd メンバーを特定している。
- etcd Pod がクラッシュループしていることを確認している。
- **cluster-admin** ロールを持つユーザーとしてクラスターにアクセスできる。
- etcd のバックアップを取得している。



重要

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

手順

1. クラッシュループしている etcd Pod を停止します。

a. クラッシュループしているノードをデバッグします。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc debug node/ip-10-0-131-183.ec2.internal 1
```

1 これを正常でないノードの名前に置き換えます。

b. ルートディレクトリーを **/host** に変更します。

```
sh-4.2# chroot /host
```

c. 既存の etcd Pod ファイルを kubelet マニフェストディレクトリーから移動します。

```
sh-4.2# mkdir /var/lib/etcd-backup
```

```
sh-4.2# mv /etc/kubernetes/manifests/etcd-pod.yaml /var/lib/etcd-backup/
```

d. etcd データディレクトリーを別の場所に移動します。

```
sh-4.2# mv /var/lib/etcd/ /tmp
```

これでノードシェルを終了できます。

2. 正常でないメンバーを削除します。

a. 影響を受けるノード上に **ない** Pod を選択します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
etcd-ip-10-0-131-183.ec2.internal           2/3     Error       7          6h9m
etcd-ip-10-0-164-97.ec2.internal            3/3     Running     0          6h6m
etcd-ip-10-0-154-204.ec2.internal           3/3     Running     0          6h6m
```

b. 実行中の etcd コンテナーに接続し、影響を受けるノードにない Pod の名前を渡します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

c. メンバーのリストを確認します。

```
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. メンバーのリストを再度表示し、メンバーが削除されたことを確認します。

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

出力例

```
+------------------+---------+----------------------------+--------------------------+---------------
-----------+
|        ID        | STATUS  |            NAME            |        PEER ADDRS        |        CLIENT
ADDRS        |
+------------------+---------+----------------------------+--------------------------+---------------
-----------+
| b78e2856655bc2eb | started |  ip-10-0-164-97.ec2.internal |  https://10.0.164.97:2380 |
https://10.0.164.97:2379 |
| d022e10b498760d5 | started | ip-10-0-154-204.ec2.internal | https://10.0.154.204:2380
| https://10.0.154.204:2379 |
+------------------+---------+----------------------------+--------------------------+---------------
-----------+
```

これでノードシェルを終了できます。

3. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

4. 削除された正常でない etcd メンバーの古いシークレットを削除します。

   a. 削除された正常でない etcd メンバーのシークレット一覧を表示します。

   ```
   $ oc get secrets -n openshift-etcd | grep ip-10-0-131-183.ec2.internal 1
   ```

   1 この手順で先ほど書き留めた正常でない etcd メンバーの名前を渡します。

   以下の出力に示されるように、ピア、サービング、およびメトリクスシークレットがあります。

   出力例

   ```
   etcd-peer-ip-10-0-131-183.ec2.internal             kubernetes.io/tls                     2      47m
   etcd-serving-ip-10-0-131-183.ec2.internal          kubernetes.io/tls                     2      47m
   etcd-serving-metrics-ip-10-0-131-183.ec2.internal  kubernetes.io/tls                     2
   47m
   ```

   b. 削除された正常でない etcd メンバーのシークレットを削除します。

      i. ピアシークレットを削除します。

      ```
      $ oc delete secret -n openshift-etcd etcd-peer-ip-10-0-131-183.ec2.internal
      ```

      ii. 提供シークレットを削除します。

      ```
      $ oc delete secret -n openshift-etcd etcd-serving-ip-10-0-131-183.ec2.internal
      ```

      iii. メトリクスシークレットを削除します。

      ```
      $ oc delete secret -n openshift-etcd etcd-serving-metrics-ip-10-0-131-
      183.ec2.internal
      ```

5. etcd の再デプロイメントを強制的に実行します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

   ```
   $ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "single-master-recovery-
   '"$( date --rfc-3339=ns )"'"}}' --type=merge 1
   ```

   1 **forceRedeploymentReason** 値は一意である必要があります。そのため、タイムスタンプが付加されます。

   etcd クラスター Operator が再デプロイを実行する場合、すべてのコントロールプレーンノードで etcd Pod が機能していることを確認します。

6. 次のコマンドを入力して、クォーラムガードをオンに戻します。

   ```
   $ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
   ```

7. 次のコマンドを入力して、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

8. 単一ノードの OpenShift を使用している場合は、ノードを再起動します。そうしないと、etcd クラスター Operator で次のエラーが発生する可能性があります。

出力例

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

検証

- 新しいメンバーが利用可能で、正常な状態にあることを確認します。

a. 再度実行中の etcd コンテナーに接続します。
クラスターにアクセスできるターミナルで、cluster-admin ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-ip-10-0-154-204.ec2.internal
```

b. すべてのメンバーが正常であることを確認します。

```
sh-4.2# etcdctl endpoint health
```

出力例

```
https://10.0.131.183:2379 is healthy: successfully committed proposal: took = 16.671434ms
https://10.0.154.204:2379 is healthy: successfully committed proposal: took = 16.698331ms
https://10.0.164.97:2379 is healthy: successfully committed proposal: took = 16.621645ms
```

5.2.4.3. マシンが実行されていないか、ノードが準備状態にない場合の正常でないベアメタル etcd メンバーの置き換え

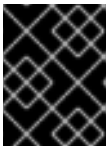
以下の手順では、マシンが実行されていないか、ノードが準備状態にない場合の正常でないベアメタル etcd メンバーを置き換える手順を説明します。

インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。それ以外の場合は、最初に作成したときと同じ方法で、新しいコントロールプレーンノードを作成する必要があります。

前提条件

- 正常でないベアメタル etcd メンバーを特定している。
- マシンが実行されていないか、ノードが準備状態にないことを確認している。

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



重要

問題が発生した場合にクラスターを復元できるように、この手順を実行する前に etcd バックアップを作成しておく。

手順

1. 正常でないメンバーを確認し、削除します。
   a. 影響を受けるノード上に **ない** Pod を選択します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd -o wide
```

出力例

```
etcd-openshift-control-plane-0   5/5   Running   11   3h56m   192.168.10.9   openshift-control-plane-0   <none>           <none>
etcd-openshift-control-plane-1   5/5   Running   0    3h54m   192.168.10.10   openshift-control-plane-1   <none>           <none>
etcd-openshift-control-plane-2   5/5   Running   0    3h58m   192.168.10.11   openshift-control-plane-2   <none>           <none>
```

   b. 実行中の etcd コンテナーに接続し、影響を受けるノードにない Pod の名前を渡します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

   c. メンバーのリストを確認します。

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

出力例

```
+------------------+---------+--------------------+--------------------------+--------------------------+--------------------+
| ID               | STATUS  | NAME               | PEER ADDRS               | CLIENT ADDRS             | IS LEARNER |
+------------------+---------+--------------------+--------------------------+--------------------------+--------------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380/ | https://192.168.10.11:2379/ | false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380/ | https://192.168.10.10:2379/ | false |
| cc3830a72fc357f9 | started | openshift-control-plane-0 | https://192.168.10.9:2380/ |
```

```
https://192.168.10.9:2379/   | false |
+-----------------+---------+-------------------+-------------------------+-----------------------
-+--------------------+
```

これらの値はこの手順で後ほど必要となるため、ID および正常でない etcd メンバーの名前を書き留めておきます。**etcdctl endpoint health** コマンドは、置き換えの手順が完了し、新規メンバーが追加されるまで、削除されたメンバーをリスト表示します。

d. ID を **etcdctl member remove** コマンドに指定して、正常でない etcd メンバーを削除します。

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

```
sh-4.2# etcdctl member remove 7a8197040a5126c8
```

出力例

```
Member 7a8197040a5126c8 removed from cluster b23536c33f2cdd1b
```

e. メンバーのリストを再度表示し、メンバーが削除されたことを確認します。

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

出力例

```
+-----------------+---------+-------------------+-------------------------+-------------------------
-+------------------------+
| ID              | STATUS  | NAME              | PEER ADDRS              | CLIENT
ADDRS              | IS LEARNER |
+-----------------+---------+-------------------+-------------------------+-------------------------
-+------------------------+
| cc3830a72fc357f9 | started | openshift-control-plane-2 | https://192.168.10.11:2380/ |
https://192.168.10.11:2379/ | false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380/ |
https://192.168.10.10:2379/ | false |
+-----------------+---------+-------------------+-------------------------+-------------------------
-+------------------------+
```

これでノードシェルを終了できます。

重要

メンバーを削除した後、残りの etcd インスタンスが再起動している間、クラスターに短時間アクセスできない場合があります。

2. 次のコマンドを入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

3. 以下のコマンドを実行して、削除された正常でない etcd メンバーの古いシークレットを削除します。

   a. 削除された正常でない etcd メンバーのシークレット一覧を表示します。

```
$ oc get secrets -n openshift-etcd | grep openshift-control-plane-2
```

この手順で先ほど書き留めた正常でない etcd メンバーの名前を渡します。

以下の出力に示されるように、ピア、サービング、およびメトリクスシークレットがあります。

```
etcd-peer-openshift-control-plane-2              kubernetes.io/tls   2   134m
etcd-serving-metrics-openshift-control-plane-2  kubernetes.io/tls   2   134m
etcd-serving-openshift-control-plane-2           kubernetes.io/tls   2   134m
```

   b. 削除された正常でない etcd メンバーのシークレットを削除します。

      i. ピアシークレットを削除します。

```
$ oc delete secret etcd-peer-openshift-control-plane-2 -n openshift-etcd

secret "etcd-peer-openshift-control-plane-2" deleted
```

      ii. 提供シークレットを削除します。

```
$ oc delete secret etcd-serving-metrics-openshift-control-plane-2 -n openshift-etcd

secret "etcd-serving-metrics-openshift-control-plane-2" deleted
```

      iii. メトリクスシークレットを削除します。

```
$ oc delete secret etcd-serving-openshift-control-plane-2 -n openshift-etcd

secret "etcd-serving-openshift-control-plane-2" deleted
```

4. コントロールプレーンマシンを削除します。
インストーラーでプロビジョニングされるインフラストラクチャーを実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。それ以外の場合は、最初に作成したときと同じ方法で、新しいコントロールプレーンノードを作成する必要があります。

   a. 正常でないメンバーのマシンを取得します。
   クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
NAME                              PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                              STATE
examplecluster-control-plane-0    Running                          3h11m   openshift-control-
plane-0   baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-
3ff2-41c5-b099-0aa41222964e   externally provisioned (1)
examplecluster-control-plane-1    Running                          3h11m   openshift-control-
plane-1   baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-
329c-475e-8d81-03b20280a3e1   externally provisioned
examplecluster-control-plane-2    Running                          3h11m   openshift-control-
plane-2   baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-
61d8-410f-be5b-6a395b056135   externally provisioned
examplecluster-compute-0          Running                          165m    openshift-compute-0
baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-4bb3-80ec-
13a31858241f          provisioned
examplecluster-compute-1          Running                          165m    openshift-compute-1
baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-4241-91dc-
e7ea72ab13b9          provisioned
```

(1) これは正常でないノードのコントロールプレーンマシンです (**examplecluster-control-plane-2**)。

b. マシン設定をファイルシステムのファイルに保存します。

```
$ oc get machine examplecluster-control-plane-2 \ (1)
    -n openshift-machine-api \
    -o yaml \
    > new-master-machine.yaml
```

(1) 正常でないノードのコントロールプレーンマシンの名前を指定します。

c. 直前の手順で作成された **new-master-machine.yaml** ファイルを編集し、新しい名前を割り当て、不要なフィールドを削除します。

i. **status** セクション全体を削除します。

```
status:
  addresses:
  - address: ""
    type: InternalIP
  - address: fe80::4adf:37ff:feb0:8aa1%ens1f1.373
    type: InternalDNS
  - address: fe80::4adf:37ff:feb0:8aa1%ens1f1.371
    type: Hostname
  lastUpdated: "2020-04-20T17:44:29Z"
  nodeRef:
    kind: Machine
    name: fe80::4adf:37ff:feb0:8aa1%ens1f1.372
    uid: acca4411-af0d-4387-b73e-52b2484295ad
  phase: Running
```

```
providerStatus:
  apiVersion: machine.openshift.io/v1beta1
  conditions:
  - lastProbeTime: "2020-04-20T16:53:50Z"
    lastTransitionTime: "2020-04-20T16:53:50Z"
    message: machine successfully created
    reason: MachineCreationSucceeded
    status: "True"
    type: MachineCreation
  instanceId: i-0fdb85790d76d0c3f
  instanceState: stopped
  kind: Machine
```

5. **metadata.name** フィールドを新規の名前に変更します。
古いマシンと同じベース名を維持し、最後の番号を次に利用可能な番号に変更することが推奨されます。この例では、**examplecluster-control-plane-2** が **examplecluster-control-plane-3** に変更されています。

   以下に例を示します。

```
apiVersion: machine.openshift.io/v1beta1
kind: Machine
metadata:
  ...
  name: examplecluster-control-plane-3
  ...
```

   a. **spec.providerID** フィールドを削除します。

```
    providerID: baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-61d8-410f-be5b-6a395b056135
```

   b. **metadata.annotations** および **metadata.generation** フィールドを削除します。

```
    annotations:
      machine.openshift.io/instance-state: externally provisioned
    ...
    generation: 2
```

   c. **spec.conditions**、**spec.lastUpdated**、**spec.nodeRef**、および **spec.phase** フィールドを削除します。

```
    lastTransitionTime: "2022-08-03T08:40:36Z"
  message: 'Drain operation currently blocked by: [{Name:EtcdQuorumOperator
  Owner:clusteroperator/etcd}]'
  reason: HookPresent
  severity: Warning
  status: "False"

  type: Drainable
  lastTransitionTime: "2022-08-03T08:39:55Z"
  status: "True"
  type: InstanceExists
```

```
lastTransitionTime: "2022-08-03T08:36:37Z"
status: "True"
type: Terminable
lastUpdated: "2022-08-03T08:40:36Z"
nodeRef:
kind: Node
name: openshift-control-plane-2
uid: 788df282-6507-4ea2-9a43-24f237ccbc3c
phase: Running
```

6. 以下のコマンドを実行して、Bare Metal Operator が利用可能であることを確認します。

```
$ oc get clusteroperator baremetal
```

出力例

```
NAME      VERSION   AVAILABLE   PROGRESSING   DEGRADED   SINCE   MESSAGE
baremetal 4.16.0    True        False         False      3d15h
```

7. 次のコマンドを実行して、古い **BareMetalHost** オブジェクトを削除します。

```
$ oc delete bmh openshift-control-plane-2 -n openshift-machine-api
```

出力例

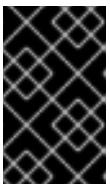
```
baremetalhost.metal3.io "openshift-control-plane-2" deleted
```

8. 次のコマンドを実行して、異常なメンバーのマシンを削除します。

```
$ oc delete machine -n openshift-machine-api examplecluster-control-plane-2
```

BareMetalHost および **Machine** オブジェクトを削除すると、**Machine** コントローラーにより **Node** オブジェクトが自動的に削除されます。

何らかの理由でマシンの削除が遅れたり、コマンドが妨げられて遅れたりする場合は、マシンオブジェクトのファイナライザーフィールドを削除することで強制的に削除できます。



重要

Ctrl+c を押してマシンの削除を中断しないでください。コマンドが完了するまで続行できるようにする必要があります。新しいターミナルウィンドウを開き、ファイナライザーフィールドを編集して削除します。

a. 次のコマンドを実行して、マシン設定を編集します。

```
$ oc edit machine -n openshift-machine-api examplecluster-control-plane-2
```

b. **Machine** カスタムリソースの次のフィールドを削除し、更新されたファイルを保存します。

```
finalizers:
- machine.machine.openshift.io
```

出力例

```
machine.machine.openshift.io/examplecluster-control-plane-2 edited
```

9. 以下のコマンドを実行して、マシンが削除されていることを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
NAME                              PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                              STATE
examplecluster-control-plane-0    Running                          3h11m   openshift-control-plane-0
baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-3ff2-41c5-b099-
0aa41222964e   externally provisioned
examplecluster-control-plane-1    Running                          3h11m   openshift-control-plane-1
baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-329c-475e-8d81-
03b20280a3e1   externally provisioned
examplecluster-compute-0          Running                          165m    openshift-compute-0
baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-4bb3-80ec-
13a31858241f         provisioned
examplecluster-compute-1          Running                          165m    openshift-compute-1
baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-4241-91dc-
e7ea72ab13b9         provisioned
```

10. 次のコマンドを実行して、ノードが削除されたことを確認します。

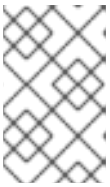
```
$ oc get nodes

NAME                       STATUS ROLES    AGE    VERSION
openshift-control-plane-0 Ready master 3h24m v1.29.4
openshift-control-plane-1 Ready master 3h24m v1.29.4
openshift-compute-0       Ready worker 176m v1.29.4
openshift-compute-1       Ready worker 176m v1.29.4
```

11. 新しい **BareMetalHost** オブジェクトとシークレットを作成して BMC 認証情報を保存します。

```
$ cat <<EOF | oc apply -f -
apiVersion: v1
kind: Secret
metadata:
  name: openshift-control-plane-2-bmc-secret
  namespace: openshift-machine-api
data:
  password: <password>
  username: <username>
type: Opaque
---
apiVersion: metal3.io/v1alpha1
kind: BareMetalHost
metadata:
  name: openshift-control-plane-2
  namespace: openshift-machine-api
spec:
```

```
  automatedCleaningMode: disabled
  bmc:
    address: redfish://10.46.61.18:443/redfish/v1/Systems/1
    credentialsName: openshift-control-plane-2-bmc-secret
    disableCertificateVerification: true
  bootMACAddress: 48:df:37:b0:8a:a0
  bootMode: UEFI
  externallyProvisioned: false
  online: true
  rootDeviceHints:
    deviceName: /dev/disk/by-id/scsi-<serial_number>
  userData:
    name: master-user-data-managed
    namespace: openshift-machine-api
EOF
```



注記

ユーザー名とパスワードは、他のベアメタルホストのシークレットで確認できます。**bmc:address** で使用するプロトコルは、他の bmh オブジェクトから取得できます。

重要

既存のコントロールプレーンホストから **BareMetalHost** オブジェクト定義を再利用する場合は、**externallyProvisioned** フィールドを **true** に設定したままにしないでください。

既存のコントロールプレーン **BareMetalHost** オブジェクトが、OpenShift Container Platform インストールプログラムによってプロビジョニングされた場合には、**externallyProvisioned** フラグが **true** に設定されている可能性があります。

検査が完了すると、**BareMetalHost** オブジェクトが作成され、プロビジョニングできるようになります。

12. 利用可能な **BareMetalHost** オブジェクトを使用して作成プロセスを確認します。

```
$ oc get bmh -n openshift-machine-api

NAME                 STATE                  CONSUMER                        ONLINE ERROR   AGE
openshift-control-plane-0 externally provisioned examplecluster-control-plane-0 true
4h48m
openshift-control-plane-1 externally provisioned examplecluster-control-plane-1 true
4h48m
openshift-control-plane-2 available          examplecluster-control-plane-3 true        47m
openshift-compute-0       provisioned        examplecluster-compute-0       true        4h48m
openshift-compute-1       provisioned        examplecluster-compute-1       true        4h48m
```

a. **new-master-machine.yaml** ファイルを使用して新規コントロールプレーンマシンを作成します。

```
$ oc apply -f new-master-machine.yaml
```

b. 新規マシンが作成されたことを確認します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例

```
NAME                                PHASE     TYPE   REGION   ZONE   AGE     NODE
PROVIDERID                                                                 STATE
examplecluster-control-plane-0      Running                          3h11m   openshift-control-
plane-0   baremetalhost:///openshift-machine-api/openshift-control-plane-0/da1ebe11-
3ff2-41c5-b099-0aa41222964e   externally provisioned (1)
examplecluster-control-plane-1      Running                          3h11m   openshift-control-
plane-1   baremetalhost:///openshift-machine-api/openshift-control-plane-1/d9f9acbc-
329c-475e-8d81-03b20280a3e1   externally provisioned
examplecluster-control-plane-2      Running                          3h11m   openshift-control-
plane-2   baremetalhost:///openshift-machine-api/openshift-control-plane-2/3354bdac-
61d8-410f-be5b-6a395b056135   externally provisioned
examplecluster-compute-0            Running                          165m    openshift-compute-
0         baremetalhost:///openshift-machine-api/openshift-compute-0/3d685b81-7410-
4bb3-80ec-13a31858241f         provisioned
examplecluster-compute-1            Running                          165m    openshift-compute-
1         baremetalhost:///openshift-machine-api/openshift-compute-1/0fdae6eb-2066-
4241-91dc-e7ea72ab13b9         provisioned
```

(1) 新規マシン **clustername-8qw5l-master-3** が作成され、**Provisioning** から **Running** にフェーズが変更されると準備状態になります。

新規マシンが作成されるまでに数分の時間がかかる場合があります。etcd クラスター Operator はマシンまたはノードが正常な状態に戻ると自動的に同期します。

c. 以下のコマンドを実行して、ベアメタルホストがプロビジョニングされ、エラーが報告されていないことを確認します。

```
$ oc get bmh -n openshift-machine-api
```

出力例

```
$ oc get bmh -n openshift-machine-api
NAME                 STATE                  CONSUMER                    ONLINE ERROR AGE
openshift-control-plane-0 externally provisioned examplecluster-control-plane-0 true
4h48m
openshift-control-plane-1 externally provisioned examplecluster-control-plane-1 true
4h48m
openshift-control-plane-2 provisioned          examplecluster-control-plane-3 true
47m
openshift-compute-0       provisioned          examplecluster-compute-0       true
4h48m
openshift-compute-1       provisioned          examplecluster-compute-1       true
4h48m
```

d. 以下のコマンドを実行して、新規ノードが追加され、Ready の状態であることを確認します。

```
$ oc get nodes
```

出力例

```
$ oc get nodes
NAME                        STATUS ROLES   AGE   VERSION
openshift-control-plane-0 Ready master 4h26m v1.29.4
openshift-control-plane-1 Ready master 4h26m v1.29.4
openshift-control-plane-2 Ready master 12m   v1.29.4
openshift-compute-0       Ready worker 3h58m v1.29.4
openshift-compute-1       Ready worker 3h58m v1.29.4
```

13. 次のコマンドを入力して、クォーラムガードをオンに戻します。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

14. 次のコマンドを入力して、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

15. 単一ノードの OpenShift を使用している場合は、ノードを再起動します。そうしないと、etcd クラスター Operator で次のエラーが発生する可能性があります。

出力例

```
EtcdCertSignerControllerDegraded: [Operation cannot be fulfilled on secrets "etcd-peer-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-sno-0": the object has been modified; please apply your changes to the latest version and try again, Operation cannot be fulfilled on secrets "etcd-serving-metrics-sno-0": the object has been modified; please apply your changes to the latest version and try again]
```

検証

1. すべての etcd Pod が適切に実行されていることを確認します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
etcd-openshift-control-plane-0     5/5     Running     0     105m
etcd-openshift-control-plane-1     5/5     Running     0     107m
etcd-openshift-control-plane-2     5/5     Running     0     103m
```

直前のコマンドの出力に 2 つの Pod のみがリスト表示される場合、etcd の再デプロイメントを手動で強制できます。クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc patch etcd cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
```
(1)

(1) **forceRedeploymentReason** 値は一意である必要があります。そのため、タイムスタンプが付加されます。

etcd メンバーがちょうど 3 つあることを確認するには、実行中の etcd コンテナーに接続し、影響を受けたノード上になかった Pod の名前を渡します。クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc rsh -n openshift-etcd etcd-openshift-control-plane-0
```

2. メンバーのリストを確認します。

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

出力例

```
+------------------+---------+--------------------+--------------------------+--------------------------+---
------------+
|        ID        | STATUS  |        NAME        |        PEER ADDRS        |       CLIENT ADDRS
|    IS LEARNER    |
+------------------+---------+--------------------+--------------------------+--------------------------+---
------------+
| 7a8197040a5126c8 | started | openshift-control-plane-2 | https://192.168.10.11:2380 |
https://192.168.10.11:2379 |    false |
| 8d5abe9669a39192 | started | openshift-control-plane-1 | https://192.168.10.10:2380 |
https://192.168.10.10:2379 |    false |
| cc3830a72fc357f9 | started | openshift-control-plane-0 | https://192.168.10.9:2380 |
https://192.168.10.9:2379 |     false |
+------------------+---------+--------------------+--------------------------+--------------------------+---
------------+
```

注記

直前のコマンドの出力に 4 つ以上の etcd メンバーが表示される場合、不要なメンバーを慎重に削除する必要があります。

3. 以下のコマンドを実行して、すべての etcd メンバーが正常であることを確認します。

```
# etcdctl endpoint health --cluster
```

出力例

```
https://192.168.10.10:2379 is healthy: successfully committed proposal: took = 8.973065ms
https://192.168.10.9:2379 is healthy: successfully committed proposal: took = 11.559829ms
https://192.168.10.11:2379 is healthy: successfully committed proposal: took = 11.665203ms
```

4. 以下のコマンドを実行して、すべてのノードが最新のリビジョンであることを確認します。

```
$ oc get etcd -o=jsonpath='{range.items[0].status.conditions[?
(@.type=="NodeInstallerProgressing")]}{.reason}{"\n"}{.message}{"\n"}'
```

```
AllNodesAtLatestRevision
```

5.2.5. 関連情報

- マシンライフサイクルフックによるクォーラム保護

5.3. 障害復旧

5.3.1. 障害復旧について

この障害復旧ドキュメントでは、OpenShift Container Platform クラスターで発生する可能性のある複数の障害のある状態からの復旧方法についての管理者向けの情報を提供しています。管理者は、クラスターの状態を機能する状態に戻すために、以下の 1 つまたは複数の手順を実行する必要がある場合があります。

重要

障害復旧には、少なくとも 1 つの正常なコントロールプレーンホストが必要です。

クラスターの直前の状態への復元

このソリューションは、管理者が重要なものを削除した場合など、クラスターを直前の状態に復元する必要がある状態に対応します。これには、大多数のコントロールプレーンホストが失われたために etcd クォーラム (定足数) が失われ、クラスターがオフラインになる状態も含まれます。etcd バックアップを取得している限り、以下の手順に従ってクラスターを直前の状態に復元できます。該当する場合は、コントロールプレーン証明書の期限切れの状態からのリカバリーが必要になる場合もあります。

警告

クラスターの直前の状態への復元は、実行中のクラスターで行う破壊的で、不安定なアクションです。この手順は、最後の手段としてのみ使用してください。

復元の実行前に、クラスターへの影響の詳細についてクラスターの復元を参照してください。

注記

大多数のマスターが依然として利用可能であり、etcd のクォーラムがある場合は、手順に従って単一の正常でない etcd メンバーの置き換えを実行します。

コントロールプレーン証明書の期限切れの状態からのリカバリー

このソリューションは、コントロールプレーン証明書の期限が切れた状態に対応します。たとえ

ば、インストールの 24 時間後に行われる最初の証明書のローテーション前にクラスターをシャットダウンする場合、証明書はローテーションされず、期限切れになります。以下の手順に従って、コントロールプレーン証明書の期限切れの状態からのリカバリーを実行できます。

5.3.2. クラスターの直前の状態への復元

クラスターを直前の状態に復元するには、スナップショットを作成して、事前に etcd データのバックアップを行っている必要があります。このスナップショットを使用して、クラスターの状態を復元します。

5.3.2.1. クラスターの状態の復元について

etcd バックアップを使用して、クラスターを直前の状態に復元できます。これは、以下の状況から回復するために使用できます。

- クラスターは、大多数のコントロールプレーンホストを失いました (クォーラムの喪失)。
- 管理者が重要なものを削除し、クラスターを復旧するために復元する必要があります。

警告

クラスターの直前の状態への復元は、実行中のクラスターで行う破壊的で、不安定なアクションです。これは、最後の手段としてのみ使用してください。

Kubernetes API サーバーを使用してデータを取得できる場合は、etcd が利用できるため、etcd バックアップを使用して復元することはできません。

etcd を効果的に復元すると、クラスターが時間内に元に戻され、すべてのクライアントは競合する並列履歴が発生します。これは、kubelet、Kubernetes コントローラーマネージャー、SDN コントローラー、永続ボリュームコントローラーなどのコンポーネントを監視する動作に影響を与える可能性があります。

etcd のコンテンツがディスク上の実際のコンテンツと一致しないと、Operator チャーンが発生し、ディスク上のファイルが etcd のコンテンツと競合すると、Kubernetes API サーバー、Kubernetes コントローラーマネージャー、Kubernetes スケジューラーなどの Operator が停止する場合があります。この場合は、問題の解決に手動のアクションが必要になる場合があります。

極端な場合、クラスターは永続ボリュームを追跡できなくなり、存在しなくなった重要なワークロードを削除し、マシンのイメージを再作成し、期限切れの証明書を使用して CA バンドルを書き換えることができます。

5.3.2.2. クラスターの直前の状態への復元

保存された **etcd** のバックアップを使用して、クラスターの以前の状態を復元したり、大多数のコントロールプレーンホストが失われたクラスターを復元したりできます。

注記

クラスターがコントロールプレーンマシンセットを使用している場合、より簡単な **etcd** リカバリー手順については、「コントロールプレーンマシンセットのトラブルシューティング」を参照してください。

重要

クラスターを復元する際に、同じ z-stream リリースから取得した **etcd** バックアップを使用する必要があります。たとえば、OpenShift Container Platform 4.7.2 クラスターは、4.7.2 から取得した **etcd** バックアップを使用する必要があります。

前提条件

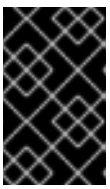
- インストール時に使用したものと同様、証明書ベースの **kubeconfig** ファイルを介して、**cluster-admin** ロールを持つユーザーとしてクラスターにアクセスします。
- リカバリーホストとして使用する正常なコントロールプレーンホストがあること。
- コントロールプレーンホストへの SSH アクセス。
- **etcd** スナップショットと静的 Pod のリソースの両方を含むバックアップディレクトリー (同じバックアップから取られるもの)。ディレクトリー内のファイル名は、**snapshot_<datetimestamp>.db** および **static_kuberesources_<datetimestamp>.tar.gz** の形式にする必要があります。

重要

非リカバリー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。**kube-apiserver** は復元プロセスの開始後にアクセスできなくなるため、コントロールプレーンノードにはアクセスできません。このため、別のターミナルで各コントロールプレーンホストに SSH 接続を確立することが推奨されます。

   重要

   この手順を完了しないと、復元手順を完了するためにコントロールプレーンホストにアクセスすることができなくなり、この状態からクラスターを回復できなくなります。

3. **etcd** バックアップディレクトリーをリカバリーコントロールプレーンホストにコピーします。この手順では、**etcd** スナップショットおよび静的 Pod のリソースを含む **backup** ディレクトリーを、リカバリーコントロールプレーンホストの /**home**/**core**/ ディレクトリーにコピーしていることを前提としています。
4. 他のすべてのコントロールプレーンノードで静的 Pod を停止します。

注記

リカバリーホストで静的 Pod を停止する必要はありません。

a. リカバリーホストではないコントロールプレーンホストにアクセスします。

b. 以下を実行して、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 -v /etc/kubernetes/manifests/etcd-pod.yaml /tmp
```

c. 以下を使用して、**etcd** Pod が停止していることを確認します。

```
$ sudo crictl ps | grep etcd | egrep -v "operator|etcd-guard"
```

このコマンドの出力が空でない場合は、数分待ってからもう一度確認してください。

d. 以下を実行して、既存の **kube-apiserver** 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 -v /etc/kubernetes/manifests/kube-apiserver-pod.yaml /tmp
```

e. 以下を実行して、**kube-apiserver** コンテナーが停止していることを確認します。

```
$ sudo crictl ps | grep kube-apiserver | egrep -v "operator|guard"
```

このコマンドの出力が空でない場合は、数分待ってからもう一度確認してください。

f. 以下を使用して、既存の **kube-controller-manager** 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 -v /etc/kubernetes/manifests/kube-controller-manager-pod.yaml /tmp
```

g. 以下を実行して、**kube-controller-manager** コンテナーが停止していることを確認します。

```
$ sudo crictl ps | grep kube-controller-manager | egrep -v "operator|guard"
```

このコマンドの出力が空でない場合は、数分待ってからもう一度確認してください。

h. 以下を使用して、既存の **kube-scheduler** 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 -v /etc/kubernetes/manifests/kube-scheduler-pod.yaml /tmp
```

i. 以下を使用して、**kube-scheduler** コンテナーが停止していることを確認します。

```
$ sudo crictl ps | grep kube-scheduler | egrep -v "operator|guard"
```

このコマンドの出力が空でない場合は、数分待ってからもう一度確認してください。

j. 次の例を使用して、**etcd** データディレクトリーを別の場所に移動します。

```
$ sudo mv -v /var/lib/etcd/ /tmp
```

k. **/etc/kubernetes/manifests/keepalived.yaml** ファイルが存在し、ノードが削除された場合は、次の手順に従います。

i. **/etc/kubernetes/manifests/keepalived.yaml** 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 -v /etc/kubernetes/manifests/keepalived.yaml /tmp
```

ii. **keepalived** デーモンによって管理されているコンテナーが停止していることを確認します。

```
$ sudo crictl ps --name keepalived
```

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。

iii. コントロールプレーンに仮想 IP (VIP) が割り当てられているかどうかを確認します。

```
$ ip -o address | egrep '<api_vip>|<ingress_vip>'
```

iv. 報告された仮想 IP ごとに、次のコマンドを実行して仮想 IP を削除します。

```
$ sudo ip address del <reported_vip> dev <reported_vip_device>
```

l. 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. **keepalived** デーモンが使用されている場合は、リカバリーコントロールプレーンノードが仮想 IP を所有していることを確認します。

```
$ ip -o address | grep <api_vip>
```

仮想 IP のアドレスが存在する場合、出力内で強調表示されます。仮想 IP が設定されていないか、正しく設定されていない場合、このコマンドは空の文字列を返します。

7. 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。

8. 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/assets/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
```

cluster-restore.sh スクリプトは、**etcd**、**kube-apiserver**、**kube-controller-manager**、および **kube-scheduler** Pod が停止され、復元プロセスの最後に開始されたことを示す必要があります。

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。

9. 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   v1.29.4
   host-172-25-75-38   Ready   infra,worker   3d20h   v1.29.4
   host-172-25-75-40   Ready   master        3d20h   v1.29.4
   host-172-25-75-65   Ready   master        3d20h   v1.29.4
   host-172-25-75-74   Ready   infra,worker   3d20h   v1.29.4
   host-172-25-75-79   Ready   worker        3d20h   v1.29.4
   host-172-25-75-86   Ready   worker        3d20h   v1.29.4
   host-172-25-75-98   Ready   infra,worker   3d20h   v1.29.4
   ```

   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
```

10. 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. 復元ホストから以下を実行します。

```
$ 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。

11. 保留中の証明書署名要求 (CSR) を承認します。

注記

単一ノードクラスターや 3 つのスケジュール可能なコントロールプレーンノードで設定されるクラスターなど、ワーカーノードを持たないクラスターには、承認する保留中の CSR はありません。この手順にリストされているすべてのコマンドをスキップできます。

a. 次のコマンドを実行して、現在の CSR のリストを取得します。

```
$ 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 kubelet 提供エンドポイントのノードによって要求される、保留中の kubelet 提供 CSR。

3 4 **node-bootstrapper** ノードのブートストラップ認証情報を使用して要求される、保留中の kubelet クライアント CSR。

b. 次のコマンドを実行して、CSR の詳細と CSR 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 のリストからの CSR の名前です。

c. 以下を実行して、有効な **node-bootstrapper** CSR をそれぞれ承認します。

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

d. user-provisioned installation の場合、以下を実行して各 kubelet service CSR を承認します。

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

12. 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. 以下を使用して、リカバリーホストから **etcd** コンテナーが実行中であることを確認します。

```
$ sudo crictl ps | grep etcd | egrep -v "operator|etcd-guard"
```

出力例

```
3ad41b7908e32
36f86e2eeaaffe662df0d21041eb22b8198e0e58abeeae8c743c3e6e977e8009
About a minute ago   Running             etcd                                    0
7c05f8af362f0
```

b. 以下を使用して、リカバリーホストから **etcd** Pod が実行されていることを確認します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

```
NAME                                READY   STATUS      RESTARTS   AGE
etcd-ip-10-0-143-125.ec2.internal               1/1     Running     1          2m47s
```

ステータスが **Pending** の場合や出力に複数の実行中の **etcd** Pod が一覧表示される場合、数分待機してから再度チェックを行います。

13. **OVNKubernetes** ネットワークプラグインを使用している場合は、**ovnkube-controlplane** Pod を再起動する必要があります。

a. 以下を実行して、すべての **ovnkube-controlplane** Pod を削除します。

```
$ oc -n openshift-ovn-kubernetes delete pod -l app=ovnkube-control-plane
```

b. 次のコマンドを使用して、すべての **ovnkube-controlplane** Pod が再デプロイされたことを確認します。

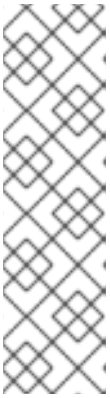
```
$ oc -n openshift-ovn-kubernetes get pod -l app=ovnkube-control-plane
```

14. OVN-Kubernetes ネットワークプラグインを使用している場合は、すべてのノードで Open Virtual Network (OVN) Kubernetes Pod を 1 つずつ再起動します。次の手順を使用して、各ノードで OVN-Kubernetes Pod を再起動します。

重要

次の順序で OVN-Kubernetes Pod を再起動します。

1. リカバリーコントロールプレーンホスト
2. 他のコントロールプレーンホスト (利用可能な場合)
3. 他のノード



注記

検証および変更用の受付 Webhook は Pod を拒否することができます。**failurePolicy** を **Fail** に設定して追加の Webhook を追加すると、Pod が拒否され、復元プロセスが失敗する可能性があります。これは、クラスターの状態の復元中に Webhook を保存および削除することで回避できます。クラスターの状態が正常に復元された後に、Webhook を再度有効にできます。

または、クラスターの状態の復元中に **failurePolicy** を一時的に **Ignore** に設定できます。クラスターの状態が正常に復元された後に、**failurePolicy** を **Fail** にすることができます。

a. ノースバウンドデータベース (nbdb) とサウスバウンドデータベース (sbdb) を削除します。Secure Shell (SSH) を使用して復元ホストと残りのコントロールプレーンノードにアクセスし、以下を実行します。

```
$ sudo rm -f /var/lib/ovn-ic/etc/*.db
```

b. OpenVSwitch サービスを再起動します。Secure Shell (SSH) を使用してノードにアクセスし、次のコマンドを実行します。

```
$ sudo systemctl restart ovs-vswitchd ovsdb-server
```

c. 次のコマンドを実行して、ノード上の **ovnkube-node** Pod を削除します。**<node>** は、再起動するノードの名前に置き換えます。

```
$ oc -n openshift-ovn-kubernetes delete pod -l app=ovnkube-node --field-
selector=spec.nodeName==<node>
```

d. 以下を使用して、**ovnkube-node** Pod が再度実行されていることを確認します。

```
$ oc -n openshift-ovn-kubernetes get pod -l app=ovnkube-node --field-
selector=spec.nodeName==<node>
```

注記

Pod が再起動するまでに数分かかる場合があります。

15. 他の非復旧のコントロールプレーンマシンを 1 つずつ削除して再作成します。マシンが再作成された後、新しいリビジョンが強制され、**etcd** が自動的にスケールアップします。

- ユーザーがプロビジョニングしたベアメタルインストールを使用する場合は、最初に作成したときと同じ方法を使用して、コントロールプレーンマシンを再作成できます。詳細は、「ユーザーがプロビジョニングしたクラスターをベアメタルにインストールする」を参照してください。

警告

リカバリーホストのマシンを削除し、再作成しないでください。

- installer-provisioned infrastructure を実行している場合、またはマシン API を使用してマシンを作成している場合は、以下の手順を実行します。

警告

リカバリーホストのマシンを削除し、再作成しないでください。

installer-provisioned infrastructure でのベアメタルインストールの場合、コントロールプレーンマシンは再作成されません。詳細は、「ベアメタルコントロールプレーンノードの交換」を参照してください。

a. 失われたコントロールプレーンホストのいずれかのマシンを取得します。
クラスターにアクセスできるターミナルで、cluster-admin ユーザーとして以下のコマンドを実行します。

```
$ oc get machines -n openshift-machine-api -o wide
```

出力例:

```
NAME                                PHASE     TYPE        REGION      ZONE         AGE
NODE                           PROVIDERID                          STATE
clustername-8qw5l-master-0          Running   m4.xlarge   us-east-1   us-east-1a
3h37m   ip-10-0-131-183.ec2.internal   aws:///us-east-1a/i-0ec2782f8287dfb7e
stopped 1
```

```
clustername-8qw5l-master-1                 Running   m4.xlarge   us-east-1   us-east-1b
3h37m   ip-10-0-143-125.ec2.internal   aws:///us-east-1b/i-096c349b700a19631
running
clustername-8qw5l-master-2                 Running   m4.xlarge   us-east-1   us-east-1c
3h37m   ip-10-0-154-194.ec2.internal   aws:///us-east-1c/i-02626f1dba9ed5bba
running
clustername-8qw5l-worker-us-east-1a-wbtgd   Running   m4.large   us-east-1   us-
east-1a   3h28m   ip-10-0-129-226.ec2.internal   aws:///us-east-1a/i-
010ef6279b4662ced   running
clustername-8qw5l-worker-us-east-1b-lrdxb   Running   m4.large   us-east-1   us-
east-1b   3h28m   ip-10-0-144-248.ec2.internal   aws:///us-east-1b/i-
0cb45ac45a166173b   running
clustername-8qw5l-worker-us-east-1c-pkg26   Running   m4.large   us-east-1   us-
east-1c   3h28m   ip-10-0-170-181.ec2.internal   aws:///us-east-1c/i-
06861c00007751b0a   running
```

1. これは、失われたコントロールプレーンホストのコントロールプレーンマシンです (**ip-10-0-131-183.ec2.internal**)。

b. 以下を実行して、マシン設定をファイルシステム上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. 以下を実行して、**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. 以下を実行して、**metadata.name** フィールドを新しい名前に変更します。
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. 以下を実行して、**spec.providerID** フィールドを削除します。

```
providerID: aws:///us-east-1a/i-0fdb85790d76d0c3f
```

iv. 以下を実行して、**metadata.annotations** フィールドと **metadata.generation** フィールドを削除します。

```
annotations:
  machine.openshift.io/instance-state: running
...
generation: 2
```

v. 以下を実行して、**metadata.resourceVersion** フィールドと **metadata.uid** フィールドを削除します。

```
resourceVersion: "13291"
uid: a282eb70-40a2-4e89-8009-d05dd420d31a
```

d. 以下を実行して、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. 以下を実行して、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. 以下を実行して、**new-master-machine.yaml** ファイルを使用してマシンを作成します。

```
$ oc apply -f new-master-machine.yaml
```

g. 以下を実行して、新しい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。

16. 次のように入力して、クォーラムガードをオフにします。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides":
{"useUnsupportedUnsafeNonHANonProductionUnstableEtcd": true}}}'
```

このコマンドにより、シークレットを正常に再作成し、静的 Pod をロールアウトできるようになります。

17. リカバリーホスト内の別のターミナルウィンドウで、以下を実行してリカバリー **kubeconfig** ファイルをエクスポートします。

```
$ export KUBECONFIG=/etc/kubernetes/static-pod-resources/kube-apiserver-
certs/secrets/node-kubeconfigs/localhost-recovery.kubeconfig
```

18. **etcd** の再デプロイメントを強制的に実行します。
リカバリー **kubeconfig** ファイルをエクスポートしたのと同じターミナルウィンドウで、以下を実行します。

```
$ 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。

19. 次のように入力して、クォーラムガードをオンに戻します。

```
$ oc patch etcd/cluster --type=merge -p '{"spec": {"unsupportedConfigOverrides": null}}'
```

20. 以下を実行すると、**unsupportedConfigOverrides** セクションがオブジェクトから削除されたことを確認できます。

```
$ oc get etcd/cluster -oyaml
```

21. 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** ステータス条件を確認し、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。

22. **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 は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、**kube-apiserver** 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を実行します。

    a. **kube-apiserver** の新規ロールアウトを強制します。

    ```
    $ 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. 次のコマンドを実行して、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します。

    ```
    $ oc patch kubecontrollermanager cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
    ```

    以下を実行して、すべてのノードが最新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. 以下を実行して、**kube-scheduler** の新規ロールアウトを強制します。

```
$ oc patch kubescheduler cluster -p='{"spec": {"forceRedeploymentReason": "recovery-'"$( date --rfc-3339=ns )"'"}}' --type=merge
```

以下を使用して、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。

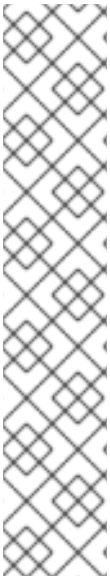
23. すべてのコントロールプレーンホストが起動しており、クラスターに参加していることを確認します。
クラスターにアクセスできるターミナルで、**cluster-admin** ユーザーとして以下のコマンドを実行します。

```
$ oc -n openshift-etcd get pods -l k8s-app=etcd
```

出力例

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

復元手順の後にすべてのワークロードが通常の動作に戻るようにするには、**kube-apiserver** 情報を保存している各 Pod を再起動します。これには、ルーター、Operator、サードパーティーコンポーネントなどの OpenShift Container Platform コンポーネントが含まれます。



注記

前の手順が完了したら、すべてのサービスが復元された状態に戻るまで数分間待つ必要がある場合があります。たとえば、**oc login** を使用した認証は、OAuth サーバー Pod が再起動するまですぐに機能しない可能性があります。

即時認証に **system:admin kubeconfig** ファイルを使用することを検討してください。この方法は、OAuth トークンではなく SSL/TLS クライアント証明書に基づいて認証を行います。以下のコマンドを実行し、このファイルを使用して認証できます。

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig
```

以下のコマンドを実行て、認証済みユーザー名を表示します。

```
$ oc whoami
```

5.3.2.3. 関連情報

- ユーザーによってプロビジョニングされるクラスターのベアメタルへのインストール
- SSH を使用して OpenShift Container Platform インスタンスおよびコントロールプレーンノードにアクセスするための bastion ホストを作成する方法
- ベアメタルコントロールプレーンノードの交換

5.3.2.4. 永続ストレージの状態復元に関する問題および回避策

OpenShift Container Platform クラスターがいずれかの形式の永続ストレージを使用する場合に、クラスターの状態は通常 etcd 外に保存されます。たとえば、Pod で実行されている Elasticsearch クラスター、または **StatefulSet** オブジェクトで実行されているデータベースなどである可能性があります。etcd バックアップから復元する場合には、OpenShift Container Platform のワークロードのステータスも復元されます。ただし、etcd スナップショットが古い場合には、ステータスは無効または期限切れの可能性があります。

重要

永続ボリューム (PV) の内容は etcd スナップショットには含まれません。etcd スナップショットから OpenShift Container Platform クラスターを復元する時に、重要ではないワークロードから重要なデータにアクセスしたり、その逆ができたりする場合があります。

以下は、古いステータスを生成するシナリオ例です。

- MySQL データベースが PV オブジェクトでバックアップされる Pod で実行されている。etcd スナップショットから OpenShift Container Platform を復元すると、Pod の起動を繰り返し試行しても、ボリュームをストレージプロバイダーに戻したり、実行中の MySQL Pod が生成したりされるわけではありません。この Pod は、ストレージプロバイダーでボリュームを復元し、次に PV を編集して新規ボリュームを参照するように手動で復元する必要があります。
- Pod P1 は、ノード X に割り当てられているボリューム A を使用している。別の Pod がノード Y にある同じボリュームを使用している場合に etcd スナップショットが作成された場合に、etcd の復元が実行されると、ボリュームがノード Y に割り当てられていることが原因で Pod P1 が正常に起動できなくなる可能性があります。OpenShift Container Platform はこの割り当

てを認識せず、ボリュームが自動的に切り離されるわけではありません。これが生じる場合には、ボリュームをノード Y から手動で切り離し、ノード X に割り当ててることで Pod P1 を起動できるようにします。

- クラウドプロバイダーまたはストレージプロバイダーの認証情報が etcd スナップショットの作成後に更新された。これが原因で、プロバイダーの認証情報に依存する CSI ドライバーまたは Operator が機能しなくなります。これらのドライバーまたは Operator で必要な認証情報を手動で更新する必要がある場合があります。

- デバイスが etcd スナップショットの作成後に OpenShift Container Platform ノードから削除されたか、名前が変更された。ローカルストレージ Operator で、**/dev/disk/by-id** または **/dev** ディレクトリーから管理する各 PV のシンボリックリンクが作成されます。この状況では、ローカル PV が存在しないデバイスを参照してしまう可能性があります。
この問題を修正するには、管理者は以下を行う必要があります。

  1. デバイスが無効な PV を手動で削除します。

  2. 各ノードからシンボリックリンクを削除します。

  3. **LocalVolume** または **LocalVolumeSet** オブジェクトを削除します (**ストレージ → 永続ストレージの設定 → ローカルボリュームを使用した永続ストレージ → ローカルストレージ Operator のリソースの削除** を参照)。

5.3.3. コントロールプレーン証明書の期限切れの状態からのリカバリー

5.3.3.1. コントロールプレーン証明書の期限切れの状態からのリカバリー

クラスターはコントロールプレーン証明書の期限切れの状態から自動的に回復できます。

ただし、kubelet 証明書を回復するために保留状態の **node-bootstrapper** 証明書署名要求 (CSR) を手動で承認する必要があります。ユーザーによってプロビジョニングされるインストールの場合は、保留中の kubelet 提供の CSR を承認しないといけない場合があります。

保留中の CSR を承認するには、以下の手順に従います。

手順

1. 現在の CSR の一覧を取得します。

```
$ oc get csr
```

出力例

```
NAME        AGE    SIGNERNAME                                    REQUESTOR
CONDITION
csr-2s94x   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending (1)
csr-4bd6t   8m3s   kubernetes.io/kubelet-serving                 system:node:<node_name>
Pending
csr-4hl85   13m    kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending (2)
csr-zhhhp   3m8s   kubernetes.io/kube-apiserver-client-kubelet
system:serviceaccount:openshift-machine-config-operator:node-bootstrapper   Pending
...
```

1. 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)。
2. 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 のリストからの 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>
```