



OpenShift Container Platform 4.2

移行

OpenShift Container Platform 3 から 4 への移行

OpenShift Container Platform 4.2 移行

OpenShift Container Platform 3 から 4 への移行

法律上の通知

Copyright © 2020 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

本書では、OpenShift Container Platform クラスターをバージョン 3 からバージョン 4 に移行する方法について説明します。

目次

第1章 OPENSIFT CONTAINER PLATFORM 3 から 4 への移行	3
1.1. OPENSIFT CONTAINER PLATFORM 3 から 4 への移行について	3
1.2. 移行の計画	3
1.3. アプリケーションワークロードの OPENSIFT CONTAINER PLATFORM 3.7 から 4.2 への移行	7
1.4. レプリケーションリポジトリの設定	10
1.5. CLUSTER APPLICATION MIGRATION (CAM) ツールのデプロイ	21
1.6. CAM WEB コンソールを使用したアプリケーションの移行	24
1.7. CONTROL PLANE MIGRATION ASSISTANT (CPMA) でのコントロールプレーン設定の移行	30
1.8. トラブルシューティング	34
第2章 OPENSIFT CONTAINER PLATFORM 4.1 から 4.2 への移行	43
2.1. アプリケーションワークロードの OPENSIFT CONTAINER PLATFORM 4.1 から 4.2 への移行	43
2.2. レプリケーションリポジトリの設定	45
2.3. CLUSTER APPLICATION MIGRATION (CAM) ツールのデプロイ	56
2.4. CAM WEB コンソールを使用したアプリケーションの移行	59
2.5. トラブルシューティング	65
第3章 OPENSIFT CONTAINER PLATFORM 4.2 クラスター間の移行	74
3.1. OPENSIFT CONTAINER PLATFORM 4.2 クラスター間のアプリケーションワークロードの移行	74
3.2. レプリケーションリポジトリの設定	76
3.3. CLUSTER APPLICATION MIGRATION (CAM) ツールのデプロイ	86
3.4. CAM WEB コンソールを使用したアプリケーションの移行	89
3.5. トラブルシューティング	95

第1章 OPENSIFT CONTAINER PLATFORM 3 から 4 への移行

1.1. OPENSIFT CONTAINER PLATFORM 3 から 4 への移行について

OpenShift Container Platform 4 には、自己管理型の柔軟で自動化されたクラスターを実現する新規のテクノロジーおよび機能が含まれています。OpenShift Container Platform 4 クラスターがデプロイされ、管理される方法は OpenShift Container Platform 3 とは大きく異なります。

OpenShift Container Platform 3 から OpenShift Container Platform 4 に正常に移行するには、以下の情報を確認することが重要になります。

移行の計画

OpenShift Container Platform のバージョン 3 と 4 の相違点について確認します。移行の前に、ストレージ、ネットワーク、ロギング、セキュリティ、およびモニタリングに関する考慮事項について確認し、準備済みであることを確認します。

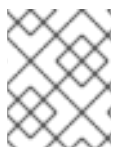
移行の実行

移行を実行するためのツールについて確認し、これらを使用します。

- アプリケーションのワークロードを移行するための Cluster Application Migration (CAM) ツール
- コントロールプレーンを移行するための Control Plane Migration Assistant (CPMA)

1.2. 移行の計画

OpenShift Container Platform 4.2 への移行を実行する前に、十分な時間を取って移行を適切に計画できるようにしてください。OpenShift Container Platform 4 ではアーキテクチャーの変更および拡張機能が導入されるため、OpenShift Container Platform 3 クラスターの管理に使用した手順は OpenShift Container Platform 4 で適用されない可能性があります。



注記

この計画ガイドでは、OpenShift Container Platform 3.11 から OpenShift Container Platform 4.2 への移行を前提としています。

本書では、最も重要な [OpenShift Container Platform 3 と OpenShift Container Platform 4 の相違点](#)と、最も注目すべき [移行に関する考慮事項](#)についての概要を説明します。OpenShift Container Platform 4 クラスターの設定についての詳細は、OpenShift Container Platform ドキュメントの該当するセクションを参照してください。新規機能および他の重要な技術上の変更点についての詳細は、『[OpenShift Container Platform 4.2 リリースノート](#)』を参照してください。

既存の OpenShift Container Platform 3 クラスターを OpenShift Container Platform 4 にアップグレードすることはできません。新規の OpenShift Container Platform 4 インストールで開始する必要があります。コントロールプレーンの設定およびアプリケーションのワークロードの移行に役立つツールを使用できます。

1.2.1. OpenShift Container Platform 3 と OpenShift Container Platform 4 の比較

OpenShift Container Platform 3 では、管理者は Red Hat Enterprise Linux (RHEL) ホストを個別にデプロイし、その後に OpenShift Container Platform をこれらのホストにインストールし、クラスターを作成しました。管理者は、これらのホストを適切に設定し、更新を実行する必要があります。

OpenShift Container Platform 4 では、これまでとは大きく異なる方法で OpenShift Container Platform クラスターがデプロイされ、管理されるようになりました。OpenShift Container Platform 4 には、Operator、MachineSet、および Red Hat Enterprise Linux CoreOS (RHCOS) などの、クラスターの操作に対するコアとなる新たなテクノロジーおよび機能が含まれます。このテクノロジーの移行により、クラスターは管理者が以前に実行していた一部の機能を自己管理できるようになります。また、プラットフォームの安定性と一貫性を確保し、インストールおよびスケーリングを単純化することが可能です。

詳細は、『[OpenShift Container Platform アーキテクチャー](#)』を参照してください。

1.2.1.1. アーキテクチャーの違い

イミュータブルなインフラストラクチャー

OpenShift Container Platform 4 は、コンテナ化されたアプリケーションを実行するために設計された Red Hat Enterprise Linux CoreOS (RHCOS) を使用し、効率的なインストール、Operator ベースの管理、および単純化されたアップグレードを可能にします。RHCOS は、RHEL のようなカスタマイズ可能なオペレーティングシステムではなく、イミュータブルなコンテナホストです。RHCOS により、OpenShift Container Platform 4 は基礎となるコンテナホストのデプロイメントを管理し、自動化できます。RHCOS は OpenShift Container Platform の一部です。これは、すべてがコンテナ内で実行され、OpenShift Container Platform を使用してデプロイされることを意味します。

OpenShift Container Platform 4 では、コントロールプレーンノードは RHCOS を実行する必要があるため、コントロールプレーンのフルスタック自動化が維持されます。これにより、OpenShift Container Platform 3 よりも簡単に更新がロールアウトされ、アップグレードが簡単になります。

詳細は、『[Red Hat Enterprise Linux CoreOS](#)』を参照してください。

Operator

Operator は、Kubernetes アプリケーションをパッケージ化し、デプロイし、管理する方法です。Operator は、ソフトウェアの他の部分を実行する際の運用上の複雑さを軽減します。Operator は環境を監視し、現在の状態を使用してリアルタイムの意思決定を行います。高度な Operator は、自動的にアップグレードし、障害に自動的に対応するように設計されています。

詳細は、『[Operator について](#)』を参照してください。

1.2.1.2. インストールおよび更新の違い

インストールプロセス

OpenShift Container Platform 3.11 をインストールするには、Red Hat Enterprise Linux (RHEL) ホストを準備し、クラスターが必要とする設定値をすべて設定してから、Ansible Playbook を実行してクラスターをインストールし、セットアップする必要がありました。

OpenShift Container Platform 4.2 では、OpenShift インストールプログラムを使用してクラスターに必要なリソースの最小セットを作成できます。クラスターの実行後に、Operator を使用してクラスターをさらに設定し、新規サービスをインストールすることができます。初回の起動後に、Red Hat Enterprise Linux CoreOS (RHCOS) システムは、OpenShift Container Platform クラスターで実行される Machine Config Operator (MCO) によって管理されます。

詳細は、『[インストールプロセス](#)』を参照してください。

RHEL ワーカーマシンを OpenShift Container Platform 4.2 クラスターに追加する場合、Ansible Playbook を使用して、クラスターの実行後に RHEL ワーカーマシンを追加します。詳細は、『[RHEL コンピュータマシンの OpenShift Container Platform クラスターへの追加](#)』を参照してください。

インフラストラクチャーオプション

OpenShift Container Platform 3.11 では、ユーザーが準備し、維持するインフラストラクチャーにクラ

スターをインストールする必要があります。OpenShift Container Platform 4 では、独自のインフラストラクチャーを提供するだけでなく、OpenShift Container Platform インストールプログラムがプロビジョニングし、クラスターが維持するインフラストラクチャーにクラスターをデプロイするオプションを提供します。

詳細は、「[OpenShift Container Platform インストールの概要](#)」を参照してください。

クラスターのアップグレード

OpenShift Container Platform 3.11 では、Ansible Playbook を実行してクラスターをアップグレードします。OpenShift Container Platform 4.2 では、クラスターが、クラスターノードの Red Hat Enterprise Linux CoreOS (RHCOS) への更新を含む独自の更新を管理します。Web コンソールまたは OpenShift CLI から **oc adm upgrade** コマンドを使用することでクラスターを容易にアップグレードでき、Operator は自動的にアップグレードされます。OpenShift Container Platform 4.2 クラスターに Red Hat Enterprise Linux ワーカーマシンがある場合、Ansible Playbook を引き続き実行してそれらのワーカーマシンをアップグレードする必要があります。

詳細は、『[クラスターの更新](#)』を参照してください。

1.2.2. 移行に関する考慮事項

OpenShift Container Platform 3.11 から OpenShift Container Platform 4 への移行に影響を与える可能性のある変更やその他の考慮事項を確認します。

1.2.2.1. ストレージに関する考慮事項

OpenShift Container Platform 3.11 から OpenShift Container Platform 4.2 に移行する際に、以下のストレージの変更を考慮してください。

ローカルボリュームの永続ストレージ

ローカルストレージは、OpenShift Container Platform 4.2 ではローカルストレージ Operator を使用する場合にのみサポートされます。OpenShift Container Platform 3.11 のローカルプロビジョナーメソッドの使用はサポートされません。

詳細は、「[ローカルボリュームを使用した永続ストレージ](#)」を参照してください。

FlexVolume 永続ボリューム

FlexVolume プラグインの場所が OpenShift Container Platform 3.11 で変更になりました。OpenShift Container Platform 4.2 の新しい場所は **/etc/kubernetes/kubelet-plugins/volume/exec** です。割り当て可能な FlexVolume プラグインはサポートされなくなりました。

詳細は、「[FlexVolume を使用した永続ストレージ](#)」を参照してください。

Container Storage Interface (CSI) 永続ストレージ

Container Storage Interface (CSI) を使用した永続ストレージは OpenShift Container Platform 3.11 では [テクノロジープレビュー](#)として利用可能でした。CSI バージョン 1.1.0 は OpenShift Container Platform 4.2 で完全にサポートされていますが、これは CSI ドライバーに同梱されません。そのため、独自のドライバをインストールする必要があります。

詳細は、「[Container Storage Interface \(CSI\) を使用した永続ストレージ](#)」を参照してください。

Red Hat OpenShift Container Storage

OpenShift Container Platform 3.11 で使用できる Red Hat OpenShift Container Storage 3 は、バックアップストレージとして Red Hat Gluster Storage を使用します。

OpenShift Container Platform 4 で使用できる Red Hat OpenShift Container Storage 4 は、バックアップストレージとして Red Hat Ceph Storage を使用します。

詳細は、「[Persistent storage using Red Hat OpenShift Container Storage](#)」および「[interoperability matrix](#)」の記事を参照してください。

サポートされていない永続ストレージオプション

OpenShift Container Platform 3.11 の以下の永続ストレージオプションのサポートが OpenShift Container Platform 4.2 で変更になりました。

- GlusterFS はサポート対象外になりました。
- スタンドアロン製品としての CephFS がサポートされなくなりました。
- スタンドアロン製品としての Ceph RBD がサポートされなくなりました。
- iSCSI がテクノロジープレビューとしてご利用いただけます。

OpenShift Container Platform 3.11 でこれらのいずれかを使用していた場合は、OpenShift Container Platform 4.2 で完全にサポートされる別の永続ストレージオプションを選択する必要があります。

詳細は、「[永続ストレージについて](#)」を参照してください。

1.2.2.2. ネットワークの考慮事項

OpenShift Container Platform 3.11 から OpenShift Container Platform 4.2 に移行する際に、考慮事項となる以下のネットワークの変更を確認してください。

ネットワーク分離モード

OpenShift Container Platform 3.11 では、ユーザーは **ovn-multitenant** を使用するように頻繁に切り替えましたが、デフォルトのネットワーク分離モードは **ovs-subnet** でした。OpenShift Container Platform 4.2 のデフォルトのネットワーク分離モードは NetworkPolicy になりました。

OpenShift Container Platform 3.11 クラスタで **ovs-subnet** または **ovs-multitenant** モードを使用していた場合、OpenShift Container Platform 4.2 クラスタでは NetworkPolicy モードに切り換えることが推奨されます。NetworkPolicy はアップストリームでサポートされ、より柔軟であり、**ovs-multitenant** が実行する機能を提供します。OpenShift Container Platform 4.2 で NetworkPolicy を使用する際に **ovs-multitenant** 動作を維持する必要がある場合、[NetworkPolicy を使用したマルチテナント分離の設定手順](#)を実行します。

詳細は、「[ネットワークポリシーについて](#)」を参照してください。

ホスト間のトラフィックの暗号化

OpenShift Container Platform 3.11 では、IPsec を使用してホスト間のトラフィックを暗号化できます。OpenShift Container Platform 4.2 は IPsec をサポートしません。サービス間で相互 TLS を有効にするために、Red Hat OpenShift Service Mesh を使用することが推奨されます。

詳細は、[Red Hat OpenShift Service Mesh](#) について参照してください。

1.2.2.3. ロギングについての考慮事項

OpenShift Container Platform 3.11 から OpenShift Container Platform 4.2 に移行する際に、考慮事項となる以下のロギングの変更を確認してください。

クラスターロギングのデプロイ

OpenShift Container Platform 4 は、クラスターロギングカスタムリソースを使用してクラスターロギングの単純なデプロイメントメカニズムを提供します。デプロイが完了すると、クラスターロギングは OpenShift Container Platform 3.11 の場合と同じように使用できます。

詳細は、「[クラスターロギングのデプロイおよび設定について](#)」を参照してください。

集計ログインデータ

集計ログインデータを OpenShift Container Platform 3.11 から新規の OpenShift Container Platform 4 クラスターに移行することはできません。

詳細は、「[クラスターログインについて](#)」を参照してください。

1.2.2.4. セキュリティーに関する考慮事項

OpenShift Container Platform 3.11 から OpenShift Container Platform 4.2 へ移行する際に、考慮事項となる以下のセキュリティーの変更を確認してください。

検出エンドポイントへの認証されていないアクセス

OpenShift Container Platform 3.11 では、認証されていないユーザーは検出エンドポイント (例: `/api/*` および `/apis/*`) にアクセスできました。セキュリティー上の理由から、検出エンドポイントへの認証されていないアクセスは OpenShift Container Platform 4.2 で許可されなくなりました。認証されていないアクセスを許可する必要がある場合は、必要に応じて RBAC を設定できます。ただし、これにより内部クラスターコンポーネントが外部ネットワークに公開される可能性があるため、セキュリティー上の影響を考慮してください。

アイデンティティープロバイダー

アイデンティティープロバイダーの設定は、以下の主な変更点を含め、OpenShift Container Platform 4 で変更されています。

- OpenShift Container Platform 4.2 の要求ヘッダーアイデンティティープロバイダーには相互 TLS が必要ですが、OpenShift Container Platform 3.11 ではこれは必要ではありませんでした。
- OpenID Connect アイデンティティープロバイダーの設定は OpenShift Container Platform 4.2 で単純化されています。OpenShift Container Platform 3.11 で以前に指定される必要のあったデータが、プロバイダーの `/.well-known/openid-configuration` エンドポイントから取得できるようになりました。

詳細は、「[アイデンティティープロバイダー設定について](#)」を参照してください。

1.2.2.5. モニタリングに関する考慮事項

OpenShift Container Platform 3.11 から OpenShift Container Platform 4.2 に移行する際に、考慮事項となる以下のモニタリングの変更を確認してください。

インフラストラクチャーの可用性についてのモニタリングアラート

モニタリング構造の可用性を確保するためにトリガーするデフォルトのアラートは OpenShift Container Platform 3.11 では **DeadMansSwitch** と呼ばれていました。この名前は OpenShift Container Platform 4 で **Watchdog** に変更されています。OpenShift Container Platform 3.11 で PagerDuty 統合をこのアラートでセットアップしている場合、OpenShift Container Platform 4 では **Watchdog** アラートについて PagerDuty 統合をセットアップする必要があります。

詳細は、「[カスタム Alertmanager 設定の適用](#)」を参照してください。

1.3. アプリケーションワークロードの OPENSIFT CONTAINER PLATFORM 3.7 から 4.2 への移行

アプリケーションワークロードを、Cluster Application Migration (CAM) ツールを使用して OpenShift Container Platform 3.7 (以降) から OpenShift Container Platform 4.2 に移行できます。CAM ツールを使用すると、移行を制御し、アプリケーションのダウンタイムを最小限に抑えることができます。

Kubernetes カスタムリソースをベースとする CAM ツールの Web コンソールおよび API により、namespace の粒度でステータフルなアプリケーションワークロードを移行できます。

オプションで、コントロールプレーン設定の移行に役立つ [Control Plane Migration Assistant \(CPMA\)](#) を使用できます。



重要

移行を開始する前に、[移行計画](#) についての情報を確認してください。

1.3.1. 移行の前提条件

- ソースクラスターは OpenShift Container Platform 3.7、3.9、3.10、または 3.11 であること。
- **podman** がインストールされていること。
- すべてのクラスターで **cluster-admin** 権限があること。
- ソースおよびターゲットクラスターには、レプリケーションリポジトリへの無制限のネットワークアクセスがなければなりません。
- Migration コントローラーがインストールされているクラスターには、他のクラスターへの無制限のアクセスが必要です。
- アプリケーションが **openshift** namespace のイメージを使用する場合、イメージの必要なバージョンがターゲットクラスターにある必要がある。
必要なイメージがない場合は、アプリケーションと互換性のある利用可能なバージョンを使用するように **imagestreamtags** 参照を更新する必要があります。**imagestreamtag** を更新できない場合、同等のイメージをアプリケーション namespace に手動でアップロードし、それらを参照するようにアプリケーションを更新できます。

以下の **imagestreamtag** は OpenShift Container Platform 4.2 から **削除** されています。

- **dotnet:1.0**、**dotnet:1.1**、**dotnet:2.0**
- **dotnet-runtime:2.0**
- **mariadb:10.1**
- **mongodb:2.4**、**mongodb:2.6**
- **mysql:5.5**、**mysql:5.6**
- **nginx:1.8**
- **nodejs:0.10**、**nodejs:4**、**nodejs:6**
- **perl:5.16**、**perl:5.20**
- **php:5.5**、**php:5.6**
- **postgresql:9.2**、**postgresql:9.4**、**postgresql:9.5**
- **python:3.3**、**python:3.4**
- **ruby:2.0**、**ruby:2.2**

1.3.2. Cluster Application Migration ツールについて

Cluster Application Migration (CAM) ツールを使用すると、CAM Web コンソールまたは Kubernetes API を使用して Kubernetes リソース、永続ボリュームデータ、および内部コンテナイメージを OpenShift Container Platform ソースクラスターから OpenShift Container Platform 4.2 ターゲットクラスターに移行できます。

CAM Web コンソールを使用してアプリケーションを移行するには、以下の手順が必要です。

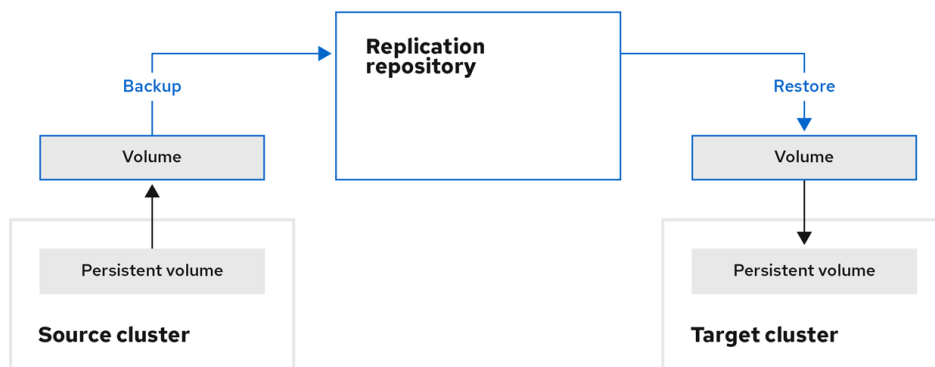
1. Cluster Application Migration Operator をすべてのクラスターにインストールします。



注記

Cluster Application Migration Operator はデフォルトで CAM ツール (CAM Web コンソールおよび Migration コントローラー) をターゲットクラスターにインストールします。CAM ツールは [OpenShift Container Platform 3](#) および [リモートクラスターにインストール](#) できます。

2. CAM ツールがデータ移行に使用する中間オブジェクトストレージであるレプリケーションリポジトリを設定します。
3. ソースクラスターを CAM Web コンソールに追加します。
4. レプリケーションリポジトリを CAM Web コンソールに追加します。
5. 以下のデータ移行オプションのいずれかを使用して、移行計画を作成します。
 - **Copy:** CAM ツールは、データをソースクラスターからレプリケーションリポジトリにコピーし、レプリケーションリポジトリからターゲットクラスターにコピーします。



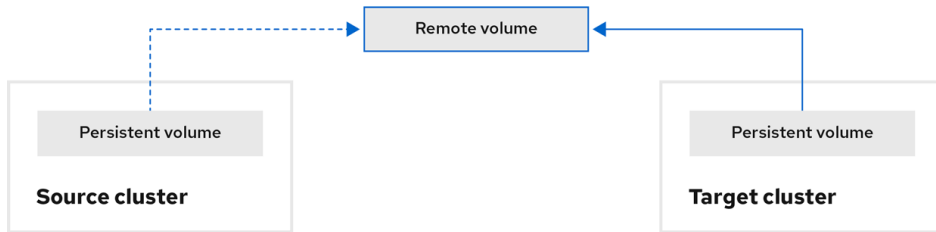
OpenShift_45_1019

- **Move:** CAM ツールはリモートボリューム (NFS など) をソースクラスターからアンマウントし、リモートボリュームをポイントするターゲットクラスターで PV リソースを作成し、その後にリモートボリュームをターゲットクラスターにマウントします。ターゲットクラスターで実行されているアプリケーションは、ソースクラスターが使用していたものと同じリモートボリュームを使用します。リモートボリュームは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。



注記

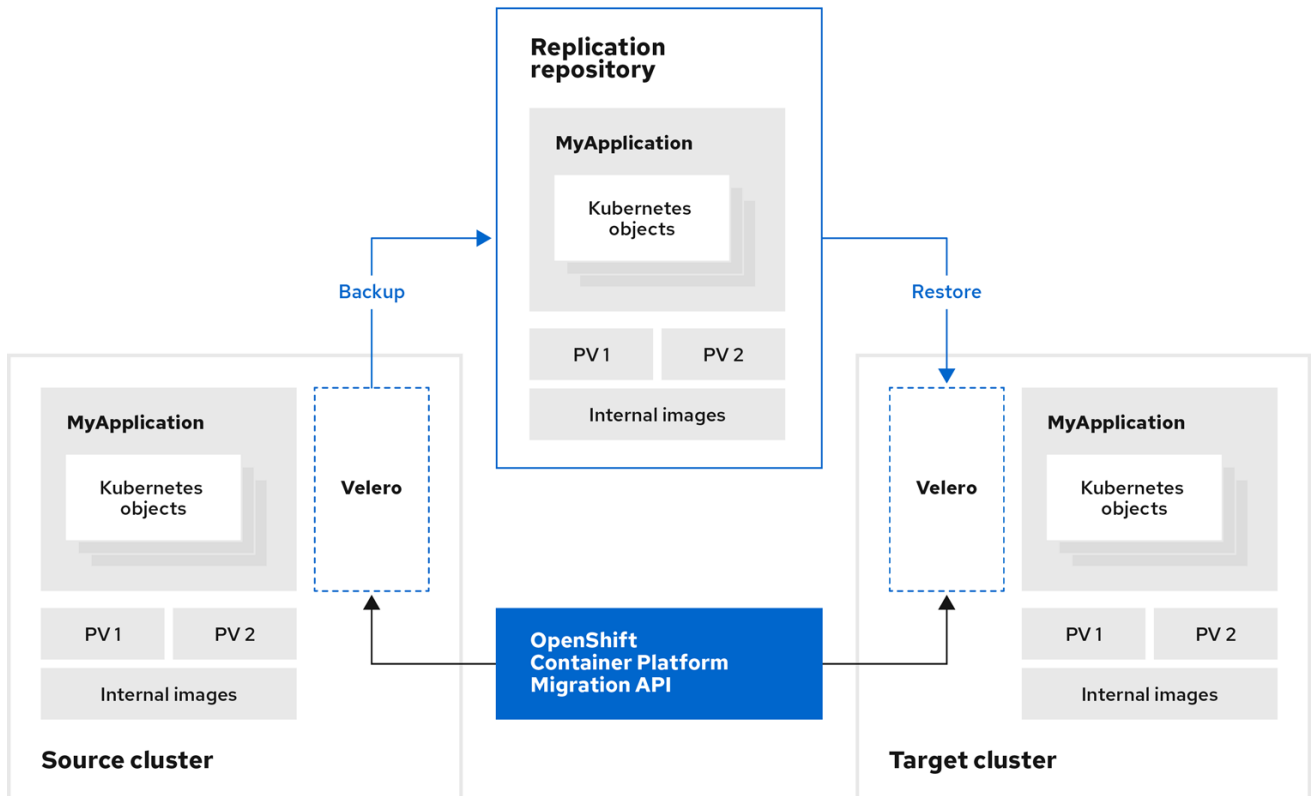
レプリケーションリポジトリはこの図には表示されていませんが、実際の移行には必要になります。



OpenShift_45_1019

6. 以下のオプションのいずれかを使用して、移行計画を実行します。

- Stage** (オプション) は、アプリケーションを停止せずにデータをターゲットクラスターにコピーします。
 ステージングは、移行前にほとんどのデータがターゲットにコピーされるように複数回実行することができます。これにより、実際の移行時間やアプリケーションのダウンタイムが最小限に抑えられます。
- Migrate** は、ソースクラスターでアプリケーションを停止し、ターゲットクラスターでそのリソースを再作成します。オプションで、アプリケーションを停止せずにワークロードを移行できます。



OpenShift_45_1019

1.4. レプリケーションリポジトリの設定

オブジェクトストレージをレプリケーションリポジトリとして使用するよう設定する必要があります。Cluster Application Migration ツールは、[ファイルシステム](#)または[スナップショット](#)のいずれかのデータの複製方法を使用して、データをソースクラスターからレプリケーションリポジトリにコピーしてから、レプリケーションリポジトリからターゲットクラスターにコピーします。

以下のストレージプロバイダーがサポートされています。

- 汎用 S3 オブジェクトストレージ (例: Minio または Ceph S3)
- Multi-Cloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

1.4.1. 移行用のデータのコピー方法について

CAM ツールは、ソースクラスターからターゲットクラスターにデータを移行するためにファイルシステムおよびスナップショットによるデータの複製方法をサポートします。ご使用の環境に適した方法で、ストレージプロバイダーでサポートされる方法を選択できます。

1.4.1.1. ファイルシステムの複製方法

CAM ツールは、データファイルをソースクラスターからレプリケーションリポジトリに複製し、そこからターゲットクラスターに複製します。

表1.1 ファイルシステムの複製方法の概要

利点	制限
<ul style="list-style-type: none"> ● クラスターで複数の異なるストレージクラスを使用することが可能 ● すべての S3 ストレージプロバイダーでサポートされている 	<ul style="list-style-type: none"> ● スナップショットの複製方法よりも遅い

1.4.1.2. スナップショットの複製方法

CAM ツールは、ソースクラスターのデータのスナップショットを、レプリケーションリポジトリとして設定されたクラウドプロバイダーのオブジェクトストレージに複製します。データはターゲットクラスターで復元されます。

AWS、Google Cloud Provider、および Microsoft Azure は、スナップショットの複製方法をサポートします。

表1.2 スナップショットの複製方法の概要

利点	制限
----	----

利点	制限
<ul style="list-style-type: none"> ● ファイルシステムのコピー方法よりも高速 	<ul style="list-style-type: none"> ● クラウドプロバイダーはスナップショットをサポートしている必要があります。 ● クラスタは同じクラウドプロバイダーになければなりません。 ● クラスタは、同じ場所またはリージョンにある必要があります。 ● クラスタには同じストレージクラスがなければなりません。 ● ストレージクラスにはスナップショットとの互換性がある必要があります。

重要

Multi-Cloud Object Gateway を移行用のレプリケーションリポジトリとして設定する機能はテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。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。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

Red Hat のテクノロジープレビュー機能のサポート範囲についての詳細は、<https://access.redhat.com/ja/support/offerings/techpreview/> を参照してください。

1.4.2. Multi-Cloud Object Gateway ストレージバケットをレプリケーションリポジトリとして設定する

OpenShift Container Storage Operator をインストールし、Multi-Cloud Object Gateway (MCG) ストレージバケットをレプリケーションリポジトリとして設定できます。

1.4.2.1. OpenShift Container Storage Operator のインストール

OpenShift Container Storage Operator は、OperatorHub からインストールできます。

手順

1. OpenShift Container Platform Web コンソールで、**Administration** → **Namespaces** をクリックします。
2. **Create Namespace** をクリックします。
3. **Name** フィールドに **openshift-storage** を入力し、**Create** をクリックします。
4. **Operators** → **OperatorHub** をクリックします。
5. **Filter by keyword** (この場合は、**OCS**) を使用し、**OpenShift Container Storage Operator** を見つけます。

6. **OpenShift Container Storage Operator** を選択し、**Install** をクリックします。
7. **Create Operator Subscription** ページで、**openshift-storage** namespace を選択します。
8. 更新チャンネルおよび承認ストラテジーを指定します。
9. **Subscribe** をクリックします。
Installed Operators ページで、**OpenShift Container Storage Operator** は、**Succeeded** のステータスと共に **openshift-storage** プロジェクトに表示されます。

1.4.2.2. Multi-Cloud Object Gateway ストレージバケットの作成

Multi-Cloud Object Gateway (MCG) ストレージバケットのカスタムリソース (CR) を作成できます。

手順

1. OpenShift Container Platform クラスタにログインします。

```
$ oc login
```

2. **NooBaa** CR 設定ファイル **noobaa.yml** を以下の内容で作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5 1
      memory: 1Gi
  coreResources:
    requests:
      cpu: 0.5 2
      memory: 1Gi
```

1 **2** 非常に小規模なクラスタの場合、**cpu** の値を **0.1** に変更できます。

3. **NooBaa** オブジェクトを作成します。

```
$ oc create -f noobaa.yml
```

4. 以下の内容で、**BackingStore** CR 設定ファイル **bs.yml** を作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: BackingStore
metadata:
  finalizers:
    - noobaa.io/finalizer
  labels:
    app: noobaa
  name: mcg-pv-pool-bs
```

```
namespace: openshift-storage
spec:
  pvPool:
    numVolumes: 3 ①
    resources:
      requests:
        storage: 50Gi ②
    storageClass: gp2 ③
  type: pv-pool
```

- ① PV プール内のボリューム数を指定します。
- ② ボリュームのサイズを指定します。
- ③ ストレージクラスを作成します。

5. **BackingStore** オブジェクトを作成します。

```
$ oc create -f bs.yml
```

6. 以下の内容で **BucketClass** CR 設定ファイル **bc.yml** を作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
    name: mcg-pv-pool-bc
  namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
      - backingStores:
          - mcg-pv-pool-bs
    placement: Spread
```

7. **BucketClass** オブジェクトを作成します。

```
$ oc create -f bc.yml
```

8. 以下の内容で **ObjectBucketClaim** CR 設定ファイル **obc.yml** を作成します。

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage
  namespace: openshift-storage
spec:
  bucketName: migstorage ①
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc
```

- 1 レプリケーションリポジトリを CAM Web コンソールに追加するために使用するバケット名を記録します。

9. **ObjectBucketClaim** オブジェクトを作成します。

```
$ oc create -f obc.yml
```

10. リソース作成プロセスを監視し、**ObjectBucketClaim** ステータスが **Bound** であることを確認します。

```
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

このプロセスには、5分から10分の時間がかかる場合があります。

11. 以下の値を取得して記録します。この値は、レプリケーションリポジトリを CAM Web コンソールに追加する際に必要になります。

- S3 エンドポイント:

```
$ oc get route -n openshift-storage s3
```

- S3 プロバイダーアクセスキー:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{.data.AWS_ACCESS_KEY_ID }}' | base64 -d
```

- S3 プロバイダーシークレットアクセスキー:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{.data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
```

1.4.3. AWS S3 ストレージバケットをレプリケーションリポジトリとして設定する

AWS S3 ストレージバケットをレプリケーションリポジトリとして設定できます。

前提条件

- AWS S3 ストレージバケットは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
- [AWS CLI](#) がインストールされていること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - EC2 Elastic Block Storage (EBS) にアクセスできる必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. AWS S3 バケットを作成します。

```
$ aws s3api create-bucket \  
  --bucket <bucket_name> \ ①  
  --region <bucket_region> ②
```

- ① S3 バケット名を指定します。
- ② S3 バケットリージョンを指定します (例: **us-east-1**)。

2. IAM ユーザー **velero** を作成します。

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

3. EC2 EBS スナップショットポリシーを作成します。

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

4. 1つまたはすべての S3 バケットの AWS S3 アクセスポリシーを作成します。

```
$ cat > velero-s3-policy.json <<EOF  
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "s3:GetObject",  
        "s3>DeleteObject",  
        "s3:PutObject",  
        "s3:AbortMultipartUpload",  
        "s3:ListMultipartUploadParts"  
      ],  
      "Resource": [  
        "arn:aws:s3:::<bucket_name>/*" ①  
      ]  
    }  
  ]  
}
```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket_name>" ❷
      ]
    }
  ]
}
EOF

```

- ❶ ❷ 単一の S3 バケットへのアクセスを付与するには、バケット名を指定します。すべての AWS S3 バケットへのアクセスを付与するには、バケット名の代わりに * を指定します。

```

"Resource": [
  "arn:aws:s3::*"
]

```

5. EC2 EBS ポリシーを **velero** に割り当てます。

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-ebs \
  --policy-document file://velero-ec2-snapshot-policy.json

```

6. AWS S3 ポリシーを **velero** に割り当てます。

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-s3 \
  --policy-document file://velero-s3-policy.json

```

7. **velero** のアクセスキーを作成します。

```

$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, ❶
    "AccessKeyId": <AWS_ACCESS_KEY_ID> ❷
  }
}

```

- ❶ ❷ AWS リポジトリを CAM Web コンソールに追加するために **AWS_SECRET_ACCESS_KEY** および **AWS_ACCESS_KEY_ID** を記録します。

1.4.4. Google Cloud Provider ストレージバケットをレプリケーションリポジトリとして設定する

Google Cloud Provider (GCP) ストレージバケットをレプリケーションリポジトリとして設定できません。

前提条件

- GCP ストレージバケットは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
- **gsutil** がインストールされていること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. **gsutil init** を実行してログインします。

```
$ gsutil init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

To continue, you must login. Would you like to login (Y/n)?
```

2. **BUCKET** 変数を設定します。

```
$ BUCKET=<bucket_name> ❶
```

- ❶ 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. **velero** サービスアカウントを作成します。

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

6. **SERVICE_ACCOUNT_EMAIL** 変数をサービスアカウントのメールアドレスに設定します。

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

7. パーミッションをサービスアカウントに付与します。

```
$ 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
)

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

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

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

8. 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
```

1.4.5. Microsoft Azure Blob ストレージコンテナをレプリケーションリポジトリとして設定

Microsoft Azure Blob ストレージコンテナをレプリケーションリポジトリとして設定できます。

前提条件

- [Azure ストレージアカウント](#)があること。
- [Azure CLI](#) がインストールされていること。
- Azure Blob ストレージコンテナがソースクラスターおよびターゲットクラスターからアクセスできること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。

- ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. **AZURE_RESOURCE_GROUP** 変数を設定します。

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

2. 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。

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

```
$ AZURE_STORAGE_ACCOUNT_ID=velerobackups
```

4. 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
```

5. **BLOB_CONTAINER** 変数を設定します。

```
$ BLOB_CONTAINER=velero
```

6. Azure Blob ストレージコンテナを作成します。

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

7. **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`
```

8. 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
```

1.5. CLUSTER APPLICATION MIGRATION (CAM) ツールのデプロイ

Cluster Application Migration (CAM) ツールをデプロイするには、Cluster Application Migration Operator を [OpenShift Container Platform 3 ソース](#) と [OpenShift Container Platform 4.2 ターゲット クラスター](#) にインストールし、[OpenShift Container Platform 3 ソースクラスター](#) でクロスオリジンリソース共有を設定する必要があります。

Cluster Application Migration Operator はデフォルトで CAM ツール (CAM Web コンソールおよび Migration コントローラー) をターゲットクラスターにインストールします。[CAM ツールは OpenShift Container Platform 3 およびリモートクラスターにインストールできます。](#)

1.5.1. OpenShift Container Platform 3 ソースクラスターへの Cluster Application Migration Operator のインストール

OpenShift Container Platform 3 は Operator Lifecycle Manager をサポートしないため、Cluster Application Migration Operator は OpenShift Container Platform 3 ソースクラスターに手動でインストールできます。

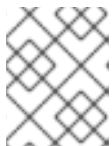
前提条件

- **podman** がインストールされていること。
- OpenShift Container Platform 3 クラスターは、イメージを [registry.redhat.io](#) からプルするように設定される必要があります。
イメージをプルするには、[imagestreamsecret](#) を作成し、これをクラスター内の各ノードにコピーする必要があります。

手順

1. Red Hat カスタマーポータルでの認証情報を使用して [registry.redhat.io](#) にログインします。

```
$ sudo podman login registry.redhat.io
```



注記

システムがルートレス Podman コンテナ用に設定されている場合は、この手順に **sudo** は必要ありません。

2. **operator.yml** ファイルをダウンロードします。

```
$ sudo podman cp $(sudo podman create registry.redhat.io/rhcam-1-2/openshift-migration-rhel7-operator:v1.2):/operator.yml ./
```

3. **controller-3.yml** ファイルをダウンロードします。

```
$ sudo podman cp $(sudo podman create registry.redhat.io/rhcam-1-2/openshift-migration-rhel7-operator:v1.2):/controller-3.yml ./
```

4. OpenShift Container Platform 3 クラスターにログインします。
5. クラスターが registry.redhat.io で認証できることを確認します。

```
$ oc run test --image registry.redhat.io/ubi8 --command sleep infinity
```

6. Cluster Application Migration Operator CR オブジェクトを作成します。

```
$ oc create -f operator.yml
namespace/openshift-migration created
rolebinding.rbac.authorization.k8s.io/system:deployers created
serviceaccount/migration-operator created
customresourcedefinition.apiextensions.k8s.io/migrationcontrollers.migration.openshift.io
created
role.rbac.authorization.k8s.io/migration-operator created
rolebinding.rbac.authorization.k8s.io/migration-operator created
clusterrolebinding.rbac.authorization.k8s.io/migration-operator created
deployment.apps/migration-operator created
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-builders" already exists 1
Error from server (AlreadyExists): error when creating "./operator.yml":
rolebindings.rbac.authorization.k8s.io "system:image-pullers" already exists 2
```

- 1 2 Error from server (AlreadyExists)** メッセージは無視することができます。これらは、Cluster Application Migration Operator が以降のリリースで提供されるリリースを OpenShift Container Platform 3 の以前のバージョン用に作成することによって生じます。

7. Migration コントローラー CR オブジェクトを作成します。

```
$ oc create -f controller-3.yml
```

8. Velero および Restic Pod が実行されていることを確認します。

```
$ oc get pods -n openshift-migration
```

1.5.2. OpenShift Container Platform 4.2 ターゲットクラスターへのクラスターアプリケーション移行 Operator のインストール

OLM を使用して OpenShift Container Platform 4.2 ターゲットクラスターに Cluster Application Migration Operator をインストールできます。

Cluster Application Migration Operator は、デフォルトで CAM ツールをターゲットクラスターにインストールします。

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 (この場合は **Migration**) を使用して **Cluster Application Migration Operator** を見つけます。
3. **Cluster Application Migration Operator** を選択し、**Install** をクリックします。
4. **Create Operator Subscription** ページで **openshift-migration** namespace を選択し、承認ストラテジーを指定します。
5. **Subscribe** をクリックします。
Installed Operators ページで、**Cluster Application Migration Operator** は、**InstallSucceeded** のステータスで **openshift-migration** プロジェクトに表示されます。
6. **Provided APIs** の下で **View 12 more...** をクリックします。
7. **Create New** → **MigrationController** をクリックします。
8. **Create** をクリックします。
9. **Workloads** → **Pods** をクリックし、Controller Manager、Migration UI、Restic、および Velero Pod が実行中であることを確認します。

1.5.3. OpenShift Container Platform 3 ソースクラスターでのクロスオリジンリソース共有の設定

ソースクラスターの API サーバーと CAM ツール間の通信を有効にするために、OpenShift Container Platform 3 ソースクラスターでクロスオリジンリソース共有を設定する必要があります。

手順

1. CAM ツールがインストールされているクラスターにログインします。
2. CORS 設定の値を取得します。

```
$ oc get -n openshift-migration route/migration -o go-template='({i})/{{ .spec.host }}(:|z){{ println }}' | sed 's,\.,\\. ,g'
```

3. OpenShift Container Platform 3 ソースクラスターにログインします。
4. CORS 設定値を、**/etc/origin/master/master-config.yaml** 設定ファイルの **corsAllowedOrigins** スタンザに追加します。

```
corsAllowedOrigins:
- ({i})//migration-openshift-migration\.apps\.cluster\.com(:|z) ❶
- ({i})//openshift\.default\.svc(:|z)
- ({i})//kubernetes\.default(:|z)
```

- ❶ CORS 設定を指定します。

5. API サーバーおよびコントローラーマネージャーを再起動して変更を適用します。
 - OpenShift Container Platform 3.7 および 3.9 では、これらのコンポーネントは **systemd** で管理されるスタンドアロンのホストプロセスとして実行され、以下のコマンドを実行して再起動します。

```
$ systemctl restart atomic-openshift-master-api atomic-openshift-master-controllers
```

- OpenShift Container Platform 3.10 および 3.11 では、これらのコンポーネントは kubelet によって管理される静的 Pod で実行され、以下のコマンドを実行して再起動します。

```
$ /usr/local/bin/master-restart api
$ /usr/local/bin/master-restart controllers
```

6. 設定を確認します。

```
$ curl -v -k -X OPTIONS \
"<cluster_url>/apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migclusters" \ 1
-H "Access-Control-Request-Method: GET" \
-H "Access-Control-Request-Headers: authorization" \
-H "Origin: https://<CAM_web_console_url>" 2
```

- 1** CORS の設定されたクラスターの URL を指定します。
- 2** CAM Web コンソールの URL を指定します。URL は CORS 設定値をベースとしています (例: <https://migration-openshift-migration.apps.cluster>)。

以下のような出力が表示されます。

```
< HTTP/2 204
< access-control-allow-credentials: true
< access-control-allow-headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-
Token, Authorization, X-Requested-With, If-Modified-Since
< access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, PATCH
< access-control-allow-origin: https://migration-openshift-migration.apps.cluster
< access-control-expose-headers: Date
< cache-control: no-store
```

1.6. CAM WEB コンソールを使用したアプリケーションの移行

1.6.1. CAM Web コンソールの起動

ブラウザで CAM Web コンソールを起動できます。

手順

1. CAM ツールがインストールされている OpenShift Container Platform クラスターにログインします。
2. 以下のコマンドを入力して CAM Web コンソール URL を取得します。

```
$ oc get -n openshift-migration route/migration -o go-template='https://{{ .spec.host }}'
https://migration-openshift-migration.apps.<cluster>.openshift.com
```

3. ブラウザーを起動し、CAM Web コンソールに移動します。

字を含めることはできません。

- **Url:** クラスターの API サーバーの URL です (例: **https://<master1.example.com>:8443**)。
- **Service account token:** ソースクラスターから取得される文字列。
- **Azure cluster:** オプション。 Azure スナップショットを使用してデータをコピーする場合はこれを選択します。
- **Azure resource group:** このフィールドは、 **Azure cluster** にチェックを付けると表示されます。

6. **Add cluster** をクリックします。
クラスターが **Clusters** セクションに表示されます。

1.6.3. CAM Web コンソールへのレプリケーションリポジトリの追加

CAM Web コンソールには、オブジェクトストレージバケットをレプリケーションリポジトリとして追加できます。

前提条件

- データを移行するには、オブジェクトストレージバケットを設定する必要があります。

手順

1. CAM Web コンソールにログインします。
2. **Replication repositories** セクションで、 **Add repository** をクリックします。
3. **Storage provider type** を選択し、以下のフィールドに入力します。
 - **AWS** (AWS S3、MCG、および汎用 S3 プロバイダーの場合):
 - **Replication repository name:** CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **S3 bucket name:** 作成した S3 バケットの名前を指定します。
 - **S3 bucket region:** S3 バケットリージョンを指定します。AWS S3 の場合に **必須** です。 **Optional** (他の S3 プロバイダーの場合)。
 - **S3 endpoint:** バケットではなく S3 サービスの URL を指定します (例: **https://<s3-storage.apps.cluster.com>**)。汎用 S3 プロバイダーの場合は **必須** です。 **https://** プレフィックスを使用する必要があります。
 - **S3 provider access key:** AWS には **<AWS_SECRET_ACCESS_KEY>** を指定するか、または MCG には S3 プロバイダーアクセスキーを指定します。
 - **S3 provider secret access key:** AWS には **<AWS_ACCESS_KEY_ID>** を指定するか、または MCG には S3 プロバイダーシークレットアクセスキーを指定します。
 - **Require SSL verification:** 汎用 S3 プロバイダーを使用している場合は、このチェックボックスをクリアします。
 - **GCP:**

- **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **GCP bucket name**: GCP バケットの名前を指定します。
 - **GCP credential JSON blob: credentials-velero** ファイルに文字列を指定します。
 - **Azure:**
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **Azure resource group**: Azure Blob ストレージのリソースグループを指定します。
 - **Azure storage account name**: Azure Blob ストレージアカウント名を指定します。
 - **Azure credentials - INI file contents credentials-velero** ファイルに文字列を指定します。
4. **Add repository** をクリックし、接続の検証を待機します。
 5. **Close** をクリックします。
新規リポジトリが **Replication repositories** セクションに表示されます。

1.6.4. 大規模な移行の場合の移行計画の制限の変更

大規模な移行の移行計画の制限を変更することができます。



重要

移行の失敗を防ぐために、まず変更についてのテストをご使用の環境で実行する必要があります。

単一の移行計画には以下のデフォルト制限があります。

- 10 namespace
この制限を超えると、CAM Web コンソールは **Namespace limit exceeded** エラーを表示し、移行計画を作成することができません。
- 100 Pod
Pod の制限を超える場合、CAM Web コンソールには、以下の例と同様の警告メッセージが表示されます。**Plan has been validated with warning condition(s). See warning message. Pod limit: 100 exceeded, found: 104**
- 100 永続ボリューム
永続ボリュームの制限を超過すると、CAM Web コンソールには同様の警告メッセージが表示されます。

手順

1. Migration コントローラー CR を編集します。

```
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h
```

```
$ oc edit migrationcontroller -n openshift-migration
```

- 以下のパラメーターを更新します。

```
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

1.6.5. CAM Web コンソールでの移行計画の作成

CAM Web コンソールで移行計画を作成できます。

前提条件

- CAM Web コンソールには以下が含まれている必要があります。
 - ソースクラスター
 - CAM ツールのインストール時に自動的に追加されるターゲットクラスター
 - レプリケーションリポジトリ
- スナップショットを使用してデータをコピーする必要がある場合、ソースおよびターゲットクラスターは、同じクラウドプロバイダー（AWS、GCP、または Azure）および同じリージョンで実行される必要があります。

手順

- CAM Web コンソールにログインします。
- Plans** セクションで、**Add plan** をクリックします。
- Plan name** を入力し、**Next** をクリックします。
Plan name には、最大 253 文字の小文字の英数字 (**a-z**、**0-9**) を含めることができます。スペースまたはアンダースコア (**_**) を含めることはできません。
- Source cluster** を選択します。
- Target cluster** を選択します。
- Replication repository** を選択します。
- 移行するプロジェクトを選択し、**Next** をクリックします。
- PV の **Copy** または **Move** を選択します。
 - Copy** は、ソースクラスターの PV のデータをレプリケーションリポジトリにコピーしてから、これを同様の特徴のある新規に作成された PV で復元します。

- **Move** は、ソースクラスターからリモートボリューム (例: NFS) をアンマウントし、リモートボリュームをポイントするターゲットクラスターで PV リソースを作成し、その後にリモートボリュームをターゲットクラスターにマウントします。ターゲットクラスターで実行されているアプリケーションは、ソースクラスターが使用していたものと同じリモートボリュームを使用します。リモートボリュームは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。

9. **Next** をクリックします。

10. PV の **Copy method** を選択します。

- **Snapshot** は、クラウドプロバイダーのスナップショット機能を使用してディスクのバックアップおよび復元を行います。この場合、**ファイルシステム**を使用する場合よりもはるかに高速になります。



注記

ストレージおよびクラスターは同じリージョンにあり、ストレージクラスには互換性がなければなりません。

- **ファイルシステム**は、データファイルをソースディスクから新規に作成されたターゲットディスクにコピーします。

11. PV の **Storage class** を選択します。

Filesystem のコピー方法を選択した場合、移行時にストレージクラスを変更できます。たとえば、Red Hat Gluster Storage または NFS ストレージから Red Hat Ceph Storage に変更できません。

12. **Finish** をクリックします。

13. **Close** をクリックします。

移行計画は **Plans** セクションに表示されます。

1.6.6. CAM Web コンソールでの移行計画の実行

CAM Web コンソールで作成した移行計画を使用してアプリケーションとデータをステージングしたり、移行したりできます。

前提条件

CAM Web コンソールには以下が含まれている必要があります。

- ソースクラスター
- CAM ツールのインストール時に自動的に追加されるターゲットクラスター
- レプリケーションリポジトリ
- 有効な移行計画

手順

1. OpenShift Container Platform 4 クラスターで CAM Web コンソールにログインします。
2. 移行計画を選択します。

3. **Stage** をクリックし、アプリケーションを停止せずにソースクラスターからターゲットクラスターにデータをコピーします。
実際の移行時間を短縮するには、**Stage** を複数回実行することができます。
4. アプリケーションのワークロードを移行する準備ができたなら、**Migrate** をクリックします。**Migrate** は、ソースクラスターでアプリケーションワークロードを停止し、ターゲットクラスターでそのリソースを再作成します。
5. オプション: **Migrate** ウィンドウで **Do not stop applications on the source cluster during migration** を選択できます。
6. **Migrate** をクリックします。
7. 移行が完了したら、アプリケーションが OpenShift Container Platform 4.2 Web コンソールで正常に移行されていることを確認します。
 - a. **Home** → **Projects** をクリックします。
 - b. 移行されたプロジェクトをクリックしてそのステータスを表示します。
 - c. **Routes** セクションで **Location** をクリックし、アプリケーションが機能していることを確認します (該当する場合)。
 - d. **Workloads** → **Pods** をクリックし、Pod が移行した namespace で実行されていることを確認します。
 - e. **Storage** → **Persistent volumes** をクリックして、移行した永続ボリュームが正常にプロビジョニングされていることを確認します。

1.7. CONTROL PLANE MIGRATION ASSISTANT (CPMA) でのコントロールプレーン設定の移行

1.7.1. Control Plane Migration Assistant について

Control Plane Migration Assistant (CPMA) は、OpenShift Container Platform 3.7 (以降) から OpenShift Container Platform 4.2 へのコントロールプレーンの移行に役立つ CLI ベースのツールです。CPMA は OpenShift Container Platform 3 設定ファイル进行处理し、OpenShift Container Platform 4.2 Operator によって使用されるカスタムリソース (CR) マニフェストファイルを生成します。

OpenShift Container Platform 3 および 4 には設定上の大きな違いがあるため、すべてのパラメーターが処理される訳ではありません。CPMA は、機能が完全にサポートされるか、部分的にサポートされるか、または全くサポートされないかについて記述するレポートを生成できます。

設定ファイル

CPMA は Kubernetes および OpenShift Container Platform API を使用して、OpenShift Container Platform 3 クラスターの以下の設定ファイルにアクセスします。

- マスター設定ファイル (デフォルト: `/etc/origin/master/master-config.yaml`)
- CRI-O 設定ファイル (デフォルト: `/etc/crio/crio.conf`)
- etcd 設定ファイル (デフォルト: `/etc/etcd/etcd.conf`)
- イメージレジストリーファイル (デフォルト: `/etc/containers/registries.conf`)

- 依存する設定ファイル:
 - パスワードファイル (HTPasswd など)
 - ConfigMap
 - シークレット

CR マニフェスト

CPMA は、以下の設定の CR マニフェストを生成します。

- API サーバー CA 証明書: **100_CPMA-cluster-config-APISecret.yaml**



注記

署名なしの API サーバー CA 証明書を使用している場合は、証明書をターゲットクラスターに手動で追加する必要があります。

- CRI-O: **100_CPMA-crio-config.yaml**
- クラスタリソースクォータ: **100_CPMA-cluster-quota-resource-x.yaml**
- プロジェクトリソースクォータ: **100_CPMA-resource-quota-x.yaml**
- 移植可能なイメージレジストリー (/etc/registries/registries.conf) および移植可能なイメージポリシー (etc/origin/master/master-config.yaml): **100_CPMA-cluster-config-image.yaml**
- OAuth プロバイダー: **100_CPMA-cluster-config-oauth.yaml**
- プロジェクト設定: **100_CPMA-cluster-config-project.yaml**
- スケジューラー: **100_CPMA-cluster-config-scheduler.yaml**
- SDN: **100_CPMA-cluster-config-sdn.yaml**

1.7.2. Control Plane Migration Assistant のインストール

Red Hat カスタマーポータルから Control Plane Migration Assistant (CPMA) バイナリーファイルをダウンロードし、Linux、MacOSX、Windows オペレーティングシステムにインストールできます。

手順

1. [Red Hat カスタマーポータル](#) の **Downloads** → **Red Hat OpenShift Container Platform** に移動します。
2. **Download Red Hat OpenShift Container Platform** ページで、**Product Variant** 一覧から **Red Hat OpenShift Container Platform** を選択します。
3. **Version** 一覧から **CPMA 1.0 for RHEL 7** を選択します。このバイナリーは、RHEL 7 および RHEL 8 で機能します。
4. **Download Now** をクリックして Linux または MacOSX の場合は **cpma** をダウンロードし、Windows の場合は **cpma.exe** をダウンロードします。

5. Linux または MacOSX の場合は **\$PATH** と定義されたディレクトリーに、Windows の場合は **%PATH%** と定義されたディレクトリーにファイルを保存します。
6. Linux の場合は、ファイルを実行可能にします。

```
$ sudo chmod +x cpma
```

1.7.3. Control Plane Migration Assistant の使用

Control Plane Migration Assistant (CPMA) は、OpenShift Container Platform 4.2 Operator によって使用される CR マニフェストを生成し、どの OpenShift Container Platform 3 機能が完全に、または部分的にサポートされるか、または全くサポートされないかを示すレポートを生成します。

CPMA はリモートモードで実行でき、SSH を使用するか、またはローカルモードで、ソースクラスターの設定ファイルのローカルコピーを使用して、ソースクラスターの設定ファイルを取得します。

前提条件

- ソースクラスターは OpenShift Container Platform 3.7 以降であること。
- ソースクラスターは、最新の同期リリースに対して更新される必要があります。
- 診断エラーや警告がないことを確認するために、環境ヘルスチェックをソースクラスターで実行する必要があります。
- CPMA バイナリーは実行可能である必要があります。
- ソースクラスターの **cluster-admin** 権限がなければなりません。

手順

1. OpenShift Container Platform 3 クラスターにログインします。

```
$ oc login https://<master1.example.com> 1
```

- 1** OpenShift Container Platform 3 マスターノード。Kubernetes および OpenShift Container Platform API のトークンを受信するには、クラスターにログインする必要があります。

2. CPMA を実行します。以下の例にあるように、各プロンプトで入力が必要になります。

```
$ cpma --manifests=false 1
? Do you wish to save configuration for future use? true
? What will be the source for OCP3 config files? Remote host 2
? Path to cri-o config file /etc/crio/crio.conf
? Path to etcd config file /etc/etcd/etcd.conf
? Path to master config file /etc/origin/master/master-config.yaml
? Path to node config file /etc/origin/node/node-config.yaml
? Path to registries config file /etc/containers/registries.conf
? Do wish to find source cluster using KUBECONFIG or prompt it? KUBECONFIG
? Select cluster obtained from KUBECONFIG contexts master1-example-com:443
? Select master node master1.example.com
? SSH login root 3
? SSH Port 22
```

```
? Path to private SSH key /home/user/.ssh/openshift_key
? Path to application data, skip to use current directory .
INFO[29 Aug 19 00:07 UTC] Starting manifest and report generation
INFO[29 Aug 19 00:07 UTC] Transform:Starting for - API
INFO[29 Aug 19 00:07 UTC] APITransform::Extract
INFO[29 Aug 19 00:07 UTC] APITransform::Transform:Reports
INFO[29 Aug 19 00:07 UTC] Transform:Starting for - Cluster
INFO[29 Aug 19 00:08 UTC] ClusterTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportQuotas
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportPVs
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportNamespaces
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportNodes
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportRBAC
INFO[29 Aug 19 00:08 UTC] ClusterReport::ReportStorageClasses
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Crio
INFO[29 Aug 19 00:08 UTC] CrioTransform::Extract
WARN[29 Aug 19 00:08 UTC] Skipping Crio: No configuration file available
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Docker
INFO[29 Aug 19 00:08 UTC] DockerTransform::Extract
INFO[29 Aug 19 00:08 UTC] DockerTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - ETCD
INFO[29 Aug 19 00:08 UTC] ETCDTransform::Extract
INFO[29 Aug 19 00:08 UTC] ETCDTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - OAuth
INFO[29 Aug 19 00:08 UTC] OAuthTransform::Extract
INFO[29 Aug 19 00:08 UTC] OAuthTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - SDN
INFO[29 Aug 19 00:08 UTC] SDNTransform::Extract
INFO[29 Aug 19 00:08 UTC] SDNTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Image
INFO[29 Aug 19 00:08 UTC] ImageTransform::Extract
INFO[29 Aug 19 00:08 UTC] ImageTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Transform:Starting for - Project
INFO[29 Aug 19 00:08 UTC] ProjectTransform::Extract
INFO[29 Aug 19 00:08 UTC] ProjectTransform::Transform:Reports
INFO[29 Aug 19 00:08 UTC] Flushing reports to disk
INFO[29 Aug 19 00:08 UTC] Report:Added: report.json
INFO[29 Aug 19 00:08 UTC] Report:Added: report.html
INFO[29 Aug 19 00:08 UTC] Successfully finished transformations
```

- 1 **--manifests=false**: CR マニフェストを生成しない
- 2 **Remote host**: リモートモード
- 3 **SSH login**: SSH ユーザーには、設定ファイルにアクセスできるように OpenShift Container Platform 3 クラスタで **sudo** パーミッションが必要です。

CPMA は、出力ディレクトリーを指定しなかった場合に、現在のディレクトリーに以下のファイルおよびディレクトリーを作成します。

- **cpma.yaml** ファイル: CPMA の実行時に指定した設定オプション
- **master1.example.com/**: マスターノードからの設定ファイル
- **report.json**: JSON でエンコードされたレポート

- **report.html**: HTML でエンコードされたレポート
3. ブラウザーで **report.html** ファイルを開き、CPMA レポートを表示します。
 4. CR マニフェストを生成する場合は、以下の例のように CR マニフェストを OpenShift Container Platform 4.2 クラスターに適用します。

```
$ oc apply -f 100_CPMA-cluster-config-secret-htpasswd-secret.yaml
```

1.8. トラブルシューティング

移行用のカスタムリソース (CR) を表示し、ログをダウンロードして失敗した移行の失敗についてのトラブルシューティングを行うことができます。

移行の失敗時にアプリケーションが停止した場合は、データ破損を防ぐために手作業でアプリケーションをロールバックする必要があります。

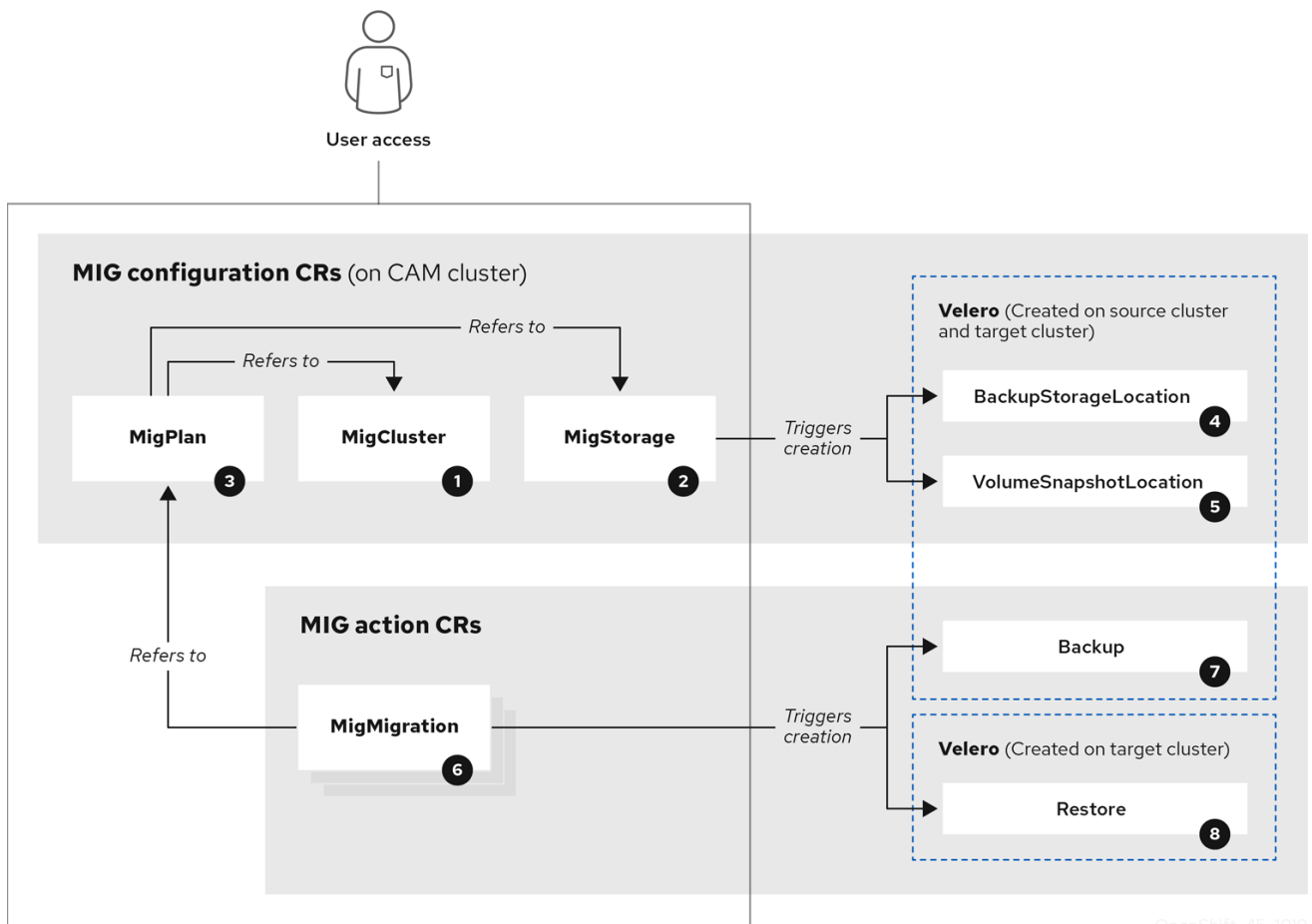


注記

移行時にアプリケーションが停止しなかった場合には、手動のロールバックは必要ありません。元のアプリケーションがソースクラスター上で依然として実行されているためです。

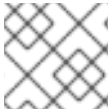
1.8.1. 移行カスタムリソース (CR) の表示

Cluster Application Migration (CAM) ツールは移行用に以下の CR を作成します。



- 1 **MigCluster** (設定、CAM クラスタ): クラスタ定義
- 2 **MigStorage** (設定、CAM クラスタ): ストレージ定義
- 3 **MigPlan** (設定、CAM クラスタ): 移行計画

MigPlan CR は移行されるソースおよびターゲットクラスタ、リポジトリ、および namespace を記述します。これは 0、1 または多数の MigMigration CR に関連付けられます。



注記

MigPlan CR を削除すると、関連付けられた MigMigration CR が削除されます。

- 4 **BackupStorageLocation** (設定、CAM クラスタ): Velero バックアップオブジェクトの場所
- 5 **VolumeSnapshotLocation** (設定、CAM クラスタ): ボリュームスナップショットの場所
- 6 **MigMigration** (アクション、CAM クラスタ): 移行、移行時に作成される

MigMigration CR は、データのステージングまたは移行を実行するたびに作成されます。各 MigMigration CR は MigPlan CR に関連付けられます。

- 7 **Backup** (アクション、ソースクラスタ): 移行計画の実行時に、MigMigration CR は各ソースクラスタに 2 つの Velero バックアップ CR を作成します。

- Kubernetes オブジェクトのバックアップ CR #1
- PV データのバックアップ CR #2

- 8 **Restore** (アクション、ターゲットクラスタ): 移行計画の実行時に、MigMigration CR はターゲットクラスタに 2 つのリストア CR を作成します。

- PV データのリストア CR #1 (バックアップ CR #2 の使用)
- Kubernetes オブジェクトのリストア CR #2 (バックアップ CR #1 の使用)

手順

1. CR 名を取得します。

```
$ oc get <cr> -n openshift-migration 1
NAME                                AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

- 1 表示する移行 CR を指定します。

2. CR を表示します。

```
$ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

出力は以下の例のようになります。

MigMigration の例

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
Labels:    <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:      MigMigration
Metadata:
  Creation Timestamp: 2019-08-29T01:01:29Z
  Generation:        20
  Resource Version:  88179
  Self Link:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  UID:               8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
  Mig Plan Ref:
    Name:      socks-shop-mig-plan
    Namespace: openshift-migration
  Quiesce Pods: true
  Stage:       false
Status:
  Conditions:
    Category:      Advisory
    Durable:       true
    Last Transition Time: 2019-08-29T01:03:40Z
    Message:       The migration has completed successfully.
    Reason:        Completed
    Status:        True
    Type:          Succeeded
  Phase:         Completed
  Start Timestamp: 2019-08-29T01:01:29Z
  Events:        <none>
```

Velero バックアップ CR #2 の例 (PV データ)

```
apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-
0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
```



```

  velero.io/storage-location: myrepo-vpzq9
name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
namespace: openshift-migration
resourceVersion: "87313"
selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims
  - namespaces
  - imagestreams
  - imagestreamtags
  - secrets
  - configmaps
  - pods
  labelSelector:
    matchLabels:
      migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  storageLocation: myrepo-vpzq9
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

```

Velero リストア CR #2 の例 (Kubernetes リソース)

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-

```

```


generation: 3
labels:
  app.kubernetes.io/part-of: migration
  migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
  migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
namespace: openshift-migration
resourceVersion: "82329"
selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

1.8.2. 移行ログのダウンロード

CAM Web コンソールで Velero、Restic、および Migration コントローラーログをダウンロードして、移行の失敗についてのトラブルシューティングを行うことができます。

手順

1. CAM Web コンソールにログインします。
2. **Plans** をクリックし、移行計画の一覧を表示します。
3. **Options** メニュー  をクリックし (特定の移行計画について)、**Logs** を選択します。
4. **Download Logs** をクリックし、すべてのクラスターの Migration コントローラー、Velero、および Restic のログをダウンロードします。
5. 特定のログをダウンロードするには、以下を実行します。
 - a. ログオプションを指定します。
 - **Cluster:** ソース、ターゲット、または CAM ホストクラスターを選択します。

- **Log source:** Velero、Restic、または Controller を選択します。
- **Pod source:** Pod 名を選択します (例: **controller-manager-78c469849c-v6wcf**)。選択したログが表示されます。

選択した内容を変更することで、ログ選択の設定をクリアできます。

- Download Selected** をクリックし、選択したログをダウンロードします。

オプションで、以下の例にあるように CLI を使用してログにアクセスできます。

```
$ oc get pods -n openshift-migration | grep controller
controller-manager-78c469849c-v6wcf      1/1   Running   0      4h49m

$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration
```

1.8.3. Restic タイムアウトエラー

Restic のタイムアウトにより移行が失敗する場合、以下のエラーが Velero ログに表示されます。

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

restic_timeout のデフォルト値は1時間です。大規模な移行では、この値を大きくすることができます。値を高くすると、エラーメッセージが返されるタイミングが送れる可能性があることに注意してください。

手順

1. OpenShift Container Platform Web コンソールで、**Operators** → **Installed Operators** に移動します。
2. **Cluster Application Migration Operator** をクリックします。
3. **MigrationController** タブで、**migration-controller** をクリックします。
4. **YAML** タブで、以下のパラメーター値を更新します。

```
spec:
  restic_timeout: 1h 1
```

- 1** 有効な単位は **h** (時間)、**m** (分)、および **s** (秒) です (例: **3h30m15s**)。

5. **Save** をクリックします。

1.8.4. 移行の手動ロールバック

移行の失敗時にアプリケーションが停止された場合は、PV でのデータの破損を防ぐために手動でこれをロールバックする必要があります。

移行時にアプリケーションが停止しなかった場合には、この手順は必要ありません。元のアプリケーションがソースクラスター上で依然として実行されているためです。

手順

1. ターゲットクラスター上で、移行したプロジェクトに切り替えます。

```
$ oc project <project>
```

2. デプロイされたリソースを取得します。

```
$ oc get all
```

3. デプロイされたリソースを削除し、アプリケーションがターゲットクラスターで実行されておらず、PVC 上にあるデータにアクセスできるようにします。

```
$ oc delete <resource_type>
```

4. これを削除せずに DaemonSet を停止するには、YAML ファイルで **nodeSelector** を更新します。

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: hello-daemonset
spec:
  selector:
    matchLabels:
      name: hello-daemonset
  template:
    metadata:
      labels:
        name: hello-daemonset
    spec:
      nodeSelector:
        role: worker ①
```

- ① いずれのノードにも存在しない **nodeSelector** 値を指定します。

5. 不要なデータが削除されるように、各 PV の回収ポリシーを更新します。移行時に、バインドされた PV の回収ポリシーは **Retain** であり、アプリケーションがソースクラスターから削除される際にデータの損失が生じないようにされます。ロールバック時にこれらの PV を削除できます。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
```

```

persistentVolumeReclaimPolicy: Retain ❶
...
status:
...

```

❶ **Recycle** または **Delete** を指定します。

6. ソースクラスターで、移行したプロジェクトに切り替え、そのデプロイされたリソースを取得します。

```

$ oc project <project>
$ oc get all

```

7. デプロイされた各リソースのレプリカを開始します。

```

$ oc scale --replicas=1 <resource_type>/<resource_name>

```

8. 手順の実行時に変更した場合は、DaemonSet の **nodeSelector** を元の値に更新します。

1.8.5. カスタマーサポートケース用のデータの収集

カスタマーサポートケースを作成する場合、**openshift-migration-must-gather-rhel8** イメージを使用して **must-gather** ツールを実行し、クラスターについての情報を収集し、これを [Red Hat カスタマーポータル](#) にアップロードできます。

openshift-migration-must-gather-rhel8 イメージは、デフォルトの **must-gather** イメージで収集されないログおよびカスタムリソースデータを収集します。

手順

1. **must-gather** データを保存するディレクトリーに移動します。
2. **oc adm must-gather** コマンドを実行します。

```

$ oc adm must-gather --image=registry.redhat.io/rhcam-1-2/openshift-migration-must-gather-rhel8

```

must-gather ツールはクラスター情報を収集し、これを **must-gather.local.<uid>** ディレクトリーに保存します。

3. 認証キーおよびその他の機密情報を **must-gather** データから削除します。
4. **must-gather.local.<uid>** ディレクトリーの内容を含むアーカイブファイルを作成します。

```

$ tar cvaf must-gather.tar.gz must-gather.local.<uid>/

```

圧縮ファイルを [Red Hat カスタマーポータル](#) 上のサポートケースに添付します。

1.8.6. 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。

- 移行中に、CAM ツールは以下の namespace アノテーションを保持します。

- openshift.io/sa.scc.mcs
- openshift.io/sa.scc.supplemental-groups
- openshift.io/sa.scc.uid-range
これらのアノテーションは UID 範囲を保持し、コンテナがターゲットクラスターのファイルシステムのパーミッションを保持できるようにします。移行された UID が、ターゲットクラスターの既存の namespace または今後の namespace 内の UID を重複させるリスクがあります。(BZ#1748440)
- S3 エンドポイントを CAM Web コンソールに追加する場合、**https://** は AWS でのみサポートされます。他の S3 プロバイダーの場合は **http://**を使用します。
- AWS バケットが CAM Web コンソールに追加された後に削除される場合、MigStorage CR は更新されないため、そのステータスは **True** のままになります。(BZ#1738564)
- Migration コントローラーが、ターゲットクラスター以外のクラスターで実行されている場合は移行に失敗します。**EnsureCloudSecretPropagated** フェーズはログに記録された警告を出して省略されます。(BZ#1757571)
- クラスターのロールバインディングおよび SCC (Security Context Constraints) を含むクラスタースコープのリソースは CAM によって処理されません。アプリケーションがクラスタースコープのリソースを必要とする場合、ターゲットクラスターでそれらを手動で作成する必要があります。(BZ#1759804)
- 移行計画の作成時に、ソースクラスターのストレージクラスが誤って表示されます。(BZ#1777869)
- CAM Web コンソールのクラスターにアクセス不可の場合、これはオープン状態の移行計画のクローズを試行します。(BZ#1758269)
- 移行に失敗すると、移行計画は休止状態の Pod のカスタム PV 設定を保持しません。移行を手動でロールバックし、移行計画を削除し、PV 設定で新たな移行計画を作成する必要があります。(BZ#1784899)

第2章 OPENSIFT CONTAINER PLATFORM 4.1 から 4.2 への移行

2.1. アプリケーションワークロードの OPENSIFT CONTAINER PLATFORM 4.1 から 4.2 への移行

アプリケーションワークロードを、Cluster Application Migration (CAM) ツールを使用して OpenShift Container Platform 4.1 から OpenShift Container Platform 4.2 に移行できます。CAM ツールを使用すると、移行を制御し、アプリケーションのダウンタイムを最小限に抑えることができます。

Kubernetes カスタムリソースをベースとする CAM ツールの Web コンソールおよび API により、namespace の粒度でステートフルおよびステートレスのアプリケーションワークロードを移行できます。

2.1.1. 移行の前提条件

- すべてのクラスターで **cluster-admin** 権限があること。
- ソースおよびターゲットクラスターには、レプリケーションリポジトリへの無制限のネットワークアクセスがなければなりません。
- Migration コントローラーがインストールされているクラスターには、他のクラスターへの無制限のアクセスが必要です。
- アプリケーションが **openshift** namespace のイメージを使用する場合、イメージの必要なバージョンがターゲットクラスターにある必要がある。
必要なイメージがない場合は、アプリケーションと互換性のある利用可能なバージョンを使用するように **imagestreamtags** 参照を更新する必要があります。**imagestreamtag** を更新できない場合、同等のイメージをアプリケーション namespace に手動でアップロードし、それらを参照するようにアプリケーションを更新できます。

以下の **imagestreamtag** は OpenShift Container Platform 4.2 から **削除** されています。

- **dotnet:1.0**、**dotnet:1.1**、**dotnet:2.0**
- **dotnet-runtime:2.0**
- **mariadb:10.1**
- **mongodb:2.4**、**mongodb:2.6**
- **mysql:5.5**、**mysql:5.6**
- **nginx:1.8**
- **nodejs:0.10**、**nodejs:4**、**nodejs:6**
- **perl:5.16**、**perl:5.20**
- **php:5.5**、**php:5.6**
- **postgresql:9.2**、**postgresql:9.4**、**postgresql:9.5**
- **python:3.3**、**python:3.4**

- **ruby:2.0、ruby:2.2**

2.1.2. Cluster Application Migration ツールについて

Cluster Application Migration (CAM) ツールを使用すると、CAM Web コンソールまたは Kubernetes API を使用して Kubernetes リソース、永続ボリュームデータ、および内部コンテナイメージを OpenShift Container Platform ソースクラスターから OpenShift Container Platform 4.2 ターゲットクラスターに移行できます。

CAM Web コンソールを使用してアプリケーションを移行するには、以下の手順が必要です。

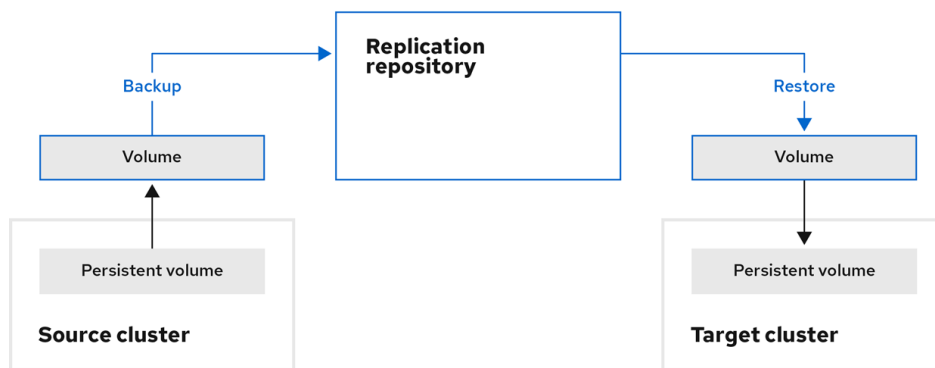
1. Cluster Application Migration Operator をすべてのクラスターにインストールします。



注記

Cluster Application Migration Operator はデフォルトで CAM ツール（CAM Web コンソールおよび Migration コントローラー）をターゲットクラスターにインストールします。CAM ツールは OpenShift Container Platform 3 およびリモートクラスターにインストールできます。

2. CAM ツールがデータ移行に使用する中間オブジェクトストレージであるレプリケーションリポジトリを設定します。
3. ソースクラスターを CAM Web コンソールに追加します。
4. レプリケーションリポジトリを CAM Web コンソールに追加します。
5. 以下のデータ移行オプションのいずれかを使用して、移行計画を作成します。
 - **Copy:** CAM ツールは、データをソースクラスターからレプリケーションリポジトリにコピーし、レプリケーションリポジトリからターゲットクラスターにコピーします。



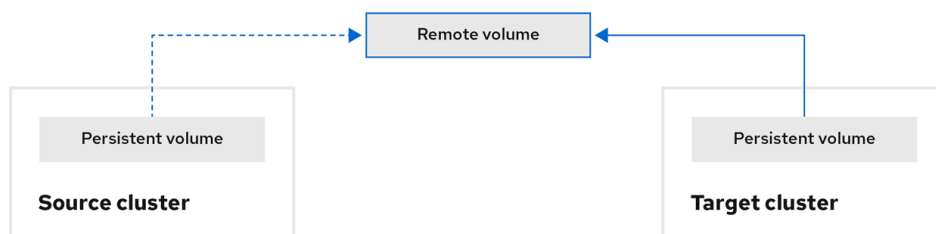
OpenShift_45_1019

- **Move:** CAM ツールはリモートボリューム (NFS など) をソースクラスターからアンマウントし、リモートボリュームをポイントするターゲットクラスターで PV リソースを作成し、その後にリモートボリュームをターゲットクラスターにマウントします。ターゲットクラスターで実行されているアプリケーションは、ソースクラスターが使用していたものと同じリモートボリュームを使用します。リモートボリュームは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。



注記

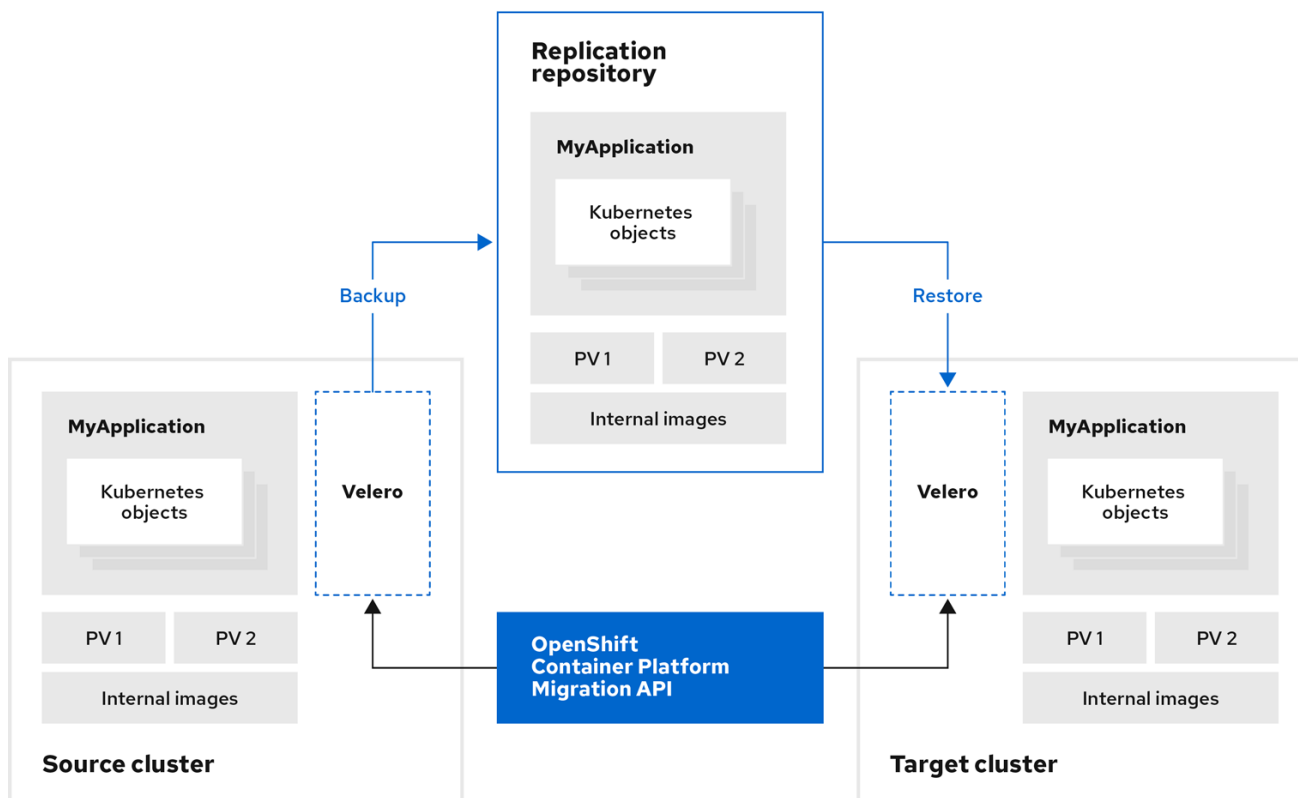
レプリケーションリポジトリはこの図には表示されていませんが、実際の移行には必要になります。



OpenShift_45_1019

6. 以下のオプションのいずれかを使用して、移行計画を実行します。

- Stage** (オプション) は、アプリケーションを停止せずにデータをターゲットクラスターにコピーします。
 ステージングは、移行前にほとんどのデータがターゲットにコピーされるように複数回実行することができます。これにより、実際の移行時間やアプリケーションのダウンタイムが最小限に抑えられます。
- Migrate** は、ソースクラスターでアプリケーションを停止し、ターゲットクラスターでそのリソースを再作成します。オプションで、アプリケーションを停止せずにワークロードを移行できます。



OpenShift_45_1019

2.2. レプリケーションリポジトリの設定

オブジェクトストレージをレプリケーションリポジトリとして使用するよう設定する必要があります。Cluster Application Migration ツールは、[ファイルシステム](#)または[スナップショット](#)のいずれかのデータの複製方法を使用して、データをソースクラスターからレプリケーションリポジトリにコピーしてから、レプリケーションリポジトリからターゲットクラスターにコピーします。

以下のストレージプロバイダーがサポートされています。

- 汎用 S3 オブジェクトストレージ (例: Minio または Ceph S3)
- Multi-Cloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

2.2.1. 移行用のデータのコピー方法について

CAM ツールは、ソースクラスターからターゲットクラスターにデータを移行するためにファイルシステムおよびスナップショットによるデータの複製方法をサポートします。ご使用の環境に適した方法で、ストレージプロバイダーでサポートされる方法を選択できます。

2.2.1.1. ファイルシステムの複製方法

CAM ツールは、データファイルをソースクラスターからレプリケーションリポジトリに複製し、そこからターゲットクラスターに複製します。

表2.1 ファイルシステムの複製方法の概要

利点	制限
<ul style="list-style-type: none"> ● クラスターで複数の異なるストレージクラスを使用することが可能 ● すべての S3 ストレージプロバイダーでサポートされている 	<ul style="list-style-type: none"> ● スナップショットの複製方法よりも遅い

2.2.1.2. スナップショットの複製方法

CAM ツールは、ソースクラスターのデータの snapshots を、レプリケーションリポジトリとして設定されたクラウドプロバイダーのオブジェクトストレージに複製します。データはターゲットクラスターで復元されます。

AWS、Google Cloud Provider、および Microsoft Azure は、snapshots の複製方法をサポートします。

表2.2 snapshots の複製方法の概要

利点	制限
----	----

利点	制限
<ul style="list-style-type: none"> ● ファイルシステムのコピー方法よりも高速 	<ul style="list-style-type: none"> ● クラウドプロバイダーはスナップショットをサポートしている必要があります。 ● クラスタは同じクラウドプロバイダーになければなりません。 ● クラスタは、同じ場所またはリージョンにある必要があります。 ● クラスタには同じストレージクラスがなければなりません。 ● ストレージクラスにはスナップショットとの互換性がある必要があります。

重要

Multi-Cloud Object Gateway を移行用のレプリケーションリポジトリとして設定する機能はテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。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。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

Red Hat のテクノロジープレビュー機能のサポート範囲についての詳細は、<https://access.redhat.com/ja/support/offerings/techpreview/> を参照してください。

2.2.2. Multi-Cloud Object Gateway ストレージバケットをレプリケーションリポジトリとして設定する

OpenShift Container Storage Operator をインストールし、Multi-Cloud Object Gateway (MCG) ストレージバケットをレプリケーションリポジトリとして設定できます。

2.2.2.1. OpenShift Container Storage Operator のインストール

OpenShift Container Storage Operator は、OperatorHub からインストールできます。

手順

1. OpenShift Container Platform Web コンソールで、**Administration** → **Namespaces** をクリックします。
2. **Create Namespace** をクリックします。
3. **Name** フィールドに **openshift-storage** を入力し、**Create** をクリックします。
4. **Operators** → **OperatorHub** をクリックします。
5. **Filter by keyword** (この場合は、**OCS**) を使用し、**OpenShift Container Storage Operator** を見つけます。

6. **OpenShift Container Storage Operator** を選択し、**Install** をクリックします。
7. **Create Operator Subscription** ページで、**openshift-storage** namespace を選択します。
8. 更新チャンネルおよび承認ストラテジーを指定します。
9. **Subscribe** をクリックします。
Installed Operators ページで、**OpenShift Container Storage Operator** は、**Succeeded** のステータスと共に **openshift-storage** プロジェクトに表示されます。

2.2.2.2. Multi-Cloud Object Gateway ストレージバケットの作成

Multi-Cloud Object Gateway (MCG) ストレージバケットのカスタムリソース (CR) を作成できます。

手順

1. OpenShift Container Platform クラスタにログインします。

```
$ oc login
```

2. **NooBaa** CR 設定ファイル **noobaa.yml** を以下の内容で作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5 1
      memory: 1Gi
  coreResources:
    requests:
      cpu: 0.5 2
      memory: 1Gi
```

1 **2** 非常に小規模なクラスタの場合、**cpu** の値を **0.1** に変更できます。

3. **NooBaa** オブジェクトを作成します。

```
$ oc create -f noobaa.yml
```

4. 以下の内容で、**BackingStore** CR 設定ファイル **bs.yml** を作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: BackingStore
metadata:
  finalizers:
    - noobaa.io/finalizer
  labels:
    app: noobaa
  name: mcg-pv-pool-bs
```

```

namespace: openshift-storage
spec:
  pvPool:
    numVolumes: 3 ❶
    resources:
      requests:
        storage: 50Gi ❷
    storageClass: gp2 ❸
  type: pv-pool

```

- ❶ PV プール内のボリューム数を指定します。
- ❷ ボリュームのサイズを指定します。
- ❸ ストレージクラスを作成します。

5. **BackingStore** オブジェクトを作成します。

```
$ oc create -f bs.yml
```

6. 以下の内容で **BucketClass** CR 設定ファイル **bc.yml** を作成します。

```

apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
  labels:
    app: noobaa
    name: mcg-pv-pool-bc
  namespace: openshift-storage
spec:
  placementPolicy:
    tiers:
      - backingStores:
          - mcg-pv-pool-bs
    placement: Spread

```

7. **BucketClass** オブジェクトを作成します。

```
$ oc create -f bc.yml
```

8. 以下の内容で **ObjectBucketClaim** CR 設定ファイル **obc.yml** を作成します。

```

apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage
  namespace: openshift-storage
spec:
  bucketName: migstorage ❶
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc

```

- 1 レプリケーションリポジトリを CAM Web コンソールに追加するために使用するバケット名を記録します。

9. **ObjectBucketClaim** オブジェクトを作成します。

```
$ oc create -f obc.yml
```

10. リソース作成プロセスを監視し、**ObjectBucketClaim** ステータスが **Bound** であることを確認します。

```
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

このプロセスには、5分から10分の時間がかかる場合があります。

11. 以下の値を取得して記録します。この値は、レプリケーションリポジトリを CAM Web コンソールに追加する際に必要になります。

- S3 エンドポイント:

```
$ oc get route -n openshift-storage s3
```

- S3 プロバイダーアクセスキー:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{.data.AWS_ACCESS_KEY_ID }}' | base64 -d
```

- S3 プロバイダーシークレットアクセスキー:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{.data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
```

2.2.3. AWS S3 ストレージバケットをレプリケーションリポジトリとして設定する

AWS S3 ストレージバケットをレプリケーションリポジトリとして設定できます。

前提条件

- AWS S3 ストレージバケットは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
- [AWS CLI](#) がインストールされていること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - EC2 Elastic Block Storage (EBS) にアクセスできる必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. AWS S3 バケットを作成します。

```
$ aws s3api create-bucket \
  --bucket <bucket_name> \ ❶
  --region <bucket_region> ❷
```

- ❶ S3 バケット名を指定します。
- ❷ S3 バケットリージョンを指定します (例: **us-east-1**)。

2. IAM ユーザー **velero** を作成します。

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

3. EC2 EBS スナップショットポリシーを作成します。

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

4. 1つまたはすべての S3 バケットの AWS S3 アクセスポリシーを作成します。

```
$ cat > velero-s3-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3>DeleteObject",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket_name>/*" ❶
      ]
    }
  ]
}
```

```

    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket_name>" ❷
      ]
    }
  ]
}
EOF

```

- ❶ ❷ 単一の S3 バケットへのアクセスを付与するには、バケット名を指定します。すべての AWS S3 バケットへのアクセスを付与するには、バケット名の代わりに * を指定します。

```

"Resource": [
  "arn:aws:s3::*"
]

```

5. EC2 EBS ポリシーを **velero** に割り当てます。

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-ebs \
  --policy-document file://velero-ec2-snapshot-policy.json

```

6. AWS S3 ポリシーを **velero** に割り当てます。

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-s3 \
  --policy-document file://velero-s3-policy.json

```

7. **velero** のアクセスキーを作成します。

```

$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, ❶
    "AccessKeyId": <AWS_ACCESS_KEY_ID> ❷
  }
}

```

- ❶ ❷ AWS リポジトリを CAM Web コンソールに追加するために **AWS_SECRET_ACCESS_KEY** および **AWS_ACCESS_KEY_ID** を記録します。

2.2.4. Google Cloud Provider ストレージバケットをレプリケーションリポジトリとして設定する

Google Cloud Provider (GCP) ストレージバケットをレプリケーションリポジトリとして設定できます。

前提条件

- GCP ストレージバケットは、ソースクラスターおよびターゲットクラスターからアクセスする必要があります。
- **gsutil** がインストールされていること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. **gsutil init** を実行してログインします。

```
$ gsutil init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

To continue, you must login. Would you like to login (Y/n)?
```

2. **BUCKET** 変数を設定します。

```
$ BUCKET=<bucket_name> ❶
```

- ❶ 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. **velero** サービスアカウントを作成します。

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

6. **SERVICE_ACCOUNT_EMAIL** 変数をサービスアカウントのメールアドレスに設定します。

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

7. パーミッションをサービスアカウントに付与します。

```
$ 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
)

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

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

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

8. 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
```

2.2.5. Microsoft Azure Blob ストレージコンテナをレプリケーションリポジトリとして設定

Microsoft Azure Blob ストレージコンテナをレプリケーションリポジトリとして設定できます。

前提条件

- [Azure ストレージアカウント](#)があること。
- [Azure CLI](#) がインストールされていること。
- Azure Blob ストレージコンテナがソースクラスターおよびターゲットクラスターからアクセスできること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。

- ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. **AZURE_RESOURCE_GROUP** 変数を設定します。

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

2. Azure リソースグループを作成します。

```
$ az group create -n $AZURE_RESOURCE_GROUP --location <CentralUS> ❶
```

- ❶ 場所を指定します。

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

```
$ AZURE_STORAGE_ACCOUNT_ID=velerobackups
```

4. 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
```

5. **BLOB_CONTAINER** 変数を設定します。

```
$ BLOB_CONTAINER=velero
```

6. Azure Blob ストレージコンテナを作成します。

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

7. **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`
```

8. 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
```

2.3. CLUSTER APPLICATION MIGRATION (CAM) ツールのデプロイ

Cluster Application Migration (CAM) ツールをデプロイするには、Cluster Application Migration Operator を [OpenShift Container Platform 4.1 ソース](#) と [OpenShift Container Platform 4.2 ターゲット](#) クラスターにインストールし、[OpenShift Container Platform 4.1 ソース](#) クラスターでクロスオリジンリソース共有を設定する必要があります。

Cluster Application Migration Operator はデフォルトで CAM ツール (CAM Web コンソールおよび Migration コントローラー) をターゲットクラスターにインストールします。[CAM ツールは OpenShift Container Platform 3 およびリモートクラスターにインストールできます。](#)

2.3.1. OpenShift Container Platform 4.1 ソースクラスターへの CAM Operator のインストール

OLM を使用して OpenShift Container Platform 4.1 ソースクラスターに Cluster Application Migration Operator をインストールできます。

手順

1. OpenShift Container Platform Web コンソールで、**Catalog** → **OperatorHub** をクリックします。
2. **Filter by keyword** フィールド (この場合は **Migration**) を使用して **Cluster Application Migration Operator** を見つけます。
3. **Cluster Application Migration Operator** を選択し、**Install** をクリックします。
4. **Create Operator Subscription** ページで **openshift-migration** namespace を選択し、承認ストラテジーを指定します。
5. **Subscribe** をクリックします。
Installed Operators ページで、**Cluster Application Migration Operator** は、**InstallSucceeded** のステータスで **openshift-migration** プロジェクトに表示されます。
6. **Provided APIs** の下で **View 12 more...** をクリックします。
7. **Create New** → **MigrationController** をクリックします。
8. **migration_controller** および **migration_ui** パラメーターを更新し、**deprecated_cors_configuration** パラメーターを **spec** スタンザに追加します。

```
spec:
  [...]
  migration_controller: false
  migration_ui: false
  [...]
  deprecated_cors_configuration: true
```

9. **Create** をクリックします。
10. **Workloads** → **Pods** をクリックし、Restic および Velero Pod が実行されていることを確認します。

2.3.2. OpenShift Container Platform 4.2 ターゲットクラスターへのクラスターアプリケーション移行 Operator のインストール

OLM を使用して OpenShift Container Platform 4.2 ターゲットクラスターに Cluster Application Migration Operator をインストールできます。

Cluster Application Migration Operator は、デフォルトで CAM ツールをターゲットクラスターにインストールします。

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 (この場合は **Migration**) を使用して **Cluster Application Migration Operator** を見つけます。
3. **Cluster Application Migration Operator** を選択し、**Install** をクリックします。
4. **Create Operator Subscription** ページで **openshift-migration** namespace を選択し、承認ストラテジーを指定します。
5. **Subscribe** をクリックします。
Installed Operators ページで、**Cluster Application Migration Operator** は、InstallSucceeded のステータスで **openshift-migration** プロジェクトに表示されます。
6. **Provided APIs** の下で **View 12 more...** をクリックします。
7. **Create New** → **MigrationController** をクリックします。
8. **Create** をクリックします。
9. **Workloads** → **Pods** をクリックし、Controller Manager、Migration UI、Restic、および Velero Pod が実行中であることを確認します。

2.3.3. OpenShift Container Platform 4.1 ソースクラスターでのクロスオリジンリソース共有の設定

ソースクラスターの API サーバーと CAM ツール間の通信を有効にするために、OpenShift Container Platform 4.1 ソースクラスターでクロスオリジンリソース共有を設定する必要があります。

手順

1. CAM ツールがインストールされているクラスターにログインします。
2. CORS 設定の値を取得します。

```
$ oc get -n openshift-migration route/migration -o go-template='{(?i)://{ .spec.host }}(:|z){{println }}' | sed 's,\. ,\.,g'
```

3. ソースクラスターにログインします。
4. OAuth サーバー CR を編集します。

```
$ oc edit authentication.operator cluster
```

5. CORS 設定値を、**spec** スタンザの **unsupportedConfigOverrides** の下にある **corsAllowedOrigins** に追加します。

```
spec:
  unsupportedConfigOverrides:
    corsAllowedOrigins:
      - (?i)//migration-openshift-migration\.apps\.cluster\.com(:|z) 1
```

- 1 CORS 設定値を指定します。

6. 変更を適用するためにファイルを保存します。
7. 以下のように Kubernetes API サーバー CR を編集します。

```
$ oc edit kubeapiserver.operator cluster
```

8. CORS 設定値を、**spec** スタンザの **unsupportedConfigOverrides** の下にある **corsAllowedOrigins** に追加します。

```
spec:
  unsupportedConfigOverrides:
    corsAllowedOrigins:
      - (?i)//migration-openshift-migration\.apps\.cluster\.com(:|z) 1
```

- 1 CORS 設定値を指定します。

9. 変更を適用するためにファイルを保存します。
10. 設定を確認します。

```
$ curl -v -k -X OPTIONS \
"<cluster_url>/apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migclusters" 1
-H "Access-Control-Request-Method: GET" \
-H "Access-Control-Request-Headers: authorization" \
-H "Origin: https://<CAM_web_console_url>" 2
```

- 1 CORS の設定されたクラスターの URL を指定します。

- 2 CAM Web コンソールの URL を指定します。URL は CORS 設定値をベースとしています (例: **https://migration-openshift-migration.apps.cluster**)。

以下のような出力が表示されます。

```
< HTTP/2 204
```

```
< access-control-allow-credentials: true
< access-control-allow-headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization, X-Requested-With, If-Modified-Since
< access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, PATCH
< access-control-allow-origin: https://migration-openshift-migration.apps.cluster
< access-control-expose-headers: Date
< cache-control: no-store
```

2.4. CAM WEB コンソールを使用したアプリケーションの移行

2.4.1. CAM Web コンソールの起動

ブラウザで CAM Web コンソールを起動できます。

手順

1. CAM ツールがインストールされている OpenShift Container Platform クラスターにログインします。
2. 以下のコマンドを入力して CAM Web コンソール URL を取得します。

```
$ oc get -n openshift-migration route/migration -o go-template='https://{{ .spec.host }}'
https://migration-openshift-migration.apps.<cluster>.openshift.com
```

3. ブラウザーを起動し、CAM Web コンソールに移動します。



注記

Cluster Application Migration Operator のインストール後すぐに CAM Web コンソールにアクセスしようとする場合、Operator は依然としてクラスターを設定し、クロスオリジンリソース共有を有効にしているため、コンソールが読み込まれないことがあります。数分待機した後に再試行します。

4. 自己署名 CA 証明書を使用している場合、ソースクラスターの API サーバーの CA 証明書を受け入れることを求めるプロンプトが出されます。Web ページは、残りの証明書を受け入れるプロセスについて説明します。
5. OpenShift Container Platform の **ユーザー名** および **パスワード** を使用してログインします。

2.4.2. CAM Web コンソールへのクラスターの追加

ソースクラスターを CAM Web コンソールに追加できます。

前提条件

- クロスオリジンリソース共有がソースクラスターで設定されている必要があります。
- Azure スナップショットを使用してデータをコピーする場合:
 - ソースクラスターの追加時に Azure リソースグループ名を指定する必要があります。
 - ソースおよびターゲットクラスターは同じ Azure リソースグループにあり、同じ場所にある必要があります。

手順

1. ソースクラスターにログインします。
2. サービスアカウントトークンを取得します。

```
$ oc sa get-token mig -n openshift-migration
eyJhbGciOiJIUzI1NiIsInR5cGU6IjY2NvdW50IiwiaWF0IjoiYjY2NvdW50Lm1pZyIsImt1YmV0ZXMuaW8vc2VydmljZWZjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6ImE1YjFiYWMMwLWMMxYmYtMTFIOS05Y2NiLTAYOWRmODYwYjMwOCIsInN1Yil6InN5c3RlbnR1bTJ2aWVudDptaWc6bWlnIn0.xqeeAINK7UXpdRqAtOj70qhBJPeMwmgLomV9iFxr5RoqUgKchZRG2J2rkqmPm6vr7K-cm7ibD1IBpdQJCcVDuoHYsFgV4mp9vgOfn9osSDp2TGikwNz4Az95e81xnjVUmzh-NjDsEpw71DH92iHV_xt2sTwtzftS49LpPW2LjrV0evtNBP_t_RfskdArt5VSv25eORI7zScqfe1CiMkcVbf2UqACQjo3LbkpfN26HAioO2oH0ECPiRzT0Xyh-KwFutJLS9Xgghyw-LD9kPKcE_xbbJ9Y4Rqajh7WdPYuB0Jd9DPVrslmzK-F6cgHHYoZEv0SvLQi-PO0rpDrcjOEEQQ
```

3. CAM Web コンソールにログインします。
4. **Clusters** セクションで、**Add cluster** をクリックします。
5. 以下のフィールドに値を入力します。
 - **Cluster name:** 小文字 (**a-z**) および数字 (**0-9**) を含めることができます。空白文字や国際文字を含めることはできません。
 - **Url:** クラスターの API サーバーの URL です (例: **https://<master1.example.com>:8443**)。
 - **Service account token:** ソースクラスターから取得される文字列。
 - **Azure cluster:** オプション。Azure スナップショットを使用してデータをコピーする場合はこれを選択します。
 - **Azure resource group:** このフィールドは、**Azure cluster** にチェックを付けると表示されます。
6. **Add cluster** をクリックします。
クラスターが **Clusters** セクションに表示されます。

2.4.3. CAM Web コンソールへのレプリケーションリポジトリの追加

CAM Web コンソールには、オブジェクトストレージバケットをレプリケーションリポジトリとして追加できます。

前提条件

- データを移行するには、オブジェクトストレージバケットを設定する必要があります。

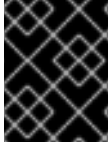
手順

1. CAM Web コンソールにログインします。

2. **Replication repositories** セクションで、**Add repository** をクリックします。
3. **Storage provider type** を選択し、以下のフィールドに入力します。
 - **AWS** (AWS S3、MCG、および汎用 S3 プロバイダーの場合):
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **S3 bucket name**: 作成した S3 バケットの名前を指定します。
 - **S3 bucket region**: S3 バケットリージョンを指定します。AWS S3 の場合に**必須**です。Optional (他の S3 プロバイダーの場合)。
 - **S3 endpoint**: バケットではなく S3 サービスの URL を指定します (例: **https://<s3-storage.apps.cluster.com>**)。汎用 S3 プロバイダーの場合は**必須**です。**https://** プレフィックスを使用する必要があります。
 - **S3 provider access key**: AWS には **<AWS_SECRET_ACCESS_KEY>** を指定するか、または MCG には S3 プロバイダーアクセスキーを指定します。
 - **S3 provider secret access key**: AWS には **<AWS_ACCESS_KEY_ID>** を指定するか、または MCG には S3 プロバイダーシークレットアクセスキーを指定します。
 - **Require SSL verification**: 汎用 S3 プロバイダーを使用している場合は、このチェックボックスをクリアします。
 - **GCP**:
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **GCP bucket name**: GCP バケットの名前を指定します。
 - **GCP credential JSON blob**: **credentials-velero** ファイルに文字列を指定します。
 - **Azure**:
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **Azure resource group**: Azure Blob ストレージのリソースグループを指定します。
 - **Azure storage account name**: Azure Blob ストレージアカウント名を指定します。
 - **Azure credentials - INI file contents**: **credentials-velero** ファイルに文字列を指定します。
4. **Add repository** をクリックし、接続の検証を待機します。
5. **Close** をクリックします。
新規リポジトリが **Replication repositories** セクションに表示されます。

2.4.4. 大規模な移行の場合の移行計画の制限の変更

大規模な移行の移行計画の制限を変更することができます。



重要

移行の失敗を防ぐために、まず変更についてのテストをご使用の環境で実行する必要があります。

単一の移行計画には以下のデフォルト制限があります。

- 10 namespace
この制限を超えると、CAM Web コンソールは **Namespace limit exceeded** エラーを表示し、移行計画を作成することができません。
- 100 Pod
Pod の制限を超える場合、CAM Web コンソールには、以下の例と同様の警告メッセージが表示されます。 **Plan has been validated with warning condition(s). See warning message. Pod limit: 100 exceeded, found: 104**
- 100 永続ボリューム
永続ボリュームの制限を超過すると、CAM Web コンソールには同様の警告メッセージが表示されます。

手順

1. Migration コントローラー CR を編集します。

```
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h

$ oc edit migrationcontroller -n openshift-migration
```

2. 以下のパラメーターを更新します。

```
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

2.4.5. CAM Web コンソールでの移行計画の作成


CAM Web コンソールで移行計画を作成できます。

前提条件

- CAM Web コンソールには以下が含まれている必要があります。
 - ソースクラスター
 - CAM ツールのインストール時に自動的に追加されるターゲットクラスター

- レプリケーションリポジトリ
- スナップショットを使用してデータをコピーする必要がある場合、ソースおよびターゲットクラスターは、同じクラウドプロバイダー（AWS、GCP、または Azure）および同じリージョンで実行される必要があります。

手順

1. CAM Web コンソールにログインします。
 2. **Plans** セクションで、**Add plan** をクリックします。
 3. **Plan name** を入力し、**Next** をクリックします。
Plan name には、最大 253 文字の小文字の英数字 (**a-z**、**0-9**) を含めることができます。スペースまたはアンダースコア (**_**) を含めることはできません。
 4. **Source cluster** を選択します。
 5. **Target cluster** を選択します。
 6. **Replication repository** を選択します。
 7. 移行するプロジェクトを選択し、**Next** をクリックします。
 8. PV の **Copy** または **Move** を選択します。
 - **Copy** は、ソースクラスターの PV のデータをレプリケーションリポジトリにコピーしてから、これを同様の特徴のある新規に作成された PV で復元します。
 - **Move** は、ソースクラスターからリモートボリューム (例: NFS) をアンマウントし、リモートボリュームをポイントするターゲットクラスターで PV リソースを作成し、その後にリモートボリュームをターゲットクラスターにマウントします。ターゲットクラスターで実行されているアプリケーションは、ソースクラスターが使用していたものと同じリモートボリュームを使用します。リモートボリュームは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
 9. **Next** をクリックします。
 10. PV の **Copy method** を選択します。
 - **Snapshot** は、クラウドプロバイダーのスナップショット機能を使用してディスクのバックアップおよび復元を行います。この場合、**ファイルシステム**を使用する場合よりもはるかに高速になります。
- 

注記

ストレージおよびクラスターは同じリージョンにあり、ストレージクラスには互換性がなければなりません。
- **ファイルシステム**は、データファイルをソースディスクから新規に作成されたターゲットディスクにコピーします。
11. PV の **Storage class** を選択します。
Filesystem のコピー方法を選択した場合、移行時にストレージクラスを変更できます。たとえば、Red Hat Gluster Storage または NFS ストレージから Red Hat Ceph Storage に変更できません。

12. **Finish** をクリックします。
13. **Close** をクリックします。
移行計画は **Plans** セクションに表示されます。

2.4.6. CAM Web コンソールでの移行計画の実行

CAM Web コンソールで作成した移行計画を使用してアプリケーションとデータをステージングしたり、移行したりできます。

前提条件

CAM Web コンソールには以下が含まれている必要があります。

- ソースクラスター
- CAM ツールのインストール時に自動的に追加されるターゲットクラスター
- レプリケーションリポジトリ
- 有効な移行計画

手順

1. OpenShift Container Platform 4 クラスターで CAM Web コンソールにログインします。
2. 移行計画を選択します。
3. **Stage** をクリックし、アプリケーションを停止せずにソースクラスターからターゲットクラスターにデータをコピーします。
実際の移行時間を短縮するには、**Stage** を複数回実行することができます。
4. アプリケーションのワークロードを移行する準備ができたなら、**Migrate** をクリックします。
Migrate は、ソースクラスターでアプリケーションワークロードを停止し、ターゲットクラスターでそのリソースを再作成します。
5. オプション: **Migrate** ウィンドウで **Do not stop applications on the source cluster during migration** を選択できます。
6. **Migrate** をクリックします。
7. 移行が完了したら、アプリケーションが OpenShift Container Platform 4.2 Web コンソールで正常に移行されていることを確認します。
 - a. **Home** → **Projects** をクリックします。
 - b. 移行されたプロジェクトをクリックしてそのステータスを表示します。
 - c. **Routes** セクションで **Location** をクリックし、アプリケーションが機能していることを確認します (該当する場合)。
 - d. **Workloads** → **Pods** をクリックし、Pod が移行した namespace で実行されていることを確認します。
 - e. **Storage** → **Persistent volumes** をクリックして、移行した永続ボリュームが正常にプロビジョニングされていることを確認します。

2.5. トラブルシューティング

移行用のカスタムリソース (CR) を表示し、ログをダウンロードして失敗した移行の失敗についてのトラブルシューティングを行うことができます。

移行の失敗時にアプリケーションが停止した場合は、データ破損を防ぐために手作業でアプリケーションをロールバックする必要があります。

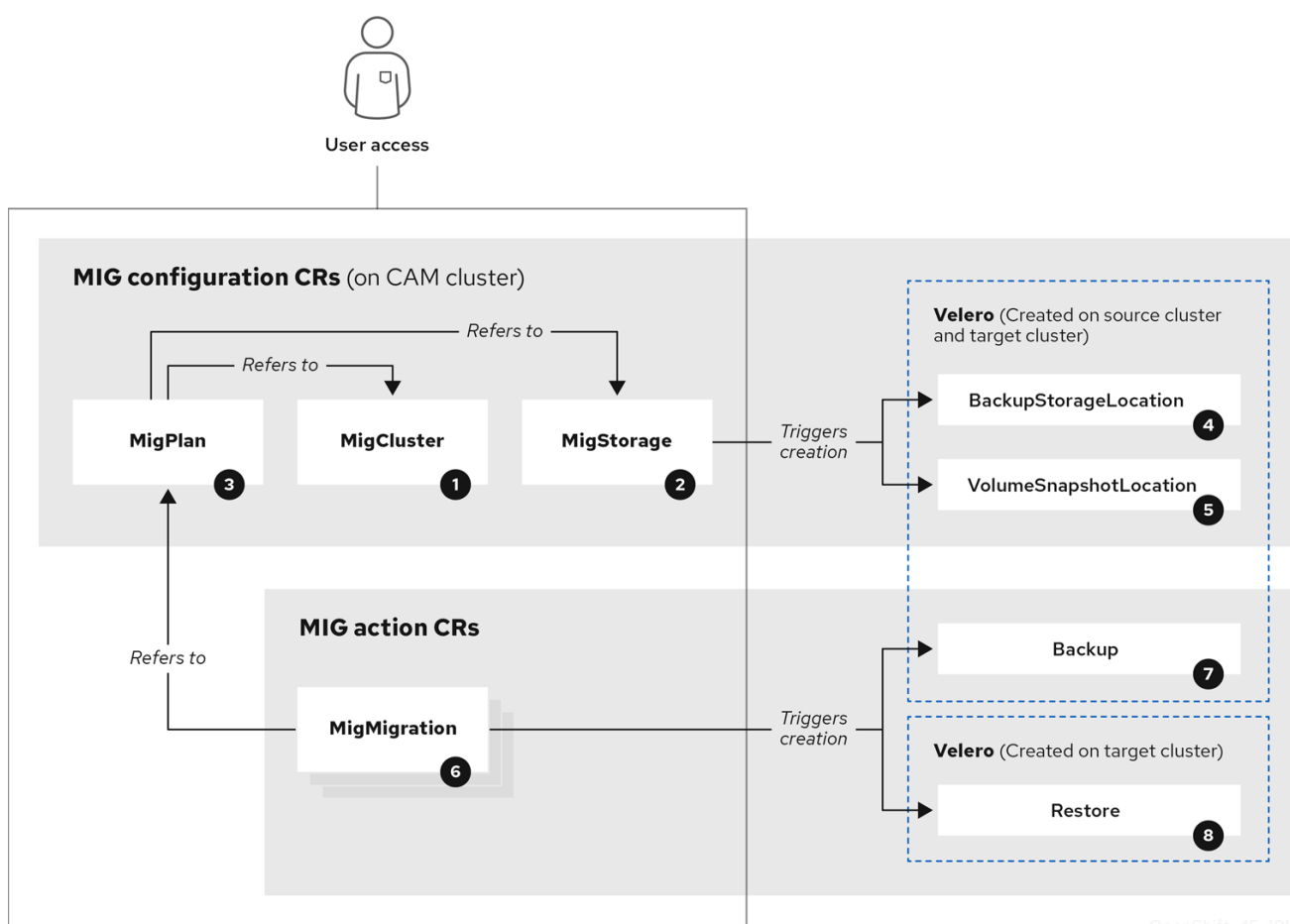


注記

移行時にアプリケーションが停止しなかった場合には、手動のロールバックは必要ありません。元のアプリケーションがソースクラスター上で依然として実行されているためです。

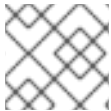
2.5.1. 移行カスタムリソース (CR) の表示

Cluster Application Migration (CAM) ツールは移行用に以下の CR を作成します。



- 1 **MigCluster** (設定、CAM クラスター): クラスター定義
- 2 **MigStorage** (設定、CAM クラスター): ストレージ定義
- 3 **MigPlan** (設定、CAM クラスター): 移行計画

MigPlan CR は移行されるソースおよびターゲットクラスター、リポジトリ、および namespace を記述します。これは 0、1 または多数の MigMigration CR に関連付けられます。



注記

MigPlan CR を削除すると、関連付けられた MigMigration CR が削除されます。

4 [BackupStorageLocation](#) (設定、CAM クラスター): Velero バックアップオブジェクトの場所

5 [VolumeSnapshotLocation](#) (設定、CAM クラスター): ボリュームスナップショットの場所

6 [MigMigration](#) (アクション、CAM クラスター): 移行、移行時に作成される

MigMigration CR は、データのステージングまたは移行を実行するたびに作成されます。各 MigMigration CR は MigPlan CR に関連付けられます。

7 [Backup](#) (アクション、ソースクラスター): 移行計画の実行時に、MigMigration CR は各ソースクラスターに 2 つの Velero バックアップ CR を作成します。

- Kubernetes オブジェクトのバックアップ CR #1
- PV データのバックアップ CR #2

8 [Restore](#) (アクション、ターゲットクラスター): 移行計画の実行時に、MigMigration CR はターゲットクラスターに 2 つのリストア CR を作成します。

- PV データのリストア CR #1 (バックアップ CR #2 の使用)
- Kubernetes オブジェクトのリストア CR #2 (バックアップ CR #1 の使用)

手順

1. CR 名を取得します。

```
$ oc get <cr> -n openshift-migration 1
NAME                                AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

- 1** 表示する移行 CR を指定します。

2. CR を表示します。

```
$ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

出力は以下の例のようになります。

MigMigration の例

```
$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
```

```

Labels:      <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:        MigMigration
Metadata:
  Creation Timestamp: 2019-08-29T01:01:29Z
  Generation:        20
  Resource Version:  88179
  Self Link:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  UID:               8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
  Mig Plan Ref:
    Name:      socks-shop-mig-plan
    Namespace: openshift-migration
  Quiesce Pods: true
  Stage:      false
Status:
  Conditions:
    Category:      Advisory
    Durable:       true
    Last Transition Time: 2019-08-29T01:03:40Z
    Message:       The migration has completed successfully.
    Reason:        Completed
    Status:        True
    Type:          Succeeded
  Phase:         Completed
  Start Timestamp: 2019-08-29T01:01:29Z
Events:         <none>

```

Velero バックアップ CR #2 の例 (PV データ)

```

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-
0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"
  selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-
5d593ce65e10-59gb7
  uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:

```

```

excludedNamespaces: []
excludedResources: []
hooks:
  resources: []
includeClusterResources: null
includedNamespaces:
- sock-shop
includedResources:
- persistentvolumes
- persistentvolumeclaims
- namespaces
- imagestreams
- imagestreamtags
- secrets
- configmaps
- pods
labelSelector:
  matchLabels:
    migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
storageLocation: myrepo-vpzq9
ttl: 720h0m0s
volumeSnapshotLocations:
- myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

```

Velero リストア CR #2 の例 (Kubernetes リソース)

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88
    migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
  name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
  namespace: openshift-migration
  resourceVersion: "82329"

```



```


selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-
d129df7f3b96-gb8nx
uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

2.5.2. 移行ログのダウンロード

CAM Web コンソールで Velero、Restic、および Migration コントローラーログをダウンロードして、移行の失敗についてのトラブルシューティングを行うことができます。

手順

1. CAM Web コンソールにログインします。
2. **Plans** をクリックし、移行計画の一覧を表示します。
3. **Options** メニュー  をクリックし (特定の移行計画について)、**Logs** を選択します。
4. **Download Logs** をクリックし、すべてのクラスターの Migration コントローラー、Velero、および Restic のログをダウンロードします。
5. 特定のログをダウンロードするには、以下を実行します。
 - a. ログオプションを指定します。
 - **Cluster:** ソース、ターゲット、または CAM ホストクラスターを選択します。
 - **Log source:** Velero、Restic、または Controller を選択します。
 - **Pod source:** Pod 名を選択します (例: **controller-manager-78c469849c-v6wcf**)。選択したログが表示されます。

選択した内容を変更することで、ログ選択の設定をクリアできます。

- a. **Download Selected** をクリックし、選択したログをダウンロードします。

オプションで、以下の例にあるように CLI を使用してログにアクセスできます。

```
$ oc get pods -n openshift-migration | grep controller
controller-manager-78c469849c-v6wcf      1/1   Running   0      4h49m

$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration
```

2.5.3. Restic タイムアウトエラー

Restic のタイムアウトにより移行が失敗する場合、以下のエラーが Velero ログに表示されます。

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all
PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

`restic_timeout` のデフォルト値は 1 時間です。大規模な移行では、この値を大きくすることができます。値を高くすると、エラーメッセージが返されるタイミングが送れる可能性があることに注意してください。

手順

1. OpenShift Container Platform Web コンソールで、**Operators** → **Installed Operators** に移動します。
2. **Cluster Application Migration Operator** をクリックします。
3. **MigrationController** タブで、**migration-controller** をクリックします。
4. **YAML** タブで、以下のパラメーター値を更新します。

```
spec:
  restic_timeout: 1h 1
```

1 有効な単位は **h** (時間)、**m** (分)、および **s** (秒) です (例: **3h30m15s**)。

5. **Save** をクリックします。

2.5.4. 移行の手動ロールバック

移行の失敗時にアプリケーションが停止された場合は、PV でのデータの破損を防ぐために手動でこれをロールバックする必要があります。

移行時にアプリケーションが停止しなかった場合には、この手順は必要ありません。元のアプリケーションがソースクラスター上で依然として実行されているためです。

手順

1. ターゲットクラスター上で、移行したプロジェクトに切り替えます。

```
$ oc project <project>
```

2. デプロイされたリソースを取得します。

```
$ oc get all
```

3. デプロイされたリソースを削除し、アプリケーションがターゲットクラスターで実行されておらず、PVC 上にあるデータにアクセスできるようにします。

```
$ oc delete <resource_type>
```

4. これを削除せずに DaemonSet を停止するには、YAML ファイルで **nodeSelector** を更新します。

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: hello-daemonset
spec:
  selector:
    matchLabels:
      name: hello-daemonset
  template:
    metadata:
      labels:
        name: hello-daemonset
    spec:
      nodeSelector:
        role: worker ①
```

- ① いずれのノードにも存在しない **nodeSelector** 値を指定します。

5. 不要なデータが削除されるように、各 PV の回収ポリシーを更新します。移行時に、バインドされた PV の回収ポリシーは **Retain** であり、アプリケーションがソースクラスターから削除される際にデータの損失が生じないようにされます。ロールバック時にこれらの PV を削除できます。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain ①
  ...
status:
  ...
```

- ① **Recycle** または **Delete** を指定します。

6. ソースクラスターで、移行したプロジェクトに切り替え、そのデプロイされたリソースを取得します。

```
$ oc project <project>
$ oc get all
```

7. デプロイされた各リソースのレプリカを開始します。

```
$ oc scale --replicas=1 <resource_type>/<resource_name>
```

8. 手順の実行時に変更した場合は、DaemonSet の **nodeSelector** を元の値に更新します。

2.5.5. カスタマーサポートケース用のデータの収集

カスタマーサポートケースを作成する場合、**openshift-migration-must-gather-rhel8** イメージを使用して **must-gather** ツールを実行し、クラスターについての情報を収集し、これを [Red Hat カスタマーポータル](#) にアップロードできます。

openshift-migration-must-gather-rhel8 イメージは、デフォルトの **must-gather** イメージで収集されないログおよびカスタムリソースデータを収集します。

手順

1. **must-gather** データを保存するディレクトリーに移動します。
2. **oc adm must-gather** コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/rhcam-1-2/openshift-migration-must-gather-rhel8
```

must-gather ツールはクラスター情報を収集し、これを **must-gather.local.<uid>** ディレクトリーに保存します。

3. 認証キーおよびその他の機密情報を **must-gather** データから削除します。
4. **must-gather.local.<uid>** ディレクトリーの内容を含むアーカイブファイルを作成します。

```
$ tar cvaf must-gather.tar.gz must-gather.local.<uid>/
```

圧縮ファイルを [Red Hat カスタマーポータル](#) 上のサポートケースに添付します。

2.5.6. 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。

- 移行中に、CAM ツールは以下の namespace アノテーションを保持します。
 - **openshift.io/sa.scc.mcs**
 - **openshift.io/sa.scc.supplemental-groups**

- **openshift.io/sa.scc.uid-range**

これらのアノテーションは UID 範囲を保持し、コンテナーがターゲットクラスターのファイルシステムのパーミッションを保持できるようにします。移行された UID が、ターゲットクラスターの既存の namespace または今後の namespace 内の UID を重複させるリスクがあります。(BZ#1748440)

- S3 エンドポイントを CAM Web コンソールに追加する場合、**https://** は AWS でのみサポートされます。他の S3 プロバイダーの場合は **http://** を使用します。
- AWS バケットが CAM Web コンソールに追加された後に削除される場合、MigStorage CR は更新されないため、そのステータスは **True** のままになります。(BZ#1738564)
- Migration コントローラーが、ターゲットクラスター以外のクラスターで実行されている場合は移行に失敗します。**EnsureCloudSecretPropagated** フェーズはログに記録された警告を出して省略されます。(BZ#1757571)
- クラスターのロールバインディングおよび SCC (Security Context Constraints) を含むクラスタースコープのリソースは CAM によって処理されません。アプリケーションがクラスタースコープのリソースを必要とする場合、ターゲットクラスターでそれらを手動で作成する必要があります。(BZ#1759804)
- 移行計画の作成時に、ソースクラスターのストレージクラスが誤って表示されます。(BZ#1777869)
- CAM Web コンソールのクラスターにアクセス不可の場合、これはオープン状態の移行計画のクローズを試行します。(BZ#1758269)
- 移行に失敗すると、移行計画は休止状態の Pod のカスタム PV 設定を保持しません。移行を手動でロールバックし、移行計画を削除し、PV 設定で新たな移行計画を作成する必要があります。(BZ#1784899)

第3章 OPENSIFT CONTAINER PLATFORM 4.2 クラスタ間の移行

3.1. OPENSIFT CONTAINER PLATFORM 4.2 クラスタ間のアプリケーションワークロードの移行

アプリケーションワークロードを、Cluster Application Migration (CAM) ツールを使用して OpenShift Container Platform 4.2 クラスタ間で移行できます。CAM ツールを使用すると、移行を制御し、アプリケーションのダウンタイムを最小限に抑えることができます。

Kubernetes カスタムリソースをベースとする CAM ツールの Web コンソールおよび API により、namespace の粒度でステートフルおよびステートレスのアプリケーションワークロードを移行できます。

3.1.1. 移行の前提条件

- すべてのクラスタで **cluster-admin** 権限があること。
- ソースおよびターゲットクラスタには、レプリケーションリポジトリへの無制限のネットワークアクセスがなければなりません。
- Migration コントローラーがインストールされているクラスタには、他のクラスタへの無制限のアクセスが必要です。

3.1.2. Cluster Application Migration ツールについて

Cluster Application Migration (CAM) ツールを使用すると、CAM Web コンソールまたは Kubernetes API を使用して Kubernetes リソース、永続ボリュームデータ、および内部コンテナイメージを OpenShift Container Platform ソースクラスタから OpenShift Container Platform 4.2 ターゲットクラスタに移行できます。

CAM Web コンソールを使用してアプリケーションを移行するには、以下の手順が必要です。

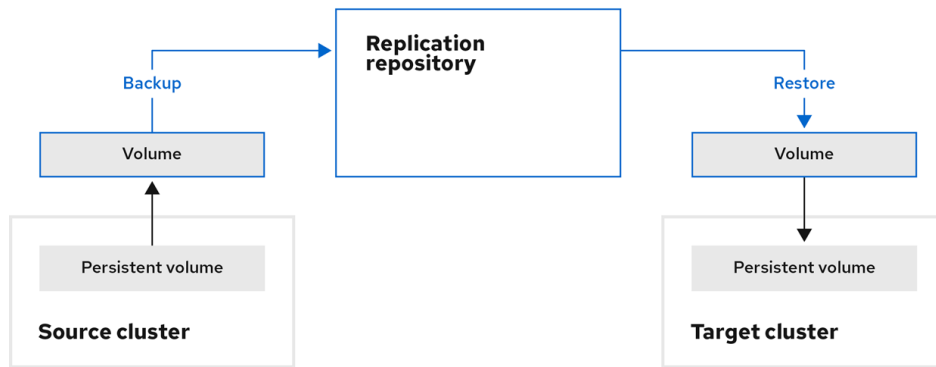
1. Cluster Application Migration Operator をすべてのクラスタにインストールします。



注記

Cluster Application Migration Operator はデフォルトで CAM ツール (CAM Web コンソールおよび Migration コントローラー) をターゲットクラスタにインストールします。[CAM ツールは OpenShift Container Platform 3 およびリモートクラスタにインストールできます。](#)

2. CAM ツールがデータ移行に使用する中間オブジェクトストレージであるレプリケーションリポジトリを設定します。
3. ソースクラスタを CAM Web コンソールに追加します。
4. レプリケーションリポジトリを CAM Web コンソールに追加します。
5. 以下のデータ移行オプションのいずれかを使用して、移行計画を作成します。
 - **Copy:** CAM ツールは、データをソースクラスタからレプリケーションリポジトリにコピーし、レプリケーションリポジトリからターゲットクラスタにコピーします。



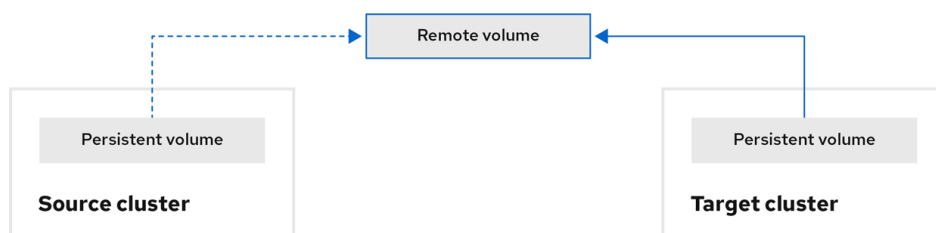
OpenShift_45_1019

- **Move:** CAM ツールはリモートボリューム (NFS など) をソースクラスターからアンマウントし、リモートボリュームをポイントするターゲットクラスターで PV リソースを作成し、その後にリモートボリュームをターゲットクラスターにマウントします。ターゲットクラスターで実行されているアプリケーションは、ソースクラスターが使用していたものと同じリモートボリュームを使用します。リモートボリュームは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。



注記

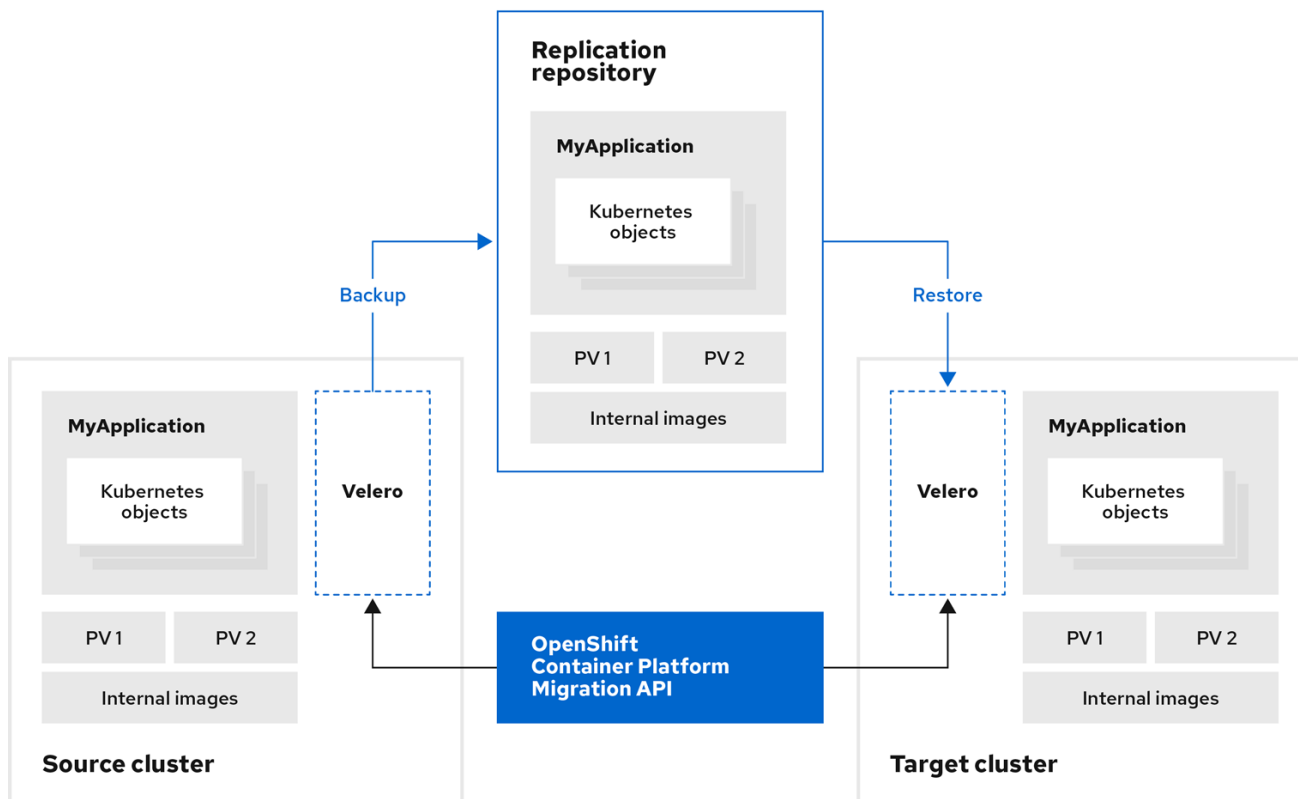
レプリケーションリポジトリはこの図には表示されていませんが、実際の移行には必要になります。



OpenShift_45_1019

6. 以下のオプションのいずれかを使用して、移行計画を実行します。

- **Stage** (オプション) は、アプリケーションを停止せずにデータをターゲットクラスターにコピーします。
ステージングは、移行前にほとんどのデータがターゲットにコピーされるように複数回実行することができます。これにより、実際の移行時間やアプリケーションのダウンタイムが最小限に抑えられます。
- **Migrate** は、ソースクラスターでアプリケーションを停止し、ターゲットクラスターでそのリソースを再作成します。オプションで、アプリケーションを停止せずにワークロードを移行できます。



OpenShift_45_1019

3.2. レプリケーションリポジトリの設定

オブジェクトストレージをレプリケーションリポジトリとして使用するよう設定する必要があります。Cluster Application Migration ツールは、[ファイルシステム](#)または[スナップショット](#)のいずれかのデータの複製方法を使用して、データをソースクラスターからレプリケーションリポジトリに複製してから、レプリケーションリポジトリからターゲットクラスターに複製します。

以下のストレージプロバイダーがサポートされています。

- 汎用 S3 オブジェクトストレージ (例: Minio または Ceph S3)
- Multi-Cloud Object Gateway (MCG)
- Amazon Web Services (AWS) S3
- Google Cloud Provider (GCP)
- Microsoft Azure

3.2.1. 移行用のデータの複製方法について

CAM ツールは、ソースクラスターからターゲットクラスターにデータを移行するためにファイルシステムおよびスナップショットによるデータの複製方法をサポートします。ご使用の環境に適した方法で、ストレージプロバイダーでサポートされる方法を選択できます。

3.2.1.1. ファイルシステムの複製方法

CAM ツールは、データファイルをソースクラスターからレプリケーションリポジトリに複製し、そこからターゲットクラスターに複製します。

表3.1 ファイルシステムのコピー方法の概要

利点	制限
<ul style="list-style-type: none"> ● クラスターで複数の異なるストレージクラスを使用することが可能 ● すべての S3 ストレージプロバイダーでサポートされている 	<ul style="list-style-type: none"> ● スナップショットのコピー方法よりも遅い

3.2.1.2. スナップショットのコピー方法

CAM ツールは、ソースクラスターのデータのスナップショットを、レプリケーションリポジトリとして設定されたクラウドプロバイダーのオブジェクトストレージにコピーします。データはターゲットクラスターで復元されます。

AWS、Google Cloud Provider、および Microsoft Azure は、スナップショットのコピー方法をサポートします。

表3.2 スナップショットのコピー方法の概要

利点	制限
<ul style="list-style-type: none"> ● ファイルシステムのコピー方法よりも高速 	<ul style="list-style-type: none"> ● クラウドプロバイダーはスナップショットをサポートしている必要があります。 ● クラスターは同じクラウドプロバイダーになければなりません。 ● クラスターは、同じ場所またはリージョンにある必要があります。 ● クラスターには同じストレージクラスがなければなりません。 ● ストレージクラスにはスナップショットとの互換性がある必要があります。

重要

Multi-Cloud Object Gateway を移行用のレプリケーションリポジトリとして設定する機能はテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は Red Hat の実稼働環境でのサービスレベルアグリーメント (SLA) ではサポートされていないため、Red Hat では実稼働環境での使用を推奨していません。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。これらの機能は、近々発表予定の製品機能をリリースに先駆けてご提供することにより、お客様は機能性をテストし、開発プロセス中にフィードバックをお寄せいただくことができます。

Red Hat のテクノロジープレビュー機能のサポート範囲についての詳細は、<https://access.redhat.com/ja/support/offerings/techpreview/> を参照してください。

3.2.2. Multi-Cloud Object Gateway ストレージバケットをレプリケーションリポジトリとして設定する

OpenShift Container Storage Operator をインストールし、Multi-Cloud Object Gateway (MCG) ストレージバケットをレプリケーションリポジトリとして設定できます。

3.2.2.1. OpenShift Container Storage Operator のインストール

OpenShift Container Storage Operator は、OperatorHub からインストールできます。

手順

1. OpenShift Container Platform Web コンソールで、**Administration** → **Namespaces** をクリックします。
2. **Create Namespace** をクリックします。
3. **Name** フィールドに **openshift-storage** を入力し、**Create** をクリックします。
4. **Operators** → **OperatorHub** をクリックします。
5. **Filter by keyword** (この場合は、OCS) を使用し、**OpenShift Container Storage Operator** を見つけます。
6. **OpenShift Container Storage Operator** を選択し、**Install** をクリックします。
7. **Create Operator Subscription** ページで、**openshift-storage** namespace を選択します。
8. 更新チャンネルおよび承認ストラテジーを指定します。
9. **Subscribe** をクリックします。
Installed Operators ページで、**OpenShift Container Storage Operator** は、**Succeeded** のステータスと共に **openshift-storage** プロジェクトに表示されます。

3.2.2.2. Multi-Cloud Object Gateway ストレージバケットの作成

Multi-Cloud Object Gateway (MCG) ストレージバケットのカスタムリソース (CR) を作成できます。

手順

1. OpenShift Container Platform クラスターにログインします。

```
$ oc login
```

2. **NooBaa** CR 設定ファイル **noobaa.yml** を以下の内容で作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: NooBaa
metadata:
  name: noobaa
  namespace: openshift-storage
spec:
  dbResources:
    requests:
      cpu: 0.5 1
```

```
memory: 1Gi
coreResources:
requests:
cpu: 0.5 ②
memory: 1Gi
```

- ① ② 非常に小規模なクラスターの場合、**cpu** の値を **0.1** に変更できます。

3. **NooBaa** オブジェクトを作成します。

```
$ oc create -f noobaa.yml
```

4. 以下の内容で、**BackingStore** CR 設定ファイル **bs.yml** を作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: BackingStore
metadata:
finalizers:
- noobaa.io/finalizer
labels:
app: noobaa
name: mcg-pv-pool-bs
namespace: openshift-storage
spec:
pvPool:
numVolumes: 3 ①
resources:
requests:
storage: 50Gi ②
storageClass: gp2 ③
type: pv-pool
```

- ① PV プール内のボリューム数を指定します。
- ② ボリュームのサイズを指定します。
- ③ ストレージクラスを作成します。

5. **BackingStore** オブジェクトを作成します。

```
$ oc create -f bs.yml
```

6. 以下の内容で **BucketClass** CR 設定ファイル **bc.yml** を作成します。

```
apiVersion: noobaa.io/v1alpha1
kind: BucketClass
metadata:
labels:
app: noobaa
name: mcg-pv-pool-bc
namespace: openshift-storage
spec:
placementPolicy:
```

```
tiers:
- backingStores:
- mcg-pv-pool-bs
placement: Spread
```

7. **BucketClass** オブジェクトを作成します。

```
$ oc create -f bc.yml
```

8. 以下の内容で **ObjectBucketClaim** CR 設定ファイル **obc.yml** を作成します。

```
apiVersion: objectbucket.io/v1alpha1
kind: ObjectBucketClaim
metadata:
  name: migstorage
  namespace: openshift-storage
spec:
  bucketName: migstorage ①
  storageClassName: openshift-storage.noobaa.io
  additionalConfig:
    bucketclass: mcg-pv-pool-bc
```

- ① レプリケーションリポジトリを CAM Web コンソールに追加するために使用するバケット名を記録します。

9. **ObjectBucketClaim** オブジェクトを作成します。

```
$ oc create -f obc.yml
```

10. リソース作成プロセスを監視し、**ObjectBucketClaim** ステータスが **Bound** であることを確認します。

```
$ watch -n 30 'oc get -n openshift-storage objectbucketclaim migstorage -o yaml'
```

このプロセスには、5分から10分の時間がかかる場合があります。

11. 以下の値を取得して記録します。この値は、レプリケーションリポジトリを CAM Web コンソールに追加する際に必要になります。

- S3 エンドポイント:

```
$ oc get route -n openshift-storage s3
```

- S3 プロバイダーアクセスキー:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_ACCESS_KEY_ID }}' | base64 -d
```

- S3 プロバイダーシークレットアクセスキー:

```
$ oc get secret -n openshift-storage migstorage -o go-template='{{
.data.AWS_SECRET_ACCESS_KEY }}' | base64 -d
```

3.2.3. AWS S3 ストレージバケットをレプリケーションリポジトリとして設定する

AWS S3 ストレージバケットをレプリケーションリポジトリとして設定できます。

前提条件

- AWS S3 ストレージバケットは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
- [AWS CLI](#) がインストールされていること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - EC2 Elastic Block Storage (EBS) にアクセスできる必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. AWS S3 バケットを作成します。

```
$ aws s3api create-bucket \
  --bucket <bucket_name> \ 1
  --region <bucket_region> 2
```

- 1** S3 バケット名を指定します。
- 2** S3 バケットリージョンを指定します (例: **us-east-1**)。

2. IAM ユーザー **velero** を作成します。

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

3. EC2 EBS スナップショットポリシーを作成します。

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

```

    }
  ]
}
EOF

```

4. 1つまたはすべての S3 バケットの AWS S3 アクセスポリシーを作成します。

```

$ cat > velero-s3-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:GetObject",
        "s3:DeleteObject",
        "s3:PutObject",
        "s3:AbortMultipartUpload",
        "s3:ListMultipartUploadParts"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket_name>/*" ❶
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListBucketMultipartUploads"
      ],
      "Resource": [
        "arn:aws:s3:::<bucket_name>" ❷
      ]
    }
  ]
}
EOF

```

- ❶ ❷ 単一の S3 バケットへのアクセスを付与するには、バケット名を指定します。すべての AWS S3 バケットへのアクセスを付与するには、バケット名の代わりに * を指定します。

```

"Resource": [
  "arn:aws:s3::*"
]

```

5. EC2 EBS ポリシーを **velero** に割り当てます。

```

$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-ebs \
  --policy-document file://velero-ec2-snapshot-policy.json

```

6. AWS S3 ポリシーを **velero** に割り当てます。

■

```
$ aws iam put-user-policy \
  --user-name velero \
  --policy-name velero-s3 \
  --policy-document file://velero-s3-policy.json
```

7. **velero** のアクセスキーを作成します。

```
$ aws iam create-access-key --user-name velero
{
  "AccessKey": {
    "UserName": "velero",
    "Status": "Active",
    "CreateDate": "2017-07-31T22:24:41.576Z",
    "SecretAccessKey": <AWS_SECRET_ACCESS_KEY>, ①
    "AccessKeyId": <AWS_ACCESS_KEY_ID> ②
  }
}
```

① ② AWS リポジトリを CAM Web コンソールに追加するために **AWS_SECRET_ACCESS_KEY** および **AWS_ACCESS_KEY_ID** を記録します。

3.2.4. Google Cloud Provider ストレージバケットをレプリケーションリポジトリとして設定する

Google Cloud Provider (GCP) ストレージバケットをレプリケーションリポジトリとして設定できません。

前提条件

- GCP ストレージバケットは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
- **gsutil** がインストールされていること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. **gsutil init** を実行してログインします。

```
$ gsutil init
Welcome! This command will take you through the configuration of gcloud.

Your current configuration has been set to: [default]

To continue, you must login. Would you like to login (Y/n)?
```

2. **BUCKET** 変数を設定します。

```
$ BUCKET=<bucket_name> ❶
```

- ❶ 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. **velero** サービスアカウントを作成します。

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

6. **SERVICE_ACCOUNT_EMAIL** 変数をサービスアカウントのメールアドレスに設定します。

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

7. パーミッションをサービスアカウントに付与します。

```
$ 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  
)  
  
gcloud iam roles create velero.server \  
--project $PROJECT_ID \  
--title "Velero Server" \  
--permissions "$(IFS=","; echo "${ROLE_PERMISSIONS[*]}")"  
  
gcloud projects add-iam-policy-binding $PROJECT_ID \  
--member serviceAccount:$SERVICE_ACCOUNT_EMAIL \  
--role projects/$PROJECT_ID/roles/velero.server  
  
gsutil iam ch serviceAccount:$SERVICE_ACCOUNT_EMAIL:objectAdmin gs://${BUCKET}
```

8. 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
```

3.2.5. Microsoft Azure Blob ストレージコンテナをレプリケーションリポジトリとして設定

Microsoft Azure Blob ストレージコンテナをレプリケーションリポジトリとして設定できます。

前提条件

- Azure ストレージアカウントがあること。
- Azure CLI がインストールされていること。
- Azure Blob ストレージコンテナがソースクラスターおよびターゲットクラスターからアクセスできること。
- スナップショットのコピー方法を使用する場合は、以下の条件を満たす必要があります。
 - ソースおよびターゲットクラスターが同じリージョンにある必要があります。
 - ソースおよびターゲットクラスターには、同じストレージクラスがある必要があります。
 - ストレージクラスはスナップショットと互換性がある必要があります。

手順

1. **AZURE_RESOURCE_GROUP** 変数を設定します。

```
$ AZURE_RESOURCE_GROUP=Velero_Backups
```

2. Azure リソースグループを作成します。

```
$ az group create -n $AZURE_RESOURCE_GROUP --location <CentralUS> ❶
```

- ❶ 場所を指定します。

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

```
$ AZURE_STORAGE_ACCOUNT_ID=velerobackups
```

4. 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
```

5. **BLOB_CONTAINER** 変数を設定します。

```
$ BLOB_CONTAINER=velero
```

- Azure Blob ストレージコンテナを作成します。

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

- 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`
```

- 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
```

3.3. CLUSTER APPLICATION MIGRATION (CAM) ツールのデプロイ

Cluster Application Migration (CAM) ツールをデプロイするには、Cluster Application Migration Operator を [OpenShift Container Platform 4.2 ソース](#) と [OpenShift Container Platform 4.2 ターゲットクラスター](#) にインストールし、[OpenShift Container Platform 4.2 ソースクラスター](#) でクロスオリジンリソース共有を設定する必要があります。

Cluster Application Migration Operator はデフォルトで CAM ツール (CAM Web コンソールおよび Migration コントローラー) をターゲットクラスターにインストールします。[CAM ツールは OpenShift Container Platform 3 およびリモートクラスターにインストールできます。](#)

3.3.1. OpenShift Container Platform 4.2 ソースクラスターへの Cluster Application Migration Operator のインストール

OLM を使用して OpenShift Container Platform 4.2 ソースクラスターに Cluster Application Migration Operator をインストールできます。

手順

- OpenShift Container Platform Web コンソールで、**Operators** → **OperatorHub** をクリックします。
- Filter by keyword** フィールド (この場合は **Migration**) を使用して **Cluster Application Migration Operator** を見つけます。

3. **Cluster Application Migration Operator** を選択し、**Install** をクリックします。
4. **Create Operator Subscription** ページで **openshift-migration** namespace を選択し、承認ストラテジーを指定します。
5. **Subscribe** をクリックします。
Installed Operators ページで、**Cluster Application Migration Operator** は、**InstallSucceeded** のステータスで **openshift-migration** プロジェクトに表示されます。
6. **Provided APIs** の下で **View 12 more...** をクリックします。
7. **Create New** → **MigrationController** をクリックします。
8. **spec** スタンザの **migration_controller** および **migration_ui** パラメーターを更新します。

```
spec:
  [...]
  migration_controller: false
  migration_ui: false
  [...]
```

9. **Create** をクリックします。
10. **Workloads** → **Pods** をクリックし、Restic および Velero Pod が実行されていることを確認します。

3.3.2. OpenShift Container Platform 4.2 ターゲットクラスターへのクラスターアプリケーション移行 Operator のインストール

OLM を使用して OpenShift Container Platform 4.2 ターゲットクラスターに Cluster Application Migration Operator をインストールできます。

Cluster Application Migration Operator は、デフォルトで CAM ツールをターゲットクラスターにインストールします。

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 (この場合は **Migration**) を使用して **Cluster Application Migration Operator** を見つけます。
3. **Cluster Application Migration Operator** を選択し、**Install** をクリックします。
4. **Create Operator Subscription** ページで **openshift-migration** namespace を選択し、承認ストラテジーを指定します。
5. **Subscribe** をクリックします。
Installed Operators ページで、**Cluster Application Migration Operator** は、**InstallSucceeded** のステータスで **openshift-migration** プロジェクトに表示されます。
6. **Provided APIs** の下で **View 12 more...** をクリックします。
7. **Create New** → **MigrationController** をクリックします。

8. **Create** をクリックします。
9. **Workloads** → **Pods** をクリックし、Controller Manager、Migration UI、Restic、および Velero Pod が実行中であることを確認します。

3.3.3. OpenShift Container Platform 4.2 ソースクラスターでのクロスオリジンリソース共有の設定

ソースクラスターの API サーバーと CAM ツール間の通信を有効にするために、OpenShift Container Platform 4.2 ソースクラスターでクロスオリジンリソース共有を設定する必要があります。

手順

1. CAM ツールがインストールされているクラスターにログインします。
2. CORS 設定の値を取得します。

```
$ oc get -n openshift-migration route/migration -o go-template='{(?i)://{ .spec.host }}(:|z){{println }}' | sed 's,\.,\\. ,g'
```

3. ソースクラスターにログインします。
4. 以下のように Kubernetes API サーバー CR を編集します。

```
$ oc edit apiserver.config.openshift.io cluster
```

5. **spec** スタンザの **additionalCORSAAllowedOrigins** に CORS 設定値を追加します。

```
spec:
  additionalCORSAAllowedOrigins:
    - (?i)//migration-openshift-migration\.apps\.cluster\.com(:|z) ❶
```

- ❶ CORS 設定値を指定します。

6. 変更を適用するためにファイルを保存します。
7. 設定を確認します。

```
$ curl -v -k -X OPTIONS \
"<cluster_url>/apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migclusters" \ ❶
-H "Access-Control-Request-Method: GET" \
-H "Access-Control-Request-Headers: authorization" \
-H "Origin: https://<CAM_web_console_url>" ❷
```

- ❶ CORS の設定されたクラスターの URL を指定します。
- ❷ CAM Web コンソールの URL を指定します。URL は CORS 設定値をベースとしています (例: **https://migration-openshift-migration.apps.cluster**)。

以下のような出力が表示されます。

```

< HTTP/2 204
< access-control-allow-credentials: true
< access-control-allow-headers: Content-Type, Content-Length, Accept-Encoding, X-CSRF-Token, Authorization, X-Requested-With, If-Modified-Since
< access-control-allow-methods: POST, GET, OPTIONS, PUT, DELETE, PATCH
< access-control-allow-origin: https://migration-openshift-migration.apps.cluster
< access-control-expose-headers: Date
< cache-control: no-store

```

3.4. CAM WEB コンソールを使用したアプリケーションの移行

3.4.1. CAM Web コンソールの起動

ブラウザで CAM Web コンソールを起動できます。

手順

1. CAM ツールがインストールされている OpenShift Container Platform クラスターにログインします。
2. 以下のコマンドを入力して CAM Web コンソール URL を取得します。

```

$ oc get -n openshift-migration route/migration -o go-template='https://{{ .spec.host }}'
https://migration-openshift-migration.apps.<cluster>.openshift.com

```

3. ブラウザーを起動し、CAM Web コンソールに移動します。



注記

Cluster Application Migration Operator のインストール後すぐに CAM Web コンソールにアクセスしようとする場合、Operator は依然としてクラスターを設定し、クロスオリジンリソース共有を有効にしているため、コンソールが読み込まれないことがあります。数分待機した後に再試行します。

4. 自己署名 CA 証明書を使用している場合、ソースクラスターの API サーバーの CA 証明書を受け入れることを求めるプロンプトが出されます。Web ページは、残りの証明書を受け入れるプロセスについて説明します。
5. OpenShift Container Platform の **ユーザー名** および **パスワード** を使用してログインします。

3.4.2. CAM Web コンソールへのクラスターの追加

ソースクラスターを CAM Web コンソールに追加できます。

前提条件

- クロスオリジンリソース共有がソースクラスターで設定されている必要があります。
- Azure スナップショットを使用してデータをコピーする場合:
 - ソースクラスターの追加時に Azure リソースグループ名を指定する必要があります。

手順

1. CAM Web コンソールにログインします。
2. **Replication repositories** セクションで、**Add repository** をクリックします。
3. **Storage provider type** を選択し、以下のフィールドに入力します。
 - **AWS** (AWS S3、MCG、および汎用 S3 プロバイダーの場合):
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **S3 bucket name**: 作成した S3 バケットの名前を指定します。
 - **S3 bucket region**: S3 バケットリージョンを指定します。AWS S3 の場合に **必須** です。Optional (他の S3 プロバイダーの場合)。
 - **S3 endpoint**: バケットではなく S3 サービスの URL を指定します (例: **https://<s3-storage.apps.cluster.com>**)。汎用 S3 プロバイダーの場合は **必須** です。**https://** プレフィックスを使用する必要があります。
 - **S3 provider access key**: AWS には **<AWS_SECRET_ACCESS_KEY>** を指定するか、または MCG には S3 プロバイダーアクセスキーを指定します。
 - **S3 provider secret access key**: AWS には **<AWS_ACCESS_KEY_ID>** を指定するか、または MCG には S3 プロバイダーシークレットアクセスキーを指定します。
 - **Require SSL verification**: 汎用 S3 プロバイダーを使用している場合は、このチェックボックスをクリアします。
 - **GCP**:
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **GCP bucket name**: GCP バケットの名前を指定します。
 - **GCP credential JSON blob: credentials-velero** ファイルに文字列を指定します。
 - **Azure**:
 - **Replication repository name**: CAM Web コンソールでレプリケーションリポジトリ名を指定します。
 - **Azure resource group**: Azure Blob ストレージのリソースグループを指定します。
 - **Azure storage account name**: Azure Blob ストレージアカウント名を指定します。
 - **Azure credentials - INI file contents credentials-velero** ファイルに文字列を指定します。
4. **Add repository** をクリックし、接続の検証を待機します。
5. **Close** をクリックします。
新規リポジトリが **Replication repositories** セクションに表示されます。

3.4.4. 大規模な移行の場合の移行計画の制限の変更

大規模な移行の移行計画の制限を変更することができます。



重要

移行の失敗を防ぐために、まず変更についてのテストをご使用の環境で実行する必要があります。

単一の移行計画には以下のデフォルト制限があります。

- 10 namespace
この制限を超えると、CAM Web コンソールは **Namespace limit exceeded** エラーを表示し、移行計画を作成することができません。
- 100 Pod
Pod の制限を超える場合、CAM Web コンソールには、以下の例と同様の警告メッセージが表示されます。 **Plan has been validated with warning condition(s). See warning message. Pod limit: 100 exceeded, found: 104**
- 100 永続ボリューム
永続ボリュームの制限を超過すると、CAM Web コンソールには同様の警告メッセージが表示されます。

手順

1. Migration コントローラー CR を編集します。

```
$ oc get migrationcontroller -n openshift-migration
NAME AGE
migration-controller 5d19h

$ oc edit migrationcontroller -n openshift-migration
```

2. 以下のパラメーターを更新します。

```
[...]
migration_controller: true

# This configuration is loaded into mig-controller, and should be set on the
# cluster where `migration_controller: true`
mig_pv_limit: 100
mig_pod_limit: 100
mig_namespace_limit: 10
[...]
```

3.4.5. CAM Web コンソールでの移行計画の作成

CAM Web コンソールで移行計画を作成できます。

前提条件

- CAM Web コンソールには以下が含まれている必要があります。
 - ソースクラスター

- CAM ツールのインストール時に自動的に追加されるターゲットクラスター
- レプリケーションリポジトリ
- スナップショットを使用してデータをコピーする必要がある場合、ソースおよびターゲットクラスターは、同じクラウドプロバイダー（AWS、GCP、または Azure）および同じリージョンで実行される必要があります。

手順

1. CAM Web コンソールにログインします。
2. **Plans** セクションで、**Add plan** をクリックします。
3. **Plan name** を入力し、**Next** をクリックします。
Plan name には、最大 253 文字の小文字の英数字 (**a-z**、**0-9**) を含めることができます。スペースまたはアンダースコア (**_**) を含めることはできません。
4. **Source cluster** を選択します。
5. **Target cluster** を選択します。
6. **Replication repository** を選択します。
7. 移行するプロジェクトを選択し、**Next** をクリックします。
8. PV の **Copy** または **Move** を選択します。
 - **Copy** は、ソースクラスターの PV のデータをレプリケーションリポジトリにコピーしてから、これを同様の特徴のある新規に作成された PV で復元します。
 - **Move** は、ソースクラスターからリモートボリューム (例: NFS) をアンマウントし、リモートボリュームをポイントするターゲットクラスターで PV リソースを作成し、その後にリモートボリュームをターゲットクラスターにマウントします。ターゲットクラスターで実行されているアプリケーションは、ソースクラスターが使用していたものと同じリモートボリュームを使用します。リモートボリュームは、ソースクラスターおよびターゲットクラスターからアクセスできる必要があります。
9. **Next** をクリックします。
10. PV の **Copy method** を選択します。
 - **Snapshot** は、クラウドプロバイダーのスナップショット機能を使用してディスクのバックアップおよび復元を行います。この場合、**ファイルシステム**を使用する場合よりもはるかに高速になります。



注記

ストレージおよびクラスターは同じリージョンにあり、ストレージクラスには互換性がなければなりません。

- **ファイルシステム**は、データファイルをソースディスクから新規に作成されたターゲットディスクにコピーします。

11. PV の **Storage class** を選択します。

--- 注記: この方法を選択した場合、移行時に PV のデータは自動的にターゲットクラスターにコピーされ、元の PV は削除されます。

Filesystem のコピー方法を選択した場合、移行時にストレージクラスを変更できます。たとえば、Red Hat Gluster Storage または NFS ストレージから Red Hat Ceph Storage に変更できます。

12. **Finish** をクリックします。
13. **Close** をクリックします。
移行計画は **Plans** セクションに表示されます。

3.4.6. CAM Web コンソールでの移行計画の実行

CAM Web コンソールで作成した移行計画を使用してアプリケーションとデータをステージングしたり、移行したりできます。

前提条件

CAM Web コンソールには以下が含まれている必要があります。

- ソースクラスター
- CAM ツールのインストール時に自動的に追加されるターゲットクラスター
- レプリケーションリポジトリ
- 有効な移行計画

手順

1. OpenShift Container Platform 4 クラスターで CAM Web コンソールにログインします。
2. 移行計画を選択します。
3. **Stage** をクリックし、アプリケーションを停止せずにソースクラスターからターゲットクラスターにデータをコピーします。
実際の移行時間を短縮するには、**Stage** を複数回実行することができます。
4. アプリケーションのワークロードを移行する準備ができれば、**Migrate** をクリックします。
Migrate は、ソースクラスターでアプリケーションワークロードを停止し、ターゲットクラスターでそのリソースを再作成します。
5. オプション: **Migrate** ウィンドウで **Do not stop applications on the source cluster during migration** を選択できます。
6. **Migrate** をクリックします。
7. 移行が完了したら、アプリケーションが OpenShift Container Platform 4.2 Web コンソールで正常に移行されていることを確認します。
 - a. **Home** → **Projects** をクリックします。
 - b. 移行されたプロジェクトをクリックしてそのステータスを表示します。
 - c. **Routes** セクションで **Location** をクリックし、アプリケーションが機能していることを確認します (該当する場合)。
 - d. **Workloads** → **Pods** をクリックし、Pod が移行した namespace で実行されていることを確認します。

- e. **Storage → Persistent volumes** をクリックして、移行した永続ボリュームが正常にプロビジョニングされていることを確認します。

3.5. トラブルシューティング

移行用のカスタムリソース (CR) を表示し、ログをダウンロードして失敗した移行の失敗についてのトラブルシューティングを行うことができます。

移行の失敗時にアプリケーションが停止した場合は、データ破損を防ぐために手作業でアプリケーションをロールバックする必要があります。

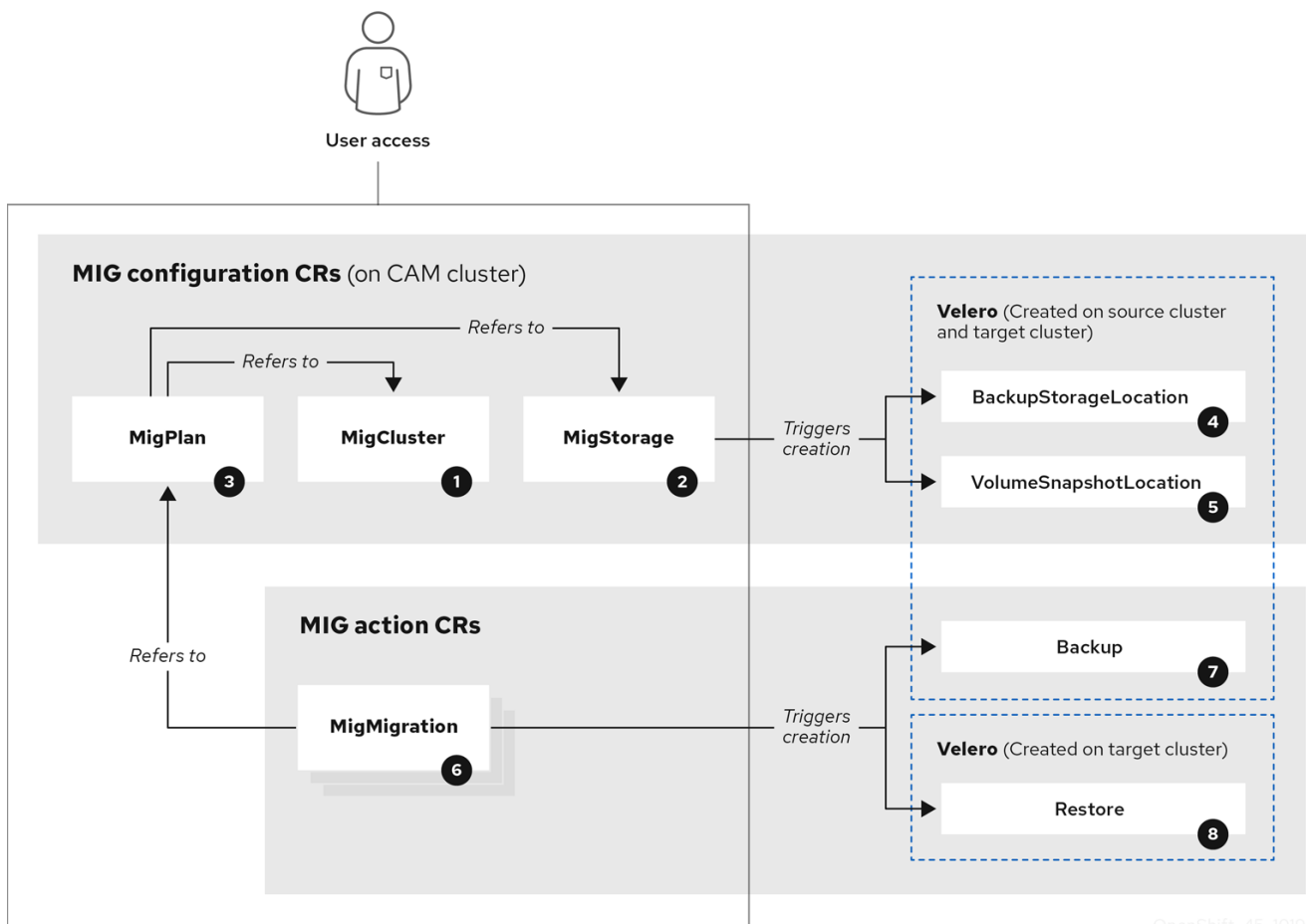


注記

移行時にアプリケーションが停止しなかった場合には、手動のロールバックは必要ありません。元のアプリケーションがソースクラスター上で依然として実行されているためです。

3.5.1. 移行カスタムリソース (CR) の表示

Cluster Application Migration (CAM) ツールは移行用に以下の CR を作成します。



1 **MigCluster** (設定、CAM クラスター): クラスター定義

2 **MigStorage** (設定、CAM クラスター): ストレージ定義

3 MigPlan (設定、CAM クラスタ): 移行計画

MigPlan CR は移行されるソースおよびターゲットクラスター、リポジトリ、および namespace を記述します。これは 0、1 または多数の MigMigration CR に関連付けられます。



注記

MigPlan CR を削除すると、関連付けられた MigMigration CR が削除されます。

4 BackupStorageLocation (設定、CAM クラスタ): Velero バックアップオブジェクトの場所

5 VolumeSnapshotLocation (設定、CAM クラスタ): ボリュームスナップショットの場所

6 MigMigration (アクション、CAM クラスタ): 移行、移行時に作成される

MigMigration CR は、データのステージングまたは移行を実行するたびに作成されます。各 MigMigration CR は MigPlan CR に関連付けられます。

7 Backup (アクション、ソースクラスター): 移行計画の実行時に、MigMigration CR は各ソースクラスターに 2 つの Velero バックアップ CR を作成します。

- Kubernetes オブジェクトのバックアップ CR #1
- PV データのバックアップ CR #2

8 Restore (アクション、ターゲットクラスター): 移行計画の実行時に、MigMigration CR はターゲットクラスターに 2 つのリストア CR を作成します。

- PV データのリストア CR #1 (バックアップ CR #2 の使用)
- Kubernetes オブジェクトのリストア CR #2 (バックアップ CR #1 の使用)

手順

1. CR 名を取得します。

```
$ oc get <cr> -n openshift-migration 1
NAME                                AGE
88435fe0-c9f8-11e9-85e6-5d593ce65e10 6m42s
```

- 1** 表示する移行 CR を指定します。

2. CR を表示します。

```
$ oc describe <cr> 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
```

出力は以下の例のようになります。

MigMigration の例

```

$ oc describe migmigration 88435fe0-c9f8-11e9-85e6-5d593ce65e10 -n openshift-migration
Name:      88435fe0-c9f8-11e9-85e6-5d593ce65e10
Namespace: openshift-migration
Labels:    <none>
Annotations: touch: 3b48b543-b53e-4e44-9d34-33563f0f8147
API Version: migration.openshift.io/v1alpha1
Kind:      MigMigration
Metadata:
  Creation Timestamp: 2019-08-29T01:01:29Z
  Generation:        20
  Resource Version:  88179
  Self Link:         /apis/migration.openshift.io/v1alpha1/namespaces/openshift-
migration/migmigrations/88435fe0-c9f8-11e9-85e6-5d593ce65e10
  UID:               8886de4c-c9f8-11e9-95ad-0205fe66cbb6
Spec:
  Mig Plan Ref:
    Name:      socks-shop-mig-plan
    Namespace: openshift-migration
  Quiesce Pods: true
  Stage:       false
Status:
  Conditions:
    Category:      Advisory
    Durable:       true
    Last Transition Time: 2019-08-29T01:03:40Z
    Message:       The migration has completed successfully.
    Reason:        Completed
    Status:        True
    Type:          Succeeded
    Phase:         Completed
    Start Timestamp: 2019-08-29T01:01:29Z
  Events:         <none>

```

Velero バックアップ CR #2 の例 (PV データ)

```

apiVersion: velero.io/v1
kind: Backup
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.105.179:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-44dd3bd5-c9f8-11e9-95ad-
0205fe66cbb6
  creationTimestamp: "2019-08-29T01:03:15Z"
  generateName: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-
  generation: 1
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    migration-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
    velero.io/storage-location: myrepo-vpzq9
  name: 88435fe0-c9f8-11e9-85e6-5d593ce65e10-59gb7
  namespace: openshift-migration
  resourceVersion: "87313"

```

```

selfLink: /apis/velero.io/v1/namespaces/openshift-migration/backups/88435fe0-c9f8-11e9-85e6-
5d593ce65e10-59gb7
uid: c80dbbc0-c9f8-11e9-95ad-0205fe66cbb6
spec:
  excludedNamespaces: []
  excludedResources: []
  hooks:
    resources: []
  includeClusterResources: null
  includedNamespaces:
  - sock-shop
  includedResources:
  - persistentvolumes
  - persistentvolumeclaims
  - namespaces
  - imagestreams
  - imagestreamtags
  - secrets
  - configmaps
  - pods
  labelSelector:
    matchLabels:
      migration-included-stage-backup: 8886de4c-c9f8-11e9-95ad-0205fe66cbb6
  storageLocation: myrepo-vpzq9
  ttl: 720h0m0s
  volumeSnapshotLocations:
  - myrepo-wv6fx
status:
  completionTimestamp: "2019-08-29T01:02:36Z"
  errors: 0
  expiration: "2019-09-28T01:02:35Z"
  phase: Completed
  startTimestamp: "2019-08-29T01:02:35Z"
  validationErrors: null
  version: 1
  volumeSnapshotsAttempted: 0
  volumeSnapshotsCompleted: 0
  warnings: 0

```

Velero リストア CR #2 の例 (Kubernetes リソース)

```

apiVersion: velero.io/v1
kind: Restore
metadata:
  annotations:
    openshift.io/migrate-copy-phase: final
    openshift.io/migrate-quiesce-pods: "true"
    openshift.io/migration-registry: 172.30.90.187:5000
    openshift.io/migration-registry-dir: /socks-shop-mig-plan-registry-36f54ca7-c925-11e9-825a-
06fa9fb68c88
  creationTimestamp: "2019-08-28T00:09:49Z"
  generateName: e13a1b60-c927-11e9-9555-d129df7f3b96-
  generation: 3
  labels:
    app.kubernetes.io/part-of: migration
    migmigration: e18252c9-c927-11e9-825a-06fa9fb68c88

```

```

migration-final-restore: e18252c9-c927-11e9-825a-06fa9fb68c88
name: e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
namespace: openshift-migration
resourceVersion: "82329"
selfLink: /apis/velero.io/v1/namespaces/openshift-migration/restores/e13a1b60-c927-11e9-9555-d129df7f3b96-gb8nx
uid: 26983ec0-c928-11e9-825a-06fa9fb68c88
spec:
  backupName: e13a1b60-c927-11e9-9555-d129df7f3b96-sz24f
  excludedNamespaces: null
  excludedResources:
  - nodes
  - events
  - events.events.k8s.io
  - backups.velero.io
  - restores.velero.io
  - resticrepositories.velero.io
  includedNamespaces: null
  includedResources: null
  namespaceMapping: null
  restorePVs: true
status:
  errors: 0
  failureReason: ""
  phase: Completed
  validationErrors: null
  warnings: 15

```

3.5.2. 移行ログのダウンロード

CAM Web コンソールで Velero、Restic、および Migration コントローラーログをダウンロードして、移行の失敗についてのトラブルシューティングを行うことができます。

手順

1. CAM Web コンソールにログインします。
2. **Plans** をクリックし、移行計画の一覧を表示します。
3. **Options** メニュー  をクリックし (特定の移行計画について)、**Logs** を選択します。
4. **Download Logs** をクリックし、すべてのクラスターの Migration コントローラー、Velero、および Restic のログをダウンロードします。
5. 特定のログをダウンロードするには、以下を実行します。
 - a. ログオプションを指定します。
 - **Cluster:** ソース、ターゲット、または CAM ホストクラスターを選択します。
 - **Log source:** Velero、Restic、または **Controller** を選択します。
 - **Pod source:** Pod 名を選択します (例: **controller-manager-78c469849c-v6wcf**)。選択したログが表示されます。

選択した内容を変更することで、ログ選択の設定をクリアできます。

- b. **Download Selected** をクリックし、選択したログをダウンロードします。

オプションで、以下の例にあるように CLI を使用してログにアクセスできます。

```
$ oc get pods -n openshift-migration | grep controller
controller-manager-78c469849c-v6wcf      1/1   Running   0      4h49m

$ oc logs controller-manager-78c469849c-v6wcf -f -n openshift-migration
```

3.5.3. Restic タイムアウトエラー

Restic のタイムアウトにより移行が失敗する場合、以下のエラーが Velero ログに表示されます。

```
level=error msg="Error backing up item" backup=velero/monitoring error="timed out waiting for all
PodVolumeBackups to complete"
error.file="/go/src/github.com/heptio/velero/pkg/restic/backupper.go:165"
error.function="github.com/heptio/velero/pkg/restic.(*backupper).BackupPodVolumes" group=v1
```

restic_timeout のデフォルト値は1時間です。大規模な移行では、この値を大きくすることができません。値を高くすると、エラーメッセージが返されるタイミングが送れる可能性があることに注意してください。

手順

1. OpenShift Container Platform Web コンソールで、**Operators** → **Installed Operators** に移動します。
2. **Cluster Application Migration Operator** をクリックします。
3. **MigrationController** タブで、**migration-controller** をクリックします。
4. **YAML** タブで、以下のパラメーター値を更新します。

```
spec:
  restic_timeout: 1h 1
```

- 1** 有効な単位は **h** (時間)、**m** (分)、および **s** (秒) です (例: **3h30m15s**)。

5. **Save** をクリックします。

3.5.4. 移行の手動ロールバック

移行の失敗時にアプリケーションが停止された場合は、PV でのデータの破損を防ぐために手動でこれをロールバックする必要があります。

移行時にアプリケーションが停止しなかった場合には、この手順は必要ありません。元のアプリケーションがソースクラスター上で依然として実行されているためです。

手順

1. ターゲットクラスター上で、移行したプロジェクトに切り替えます。


```
$ oc project <project>
```

2. デプロイされたリソースを取得します。

```
$ oc get all
```

3. デプロイされたリソースを削除し、アプリケーションがターゲットクラスターで実行されておらず、PVC 上にあるデータにアクセスできるようにします。

```
$ oc delete <resource_type>
```

4. これを削除せずに DaemonSet を停止するには、YAML ファイルで **nodeSelector** を更新します。

```
apiVersion: extensions/v1beta1
kind: DaemonSet
metadata:
  name: hello-daemonset
spec:
  selector:
    matchLabels:
      name: hello-daemonset
  template:
    metadata:
      labels:
        name: hello-daemonset
    spec:
      nodeSelector:
        role: worker ❶
```

- ❶ いずれのノードにも存在しない **nodeSelector** 値を指定します。

5. 不要なデータが削除されるように、各 PV の回収ポリシーを更新します。移行時に、バインドされた PV の回収ポリシーは **Retain** であり、アプリケーションがソースクラスターから削除される際にデータの損失が生じないようにされます。ロールバック時にこれらの PV を削除できます。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: pv0001
spec:
  capacity:
    storage: 5Gi
  accessModes:
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain ❶
  ...
status:
  ...
```

- ❶ **Recycle** または **Delete** を指定します。

6. ソースクラスターで、移行したプロジェクトに切り替え、そのデプロイされたリソースを取得します。

```
$ oc project <project>
$ oc get all
```

7. デプロイされた各リソースのレプリカを開始します。

```
$ oc scale --replicas=1 <resource_type>/<resource_name>
```

8. 手順の実行時に変更した場合は、DaemonSet の **nodeSelector** を元の値に更新します。

3.5.5. カスタマーサポートケース用のデータの収集

カスタマーサポートケースを作成する場合、**openshift-migration-must-gather-rhel8** イメージを使用して **must-gather** ツールを実行し、クラスターについての情報を収集し、これを [Red Hat カスタマーポータル](#) にアップロードできます。

openshift-migration-must-gather-rhel8 イメージは、デフォルトの **must-gather** イメージで収集されないログおよびカスタムリソースデータを収集します。

手順

1. **must-gather** データを保存するディレクトリーに移動します。
2. **oc adm must-gather** コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/rhcam-1-2/openshift-migration-must-gather-rhel8
```

must-gather ツールはクラスター情報を収集し、これを **must-gather.local.<uid>** ディレクトリーに保存します。

3. 認証キーおよびその他の機密情報を **must-gather** データから削除します。
4. **must-gather.local.<uid>** ディレクトリーの内容を含むアーカイブファイルを作成します。

```
$ tar cvaf must-gather.tar.gz must-gather.local.<uid>/
```

圧縮ファイルを [Red Hat カスタマーポータル](#) 上のサポートケースに添付します。

3.5.6. 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。

- 移行中に、CAM ツールは以下の namespace アノテーションを保持します。

- **openshift.io/sa.scc.mcs**
- **openshift.io/sa.scc.supplemental-groups**
- **openshift.io/sa.scc.uid-range**

これらのアノテーションは UID 範囲を保持し、コンテナーがターゲットクラスターのファイルシステムのパーミッションを保持できるようにします。移行された UID が、ターゲッ

トクラスターの既存の namespace または今後の namespace 内の UID を重複させるリスクがあります。(BZ#1748440)

- S3 エンドポイントを CAM Web コンソールに追加する場合、**https://** は AWS でのみサポートされます。他の S3 プロバイダーの場合は **http://** を使用します。
- AWS バケットが CAM Web コンソールに追加された後に削除される場合、MigStorage CR は更新されないため、そのステータスは **True** のままになります。(BZ#1738564)
- Migration コントローラーが、ターゲットクラスター以外のクラスターで実行されている場合は移行に失敗します。**EnsureCloudSecretPropagated** フェーズはログに記録された警告を出して省略されます。(BZ#1757571)
- クラスターのロールバインディングおよび SCC (Security Context Constraints) を含むクラスタースコープのリソースは CAM によって処理されません。アプリケーションがクラスタースコープのリソースを必要とする場合、ターゲットクラスターでそれらを手動で作成する必要があります。(BZ#1759804)
- 移行計画の作成時に、ソースクラスターのストレージクラスが誤って表示されます。(BZ#1777869)
- CAM Web コンソールのクラスターにアクセス不可の場合、これはオープン状態の移行計画のクローズを試行します。(BZ#1758269)
- 移行に失敗すると、移行計画は休止状態の Pod のカスタム PV 設定を保持しません。移行を手動でロールバックし、移行計画を削除し、PV 設定で新たな移行計画を作成する必要があります。(BZ#1784899)