Red Hat OpenShift Service on AWS 4

ストレージおよびレジストリー

Red Hat OpenShift Service on AWS クラスターのストレージの設定

Last Updated: 2022-12-05

Red Hat OpenShift Service on AWS 4 ストレージおよびレジストリー

Red Hat OpenShift Service on AWS クラスターのストレージの設定

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律上の通知

概要

本書では、Red Hat OpenShift Service on AWS (ROSA) クラスターへのストレージの設定について説明します。

目次

第1章 永続ストレージ

1.1. AWS ELASTIC BLOCK STORE (EBS) を使用した永続ストレージ

Red Hat OpenShift Service on AWS (ROSA) クラスターは、AWS Elastic Block Store (EBS) ボリュームを使用する 2 つのストレージクラスで事前に構築されています。これらのストレージクラスはすぐに使用でき、Kubernetes と AWS にある程度精通していることを前提としています。

以下は、2 つのビルド済みストレージクラスです。

Name	プロビジョナー
gp2	kubernetes.io/aws-ebs
gp2-csi	ebs.csi.aws.com

gp2 ストレージクラスがデフォルトとして設定されています。ただし、デフォルトのストレージクラスとしていずれかを選択できます。

Kubernetes 永続ボリュームフレームワークは、管理者がクラスターのプロビジョニングを永続ストレージを使用して実行できるようにし、ユーザーが基礎となるインフラストラクチャーの知識がなくてもこれらのリソースを要求できるようにします。AWS EBS ボリュームを動的にプロビジョニングできます。永続ボリュームは、単一のプロジェクトまたは名前空間にバインドされていません。したがって、ボリュームは ROSA クラスター間で共有できます。永続ボリューム要求はプロジェクトまたは namespace に固有のもので、ユーザーによって要求されます。

重要

- ROSA はデフォルトで、ツリー内または非 Container Storage Interface (CSI) プラグインを使用して AWS EBS ストレージをプロビジョニングします。今後の ROSA バージョンでは、既存の in-tree プラグインを使用してプロビジョニングされるボリュームは、同等の CSI ドライバーに移行される予定です。完全な移行後、ツリー内プラグインは ROSA の将来のバージョンから削除される予定です。
- インフラストラクチャーにおけるストレージの高可用性は、基礎となるストレージのプロバイダーに委ねられています。

1.1.1. 永続ボリュームのフォーマット

ROSA クラスターは、ボリュームをマウントしてコンテナーに渡す前に、永続ボリューム定義の **fsType** パラメーターで指定されたファイルシステムがボリュームにあるかどうかを確認します。デバイスが指定されたファイルシステムでフォーマットされていない場合は、デバイスのデータがすべて消去され、デバイスはそのファイルシステムで自動的にフォーマットされます。この検証により、ROSA クラスターが最初に使用する前に AWS ボリュームをフォーマットするため、フォーマットされていない AWS ボリュームを永続ボリュームとして使用できます。

1.1.2. ノード上の EBS ボリュームの容量

デフォルトでは、ROSA クラスターは 1 つのノードに接続された最大 39 の EBS ボリュームをサポートします。この制限は、AWS ボリュームの制限 に合致します。ボリュームの制限は、インスタンスのタイプによって異なります。

重要

In-tree または CSI ボリュームのいずれかと、それぞれのストレージクラスを使用する必要がありますが、ボリュームの両方のタイプを同時に使用することはできません。割り当てられている EBS ボリュームの最大数は、in-tree および CSI ボリュームについて別々にカウントされるため、各タイプの EBS ボリュームを最大 39 個持つことができます。

In-tree (インツリー) ボリュームプラグインでは不可能なボリュームスナップショットなどの追加ストレージオプションへのアクセスに関する詳細は、Elastic Block Store CSI Driver Operator を参照してください。

1.1.3. 永続ボリューム要求の作成

前提条件

ストレージは、ボリュームとして ROSA クラスターにマウントされる前に基礎となるインフラストラクチャーになければなりません。

手順

1. OpenShift Cluster コンソールで、**Storage → Persistent Volume Claims**をクリックします。
2. 永続ボリューム要求の概要で、**Create Persistent Volume Claim**をクリックします。
3. 表示されるページで必要なオプションを定義します。
   a. ドロップダウンメニューから以前に作成したストレージクラスを選択します。
   b. ストレージ要求の一意の名前を入力します。
   c. アクセスモードを選択します。この選択により、ストレージクレームの読み取りおよび書き込みアクセスが決定されます。
   d. ストレージ要求のサイズを定義します。
4. **Create** をクリックして永続ボリューム要求を作成し、永続ボリュームを生成します。

1.2. RED HAT OPENSHIFT SERVICE ON AWS への AWS EFS の設定

警告

この手順は、Amazon Web Services Elastic File System (AWS EFS) community Operator に固有のものであり、Red Hat OpenShift Service on AWS 4.9 までのバージョンにのみ適用されます。

Amazon Web Services Elastic File System (AWS EFS) は、Red Hat OpenShift Service on AWS クラスターでプロビジョニングできる Network File System (NFS) です。AWS は、Kubernetes ワークロードがこの共有ファイルストレージを活用できるようにする Kubernetes で使用する CSI EFS ドライバーも提供し、サポートしています。

本書では、Red Hat OpenShift Service on AWS で使用される EFS を準備するために必要な基本的な手順について説明します。AWS EFS の詳細は、AWS EFS のドキュメント を参照してください。

重要

Red Hat は、バックアップや復元を含む、この機能に対する正式なサポートを提供しません。お客様は EFS データのバックアップを作成し、停止またはデータ損失が発生したときに復元します。

クラスターで EFS を有効にする高レベルのプロセスは、以下のとおりです。

1. クラスターで使用される AWS アカウントに AWS EFS を作成します。
2. OperatorHub から AWS EFS Operator をインストールします。
3. **SharedVolume** カスタムリソースを作成します。
4. Pod の **spec.volumes** で生成された永続ボリューム要求を使用します。

1.2.1. 前提条件

- Red Hat OpenShift Service on AWS クラスター
- そのクラスターの AWS アカウントへの管理者アクセス

1.2.2. AWS アカウントの設定

Red Hat OpenShift Service on AWS で使用する AWS EFS を準備するには、AWS アカウントを設定します。

手順

1. AWS EC2 コンソール にログインします。
2. クラスターリージョンに一致するリージョンを選択します。
3. ワーカー EC2 インスタンスのみをフィルターし、インスタンスを選択します。VPC ID およびセキュリティーグループ ID を書き留めます。これらの値は、プロセスの後半に必要です。
4. **Security** タブをクリックし、Security Group Name をクリックします。
5. **Actions** ドロップダウンメニューから、**Edit Inbound Rules** をクリックします。下にスクロールし、**Add Rule** をクリックします。
6. VPC プライベート CIDR から NFS トラフィックを許可する NFS ルールを追加します。
7. Amazon EFS ページ を開きます。EFS を作成するには、**Create file system** をクリックします。
8. **Customize** をクリックし、ウィザードに進みます。
    a. **Step 2:** で、ネットワークアクセスを設定します。
        i. 書き留めておいたクラスターの VPC をクリックします。
        ii. プライベートサブネットが選択されていることを確認します。

iii. EC2 ワーカーインスタンス用に以前に書き留めた Security Group Name を選択します。

iv. **Next** をクリックします。

b. **Step 3:** で、クライアントアクセスを設定します。

i. **Add access point** をクリックします。

ii. **/access_point_1** などの一意のパスを入力します。

iii. ワーカー Pod への書き込みアクセスを許可する所有権またはパーミッションで Owner フィールドを設定します。たとえば、ワーカー Pod をグループ ID **100** で実行する場合は、その ID を **Owner Group ID** として設定し、パーミッションに **g+rwx** が含まれるようにします。

9. ウィザードの手順を実行し、**ファイルシステムの作成** をクリックします。

10. ファイルシステムの作成後は、以下を行います。

a. 後で使用できるようにファイルシステム ID を書き留めます。

b. **Manage client access** をクリックし、アクセスポイント ID を書き留めます。

ステップ 5 ~ 10 を使用してさらに NFS ルールを追加して、別の共有データストアを作成できます。それぞれのケースで、対応するファイルシステム ID およびアクセスポイント ID を書き留めます。

1.2.3. EFS Operator のインストール

手順

1. クラスターの OpenShift Web UI にログインします。

2. **Operators → OperatorHub** をクリックします。

3. AWS EFS Operator を検索して、選択します。**Install** をクリックします。

4. デフォルト設定を受け入れ、**Subscribe** をクリックします。

1.2.4. コンソールを使用した SharedVolume リソースの作成

ファイルシステムとアクセスポイントのペアごとに 1 つの **SharedVolume** リソースを作成する必要があります。このペアは、Pod がこれにアクセスする各プロジェクトに作成します。

手順

1. OpenShift Web コンソールで、プロジェクトを作成し、これに移動します。

2. **Operators → Installed Operators** をクリックします。AWS EFS Operator のエントリーを見つけ、Provided APIs の下にある **SharedVolume** をクリックします。

3. **Create SharedVolume** をクリックします。

4. サンプル YAML を編集します。

a. **name** に適切な値を入力します。

b. **accessPointID** および **fileSystemID** の値は、先に作成した EFS リソースの値に置き換えてください。

```
apiVersion: aws-efs.managed.openshift.io/v1alpha1
kind: SharedVolume
metadata:
  name: sv1
  namespace: efsop2
spec:
  accessPointID: fsap-0123456789abcdef
  fileSystemID: fs-0123cdef
```

5. **Create** をクリックします。
SharedVolume リソースが作成され、AWS EFS Operator がトリガーされて PersistentVolume:PersistentVolumeClaim ペアが生成され、指定した EFS アクセスポイントに関連付けられます。

6. 永続ボリューム要求 (PVC) が存在し、バインドされていることを確認するには、**Storage → Persistent Volume Claims** をクリックします。
PVC 名は **pvc-<shared_volume_name>** です。関連付けられた PV 名は **pv-<project_name>-<shared_volume_name>** です。

1.2.5. CLI を使用した SharedVolume リソースの作成

ファイルシステムとアクセスポイントのペアごとに 1 つの **SharedVolume** リソースを作成する必要があります。このペアは、Pod がこれにアクセスする各プロジェクトに作成します。YAML または JSON 定義を入力するか、またはファイルをエディターにドラッグアンドドロップして、SharedVolume を手動で作成できます。

手順

1. **oc** CLI を使用して、先に作成した EFS リソースから **accessPointID** および **fileSystemID** 値を使用して YAML ファイルを作成します。

```
apiVersion: aws-efs.managed.openshift.io/v1alpha1
kind: SharedVolume
metadata:
  name: sv1
  namespace: efsop2
spec:
  accessPointID: fsap-0123456789abcdef
  fileSystemID: fs-0123cdef
```

2. 以下のコマンドを使用して、ファイルをクラスターに適用します。

```
$ oc apply -f <filename>.yaml
```

SharedVolume リソースが作成され、AWS EFS Operator がトリガーされて PersistentVolume:PersistentVolumeClaim ペアが生成され、指定した EFS アクセスポイントに関連付けられます。

3. PVC が存在し、バインドされていることを確認するには、**Storage** > **Persistent Volume Claims** に移動します。

PVC 名は **pvc-{shared_volume_name}** です。関連付けられた PV 名は **pv-{project_name}-{shared_volume_name}** です。

1.2.6. Pod の接続

プロジェクトで作成された永続ボリューム要求 (PVC) を使用する準備が整いました。このサンプル Pod を作成してこの PVC をテストできます。

手順

1. プロジェクトを作成し、これに移動します。
2. **Workloads → Pods → Create Pod** をクリックします。
3. YAML 情報を入力します。**PersistentVolumeClaim** オブジェクトの名前を **.spec.volumes[].persistentVolumeClaim.claimName** で使用します。

   例

```
apiVersion: v1
kind: Pod
metadata:
 name: test-efs
spec:
 volumes:
   - name: efs-storage-vol
     persistentVolumeClaim:
       claimName: pvc-sv1
 containers:
   - name: test-efs
     image: centos:latest
     command: [ "/bin/bash", "-c", "--" ]
     args: [ "while true; do touch /mnt/efs-data/verify-efs && echo 'hello efs' && sleep 30;
done;" ]
     volumeMounts:
       - mountPath: "/mnt/efs-data"
         name: efs-storage-vol
```

4. Pod の作成後に、**Workloads → Pods → Logs** をクリックし、Pod のログを確認します。

1.2.7. EFS Operator のアンインストール

手順

Operator をクラスターから削除するには、以下を実行します。

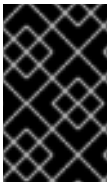
1. Operator によって生成された永続ボリューム要求を使用してすべてのワークロードを削除します。
2. すべての namespace から共有ボリュームをすべて削除します。Operator は関連付けられた永続ボリュームおよび永続ボリューム要求を自動的に削除します。
3. Operator をアンインストールします。
   a. **Operators → Installed Operators** をクリックします。

b. AWS EFS Operator のエントリーを見つけ、Operator の右側のメニューボタンをクリックします。

c. **Uninstall** をクリックし、削除を確定します。

4. 共有ボリューム CRD を削除します。このアクションは、残りの Operator 所有リソースの削除をトリガーします。

1.3. AWS ELASTIC FILE SERVICE CSI ドライバー OPERATOR の設定



重要

この手順は、Amazon Web Services Elastic File System (AWS EFS) CSI Driver Operator に固有のものであり、Red Hat OpenShift Service on AWS 4.10 以降のバージョンにのみ適用されます。

1.3.1. 概要

Red Hat OpenShift Service on AWS は、AWS Elastic File Service (EFS) の Container Storage Interface (CSI) ドライバーを使用して永続ボリューム (PV) をプロビジョニングできます。

CSI Operator およびドライバーを使用する場合、永続ストレージ および CSI ボリュームの設定 について理解しておくことが推奨されます。

AWS EFS CSI Driver Operator をインストールした後、Red Hat OpenShift Service on AWS は、デフォルトで AWS EFS CSI Operator と AWS EFS CSI ドライバーを **openshift-cluster-csi-drivers** namespace にインストールします。これにより、AWS EFS CSI ドライバー Operator は、AWS EFS アセットにマウントする CSI がプロビジョニングする PV を作成することができます。

- **AWS EFS CSI ドライバー Operator**をインストールしても、永続ボリューム要求 (PVC) の作成に使用するストレージクラスがデフォルトで作成されません。ただし、AWS EFS **StorageClass** を手動で作成することは可能です。AWS EFS CSI ドライバー Operator は、ストレージボリュームをオンデマンドで作成できるようにし、クラスター管理者がストレージを事前にプロビジョニングする必要がなくすことで、動的ボリュームのプロビジョニングをサポートします。

- **AWS EFS CSI ドライバー** を使用すると、AWS EFS PV を作成し、マウントできます。

注記

AWS EFS はリージョナルボリュームのみをサポートしており、ゾーンボリュームはサポートしていません。

1.3.2. CSI について

ストレージベンダーはこれまで Kubernetes の一部としてストレージドライバーを提供してきました。Container Storage Interface (CSI) の実装では、サードパーティーのプロバイダーは、コア Kubernetes コードを変更せずに標準のインターフェイスを使用してストレージプラグインを提供できます。

CSI Operator は、in-tree (インツリー) ボリュームプラグインでは不可能なボリュームスナップショットなどのストレージオプションを Red Hat OpenShift Service on AWS ユーザーに付与します。

1.3.3. AWS EFS CSI ドライバー Operator のインストール

AWS EFS CSI Driver Operator は、デフォルトでは Red Hat OpenShift Service on AWS にインストールされません。以下の手順を使用して、クラスター内で AWS EFS CSI ドライバー Operator をインストールおよび設定します。

前提条件

- Red Hat OpenShift Service on AWS Web コンソールにアクセスできる。

手順

Web コンソールから AWS EFS CSI ドライバー Operator をインストールするには、以下を実行します。

1. Web コンソールにログインします。
2. AWS EFS CSI Operator をインストールします。
   a. **Operators → OperatorHub** をクリックします。
   b. フィルターボックスに**AWS EFS CSI**と入力して、AWS EFS CSI Operator を探します。
   c. **AWS EFS CSI Driver Operator**ボタンをクリックします。

重要

AWS EFS Operatorではなく**AWS EFS CSI Driver Operator**を必ず選択してください。**AWS EFS Operator**はコミュニティー Operator であり、Red Hat ではサポートしていません。

   d. **AWS EFS CSI Driver Operator**ページで**Install**をクリックします。
   e. **Install Operator**のページで、以下のことを確認してください。
      - **All namespaces on the cluster (default)**が選択されている。
      - **Installed Namespace**が**openshift-cluster-csi-drivers**に設定されている。
   f. **Install** をクリックします。
      インストールが終了すると、AWS EFS CSI Operator が Web コンソールの**Installed Operators**に表示されます。
3. AWS EFS CSI ドライバーをインストールします。
   a. **administration → CustomResourceDefinitions → ClusterCSIDriver**をクリックします。
   b. **Instances**タブで**Create ClusterCSIDriver**をクリックします。
   c. 以下の YAML ファイルを使用します。

```
apiVersion: operator.openshift.io/v1
kind: ClusterCSIDriver
metadata:
    name: efs.csi.aws.com
spec:
  managementState: Managed
```

   d. **Create** をクリックします。

e. 以下の条件が "true"に変わるのを待ちます。

- AWSEFSDriverCredentialsRequestControllerAvailable
- AWSEFSDriverNodeServiceControllerAvailable
- AWSEFSDriverControllerServiceControllerAvailable

関連情報

- AWS EFS CSI ドライバーと STS の設定

1.3.4. AWS EFS CSI ドライバー Operator と Secure Token Service の設定

この手順では、AWS Secure Token Service (STS) で Red Hat OpenShift Service on AWS を使用して AWS EFS CSI Driver Operator を設定する方法を説明します。

AWS EFS CSI Operator のインストール後に、**AWS EFS CSI ドライバー Operator のインストール**手順の一部として AWS EFS CSI ドライバーをインストールする前に、以下の手順を実行します。ドライバーのインストールおよびボリュームの作成後にこの手順を実行すると、ボリュームの Pod へのマウントに失敗します。

前提条件

- AWS アカウントの認証情報

手順

AWS EFS CSI ドライバー Operator と STS を設定するには、以下を実行します。

1. STS でクラスターをインストールするために使用した Red Hat OpenShift Service on AWS リリースイメージから、CCO ユーティリティー (**ccoctl**) バイナリーを展開します。詳細は、Cloud Credential Operator ユーティリティーの設定を参照してください。
2. 以下の例に示されているように EFS **CredentialsRequest** YAML ファイルを作成および保存してから、それを **credrequests** ディレクトリーに配置します。

   例

```
apiVersion: cloudcredential.openshift.io/v1
kind: CredentialsRequest
metadata:
  name: openshift-aws-efs-csi-driver
  namespace: openshift-cloud-credential-operator
spec:
  providerSpec:
    apiVersion: cloudcredential.openshift.io/v1
    kind: AWSProviderSpec
    statementEntries:
    - action:
      - elasticfilesystem:*
      effect: Allow
      resource: '*'
  secretRef:
    name: aws-efs-cloud-credentials
    namespace: openshift-cluster-csi-drivers
```

```
serviceAccountNames:
- aws-efs-csi-driver-operator
- aws-efs-csi-driver-controller-sa
```

3. **ccoctl** ツールを実行して AWS に新規の IAM ロールを生成し、その YAML ファイルをローカルファイルシステムに作成します (**<path_to_ccoctl_output_dir>/manifests/openshift-cluster-csi-drivers-aws-efs-cloud-credentials-credentials.yaml**)。

```
$ ccoctl aws create-iam-roles --name=<name> --region=<aws_region> --credentials-requests-dir=<path_to_directory_with_list_of_credentials_requests>/credrequests --identity-provider-arn=arn:aws:iam::<aws_account_id>:oidc-provider/<name>-oidc.s3.<aws_region>.amazonaws.com
```

- **name=<name>** は、追跡用に作成されたクラウドリソースにタグを付けるために使用される名前です。
- **region=<aws_region>** は、クラウドリソースが作成される AWS リージョンです。
- **dir=<path_to_directory_with_list_of_credentials_requests>/credrequests** は、前のステップの EFS CredentialsRequest ファイルが含まれるディレクトリーです。
- **<aws_account_id>** は AWS アカウント ID です。

例

```
$ ccoctl aws create-iam-roles --name my-aws-efs --credentials-requests-dir credrequests --identity-provider-arn arn:aws:iam::123456789012:oidc-provider/my-aws-efs-oidc.s3.us-east-2.amazonaws.com
```

出力例

```
2022/03/21 06:24:44 Role arn:aws:iam::123456789012:role/my-aws-efs -openshift-cluster-csi-drivers-aws-efs-cloud- created
2022/03/21 06:24:44 Saved credentials configuration to: /manifests/openshift-cluster-csi-drivers-aws-efs-cloud-credentials-credentials.yaml
2022/03/21 06:24:45 Updated Role policy for Role my-aws-efs-openshift-cluster-csi-drivers-aws-efs-cloud-
```

4. AWS EFS クラウド認証情報およびシークレットを作成します。

```
$ oc create -f <path_to_ccoctl_output_dir>/manifests/openshift-cluster-csi-drivers-aws-efs-cloud-credentials-credentials.yaml
```

例

```
$ oc create -f /manifests/openshift-cluster-csi-drivers-aws-efs-cloud-credentials-credentials.yaml
```

出力例

```
secret/aws-efs-cloud-credentials created
```

関連情報

- AWS EFS CSI ドライバー Operator のインストール
- Cloud Credential Operator ユーティリティーの設定

1.3.5. AWS EFS ストレージクラスの作成

ストレージクラスを使用すると、ストレージのレベルや使用状況を区別し、記述することができます。ストレージクラスを定義することにより、ユーザーは動的にプロビジョニングされた永続ボリュームを取得できます。

AWS EFS CSI ドライバー Operatorをインストールしても、ストレージクラスがデフォルトで作成されません。ただし、AWS EFS ストレージクラスを手動で作成することは可能です。

1.3.6. AWS における EFS ボリュームへのアクセスの作成と設定

この手順では、Red Hat OpenShift Service on AWS で使用できるように、AWS で EFS ボリュームを作成および設定する方法を説明します。

前提条件

- AWS アカウントの認証情報

手順

AWS で EFS ボリュームへのアクセスを作成および設定するには、以下の手順を実施します。

1. AWS のコンソールで、https://console.aws.amazon.com/efs を開きます。
2. **Create file system**をクリックします。
   - ファイルシステムの名前を入力します。
   - **Virtual Private Cloud (VPC)**の場合、Red Hat OpenShift Service on AWS の仮想プライベートクラウド (VPC) を選択します。
   - その他の選択項目については、デフォルト設定を受け入れます。
3. ボリュームとマウントターゲットが完全に作成され終わるのを待ちます。
   a. https://console.aws.amazon.com/efs#/file-systems にアクセスしてください。
   b. ボリュームをクリックし、**Network**タブで、すべてのマウントターゲットが利用可能になるまで待ちます (最長で 1 - 2 分間)。
4. **Network**タブでセキュリティーグループ ID をコピーします (次のステップで必要になります)。
5. https://console.aws.amazon.com/ec2/v2/home#SecurityGroups にアクセスし、EFS ボリュームで使用されているセキュリティーグループを探します。
6. **Inbound rules** タブで **Edit inbound rules** をクリックし、次の設定で新しいルールを追加して、Red Hat OpenShift Service on AWS ノードが EFS ボリュームにアクセスできるようにします。
   - **Type**: NFS
   - **Protocol**: TCP

- **Port range**: 2049
- **Source**: ノードのカスタム/IP アドレス範囲 (例:"10.0.0.0/16")
  この手順により、Red Hat OpenShift Service on AWS がクラスターから NFS ポートを使用できるようになります。

7. ルールを保存します。

1.3.7. AWS EFS の動的プロビジョニング

AWS EFS CSI ドライバーは、他の CSI ドライバーとは異なる形態の動的プロビジョニングをサポートしています。既存の EFS ボリュームのサブディレクトリーとして新しい PV をプロビジョニングします。PV はお互いに独立しています。しかし、これらはすべて同じ EFS ボリュームを共有しています。ボリュームが削除されると、そのボリュームからプロビジョニングされたすべての PV も削除されます。EFS CSI ドライバーは、そのようなサブディレクトリーごとに AWS アクセスポイントを作成します。AWS アクセスポイントの制限により、1 つの **StorageClass**/EFS ボリュームから動的にプロビジョニングできる PV は 120 までとなります。

重要

なお、**PVC.spec.resources** は EFS では強制されません。

以下の例では、5GiB の容量を要求しています。しかし、作成された PV は無限であり、どんな量のデータ (ペタバイトのような) も保存することができます。ボリュームに大量のデータを保存してしまうと、壊れたアプリケーション、あるいは不正なアプリケーションにより、多額の費用が発生します。

AWS の EFS ボリュームサイズのモニタリングを使用することを強く推奨します。

前提条件

- AWS EFS ボリュームを作成している。
- AWS EFS ストレージクラスを作成している。

手順

動的プロビジョニングを有効にするには、以下の手順を実施します。

- 上記で作成した **StorageClass** を参照して、通常通り PVC(または StatefulSet や Template) を作成します。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: test
spec:
  storageClassName: efs-sc
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 5Gi
```

動的プロビジョニングのセットアップに問題がある場合は、AWS EFS のトラブルシューティング を参照してください。

1.3.8. AWS EFS による静的 PV の作成

動的プロビジョニングを行わずに、単一の PV として AWS EFS ボリュームを使用することが可能です。ボリューム全体が Pod にマウントされます。

前提条件

- AWS EFS ボリュームを作成している。

手順

- 以下の YAML ファイルで PV を作成します。

```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: efs-pv
spec:
  capacity: 1
    storage: 5Gi
  volumeMode: Filesystem
  accessModes:
    - ReadWriteMany
    - ReadWriteOnce
  persistentVolumeReclaimPolicy: Retain
  csi:
    driver: efs.csi.aws.com
    volumeHandle: fs-ae66151a 2
    volumeAttributes:
      encryptInTransit: "false" 3
```

1. **spec.capacity** には意味がなく、CSI ドライバーでは無視されます。PVC へのバインディング時にのみ使用されます。アプリケーションは、ボリュームに任意の量のデータを保存することができます。
2. **volumeHandle** は、AWS で作成した EFS ボリュームと同じ ID である必要があります。独自のアクセスポイントを提供する場合、**volumeHandle** は **<EFS volume ID>::<access point ID>** とします。例:**fs-6e633ada::fsap-081a1d293f0004630**
3. 必要に応じて、転送中の暗号化を無効にすることができます。デフォルトでは、暗号化が有効になっています。

静的 PV の設定に問題がある場合は、AWS EFS のトラブルシューティング を参照してください。

1.3.9. AWS EFS のセキュリティー

以下の情報は、AWS EFS のセキュリティーにとって重要です。

前述の動的プロビジョニングなどでアクセスポイントを使用する場合、Amazon はファイルの GID をアクセスポイントの GID に自動的に置き換えます。また、EFS では、ファイルシステムの権限を評価する際に、アクセスポイントのユーザー ID、グループ ID、セカンダリーグループ ID を考慮します。EFS は、NFS クライアントの ID を無視します。アクセスポイントの詳細については、https://docs.aws.amazon.com/efs/latest/ug/efs-access-points.html を参照してください。

結果として、EFS ボリュームは暗黙のうちに FSGroup を無視します。Red Hat OpenShift Service on AWS は、ボリューム上のファイルの GID を FSGroup に置き換えることができません。マウントされた EFS アクセスポイントにアクセスできる Pod は、そこにあるすべてのファイルにアクセスできます。

これとは関係ありませんが、転送中の暗号化はデフォルトで有効になっています。詳しくは、https://docs.aws.amazon.com/efs/latest/ug/encryption-in-transit.html を参照してください。

1.3.10. AWS EFS のトラブルシューティング

以下の情報は、AWS EFS の問題をトラブルシューティングするためのガイダンスです。

- AWS EFS Operator と CSI ドライバーは、namespace **openshift-cluster-csi-drivers** で実行されます。
- AWS EFS Operator と CSI ドライバーのログ収集を開始するには、以下のコマンドを実行します。

```
$ oc adm must-gather
[must-gather      ] OUT Using must-gather plugin-in image: quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:125f183d13601537ff15b3239df95d47f0a604da2847b561151fedd699f5e3a5
[must-gather      ] OUT namespace/openshift-must-gather-xm4wq created
[must-gather      ] OUT clusterrolebinding.rbac.authorization.k8s.io/must-gather-2bd8x created
[must-gather      ] OUT pod for plug-in image quay.io/openshift-release-dev/ocp-v4.0-art-dev@sha256:125f183d13601537ff15b3239df95d47f0a604da2847b561151fedd699f5e3a5 created
```

- AWS EFS Operator のエラーを表示するには、**ClusterCSIDriver** のステータスを表示します。

```
$ oc get clustercsidriver efs.csi.aws.com -o yaml
```

- Pod にボリュームをマウントできない場合 (以下のコマンドの出力に示す):

```
$ oc describe pod
...
  Type     Reason       Age    From               Message
  ----     ------       ----   ----               -------
  Normal   Scheduled    2m13s  default-scheduler  Successfully assigned default/efs-app to ip-10-0-135-94.ec2.internal
  Warning  FailedMount  13s    kubelet            MountVolume.SetUp failed for volume "pvc-d7c097e6-67ec-4fae-b968-7e7056796449" : rpc error: code = DeadlineExceeded desc = context deadline exceeded (1)
  Warning  FailedMount  10s    kubelet            Unable to attach or mount volumes: unmounted volumes=[persistent-storage], unattached volumes=[persistent-storage kube-api-access-9j477]: timed out waiting for the condition
```

(1) ボリュームがマウントされていないことを示す警告メッセージ。

このエラーは、AWS が Red Hat OpenShift Service on AWS ノードと AWS EFS の間でパケットをドロップすることで頻繁に発生します。

以下が正しいことを確認します。

- AWS のファイアウォールとセキュリティーグループ
- ネットワーク: ポート番号と IP アドレス

1.3.11. AWS EFS CSI ドライバー Operator のアンインストール

AWS EFS CSI ドライバー Operator をアンインストールすると、すべての EFS PV にアクセスできなくなる。

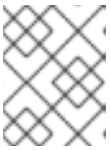
前提条件

- Red Hat OpenShift Service on AWS Web コンソールにアクセスできる。

手順

Web コンソールから AWS EFS CSI ドライバー Operator をアンインストールするには、以下を実行します。

1. Web コンソールにログインします。
2. AWS EFS PV を使用するすべてのアプリケーションを停止します。
3. すべての AWS EFS PV を削除します。
   a. **Storage → PersistentVolumeClaims** をクリックします。
   b. AWS EFS CSI ドライバー Operator が使用している各 PVC を選択し、PVC の右端にあるドロップダウンメニューをクリックして、**Delete PersistentVolumeClaims**をクリックします。
4. AWS EFS CSI ドライバーをアンインストールします。

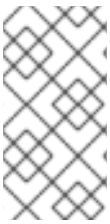


注記

Operator をアンインストールする前に、まず CSI ドライバーを削除する必要があります。

   a. **administration → CustomResourceDefinitions → ClusterCSIDriver**をクリックします。
   b. **Instances**タブの**efs.csi.aws.com**の左端にあるドロップダウンメニューをクリックし、**Delete ClusterCSIDriver**をクリックします。
   c. プロンプトが表示されたら、**Delete**をクリックします。
5. AWS EFS CSI Operator をアンインストールします。
   a. **Operators → Installed Operators** をクリックします。
   b. **Installed Operators**ページで、スクロールするか、**Search by name**ボックスに AWS EFS CSI と入力してオペレーターを見つけ、クリックします。
   c. **Installed Operators > Operator details**ページの右上にある**Actions → Uninstall Operator**をクリックします。

d. **Uninstall Operator** ウィンドウでプロンプトが表示されたら、**Uninstall** ボタンをクリックして namespace から Operator を削除します。Operator によってクラスターにデプロイされたアプリケーションは手動でクリーンアップする必要があります。
アンインストールすると、AWS EFS CSI ドライバー Operator が Web コンソールの **Installed Operators** セクションに一覧表示されなくなります。



注記

クラスターを破棄 (**openshift-install destroy cluster**) する前に、AWS の EFS ボリュームを削除する必要があります。クラスターの VPC を使用する EFS ボリュームがある場合は、Red Hat OpenShift Service on AWS クラスターを破棄できません。Amazon はこのような VPC の削除を許可していません。

1.3.12. 関連情報

- CSI ボリュームの設定