



Red Hat OpenShift Data Foundation

4.12

単一ノードの OpenShift クラスターでの論理ボリュームマネージャストレージのデプロイと管理

単一ノードの OpenShift クラスターで論理ボリュームマネージャストレージをデプロイおよび管理するための手順。

Red Hat OpenShift Data Foundation 4.12 単一ノードの OpenShift クラスターでの論理ボリュームマネージャストレージのデプロイと管理

単一ノードの OpenShift クラスターで論理ボリュームマネージャストレージをデプロイおよび管理するための手順。

法律上の通知

Copyright © 2023 Red Hat, Inc.

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

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

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

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

概要

単一ノードの OpenShift クラスターに論理ボリュームマネージャストレージをインストールする手順については、このドキュメントをお読みください。

目次

多様性を受け入れるオープンソースの強化	3
RED HAT ドキュメントへのフィードバック (英語のみ)	4
はじめに	5
第1章 RHACM を使用して単一ノードの OPENSIFT クラスターに論理ボリュームマネージャーストレージをデプロイする	6
1.1. RHACM を使用して論理ボリュームマネージャーストレージをデプロイするための要件	6
1.2. RHACM を使用した論理ボリュームマネージャーストレージのインストール	6
1.3. RHACM を使用してインストールされた論理ボリュームマネージャーストレージのアンインストール	9
第2章 OPENSIFT WEB コンソールを使用して、単一ノードの OPENSIFT クラスターに論理ボリュームマネージャーストレージをデプロイする	16
2.1. OPENSIFT WEB コンソールを使用した論理ボリュームマネージャーストレージのインストール	16
2.2. 論理ボリュームマネージャークラスターの作成	16
2.3. OPENSIFT WEB コンソールを使用してインストールされた論理ボリュームマネージャーストレージのアンインストール	18
第3章 論理ボリュームマネージャーストレージを使用したストレージのプロビジョニング	20
第4章 論理ボリュームマネージャーストレージのモニタリング	22
第5章 単一ノード OPENSIFT クラスターのストレージのスケールアップ	24
5.1. RHACM を使用して単一ノードの OPENSIFT クラスターに容量を追加することによるストレージのスケールアップ	24
5.2. 単一ノードの OPENSIFT クラスターに容量を追加することによるストレージのスケールアップ	27
第6章 単一ノード OPENSIFT クラスターでの論理ボリュームマネージャーストレージのアップグレード ...	29
第7章 シングルノード OPENSIFT のボリュームスナップショット	30
7.1. シングルノード OPENSIFT でのボリュームスナップショットの作成	30
7.2. シングルノード OPENSIFT でのボリュームスナップショットの復元	30
7.3. シングルノード OPENSIFT でのボリュームスナップショットの削除	31
第8章 シングルノード OPENSIFT のボリュームクローン作成	33
8.1. シングルノード OPENSIFT でのボリュームクローンの作成	33
8.2. シングルノード OPENSIFT でクローンボリュームの削除	33
第9章 MUST-GATHER を使用したログファイルおよび診断情報のダウンロード	34
第10章 参照資料	35

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があれば、ぜひお知らせください。

フィードバックを送信するには、Bugzilla チケットを作成します。

1. [Bugzilla](#) の Web サイトに移動します。
2. **Component** セクションで、**documentation** を選択します。
3. **Description** フィールドに、ドキュメントの改善に向けたご提案を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

はじめに

論理ボリュームマネージャストレージは、TopoLVM CSI ドライバーを使用して、単一ノードの OpenShift (SNO) クラスタでローカルストレージを動的にプロビジョニングします。

論理ボリュームマネージャストレージは、論理ボリュームマネージャを使用してシンプロビジョニングボリュームを作成し、リソースが制限された単一ノード SNO クラスタでブロックストレージの動的プロビジョニングを提供します。

単一ノードの Openshift ベアメタルまたはユーザープロビジョニングインフラストラクチャクラスターに論理ボリュームマネージャストレージをデプロイし、ワークロード用にストレージを動的にプロビジョニングするように設定できます。

論理ボリュームマネージャストレージは、使用可能なすべての未使用ディスクを使用してボリュームグループを作成し、ボリュームグループの 90% のサイズを持つ単一のシンプールを作成します。ボリュームグループの残りの 10% は、必要に応じてシンプールを拡張することにより、データ回復を可能にするために自由に使用できます。このようなりカバリーは手動で実行する必要がある場合があります。

永続ボリュームクレーム (PVC) と論理ボリュームマネージャストレージによってプロビジョニングされたボリュームスナップショットを使用して、ストレージをリクエストし、ボリュームスナップショットを作成できます。

論理ボリュームマネージャストレージは、シンプロビジョニング機能を利用するために、デフォルトのオーバープロビジョニング制限を 10 に設定します。シングルノード OpenShift クラスタで作成できるボリュームとボリュームスナップショットの合計サイズは、シンプールのサイズの 10 倍です。

次のいずれかを使用して、単一ノードの OpenShift クラスタに論理ボリュームマネージャストレージをデプロイできます。

- Red Hat Advanced Cluster Management for Kubernetes (RHACM)
- OpenShift Web コンソール

第1章 RHACM を使用して単一ノードの OPENSIFT クラスターに論理ボリュームマネージャーストレージをデプロイする

1.1. RHACM を使用して論理ボリュームマネージャーストレージをデプロイするための要件

単一ノード Openshift (SNO) クラスターに論理ボリュームマネージャーストレージのデプロイを開始する前に、次の要件が満たされていることを確認してください。

1. OpenShift クラスターに Red Hat Advanced Cluster Management for Kubernetes (RHACM) がインストールされている。詳細は [Red Hat Advanced Cluster Management for Kubernetes: インストール](#) を参照してください。
2. すべてのマネージド SNO クラスターには、ストレージのプロビジョニングに使用される専用のディスクがあります。

単一ノード Openshift (SNO) クラスターに論理ボリュームマネージャーストレージをデプロイする前に、次の制限事項に注意してください。

1. OpenShift Container Platform クラスターで作成できる LVMCluster のインスタンスは1つだけです。
2. LVMCluster で作成できる deviceClass エントリは1つだけです。
3. デバイスが LVMCluster の一部になると、削除できなくなります。

1.2. RHACM を使用した論理ボリュームマネージャーストレージのインストール

論理ボリュームマネージャーストレージは、Red Hat Advanced Cluster Management for Kubernetes (RHACM) を使用して、単一ノードの OpenShift (SNO) クラスターにデプロイされます。RHACM に Policy を作成して、**PlacementRule** で指定したセクターに一致するマネージドクラスターに適用される場合に Operator をデプロイし、設定します。このポリシーは、後にインポートされ、**PlacementRule** を満たすクラスターにも適用されます。

前提条件

- **cluster-admin** および Operator のインストールパーミッションを持つアカウントを使用して RHACM クラスターにアクセスできる。
- 論理ボリュームマネージャーストレージによって使用される各 SNO クラスターの専用ディスク。
- SNO クラスターは、インポートまたは作成されたものにかかわらず、RHACM によって管理される必要があります。

手順

1. OpenShift 認証情報を使用して RHACM CLI にログインします。
詳細は、[Red Hat Advanced Cluster Management for Kubernetes のインストール](#) を参照してください。
2. ポリシーを作成する namespace を作成します。

```
# oc create ns lvms-policy-ns
```

3. 次の YAML を **policy-lvms-operator.yaml** などの名前でファイルに保存して、ポリシーを作成します。

- ボリュームグループを優先ディスクに制御または制限するには、**LVMCluster** YAML の **deviceSelector** セクションでディスクのローカルパスを手動で指定します。
- **PlacementRule.spec.clusterSelector** のキーと値を置き換えて、論理ボリュームマネージャストレージをインストールする SNO クラスターに設定されたラベルと一致させます。
- OpenShift Container Platform は、ユーザーがプロビジョニングしたベアメタルインフラストラクチャー上の単一ノード OpenShift クラスターの追加のワーカーノードをサポートします。詳細は、[単一ノードの OpenShift クラスターのワーカーノード](#) を参照してください。

論理ボリュームマネージャストレージは、新しいノードが表示されると、新しい追加のワーカーノードを検出して使用します。追加のワーカーノードのサブセットであるノードフィルターを追加するには、**nodeSelector** セクションに必要なフィルターを指定します。このノードフィルターの一致は、Pod ラベルの一致と同じではないことに注意してください。

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-install-lvms
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    - key: mykey
      operator: In
      values:
      - myvalue
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-install-lvms
placementRef:
  apiGroup: apps.open-cluster-management.io
  kind: PlacementRule
  name: placement-install-lvms
subjects:
- apiGroup: policy.open-cluster-management.io
  kind: Policy
  name: install-lvms
---
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  annotations:
    policy.open-cluster-management.io/categories: CM Configuration Management
    policy.open-cluster-management.io/controls: CM-2 Baseline Configuration
```

```
  policy.open-cluster-management.io/standards: NIST SP 800-53
  name: install-lvms
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
    apiVersion: policy.open-cluster-management.io/v1
    kind: ConfigurationPolicy
    metadata:
      name: install-lvms
    spec:
      object-templates:
      - complianceType: musthave
        objectDefinition:
          apiVersion: v1
          kind: Namespace
          metadata:
            labels:
              openshift.io/cluster-monitoring: "true"
              pod-security.kubernetes.io/enforce: privileged
              pod-security.kubernetes.io/audit: privileged
              pod-security.kubernetes.io/warn: privileged
            name: openshift-storage
        - complianceType: musthave
          objectDefinition:
            apiVersion: operators.coreos.com/v1
            kind: OperatorGroup
            metadata:
              name: openshift-storage-operatorgroup
              namespace: openshift-storage
            spec:
              targetNamespaces:
              - openshift-storage
        - complianceType: musthave
          objectDefinition:
            apiVersion: operators.coreos.com/v1alpha1
            kind: Subscription
            metadata:
              name: lvms
              namespace: openshift-storage
            spec:
              installPlanApproval: Automatic
              name: lvms-operator
              source: redhat-operators
              sourceNamespace: openshift-marketplace
            remediationAction: enforce
            severity: low
        - objectDefinition:
          apiVersion: policy.open-cluster-management.io/v1
          kind: ConfigurationPolicy
          metadata:
            name: lvms
          spec:
            object-templates:
            - complianceType: musthave
```

```

objectDefinition:
  apiVersion: lvm.topolvm.io/v1alpha1
  kind: LVMCluster
  metadata:
    name: my-lvmcluster
    namespace: openshift-storage
  spec:
    storage:
      deviceClasses:
        - name: vg1
      deviceSelector:
        paths:
          - /dev/disk/by-path/pci-0000:87:00.0-nvme-1
          - /dev/disk/by-path/pci-0000:88:00.0-nvme-1
      thinPoolConfig:
        name: thin-pool-1
        sizePercent: 90
        overprovisionRatio: 10
      nodeSelector:
        nodeSelectorTerms:
          - matchExpressions:
              - key: app
                operator: In
                values:
                  - test1
      remediationAction: enforce
      severity: low

```

さまざまなフィールドの説明については、[リファレンス](#) を参照してください。

4. 次のコマンドを実行して、namespace にポリシーを作成します。

```
# oc create -f policy-lvms-operator.yaml -n lvms-policy-ns
```

ここで、**policy-lvms-operator.yaml** は、ポリシーが保存されるファイルの名前です。

これにより、namespace **lvms-policy-ns** に **Policy**、**PlacementRule**、および **PlacementBinding** が作成されます。**Policy** は、PlacementRule に一致するクラスター上に **Namespace**、**OperatorGroup**、**Subscription**、および **LVMCluster** リソースを作成します。これにより、選択基準に一致する SNO クラスターで Operator をデプロイし、ストレージをプロビジョニングするのに必要なリソースを設定するように設定します。operator は、**LVMCluster** で指定されたすべてのディスクを使用します。ディスクが指定されていない場合、operator は SNO ノード上のすべての未使用ディスクを使用します。デバイスが LVMCluster に追加された後は、削除できないことに注意してください。

1.3. RHACM を使用してインストールされた論理ボリュームマネージャーストレージのアンインストール

RHACM を使用して operator をインストールしたときに論理ボリュームマネージャーストレージをアンインストールするには、operator のデプロイと設定のために作成した ACM ポリシーを削除する必要があります。ただし、ACM ポリシーを削除しても、ポリシーが作成したリソースは削除されません。リソースを削除する追加のポリシーを作成する必要があります。

ポリシーを削除しても作成されたリソースは削除されないため、次の手順を実行する必要があります。

- 論理ボリュームマネージャーストレージによってプロビジョニングされたすべての PVC およびボリュームスナップショットを削除します。
- **LVMCluster** リソースを削除して、ディスク上に作成された Logical Volume Manager リソースをクリーンアップします。
- operator をアンインストールするための追加のポリシーを作成します。

前提条件

- ポリシーを削除する前に、以下が削除されていることを確認してください。
 - 論理ボリュームマネージャーストレージによってプロビジョニングされたストレージを使用している、管理対象クラスター上のすべてのアプリケーション。
 - 論理ボリュームマネージャーストレージを使用してプロビジョニングされる永続ボリューム要求 (PVC) および永続ボリューム (PV)。
 - 論理ボリュームマネージャーストレージによってプロビジョニングされたすべてのボリュームスナップショット。
- **cluster-admin** ロールを持つアカウントを使用した RHACM クラスターへのアクセス。

手順

1. OpenShift コマンドラインインターフェイスで、次のコマンドを使用して、ハブクラスターに論理ボリュームマネージャーストレージをデプロイおよび設定するために作成した ACM ポリシーを削除します。

```
# oc delete -f policy-lvms-operator.yaml -n lvms-policy-ns
```

2. 次の YAML を **lvms-remove-policy.yaml** などの名前でファイルに保存して、**LVMCluster** を削除するためのポリシーを作成します。これにより、operator はクラスター上に作成したすべての Logical Volume Manager リソースをクリーンアップできます。**PlacementRule.spec.clusterSelector** の値を設定して、論理ボリュームマネージャーストレージをアンインストールするクラスターを選択します。

```
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-lvmcluster-delete
  annotations:
    policy.open-cluster-management.io/standards: NIST SP 800-53
    policy.open-cluster-management.io/categories: CM Configuration Management
    policy.open-cluster-management.io/controls: CM-2 Baseline Configuration
spec:
  remediationAction: enforce
  disabled: false
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: policy-lvmcluster-removal
      spec:
        remediationAction: enforce # the policy-template spec.remediationAction is overridden
```

```

by the preceding parameter value for spec.remediationAction.
severity: low
object-templates:
  - complianceType: mustnothave
    objectDefinition:
      kind: LVMCluster
      apiVersion: lvm.topolvm.io/v1alpha1
      metadata:
        name: my-lvmcluster
        namespace: openshift-storage # must have namespace 'openshift-storage'
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-policy-lvmcluster-delete
placementRef:
  apiGroup: apps.open-cluster-management.io
  kind: PlacementRule
  name: placement-policy-lvmcluster-delete
subjects:
  - apiGroup: policy.open-cluster-management.io
    kind: Policy
    name: policy-lvmcluster-delete
---
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-policy-lvmcluster-delete
spec:
  clusterConditions:
    - status: 'True'
      type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      - key: mykey
        operator: In
        values:
          - myvalue

```

さまざまなフィールドの説明については、[リファレンス](#) を参照してください。

3. 次のコマンドを実行してポリシーを作成します。

```
# oc create -f lvms-remove-policy.yaml -n lvms-policy-ns
```

4. 次の YAML を **check-lvms-remove-policy.yaml** などの名前でファイルに保存して、**LVMCluster** CR が削除されたかどうかを確認するポリシーを作成します。

```

apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  name: policy-lvmcluster-inform
annotations:
  policy.open-cluster-management.io/standards: NIST SP 800-53
  policy.open-cluster-management.io/categories: CM Configuration Management
  policy.open-cluster-management.io/controls: CM-2 Baseline Configuration

```

```

spec:
  remediationAction: inform
  disabled: false
  policy-templates:
    - objectDefinition:
        apiVersion: policy.open-cluster-management.io/v1
        kind: ConfigurationPolicy
        metadata:
          name: policy-lvmcluster-removal-inform
        spec:
          remediationAction: inform # the policy-template spec.remediationAction is overridden
          by the preceding parameter value for spec.remediationAction.
          severity: low
        object-templates:
          - complianceType: mustnothave
            objectDefinition:
              kind: LVMCluster
              apiVersion: lvm.topolvm.io/v1alpha1
              metadata:
                name: my-lvmcluster
                namespace: openshift-storage # must have namespace 'openshift-storage'
    ---
  apiVersion: policy.open-cluster-management.io/v1
  kind: PlacementBinding
  metadata:
    name: binding-policy-lvmcluster-check
  placementRef:
    apiGroup: apps.open-cluster-management.io
    kind: PlacementRule
    name: placement-policy-lvmcluster-check
  subjects:
    - apiGroup: policy.open-cluster-management.io
      kind: Policy
      name: policy-lvmcluster-inform
    ---
  apiVersion: apps.open-cluster-management.io/v1
  kind: PlacementRule
  metadata:
    name: placement-policy-lvmcluster-check
  spec:
    clusterConditions:
      - status: 'True'
        type: ManagedClusterConditionAvailable
    clusterSelector:
      matchExpressions:
        - key: mykey
          operator: In
          values:
            - myvalue

```

5. 次のコマンドを実行してポリシーを作成します。

```
# oc create -f check-lvms-remove-policy.yaml -n lvms-policy-ns
```

6. ポリシーのステータスを確認します。

■


```
# oc get policy -n lvms-policy-ns
NAME                                REMEDIATION ACTION  COMPLIANCE STATE  AGE
policy-lvmcluster-delete            enforce             Compliant         15m
policy-lvmcluster-inform            inform              Compliant         15m
```

7. 両方のポリシーに準拠したら、次の YAML を **lvms-uninstall-policy.yaml** などの名前のファイルに保存して、論理ボリュームマネージャストレージをアンインストールするポリシーを作成します。

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-uninstall-lvms
spec:
  clusterConditions:
  - status: "True"
    type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
    - key: mykey
      operator: In
      values:
      - myvalue
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-uninstall-lvms
placementRef:
  apiGroup: apps.open-cluster-management.io
  kind: PlacementRule
  name: placement-uninstall-lvms
subjects:
- apiGroup: policy.open-cluster-management.io
  kind: Policy
  name: uninstall-lvms
---
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  annotations:
    policy.open-cluster-management.io/categories: CM Configuration Management
    policy.open-cluster-management.io/controls: CM-2 Baseline Configuration
    policy.open-cluster-management.io/standards: NIST SP 800-53
  name: uninstall-lvms
spec:
  disabled: false
  policy-templates:
  - objectDefinition:
      apiVersion: policy.open-cluster-management.io/v1
      kind: ConfigurationPolicy
      metadata:
        name: uninstall-lvms
      spec:
        object-templates:
        - complianceType: mustnothave
```

```

objectDefinition:
  apiVersion: v1
  kind: Namespace
  metadata:
    name: openshift-storage
- complianceType: mustnothave
objectDefinition:
  apiVersion: operators.coreos.com/v1
  kind: OperatorGroup
  metadata:
    name: openshift-storage-operatorgroup
    namespace: openshift-storage
  spec:
    targetNamespaces:
      - openshift-storage
- complianceType: mustnothave
objectDefinition:
  apiVersion: operators.coreos.com/v1alpha1
  kind: Subscription
  metadata:
    name: lvms-operator
    namespace: openshift-storage
  remediationAction: enforce
  severity: low
- objectDefinition:
  apiVersion: policy.open-cluster-management.io/v1
  kind: ConfigurationPolicy
  metadata:
    name: policy-remove-lvms-crds
  spec:
    object-templates:
      - complianceType: mustnothave
        objectDefinition:
          apiVersion: apiextensions.k8s.io/v1
          kind: CustomResourceDefinition
          metadata:
            name: logicalvolumes.topolvm.io
      - complianceType: mustnothave
        objectDefinition:
          apiVersion: apiextensions.k8s.io/v1
          kind: CustomResourceDefinition
          metadata:
            name: lvmlusters.lvm.topolvm.io
      - complianceType: mustnothave
        objectDefinition:
          apiVersion: apiextensions.k8s.io/v1
          kind: CustomResourceDefinition
          metadata:
            name: lvmvolumegroupnodestatuses.lvm.topolvm.io
      - complianceType: mustnothave
        objectDefinition:
          apiVersion: apiextensions.k8s.io/v1
          kind: CustomResourceDefinition
          metadata:

```

```
name: lvmvolumegroups.lvm.topolvm.io
remediationAction: enforce
severity: high
```

8. 次のコマンドを実行してポリシーを作成します。

```
# oc create -f lvms-uninstall-policy.yaml -ns lvms-policy-ns
```

第2章 OPENSIFT WEB コンソールを使用して、単一ノードの OPENSIFT クラスターに論理ボリュームマネージャーストレージをデプロイする

2.1. OPENSIFT WEB コンソールを使用した論理ボリュームマネージャーストレージのインストール

Red Hat OpenShift Container Platform Operator Hub を使用して論理ボリュームマネージャーストレージをインストールできます。

前提条件

- **cluster-admin** および Operator インストール権限を持つアカウントを使用して、OpenShift Container Platform 単一ノード OpenShift (SNO) クラスターにアクセスします。

手順

1. OpenShift Web コンソールにログインします。
2. **Operators** → **OperatorHub** をクリックします。
3. **LVM Storage** をスクロールするか、**Filter by keyword** ボックスに入力して、論理ボリュームマネージャーストレージを見つけます。
4. **Install** をクリックします。
5. **Install Operator** ページで、以下のオプションを設定します。
 - a. Channel を **stable-4.12** として更新します。
 - b. Installation Mode オプションに **A specific namespace on the cluster** を選択します。
 - c. Installed Namespace に **Operator recommended namespace openshift-storage** を選択します。namespace **openshift-storage** が存在しない場合、これは Operator のインストール時に作成されます。
 - d. 承認ストラテジーを **Automatic** または **Manual** として選択します。
Automatic (自動) 更新を選択した場合、Operator Lifecycle Manager (OLM) は介入なしに、Operator の実行中のインスタンスを自動的にアップグレードします。

Manual 更新を選択した場合、OLM は更新要求を作成します。クラスター管理者は、Operator を新しいバージョンに更新できるように更新要求を手動で承認する必要があります。
 - e. **Install** をクリックします。

検証手順

- 論理ボリュームマネージャーストレージに、インストールの成功を示す緑色のチェックマークが表示されていることを確認します。

2.2. 論理ボリュームマネージャークラスターの作成

論理ボリュームマネージャストレージをインストールした後、論理ボリュームマネージャクラスターを作成します。

- OpenShift Container Platform は、ユーザーがプロビジョニングしたベアメタルインフラストラクチャー上の単一ノード OpenShift クラスタの追加のワーカーノードをサポートします。詳細は、[単一ノードの OpenShift クラスタのワーカーノード](#) を参照してください。論理ボリュームマネージャストレージは、新しいノードが表示されると、新しい追加のワーカーノードを検出して使用します。追加のワーカーノードにノードフィルターを設定する必要がある場合は、クラスタの作成中に YAML ビューを使用できます。このノードフィルターの一致は、Pod ラベルの一致と同じではないことに注意してください。

前提条件

- 論理ボリュームマネージャストレージは、operator ハブからインストールする必要があります。

手順

1. OpenShift Web コンソールで、**Operators → Installed Operators** をクリックし、インストールされた Operator を表示します。
選択された **Project** が **openshift-storage** であることを確認します。
2. **LVM Storage** をクリックし、LVMCluster の下の **Create LVMCluster** をクリックします。
3. LVMCluster の作成ページで、**Form view** または **YAML view** のいずれかを選択します。
4. クラスタの名前を入力します。
5. **Create** をクリックします。
6. (オプション) ノードフィルターを追加するには、**YAML view** をクリックし、**nodeSelector** セクションでフィルターを指定します。

```
apiVersion: lvm.topolvm.io/v1alpha1
kind: LVMCluster
metadata:
  name: my-lvmcluster
spec:
  storage:
    deviceClasses:
      - name: vg1
        thinPoolConfig:
          name: thin-pool-1
          sizePercent: 90
          overprovisionRatio: 10
    nodeSelector:
      nodeSelectorTerms:
        - matchExpressions:
            - key: app
              operator: In
          Values:
            - test1
```

7. (オプション) ディスクのローカルデバイスパスを編集するには、YAML ビューをクリックし、**deviceSelector** セクションでデバイスパスを指定します。

```
spec:
  storage:
    deviceClasses:
      - name: vg1
        deviceSelector:
          paths:
            - /dev/disk/by-path/pci-0000:87:00.0-nvme-1
            - /dev/disk/by-path/pci-0000:88:00.0-nvme-1
        thinPoolConfig:
          name: thin-pool-1
          sizePercent: 90
          overprovisionRatio: 10
```

さまざまなフィールドの説明については、[リファレンス](#) を参照してください。

詳細は、[単一ノード OpenShift クラスターのストレージのスケーリング](#) を参照してください。

検証手順

1. OpenShift Web コンソールの左側のペインから **Storage → Storage Classes** をクリックします。
2. LVMCluster の作成で **lvms-<device-class-name>** ストレージクラスが作成されていることを確認します。デフォルトでは、**vg1** は device-class-name になります。

2.3. OPENSIFT WEB コンソールを使用してインストールされた論理ボリュームマネージャーストレージのアンインストール

前提条件

- ポリシーを削除する前に、以下が削除されていることを確認してください。
 - 論理ボリュームマネージャーストレージによってプロビジョニングされたストレージを使用している、クラスター上のすべてのアプリケーション。
 - 論理ボリュームマネージャーストレージを使用してプロビジョニングされる永続ボリューム要求 (PVC) および永続ボリューム (PV)。
 - 論理ボリュームマネージャーストレージによってプロビジョニングされたすべてのボリュームスナップショット。
- **oc get logicalvolume** コマンドを使用して、論理ボリュームリソースが存在しないことを確認します。
- **cluster-admin** パーミッションを持つアカウントを使用して、OpenShift Container Platform 単一ノード OpenShift (SNO) クラスターにアクセスします。

installed operators → lvm → lvmcluster タブ → 右端の 3 つのドットをクリック → lvm cluster の削除

手順

1. **Operators → Installed Operators** ページから、**LVM Storage** までスクロールするか、**Filter by name** に **LVM Storage** と入力して検索し、クリックします。
2. LVMCluster タブをクリックします。

3. LVMCluster ページの右側で、Actions ドロップダウンメニューから Delete LVMCluster を選択します。
4. Details タブをクリックします。
5. **Operator Details** ページの右側で、Actions ドロップダウンメニューから **Uninstall Operator** を選択します。
6. **Remove** を選択します。論理ボリュームマネージャストレージは実行を停止し、完全に削除されます。

第3章 論理ボリュームマネージャーストレージを使用したストレージのプロビジョニング

Operator のインストール中に作成されるストレージクラスを使用して、永続ボリュームクレーム (PVC) をプロビジョニングできます。ブロックおよびファイル PVC をプロビジョニングできますが、ストレージは、PVC を使用する Pod が作成された場合にのみ割り当てられます。



注記

論理ボリュームマネージャーストレージは、PVC を 1GiB 単位でプロビジョニングします。要求されたストレージは、最も近い GiB に切り上げられます。

手順

1. 論理ボリュームマネージャーストレージのデプロイ時に作成される StorageClass を特定します。
StorageClass 名の形式は **lvms-<device-class-name>** です。**device-class-name** は、ポリシー YAML の LVMCluster で指定したデバイスクラスの名前です。たとえば、deviceClass が **vg1** と呼ばれる場合、storageClass 名は **lvms-vg1** です。

ストレージクラスの **volumeBindingMode** は **WaitForFirstConsumer** に設定されます。

2. 次の YAML を **pvc.yaml** などの名前のファイルに保存して、アプリケーションがストレージを必要とする PVC を作成します。

```
# Sample YAML to create a PVC
# block pvc
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: lvm-block-1
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Block
  resources:
    requests:
      storage: 10Gi
  storageClassName: lvms-vg1
---
# file pvc
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: lvm-file-1
  namespace: default
spec:
  accessModes:
    - ReadWriteOnce
  volumeMode: Filesystem
  resources:
    requests:
      storage: 10Gi
  storageClassName: lvms-vg1
```


-
- 3. 以下のコマンドを実行して PVC を作成します。

```
# oc create -f pvc.yaml -ns <application namespace>
```

作成された PVC は、それらを使用する Pod をデプロイするまで **pending** 状態のままになります。

第4章 論理ボリュームマネージャーストレージのモニタリング

OpenShift Web コンソールを使用して論理ボリュームマネージャーストレージをインストールすると、デフォルトで、コンソールの **Block and File** ダッシュボードを使用してクラスターをモニターできます。ただし、RHACM を使用して論理ボリュームマネージャーストレージをインストールする場合は、RHACM Observability を設定して、すべての SNO クラスターを1か所からモニターする必要があります。

RHACM ダッシュボードで operator によってエクスポートされたメトリックとトリガーされたアラートを表示することで、論理ボリュームマネージャーストレージをモニターできます。[可観測性](#) ガイドの説明に従って、RHACM 可観測性を有効にします。

メトリクス

- [カスタムメトリクスの追加](#) セクションで指定されているように、次の **topolvm** メトリクスを許可リストに追加します。

```
topolvm_thinpool_data_percent
topolvm_thinpool_metadata_percent
topolvm_thinpool_size_bytes
```



注記

メトリクスは、10 分ごとに更新されるか、新しい論理ボリュームの作成など、シンプルに変更があったときに更新されます。

アラート

シンプルとボリュームグループがいっぱいになると、それ以降の操作は失敗し、データが失われる可能性があります。論理ボリュームマネージャーストレージは、シンプルとボリュームグループの使用量が特定の値を超えると、次のアラートを送信します。

表4.1 Red Hat Advanced Cluster Management for Kubernetes の Logical Volume Manager クラスターのアラート

アラート	説明
VolumeGroupUsageAtThresholdNearFull	このアラートは、ボリュームグループとシンプールの使用率の両方がノードで 75% を超えたときにトリガーされます。データの削除またはボリュームグループの拡張が必要です。
VolumeGroupUsageAtThresholdCritical	このアラートは、ボリュームグループとシンプールの両方の使用率がノードで 85% を超えたときにトリガーされます。VolumeGroup が非常にいっぱいです。データの削除またはボリュームグループの拡張が必要です。
ThinPoolDataUsageAtThresholdNearFull	このアラートは、ボリュームグループのシンプールのデータ使用率がノード上で 75% を超えるとトリガーされます。データの削除またはシンプールの拡張が必要です。

アラート	説明
ThinPoolDataUsageAtThresholdCritical	このアラートは、ボリュームグループのシンプルのデータ使用率がノード上で 85% を超えるとトリガーされます。データの削除またはシンプルの拡張が必要です。
ThinPoolMetaDataUsageAtThresholdNearFull	このアラートは、ボリュームグループのシンプルのメタデータ使用率がノードで 75% を超えると発生します。データの削除またはシンプルの拡張が必要です。
ThinPoolMetaDataUsageAtThresholdCritical	このアラートは、ボリュームグループのシンプルのメタデータ使用率がノード上で 85% を超えるとトリガーされます。データの削除またはシンプルの拡張が必要です。

第5章 単一ノード OPENSIFT クラスターのストレージのスケールリング

OpenShift Container Platform は、ユーザーがプロビジョニングしたベアメタルインフラストラクチャー上の単一ノード OpenShift クラスターの追加のワーカーノードをサポートします。詳細は、[単一ノードの OpenShift クラスターのワーカーノード](#) を参照してください。論理ボリュームマネージャーストレージは、新しいノードが表示されると、新しい追加のワーカーノードを検出して使用します。

単一ノード OpenShift クラスターで設定されたワーカーノードのストレージ容量をスケールリングするには、ディスクを追加して容量を増やすことができます。

5.1. RHACM を使用して単一ノードの OPENSIFT クラスターに容量を追加することによるストレージのスケールアップ

前提条件

- cluster-admin パーミッションのあるアカウントを使用した RHACM クラスターへのアクセス
- 論理ボリュームマネージャーストレージによって使用される、各 SNO クラスター上の追加の未使用ディスク。

手順

1. OpenShift 認証情報を使用して RHACM CLI にログインします。
詳細は、[Red Hat Advanced Cluster Management for Kubernetes のインストール](#) を参照してください。
2. 追加するディスクを見つめます。追加するディスクは、既存のディスクのデバイス名およびパスと一致するようにする必要があります。
3. 単一ノードの OpenShift クラスターに容量を追加するには、既存のポリシー YAML の **deviceSelector** セクション (**policy-lvms-operator.yaml** など) を編集します。



注記

LVMCluster の作成中に **deviceSelector** が含まれていない場合は、**deviceSelector** セクションを CR に追加することはできません。LVMCluster を削除してから、新しい CR から再作成する必要があります。

```
apiVersion: apps.open-cluster-management.io/v1
kind: PlacementRule
metadata:
  name: placement-install-lvms
spec:
  clusterConditions:
    - status: "True"
      type: ManagedClusterConditionAvailable
  clusterSelector:
    matchExpressions:
      - key: mykey
        operator: In
        values:
          - myvalue
```

```
---
apiVersion: policy.open-cluster-management.io/v1
kind: PlacementBinding
metadata:
  name: binding-install-lvms
placementRef:
  apiGroup: apps.open-cluster-management.io
  kind: PlacementRule
  name: placement-install-lvms
subjects:
- apiGroup: policy.open-cluster-management.io
  kind: Policy
  name: install-lvms
---
apiVersion: policy.open-cluster-management.io/v1
kind: Policy
metadata:
  annotations:
    policy.open-cluster-management.io/categories: CM Configuration Management
    policy.open-cluster-management.io/controls: CM-2 Baseline Configuration
    policy.open-cluster-management.io/standards: NIST SP 800-53
  name: install-lvms
spec:
  disabled: false
  remediationAction: enforce
  policy-templates:
  - objectDefinition:
    apiVersion: policy.open-cluster-management.io/v1
    kind: ConfigurationPolicy
    metadata:
      name: install-lvms
    spec:
      object-templates:
      - complianceType: musthave
        objectDefinition:
          apiVersion: v1
          kind: Namespace
          metadata:
            labels:
              openshift.io/cluster-monitoring: "true"
              pod-security.kubernetes.io/enforce: privileged
              pod-security.kubernetes.io/audit: privileged
              pod-security.kubernetes.io/warn: privileged
            name: openshift-storage
      - complianceType: musthave
        objectDefinition:
          apiVersion: operators.coreos.com/v1
          kind: OperatorGroup
          metadata:
            name: openshift-storage-operatorgroup
            namespace: openshift-storage
          spec:
            targetNamespaces:
            - openshift-storage
      - complianceType: musthave
        objectDefinition:
```

```

  apiVersion: operators.coreos.com/v1alpha1
  kind: Subscription
  metadata:
    name: lvms
    namespace: openshift-storage
  spec:
    installPlanApproval: Automatic
    name: lvms-operator
    source: redhat-operators
    sourceNamespace: openshift-marketplace
  remediationAction: enforce
  severity: low
- objectDefinition:
  apiVersion: policy.open-cluster-management.io/v1
  kind: ConfigurationPolicy
  metadata:
    name: lvms
  spec:
    object-templates:
      - complianceType: musthave
        objectDefinition:
          apiVersion: lvm.topolvm.io/v1alpha1
          kind: LVMCluster
          metadata:
            name: my-lvmcluster
            namespace: openshift-storage
          spec:
            storage:
              deviceClasses:
                - name: vg1
                  deviceSelector:
                    paths:
                      - /dev/disk/by-path/pci-0000:87:00.0-nvme-1
                      - /dev/disk/by-path/pci-0000:88:00.0-nvme-1
                      - /dev/disk/by-path/pci-0000:89:00.0-nvme-1 # new disk is added
            thinPoolConfig:
              name: thin-pool-1
              sizePercent: 90
              overprovisionRatio: 10
            nodeSelector:
              nodeSelectorTerms:
                - matchExpressions:
                    - key: app
                      operator: In
                      values:
                        - test1
            remediationAction: enforce
            severity: low

```

さまざまなフィールドの説明については、[リファレンス](#) を参照してください。

4. 以下のコマンドを実行してポリシーを編集します。

```
# oc edit -f policy-lvms-operator.yaml -ns lvms-policy-ns
```

ここで、**policy-lvms-operator.yaml** は既存のポリシーの名前です。

これは、**LVMCluster** で指定された新しいディスクを使用してストレージをプロビジョニングします。

5.2. 単一ノードの OPENSIFT クラスターに容量を追加することによるストレージのスケールアップ

前提条件

- 論理ボリュームマネージャストレージによって使用される、各 SNO クラスター上の追加の未使用ディスク。

手順

1. SNO クラスターの OpenShift コンソールにログインします。
2. **Operators** → **Installed Operators** ページで、**openshift-storage** namespace の **LVM Storage operator** をクリックします。
3. **LVMCluster** タブをクリックして、クラスターで作成された LVMCluster を一覧表示します。
4. Actions ドロップダウンメニューから **Edit LVMCluster** を選択します。
5. **YAML** タブをクリックします。
6. LVMCluster YAML を編集して、**deviceSelector** セクションに新しいデバイスパスを追加します。

```
[...]

apiVersion: lvm.topolvm.io/v1alpha1
kind: LVMCluster
metadata:
  name: my-lvmcluster
spec:
  storage:
    deviceClasses:
      - name: vg1
        deviceSelector:
          paths:
            - /dev/disk/by-path/pci-0000:87:00.0-nvme-1 # path can be added by name (/dev/sdb)
          or by path
            - /dev/disk/by-path/pci-0000:88:00.0-nvme-1
            - /dev/disk/by-path/pci-0000:89:00.0-nvme-1 # new disk is added
    thinPoolConfig:
      name: thin-pool-1
      sizePercent: 90
      overprovisionRatio: 10

[...]
```

さまざまなフィールドの説明については、[リファレンス](#) を参照してください。



注記

LVMCluster の作成中に **deviceSelector** が含まれていない場合は、**deviceSelector** セクションを CR に追加することはできません。LVMCluster を削除してから、新しい CR から再作成する必要があります。

第6章 単一ノード OPENSIFT クラスターでの論理ボリュームマネージャーストレージのアップグレード

現在、単一ノードの OpenShift クラスターで OpenShift Data Foundation Logical Volume Manager Operator 4.11 から論理ボリュームマネージャーストレージ 4.12 にアップグレードすることはできません。以下を実行する必要があります。

- PVC で保持する必要があるデータをバックアップします。
- OpenShift Data Foundation Logical Volume Manager Operator とその Pod によってプロビジョニングされたすべての PVC を削除します。
- OpenShift Container Platform 4.12 に論理ボリュームマネージャーストレージを再インストールします。
- ワークロードを再作成します。



注記

このプロセスではデータが保持されないため、必ずデータをバックアップし、4.12 へのアップグレード後に作成された PVC にコピーしてください。

第7章 シングルノード OPENSIFT のボリュームスナップショット

論理ボリュームマネージャーストレージによってプロビジョニングされた永続ボリューム (PV) のボリュームスナップショットを取得できます。複製されたボリュームのボリュームスナップショットを作成することもできます。ボリュームスナップショットは、次のことに役立ちます。

- アプリケーションデータをバックアップします (ボリュームスナップショットはバックアップではありません)
- ボリュームスナップショットが作成された状態に戻します

シンプルな使用可能な容量とオーバープロビジョニングの制限に基づいて、ボリュームスナップショットを作成できます。論理ボリュームマネージャーストレージは、`lvms-<deviceclass-name>` という名前の **VolumeSnapshotClass** を作成します。

7.1. シングルノード OPENSIFT でのボリュームスナップショットの作成

前提条件

- 一貫性のあるスナップショットを作成するには、PVC がバインド状態になっていることを確認してください。また、スナップショットを作成する前に、PVC へのすべての I/O が停止していることを確認してください。

手順

1. `oc` コマンドを実行する必要がある OpenShift 単一ノードクラスターにログインします。
2. 次の YAML を `lvms-vol-snapshot.yaml` などの名前で作成してファイルに保存します。

```
# Sample YAML to create a volume snapshot

apiVersion: snapshot.storage.k8s.io/v1
kind: VolumeSnapshot
metadata:
  name: lvm-block-1-snap
spec:
  volumeSnapshotClassName: lvms-vg1
  source:
    persistentVolumeClaimName: lvm-block-1
```

3. PVC と同じ namespace で次のコマンドを実行して、スナップショットを作成します。

```
# oc create -f lvms-vol-snapshot.yaml
```

PVC の読み取り専用コピーがボリュームスナップショットとして作成されます。

7.2. シングルノード OPENSIFT でのボリュームスナップショットの復元

ボリュームスナップショットを復元する際に、新規の Persistent Volume Claim (永続ボリューム要求、PVC) が作成されます。復元される PVC はボリュームスナップショットおよびソース PVC とは切り離されています。

前提条件

- ストレージクラスは、ソース PVC のストレージクラスと同じである必要がある。
- 要求された PVC のサイズは、スナップショットのソースボリュームのサイズと同じである必要がある。

手順

1. ソース PVC のストレージクラス名とボリュームスナップショット名を特定します。
2. 次の YAML を **lvms-vol-restore.yaml** などの名前ファイルに保存して、スナップショットを復元します。

```
# Sample YAML to restore a PVC.

kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: lvm-block-1-restore
spec:
  accessModes:
  - ReadWriteOnce
  volumeMode: Block
Resources:
  Requests:
    storage: 2Gi
storageClassName: lvms-vg1
dataSource:
  name: lvm-block-1-snap
  kind: VolumeSnapshot
  apiGroup: snapshot.storage.k8s.io
```

3. スナップショットと同じ namespace で次のコマンドを実行して、ポリシーを作成します。

```
# oc create -f lvms-vol-restore.yaml
```

7.3. シングルノード OPENSIFT でのボリュームスナップショットの削除

手順

- ボリュームスナップショットを削除するには、ボリュームスナップショットリソースを削除します。

```
# oc delete volumesnapshot <volume-snapshot-name> -n <namespace>
```



注記

永続ボリューム要求 (PVC) を削除しても、PVC のスナップショットは削除されません。

- 復元されたボリュームスナップショットを削除するには、ボリュームスナップショットを復元するために作成された PVC を削除します。

```
# oc delete pvc <pvc-name> -n <namespace>
```

第8章 シングルノード OPENSIFT のボリュームクローン作成

クローンは、既存のストレージボリュームの複製であり、他の標準ボリュームと同じように使用できます。ボリュームのクローンを作成し、データの特定の時点のコピーを作成します。永続ボリューム要求 (PVC) は別のサイズでクローンできません。

8.1. シングルノード OPENSIFT でのボリュームクローンの作成

前提条件

- ソース PVC が Bound 状態であり、使用中でないことを確認する。
- StorageClass がソース PVC のものと同じであることを確認する。

手順

1. ソース PVC のストレージクラスを特定します。
2. 次の YAML を **lvms-vol-clone.yaml** などの名前でファイルに保存して、ボリュームクローンを作成します。

```
# Sample YAML to clone a volume
# pvc-clone.yaml
apiVersion: v1
kind: PersistentVolumeClaim
Metadata:
  name: lvm-block-1-clone
Spec:
  storageClassName: lvms-vg1
  dataSource:
    name: lvm-block-1
    kind: PersistentVolumeClaim
  accessModes:
    - ReadWriteOnce
  volumeMode: Block
Resources:
  Requests:
    storage: 2Gi
The cloned PVC has write access.
```

3. ソース PVC と同じ ns で次のコマンドを実行して、ポリシーを作成します。

```
# oc create -f lvms-vol-clone.yaml
```

8.2. シングルノード OPENSIFT でクローンボリュームの削除

手順

- クローンボリュームを削除するには、クローン PVC を削除します。

```
# oc delete pvc <clone-pvc-name> -n <namespace>
```

第9章 MUST-GATHER を使用したログファイルおよび診断情報のダウンロード

論理ボリュームマネージャーストレージが問題を自動的に解決できない場合は、must-gather ツールを使用してログファイルと診断情報を収集し、ユーザーまたは Red Hat サポートが問題を確認して解決策を決定できるようにします。

- 論理ボリュームマネージャーストレージクラスターに接続されているクライアントから、must-gather コマンドを実行します。

```
$ oc adm must-gather --image=registry.redhat.io/odf4/ocs-must-gather-rhel8:v4.12 --dest-dir=<directory-name>
```

詳細は、[クラスターに関するデータの収集](#) を参照してください。

第10章 参照資料

すべてのフィールドを記述したサンプル LVMCluster YAML ファイル:

```

apiVersion: lvm.topolvm.io/v1alpha1
kind: LVMCluster
metadata:
  name: my-lvmcluster
spec:
  tolerations:
    - effect: NoSchedule
      key: xyz
      operator: Equal
      value: "true"
  storage:
    deviceClasses: # The lvm volume groups to be created on the cluster. Currently, only a single
deviceClass is supported.
    - name: vg1 # The name of the lvm volume group to be created on the nodes
      nodeSelector: # Determines the nodes on which to create the lvm volume group. If empty, all
nodes are considered.
        nodeSelectorTerms: #A list of node selector requirements
          - matchExpressions:
            - key: mykey
              operator: In
              values:
                - ssd
        deviceSelector: # A list of device paths which would be used to create the lvm volume group. If
this field is missing, all unused disks on the node will be used
          paths:
            - /dev/disk/by-path/pci-0000:87:00.0-nvme-1
            - /dev/disk/by-path/pci-0000:88:00.0-nvme-1
            - /dev/disk/by-path/pci-0000:89:00.0-nvme-1
        thinPoolConfig: # The lvm thin pool configuration
          name: thin-pool-1 # The name of the thinpool to be created in the lvm volume group
          sizePercent: 90 # The percentage of remaining space in the lvm volume group that should be
used for creating the thin pool.
        overprovisionRatio: 10 # The factor by which additional storage can be provisioned compared to
the available storage in the thin pool.
    status:
      deviceClassStatuses: #The status of the deviceClass
        - name: vg1
          nodeStatus: # The status of the lvm volume group on each node
        - devices: # The list of devices used to create the lvm volume group
          - /dev/nvme0n1
          - /dev/nvme1n1
          - /dev/nvme2n1
        node: my-node.example.com #Node on which the deviceClass has been created
        status: Ready # Status of the lvm volume group on this node
      ready: true # deprecated
      state: Ready # The status of the LVMCluster

```