



# Red Hat OpenShift Data Foundation 4.12

## Deploying and managing logical volume manager storage on single node OpenShift clusters

Instructions for deploying and managing logical volume manager storage on single node OpenShift clusters.



## Red Hat OpenShift Data Foundation 4.12 Deploying and managing logical volume manager storage on single node OpenShift clusters

---

Instructions for deploying and managing logical volume manager storage on single node OpenShift clusters.

## Legal Notice

Copyright © 2024 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## Abstract

Read this document for instructions regarding installing logical volume manager storage on single node OpenShift clusters.

## Table of Contents

<b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....	<b>3</b>
<b>PROVIDING FEEDBACK ON RED HAT DOCUMENTATION</b> .....	<b>4</b>
<b>PREFACE</b> .....	<b>5</b>
<b>CHAPTER 1. DEPLOYING LOGICAL VOLUME MANAGER STORAGE ON SINGLE NODE OPENSIFT CLUSTERS USING RHACM</b> .....	<b>6</b>
1.1. REQUIREMENTS FOR DEPLOYING LOGICAL VOLUME MANAGER STORAGE USING RHACM	6
1.2. INSTALLING LOGICAL VOLUME MANAGER STORAGE USING RHACM	6
1.3. UNINSTALLING LOGICAL VOLUME MANAGER STORAGE INSTALLED USING RHACM	9
<b>CHAPTER 2. DEPLOYING LOGICAL VOLUME MANAGER STORAGE ON SINGLE NODE OPENSIFT CLUSTERS USING OPENSIFT WEB CONSOLE</b> .....	<b>15</b>
2.1. INSTALLING LOGICAL VOLUME MANAGER STORAGE USING OPENSIFT WEB CONSOLE	15
2.2. CREATING LOGICAL VOLUME MANAGER CLUSTER	15
2.3. UNINSTALLING LOGICAL VOLUME MANAGER STORAGE INSTALLED USING THE OPENSIFT WEB CONSOLE	17
<b>CHAPTER 3. PROVISIONING STORAGE USING LOGICAL VOLUME MANAGER STORAGE</b> .....	<b>19</b>
<b>CHAPTER 4. MONITORING THE LOGICAL VOLUME MANAGER STORAGE</b> .....	<b>21</b>
<b>CHAPTER 5. SCALING STORAGE OF SINGLE NODE OPENSIFT CLUSTER</b> .....	<b>23</b>
5.1. SCALING UP STORAGE BY ADDING CAPACITY TO YOUR SINGLE NODE OPENSIFT CLUSTER USING RHACM	23
5.2. SCALING UP STORAGE BY ADDING CAPACITY TO YOUR SINGLE NODE OPENSIFT CLUSTER	26
<b>CHAPTER 6. UPGRADING LOGICAL VOLUME MANAGER STORAGE ON SINGLE NODE OPENSIFT CLUSTERS</b> .....	<b>27</b>
<b>CHAPTER 7. VOLUME SNAPSHOTS FOR SINGLE NODE OPENSIFT</b> .....	<b>28</b>
7.1. CREATING VOLUME SNAPSHOTS IN SINGLE NODE OPENSIFT	28
7.2. RESTORING VOLUME SNAPSHOTS IN SINGLE NODE OPENSIFT	28
7.3. DELETING VOLUME SNAPSHOTS IN SINGLE NODE OPENSIFT	29
<b>CHAPTER 8. VOLUME CLONING FOR SINGLE NODE OPENSIFT</b> .....	<b>30</b>
8.1. CREATING VOLUME CLONES IN SINGLE NODE OPENSIFT	30
8.2. DELETING CLONED VOLUMES IN SINGLE NODE OPENSIFT	30
<b>CHAPTER 9. DOWNLOADING LOG FILES AND DIAGNOSTIC INFORMATION USING MUST-GATHER</b> ...	<b>32</b>
<b>CHAPTER 10. REFERENCE</b> .....	<b>33</b>



## MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

## PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your input on our documentation. Do let us know how we can make it better.

To give feedback, create a Bugzilla ticket:

1. Go to the [Bugzilla](#) website.
2. In the **Component** section, choose **documentation**.
3. Fill in the **Description** field with your suggestion for improvement. Include a link to the relevant part(s) of documentation.
4. Click **Submit Bug**.



## PREFACE

The logical volume manager storage uses the TopoLVM CSI driver to dynamically provision local storage on single node OpenShift (SNO) clusters.

The logical volume manager storage creates thin-provisioned volumes using the Logical Volume Manager and provides dynamic provisioning of block storage on a single node, limited resources SNO cluster.

You can deploy the logical volume manager storage on a single node Openshift bare metal or user provisioned infrastructure cluster and configure it to dynamically provision storage for your workloads.

The logical volume manager storage creates a volume group using all the available unused disks and creates a single thin pool with a size of 90% of the volume group. The remaining 10% of the volume group is left free to enable data recovery by expanding the thin pool when required. You might need to manually perform such recovery.

You can use persistent volume claims (PVCs) and volume snapshots provisioned by the logical volume manager storage to request storage and create volume snapshots.

The logical volume manager storage configures a default overprovisioning limit of 10 to take advantage of the thin-provisioning feature. The total size of the volumes and volume snapshots that can be created on the single node OpenShift clusters is 10 times the size of the thin pool.

You can deploy logical volume manager storage on single node OpenShift clusters using one of the following:

- Red Hat Advanced Cluster Management for Kubernetes (RHACM)
- OpenShift Web Console

# CHAPTER 1. DEPLOYING LOGICAL VOLUME MANAGER STORAGE ON SINGLE NODE OPENSIFT CLUSTERS USING RHACM

## 1.1. REQUIREMENTS FOR DEPLOYING LOGICAL VOLUME MANAGER STORAGE USING RHACM

Before you begin deploying logical volume manager storage on single node Openshift (SNO) clusters, ensure that the following requirements are met:

1. You have installed Red Hat Advanced Cluster Management for Kubernetes (RHACM) on an OpenShift cluster. For information, see [Red Hat Advanced Cluster Management for Kubernetes: Install](#).
2. Every managed SNO cluster has dedicated disks that are used to provision storage.

Before you deploy logical volume manager storage on single node Openshift (SNO) clusters, be aware of the following limitations:

1. You can only create a single instance of the LVMCluster on an OpenShift Container Platform cluster
2. You can make only a single deviceClass entry in the LVMCluster.
3. When a device becomes part of the LVMCluster, it cannot be removed.

## 1.2. INSTALLING LOGICAL VOLUME MANAGER STORAGE USING RHACM

The logical volume manager storage is deployed on single node OpenShift (SNO) clusters using Red Hat Advanced Cluster Management for Kubernetes (RHACM). You create a Policy on RHACM that deploys and configures the operator when it is applied to managed clusters which match the selector specified in the **PlacementRule**. The policy is also applied to clusters that are imported later and satisfy the **PlacementRule**.

### Prerequisites

- Access to the RHACM cluster using an account with **cluster-admin** and operator installation permissions.
- Dedicated disks on each SNO cluster to be used by logical volume manager storage.
- The SNO cluster needs to be managed by the RHACM, either imported or created.

### Procedure

1. Log in to the RHACM CLI using your OpenShift credentials.  
For more information, see [Install Red Hat Advanced Cluster Management for Kubernetes](#) .
2. Create a namespace in which you will create policies.

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

3. Save the following YAML to a file with a name such as **policy-lvms-operator.yaml** to create a policy.
  - To control or restrict the volume group to your preferred disks, you can manually specify the local paths of the disks in the **deviceSelector** section of the **LVMCluster** YAML.
  - Replace the key and value in **PlacementRule.spec.clusterSelector** to match the labels set on the SNO clusters on which you want to install logical volume manager storage.
  - OpenShift Container Platform supports additional worker nodes for single node OpenShift clusters on bare metal user provisioned infrastructure. For more information, see [Worker nodes for single-node OpenShift clusters](#).

The logical volume manager storage detects and uses the new additional worker nodes when the new nodes show up. To add a node filter which is a subset of the additional worker nodes, specify the required filter in the **nodeSelector** section. Note that this node filter matching is not the same as the pod label matching.

```

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

```

For descriptions of different fields, see [Reference](#).

4. Create the policy in the namespace by running the following command:

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

where, ***policy-lvms-operator.yaml*** is the name of the file to which the policy is saved.

This creates a **Policy**, a **PlacementRule**, and a **PlacementBinding** in the namespace, **lvms-policy-ns**. The **Policy** creates a **Namespace**, **OperatorGroup**, **Subscription**, and **LVMCluster** resource on the clusters matching the PlacementRule. This deploys the operator on the SNO clusters which match the selection criteria and configures it to set up the required resources to provision storage. The operator uses all the disks specified in the **LVMCluster**. If no disks are specified, the operator uses all the unused disks on the SNO node. Note that after a device has been added to the LVMCluster, it cannot be removed.

### 1.3. UNINSTALLING LOGICAL VOLUME MANAGER STORAGE INSTALLED USING RHACM

To uninstall logical volume manager storage when you have installed the operator using RHACM, you need to delete the ACM policy that you created for deploying and configuring the operator. However, when you delete the ACM policy, the resources that the policy has created are not removed. You need to create additional policies to remove the resources.

As the resources that are created are not removed when you delete the policy, you need to perform the following steps:

- Remove all the PVCs and volume snapshots provisioned by the logical volume manager storage.
- Remove the **LVMCluster** resources to clean up the Logical Volume Manager resources created on the disks.

- Create an additional policy to uninstall the operator.

## Prerequisites

- Ensure that the following are deleted before deleting the policy:
  - All the applications on the managed clusters that are using the storage provisioned by the logical volume manager storage.
  - Persistent volume claims (PVCs) and persistent volumes (PVs) provisioned using the logical volume manager storage.
  - All volume snapshots provisioned by the logical volume manager storage.
- Access to the RHACM cluster using an account with a **cluster-admin** role.

## Procedure

1. In the OpenShift command-line interface, delete the ACM policy that you created for deploying and configuring the logical volume manager storage on the hub cluster by using the following command:

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

2. Save the following YAML to a file with a name such as **lvms-remove-policy.yaml** to create a policy for removing the **LVMCluster**. This enables the operator to clean up all the Logical Volume Manager resources that it created on the cluster. Set the value of **PlacementRule.spec.clusterSelector** to select the clusters from which to uninstall logical volume manager storage.

```
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

```

For descriptions of different fields, see [Reference](#).

3. Create the policy by running the following command:

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

4. Save the following YAML to a file with a name such as **check-lvms-remove-policy.yaml** to create a policy to check if the **LVMCluster** CR has been removed.

```

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. Create the policy by running the following command:

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

6. Check the policy status.

```
# 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. After both the policies are compliant, save the following YAML to a file with a name such as **lvms-uninstall-policy.yaml** to create a policy to uninstall the logical volume manager storage.

```

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: lvmclusters.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. Create the policy by running the following command:

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

## CHAPTER 2. DEPLOYING LOGICAL VOLUME MANAGER STORAGE ON SINGLE NODE OPENSIFT CLUSTERS USING OPENSIFT WEB CONSOLE

### 2.1. INSTALLING LOGICAL VOLUME MANAGER STORAGE USING OPENSIFT WEB CONSOLE

You can install logical volume manager storage using the Red Hat OpenShift Container Platform Operator Hub.

#### Prerequisites

- Access to the OpenShift Container Platform single node OpenShift (SNO) cluster using an account with **cluster-admin** and Operator installation permissions.

#### Procedure

1. Log in to the OpenShift Web Console.
2. Click **Operators** → **OperatorHub**.
3. Scroll or type **LVM Storage** into the **Filter by keyword** box to find the logical volume manager storage.
4. Click **Install**.
5. Set the following options on the **Install Operator** page:
  - a. Update Channel as **stable-4.12**.
  - b. Installation Mode as **A specific namespace on the cluster**
  - c. Installed Namespace as **Operator recommended namespace openshift-storage**. If Namespace **openshift-storage** does not exist, it is created during the operator installation.
  - d. Select Approval Strategy as **Automatic** or **Manual**.  
If you select **Automatic** updates, then the Operator Lifecycle Manager (OLM) automatically upgrades the running instance of your Operator without any intervention.  
  
If you select **Manual** updates, then the OLM creates an update request. As a cluster administrator, you must then manually approve that update request to update the Operator to a newer version.
  - e. Click **Install**.

#### Verification steps

- Verify that the logical volume manager storage shows a green tick indicating successful installation.

### 2.2. CREATING LOGICAL VOLUME MANAGER CLUSTER

Create a Logical Volume Manager cluster after you install the logical volume manager storage.

- OpenShift Container Platform supports additional worker nodes for single node OpenShift clusters on bare metal user provisioned infrastructure. For more information, see [Worker nodes for single-node OpenShift clusters](#).

The logical volume manager storage detects and uses the new additional worker nodes when the new nodes show up. In case you need to set a node filter for the additional worker nodes, you can use the YAML view while creating the cluster. Note that this node filter matching is not the same as the pod label matching.

## Prerequisites

- The logical volume manager storage must be installed from the Operator Hub.

## Procedure

1. In the OpenShift Web Console, click **Operators → Installed Operators** to view all the installed operators.  
Ensure that the **Project** selected is **openshift-storage**.
2. Click on the **LVM Storage**, and then click **Create LVMCluster** under LVMCluster.
3. In the Create LVMCluster page, select either **Form view** or **YAML view**.
4. Enter a name for the cluster.
5. Click **Create**.
6. (Optional) To add a node filter, click **YAML view** and specify the filter in the **nodeSelector** section:

```
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. (Optional) To edit the local device path of the disks, click **YAML view** and specify the device path in the **deviceSelector** section:

```
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

```

For descriptions of different fields, see [Reference](#).

For more information, see [Scaling storage of single node OpenShift cluster](#).

### Verification Steps

1. Click **Storage** → **Storage Classes** from the left pane of the OpenShift Web Console.
2. Verify that the **lvms-<device-class-name>** storage class is created with the LVMCluster creation. By default, **vg1** is the device-class-name.

## 2.3. UNINSTALLING LOGICAL VOLUME MANAGER STORAGE INSTALLED USING THE OPENSIFT WEB CONSOLE

### Prerequisites

- Ensure that the following are deleted before deleting the policy:
  - All the applications on the clusters that are using the storage provisioned by the logical volume manager storage.
  - Persistent volume claims (PVCs) and persistent volumes (PVs) provisioned using the logical volume manager storage.
  - All volume snapshots provisioned by the logical volume manager storage.
- Ensure that no logical volume resources exist by using the **oc get logicalvolume** command.
- Access to the OpenShift Container Platform single node OpenShift (SNO) cluster using an account with **cluster-admin** permissions.

installed operators → lvm → lvmcluster tab → click on 3 dots at the right end → delete lvm cluster

### Procedure

1. From the **Operators** → **Installed Operators** page, scroll to **LVM Storage** or type **LVM Storage** into the **Filter by name** to find and click on it.
2. Click on the LVMCluster tab.
3. On the right-hand side of the LVMCluster page, select Delete LVMCluster from the Actions drop-down menu.
4. Click on the Details tab.

5. On the right-hand side of the **Operator Details** page, select **Uninstall Operator** from the Actions drop-down menu.
6. Select **Remove**. The logical volume manager storage stops running and is completely removed.

## CHAPTER 3. PROVISIONING STORAGE USING LOGICAL VOLUME MANAGER STORAGE

You can provision persistent volume claims (PVCs) using the storage class that gets created during the operator installation. You can provision block and file PVCs, however, the storage is allocated only when a pod that uses the PVC is created.



### NOTE

The logical volume manager storage provisions PVCs in units of 1 GiB. The requested storage is rounded up to the nearest GiB.

### Procedure

1. Identify the StorageClass that is created when logical volume manager storage is deployed. The StorageClass name is in the format, **lvms-<device-class-name>**. **device-class-name** is the name of the device class that you provided in the LVMCluster of the policy YAML. For example, if the deviceClass is called **vg1**, then the storageClass name is **lvms-vg1**.

The **volumeBindingMode** of the storage class is set to **WaitForFirstConsumer**.

2. Save the following YAML to a file with a name such as **pvc.yaml** to create a PVC where the application requires storage.

```
# 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. Create the PVC by running the following command:

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

The PVCs that are created will remain in the **pending** state until you deploy the pods that use them.



## CHAPTER 4. MONITORING THE LOGICAL VOLUME MANAGER STORAGE

When the logical volume manager storage is installed using the OpenShift Web Console, you can monitor the cluster using the **Block and File** dashboard in the console by default. However, when you use RHACM to install the logical volume manager storage, you need to configure the RHACM Observability to monitor all the SNO clusters from one place.

You can monitor the logical volume manager storage by viewing the metrics exported by the operator on the RHACM dashboards and the alerts that are triggered. Enable RHACM Observability as described in the [Observability](#) guide.

### Metrics

- Add the following **topolvm** metrics to the allow list as specified in the [Adding custom metrics](#) section:

```
topolvm_thinpool_data_percent
topolvm_thinpool_metadata_percent
topolvm_thinpool_size_bytes
```



### NOTE

Metrics are updated every 10 minutes or when there is a change in the thin-pool, such as a new logical volume creation.

### Alerts

When the thin pool and volume group are filled up, further operations fail and might lead to data loss. The logical volume manager storage sends the following alerts the usage of the thin pool and volume group crosses certain value:

**Table 4.1. Alerts for Logical Volume Manager cluster in Red Hat Advanced Cluster Management for Kubernetes**

Alert	Description
VolumeGroupUsageAtThresholdNearFull	This alert is triggered when both the volume group and thin pool utilization cross 75% on nodes. Data deletion or volume group expansion is required.
VolumeGroupUsageAtThresholdCritical	This alert is triggered when both the volume group and thin pool utilization cross 85% on nodes. VolumeGroup is critically full. Data deletion or volume group expansion is required.
ThinPoolDataUsageAtThresholdNearFull	This alert is triggered when the thin pool data utilization in the volume group crosses 75% on nodes. Data deletion or thin pool expansion is required.

Alert	Description
ThinPoolDataUsageAtThresholdCritical	This alert is triggered when the thin pool data utilization in the volume group crosses 85% on nodes. Data deletion or thin pool expansion is required.
ThinPoolMetaDataUsageAtThresholdNearFull	This alert is triggered when the thin pool metadata utilization in the volume group crosses 75% on nodes. Data deletion or thin pool expansion is required.
ThinPoolMetaDataUsageAtThresholdCritical	This alert is triggered when the thin pool metadata utilization in the volume group crosses 85% on nodes. Data deletion or thin pool expansion is required.

## CHAPTER 5. SCALING STORAGE OF SINGLE NODE OPENSIFT CLUSTER

OpenShift Container Platform supports additional worker nodes for single node OpenShift clusters on bare metal user provisioned infrastructure. For more information, see [Worker nodes for single-node OpenShift clusters](#). The logical volume manager storage detects and uses the new additional worker nodes when the new nodes show up.

To scale the storage capacity of your configured worker nodes on Single Node OpenShift cluster, you can increase the capacity by adding disks.

### 5.1. SCALING UP STORAGE BY ADDING CAPACITY TO YOUR SINGLE NODE OPENSIFT CLUSTER USING RHACM

#### Prerequisites

- Access to the RHACM cluster using an account with cluster-admin permissions.
- Additional unused disks on each SNO cluster to be used by logical volume manager storage.

#### Procedure

1. Log in to the RHACM CLI using your OpenShift credentials.  
For more information, see [Install Red Hat Advanced Cluster Management for Kubernetes](#) .
2. Find the disk that you want to add. The disk to be added needs to match with the device name and path of the existing disks.
3. To add capacity to the Single Node OpenShift cluster, edit the **deviceSelector** section of the existing policy YAML, for example, **policy-lvms-operator.yaml**.



#### NOTE

In case **deviceSelector** is not included during the LVMCluster creation, it is not possible to add the **deviceSelector** section to the CR. You need to remove the LVMCluster and then recreate from the new CR.

```
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

```

For descriptions of different fields, see [Reference](#).

4. Edit the policy by running the following command:

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

where, ***policy-lvms-operator.yaml*** is the name of the existing policy.

This uses the new disk specified in the **LVMCluster** to provision storage.

## 5.2. SCALING UP STORAGE BY ADDING CAPACITY TO YOUR SINGLE NODE OPENSIFT CLUSTER

### Prerequisites

- Additional unused disks on each SNO cluster to be used by logical volume manager storage.

### Procedure

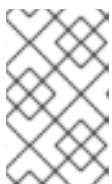
1. Log in to OpenShift console of the SNO cluster.
2. From the **Operators** → **Installed Operators** page, click on the **LVM Storage** operator in the **openshift-storage** namespace.
3. Click on the **LVMCluster** tab to list the LVMCluster created on the cluster.
4. Select **Edit LVMCluster** from the Actions drop-down menu.
5. Click on the **YAML** tab.
6. Edit the LVMCluster YAML to add the new device path in the **deviceSelector** section:

```
[...]

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

[...]
```

For descriptions of different fields, see [Reference](#).



### NOTE

In case **deviceSelector** is not included during the LVMCluster creation, it is not possible to add the **deviceSelector** section to the CR. You need to remove the LVMCluster and then recreate from the new CR.

## CHAPTER 6. UPGRADING LOGICAL VOLUME MANAGER STORAGE ON SINGLE NODE OPENSIFT CLUSTERS

Currently, it is not possible to upgrade from OpenShift Data Foundation Logical Volume Manager Operator 4.11 to logical volume manager storage 4.12 on single node OpenShift clusters. You need to perform the following:

- Backup any data that you want to preserve on the PVCs.
- Delete all PVCs provisioned by the OpenShift Data Foundation Logical Volume Manager Operator and their pods.
- Reinstall the logical volume manager storage on OpenShift Container Platform 4.12.
- Recreate the workloads.



### NOTE

Ensure that you backup your data and copy it to the PVCs created after upgrading to 4.12 as the data will not be preserved during this process.

## CHAPTER 7. VOLUME SNAPSHOTS FOR SINGLE NODE OPENSIFT

You can take volume snapshots of persistent volumes (PVs) that are provisioned by the logical volume manager storage. You can also create volume snapshots of the cloned volumes. Volume snapshots help you to:

- Back up your application data (volume snapshots are not backups)
- Revert to a state at which the volume snapshot was taken

You can create volume snapshots based on the available capacity of the thin pool and overprovisioning limits. The logical volume manager storage creates a **VolumeSnapshotClass** with the name **lvms-`<deviceclass-name>`**.

### 7.1. CREATING VOLUME SNAPSHOTS IN SINGLE NODE OPENSIFT

#### Prerequisites

- For a consistent snapshot, ensure that the PVC is in Bound state. Also, ensure that all the I/O to the PVC is stopped before taking the snapshot.

#### Procedure

1. Log in to the OpenShift single node cluster for which you need to run the **oc** command.
2. Save the following YAML to a file with a name such as **lvms-vol-snapshot.yaml**.

```
# 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. Create the snapshot by running the following command in the same namespace as the PVC:

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

A read only copy of the PVC is created as a volume snapshot.

### 7.2. RESTORING VOLUME SNAPSHOTS IN SINGLE NODE OPENSIFT

When you restore a volume snapshot, a new Persistent Volume Claim (PVC) gets created. The restored PVC is independent of the volume snapshot and the source PVC.

#### Prerequisites

- The storage class must be the same as that of the source PVC.



- The size of the requested PVC must be the same as that of the source volume of the snapshot.

### Procedure

1. Identify the storage class name of the source PVC and volume snapshot name.
2. Save the following YAML to a file with a name such as **lvms-vol-restore.yaml** to restore the snapshot.

```
# 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. Create the policy by running the following command in the same namespace as the snapshot:

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

## 7.3. DELETING VOLUME SNAPSHOTS IN SINGLE NODE OPENSIFT

### Procedure

- To delete the volume snapshot, delete the volume snapshot resource.

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



#### NOTE

When you delete a persistent volume claim (PVC), the snapshots of the PVC are not deleted.

- To delete the restored volume snapshot, delete the PVC that was created to restore the volume snapshot.

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

## CHAPTER 8. VOLUME CLONING FOR SINGLE NODE OPENSIFT

A clone is a duplicate of an existing storage volume that can be used like any standard volume. You create a clone of a volume to make a point in time copy of the data. A persistent volume claim (PVC) cannot be cloned with a different size.

### 8.1. CREATING VOLUME CLONES IN SINGLE NODE OPENSIFT

#### Prerequisites

- Ensure that the source PVC is in Bound state and not in use.
- Ensure that the StorageClass is the same as that of the source PVC.

#### Procedure

1. Identify the storage class of the source PVC.
2. Save the following YAML to a file with a name such as **lvms-vol-clone.yaml** to create a volume clone.

```
# 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. Create the policy by running the following command in the same ns as the source PVC:

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

### 8.2. DELETING CLONED VOLUMES IN SINGLE NODE OPENSIFT

#### Procedure

- To delete the cloned volume, you can delete the cloned PVC.

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

-

## CHAPTER 9. DOWNLOADING LOG FILES AND DIAGNOSTIC INFORMATION USING MUST-GATHER

When logical volume manager storage is unable to automatically resolve a problem, use the must-gather tool to collect the log files and diagnostic information so that you or the Red Hat support can review the problem and determine a solution.

- Run the must-gather command from the client connected to the logical volume manager storage cluster:

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

For more information, see [Gathering data about your cluster](#).

## CHAPTER 10. REFERENCE

A Sample LVMCluster YAML file that describes all the fields:

```

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

```