



# OpenShift Dedicated 4

## Storage

Configuring storage for OpenShift Dedicated clusters



# OpenShift Dedicated 4 Storage

---

Configuring storage for OpenShift Dedicated clusters

## Legal Notice

Copyright © 2022 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

This document provides information about setting up storage for OpenShift Dedicated clusters.

---

## Table of Contents

<b>CHAPTER 1. PERSISTENT STORAGE .....</b>	<b>3</b>
1.1. SETTING UP AWS EFS FOR OPENSIFT DEDICATED	3
1.1.1. Prerequisites	3
1.1.2. Configuring the AWS account	3
1.1.3. Installing the EFS Operator	4
1.1.4. Creating SharedVolume resources using the console	4
1.1.5. Creating SharedVolume resources using the CLI	5
1.1.6. Connecting pods	6
1.1.7. Uninstalling the EFS Operator	6

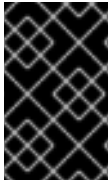


# CHAPTER 1. PERSISTENT STORAGE

## 1.1. SETTING UP AWS EFS FOR OPENSIFT DEDICATED

The Amazon Web Services Elastic File System (AWS EFS) is a Network File System (NFS) that can be provisioned on OpenShift Dedicated clusters. AWS also provides and supports a CSI EFS Driver to be used with Kubernetes that allows Kubernetes workloads to leverage this shared file storage.

This document describes the basic steps needed to set up your AWS account to prepare EFS to be used by OpenShift Dedicated. For more information about AWS EFS, see the [AWS EFS documentation](#).



### IMPORTANT

Red Hat does not provide official support for this feature, including backup and recovery. The customer is responsible for backing up the EFS data and recovering it in the event of an outage or data loss.

The high-level process to enable EFS on a cluster is:

1. Create an AWS EFS in the AWS account used by the cluster.
2. Install the AWS EFS Operator from OperatorHub.
3. Create **SharedVolume** custom resources.
4. Use the generated persistent volume claims in pod **spec.volumes**.

### 1.1.1. Prerequisites

- Customer Cloud Subscription (CCS) for an OpenShift Dedicated cluster
- Administrator access to the AWS account of that cluster

### 1.1.2. Configuring the AWS account

Set up your AWS account to prepare AWS EFS for use by OpenShift Dedicated.

#### Procedure

1. Log in to the [AWS EC2 Console](#).
2. Select the region that matches the cluster region.
3. Filter only worker EC2 instances, and select an instance. Note the VPC ID and security group ID. These values are required later in the process.
4. Click the **Security** tab, and click the Security Group Name.
5. From the **Actions** dropdown menu, click **Edit Inbound Rules**. Scroll to the bottom, and click **Add Rule**.
6. Add an NFS rule that allows NFS traffic from the VPC private CIDR.
7. Open the [Amazon EFS page](#). To create the EFS, click **Create file system**

8. Click **Customize** and proceed through the wizard.
  - a. In **Step 2:**, configure the network access:
    - i. Click the VPC of the cluster that you noted previously.
    - ii. Ensure that the private subnets are selected.
    - iii. Select the Security Group Name that you noted previously for the EC2 worker instances.
    - iv. Click **Next**.
  - b. In **Step 3:**, configure the client access:
    - i. Click **Add access point**
    - ii. Enter a unique Path such as **/access\_point\_1**.
    - iii. Configure the Owner fields with ownership or permissions that allow write access for your worker pods. For example, if your worker pods run with group ID **100**, you can set that ID as your **Owner Group ID** and ensure the permissions include **g+rwX**.
9. Continue through the wizard steps, and click **Create File System**.
10. After the file system is created:
  - a. Note the file system ID for later use.
  - b. Click **Manage client access** and note the access point ID.

You can add more NFS rules, using steps 5-10, to create separate shared data stores. In each case, make note of the corresponding file system ID and access point ID.

### 1.1.3. Installing the EFS Operator

#### Procedure

1. Log in to the OpenShift Web UI for your cluster.
2. Click **Operators** → **OperatorHub**.
3. Search for and select the AWS EFS Operator. Click **Install**.
4. Accept the default settings, and click **Subscribe**.

### 1.1.4. Creating SharedVolume resources using the console

You must create one **SharedVolume** resource per file system:access point pair in each project from which you want pods to access it.

#### Procedure

1. In the OpenShift web console, create and navigate to a project.



2. Click **Operators** → **Installed Operators**. Find the entry for AWS EFS Operator, and click **SharedVolume** under Provided APIs.
3. Click **Create SharedVolume**.
4. Edit the sample YAML:
  - a. Type a suitable value for **name**.
  - b. Replace the values of **accessPointID** and **fileSystemID** with the values from the EFS resources you created earlier.

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

5. Click **Create**.  
The **SharedVolume** resource is created, and triggers the AWS EFS Operator to generate and associate a PersistentVolume:PersistentVolumeClaim pair with the specified EFS access point.
6. To verify that the persistent volume claim (PVC) exists and is bound, click **Storage** → **Persistent Volume Claims**  
The PVC name is **pvc-<shared\_volume\_name>**. The associated PV name is **pv-<project\_name>-<shared\_volume\_name>**.

### 1.1.5. Creating SharedVolume resources using the CLI

You must create one **SharedVolume** resource per file system:access point pair in each project from which you want pods to access it. You can create a SharedVolume manually by entering YAML or JSON definitions, or by dragging and dropping a file into an editor.

#### Procedure

1. Using the **oc** CLI, create the YAML file using the **accessPointID** and **fileSystemID** values from the EFS resources you created earlier.

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

2. Apply the file to the cluster using the following command:

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

The **SharedVolume** resource is created, and triggers the AWS EFS Operator to generate and associate a PersistentVolume:PersistentVolumeClaim pair with the specified EFS access point.

3. To verify that the PVC exists and is bound, navigate to **Storage > Persistent Volume Claims**. The PVC name is **pvc-{shared\_volume\_name}**. The associated PV name is **pv-{project\_name}-{shared\_volume\_name}**.

### 1.1.6. Connecting pods

The persistent volume claim (PVC) that was created in your project is ready for use. You can create a sample pod to test this PVC.

#### Procedure

1. Create and navigate to a project.
2. Click **Workloads → Pods → Create Pod**.
3. Enter the YAML information. Use the name of your **PersistentVolumeClaim** object under **.spec.volumes[].persistentVolumeClaim.claimName**.

#### Example

```
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. After the pods are created, click **Workloads → Pods → Logs** to verify the pod logs.

### 1.1.7. Uninstalling the EFS Operator

#### Procedure

To remove the Operator from your cluster:

1. Delete all of the workloads using the persistent volume claims that were generated by the Operator.
2. Delete all of the shared volumes from all of the namespaces. The Operator automatically removes the associated persistent volumes and persistent volume claims.

3. Uninstall the Operator:
  - a. Click **Operators** → **Installed Operators**.
  - b. Find the entry for AWS EFS Operator, and click the menu button on the right-hand side of the Operator.
  - c. Click **Uninstall** and confirm the deletion.
4. Delete the shared volume CRD. This action triggers the deletion of the remaining Operator-owned resources.