




# OpenShift Dedicated 4

## Operator

OpenShift Dedicated 4  Operator



# OpenShift Dedicated 4 Operator

---

OpenShift Dedicated 4  $\mathcal{O}$  Operator

## 法律上の通知

Copyright © 2020 Red Hat, Inc.

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

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

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

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

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

Linux<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.

## 概要

本書では、OpenShift Dedicated で Operator を使用方法について説明します。

---

## 目次

<b>第1章 OPERATOR について</b> .....	<b>3</b>
1.1. OPERATOR を使用する理由	3
1.2. OPERATOR FRAMEWORK	4
1.3. OPERATOR 成熟度モデル	4
<b>第2章 OPERATOR のクラスターへの追加</b> .....	<b>6</b>
2.1. OPERATORHUB からの OPERATOR のインストール	6
<b>第3章 クラスターからの OPERATOR の削除</b> .....	<b>10</b>
3.1. WEB コンソールの使用によるクラスターからの OPERATOR の削除	10
3.2. CLI の使用によるクラスターからの OPERATOR の削除	10
<b>第4章 インストールされた OPERATOR からのアプリケーションの作成</b> .....	<b>12</b>
4.1. OPERATOR を使用した ETCD クラスターの作成	12
<b>第5章 OPERATOR ステータスの表示</b> .....	<b>15</b>
5.1. 条件のタイプ	15
5.2. CLI を使用した OPERATOR ステータスの表示	15
<b>第6章 CRD</b> .....	<b>16</b>



# 第1章 OPERATOR について

概念的に、**Operator** は人間の運用上のナレッジを使用し、これをコンシューマーと簡単に共有できるソフトウェアにエンコードします。

Operator は、ソフトウェアの他の部分を実行する運用上の複雑さを軽減するソフトウェアの特定の部分で構成されます。Operator はソフトウェアベンダーのエンジニアリングチームの拡張機能のように動作し、(OpenShift Dedicated などの) Kubernetes 環境を監視し、その最新状態に基づいてリアルタイムの意思決定を行います。高度な Operator はアップグレードをシームレスに実行し、障害に自動的に対応するように設計されており、時間の節約のためにソフトウェアのバックアッププロセスを省略するなどのショートカットを実行することはありません。

技術的には、**Operator** は Kubernetes アプリケーションをパッケージ化し、デプロイし、管理する方法です。

Kubernetes アプリケーションは、Kubernetes にデプロイされ、Kubernetes API および **kubectl** または **oc** ツールを使用して管理されるアプリケーションです。Kubernetes を最大限に活用するには、Kubernetes 上で実行されるアプリケーションを提供し、管理するために拡張できるように一連の総合的な API が必要です。Operator は、Kubernetes 上でこのタイプのアプリケーションを管理するランタイムと見なすことができます。

## 1.1. OPERATOR を使用する理由

Operator は以下を提供します。

- インストールおよびアップグレードの反復性。
- すべてのシステムコンポーネントの継続的なヘルスチェック。
- OpenShift コンポーネントおよび ISV コンテンツの OTA (Over-the-air) 更新。
- フィールドエンジニアからの知識をカプセル化し、1または2ユーザーだけでなく、すべてのユーザーに展開する場所。

### Kubernetes にデプロイする理由

Kubernetes (延長線上で考えると OpenShift Dedicated も含まれる) には、シークレットの処理、負荷分散、サービスの検出、自動スケーリングなどの、オンプレミスおよびクラウドプロバイダーで機能する、複雑な分散システムをビルドするために必要なすべてのプリミティブが含まれます。

### アプリケーションを Kubernetes API および **kubectl** ツールで管理する理由

これらの API は機能的に充実しており、すべてのプラットフォームのクライアントを持ち、クラスターのアクセス制御/監査機能にプラグインします。Operator は Kubernetes の拡張メカニズム、カスタムリソース定義 (CRD、Custom Resource Definition) を使用するので、**MongoDB** などのカスタムオブジェクトはビルトインされた、ネイティブ Kubernetes オブジェクトのように表示され、機能します。

### Operator とサービスブローカーとの比較

サービスブローカーは、アプリケーションのプログラムによる検出およびデプロイメントを行うための1つの手段です。ただし、これは長期的に実行されるプロセスではないため、アップグレード、フェイルオーバー、またはスケーリングなどの Day 2 オペレーションを実行できません。カスタマイズおよびチューニング可能なパラメーターはインストール時に提供されるのに対し、Operator はクラスターの最新の状態を常に監視します。クラスター外のサービスを使用する場合は、これらをサービスブローカーで使用できますが、Operator もこれらのクラスター外のサービスに使用できません。

## 1.2. OPERATOR FRAMEWORK

Operator Framework は、上記のカスタマーエクスペリエンスに関連して提供されるツールおよび機能のファミリーです。これは、コードを作成するためだけにあるのではなく、Operator のテスト、実行、および更新などの重要な機能を実行します。Operator Framework コンポーネントは、これらの課題に対応するためのオープンソースツールで構成されています。

### Operator SDK

Operator SDK は Kubernetes API の複雑性を把握していなくても、それぞれの専門知識に基づいて独自の Operator のブートストラップ、ビルド、テストおよびパッケージ化を実行できるように Operator の作成者を支援します。

### Operator Lifecycle Manager

Operator Lifecycle Manager は、クラスター内の Operator のインストール、アップグレード、ロールベースのアクセス制御 (RBAC) を制御します。OpenShift Dedicated 4.3 ではデフォルトでデプロイされます。

### Operator レジストリー

Operator レジストリーは、クラスターで作成するための ClusterServiceVersion (CSV) およびカスタムリソース定義 (CRD) を保存し、パッケージおよびチャネルについての Operator メタデータを保存します。これは Kubernetes または OpenShift クラスターで実行され、この Operator カタログデータを OLM に指定します。

### OperatorHub

OperatorHub は、クラスター管理者がクラスター上にインストールする Operator を検出し、選択するための Web コンソールです。OpenShift Dedicated ではデフォルトでデプロイされます。

### Operator Metering

Operator Metering は、クラスター上で Day 2 管理についての Operator の運用上のメトリクスを収集し、使用状況のメトリクスを集計します。

これらのツールは組み立て可能なツールとして設計されているため、役に立つと思われるツールを使用できます。

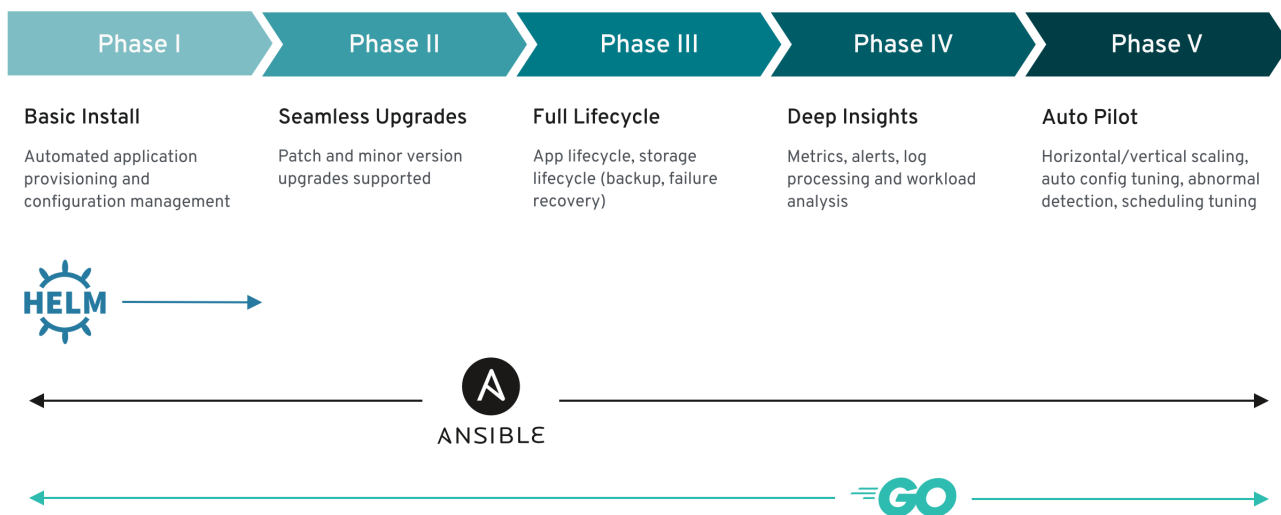
## 1.3. OPERATOR 成熟度モデル

Operator 内にカプセル化されている管理ロジックの複雑さのレベルはさまざまです。また、このロジックは通常 Operator によって表されるサービスのタイプによって大きく変わります。

ただし、大半の Operator に含まれる特定の機能セットについては、Operator のカプセル化された操作の成熟度を一般化することができます。このため、以下の Operator 成熟度モデルは、Operator の一般的な Day 2 オペレーションについての 5 つのフェーズの成熟度を定義しています。



図1.1 Operator 成熟度モデル



上記のモデルでは、これらの機能を Operator SDK の Helm、Go、および Ansible 機能で最適に開発する方法も示します。

## 第2章 OPERATOR のクラスターへの追加

以下では、クラスター管理者を対象に、Operator の OpenShift Dedicated クラスターへのインストールおよび Operator を namespace にサブスクライブする方法について説明します。

### 2.1. OPERATORHUB からの OPERATOR のインストール

クラスター管理者は、OpenShift Dedicated Web コンソールを使用して OperatorHub から Operator をインストールできます。その後、Operator をデフォルトの **openshift-operators** namespace にサブスクライブし、これをクラスターで開発者が使用できるようにできます。

OpenShift Dedicated クラスターでは、Operator のキュレートされた一覧を OperatorHub からインストールできるようになります。管理者は Operator をデフォルトの **openshift-operators** namespace のみにインストールできます。ただし、**openshift-logging** namespace を必要とする Logging Operator はこの例外です。



#### 注記

特権付き Operator およびカスタム Operator はインストールできません。

インストール時に、Operator の以下の初期設定を判別する必要があります。

#### インストールモード

OpenShift Dedicated クラスターでは、**All namespaces on the cluster (default)** を選択し、Operator をすべての namespace にインストールできます。これにより、Operator をすべてのユーザーおよびプロジェクトで利用可能にできます。

#### 更新チャンネル

Operator が複数のチャンネルで利用可能な場合、サブスクライブするチャンネルを選択できます。たとえば、(利用可能な場合に) **stable** チャンネルからデプロイするには、これを一覧から選択します。

#### 承認ストラテジー

自動 (Automatic) または手動 (Manual) のいずれかの更新を選択します。インストールされた Operator について自動更新を選択する場合、Operator の新規バージョンが利用可能になると、Operator Lifecycle Manager (OLM) は人の介入なしに、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。手動更新を選択する場合、Operator の新規バージョンが利用可能になると、OLM は更新要求を作成します。クラスター管理者は、Operator が新規バージョンに更新されるように更新要求を手動で承認する必要があります。

#### 2.1.1. Web コンソールを使用した OperatorHub からのインストール

この手順では、Couchbase Operator をサンプルとして使用し、OpenShift Dedicated Web コンソールを使用して、OperatorHub から Operator をインストールし、これにサブスクライブします。

#### 前提条件

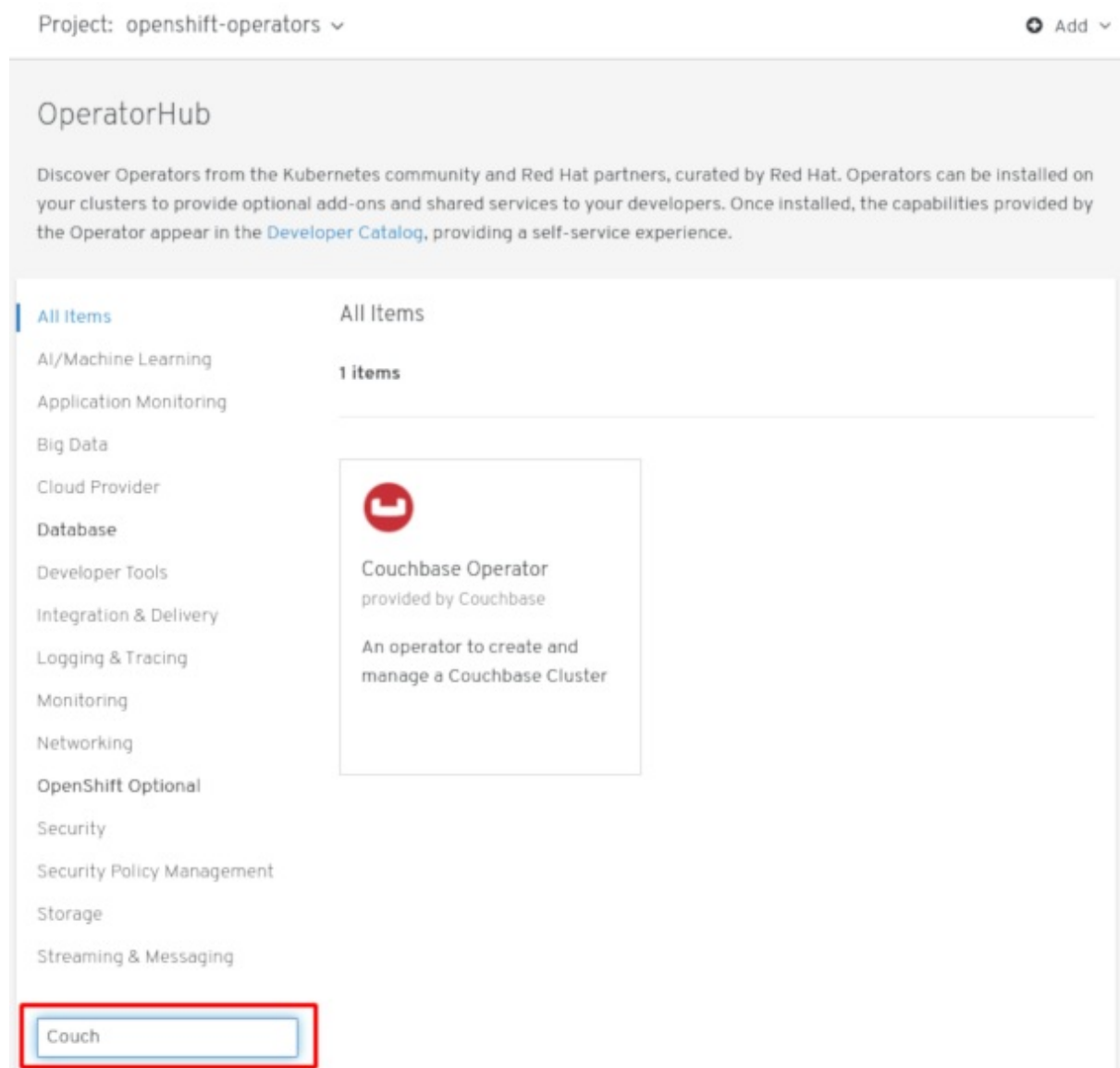
- **dedicated-admins-cluster** パーミッションを持つアカウントを使用して OpenShift Dedicated クラスターにアクセスできること。

#### 手順

1. Web コンソールで、Operators → OperatorHub ページに移動します。

- スクロールするか、またはキーワードを **Filter by keyword** ボックスに入力し (この場合は **Couchbase**)、必要な Operator を見つけます。

図2.1 キーワードによる Operator のフィルター



- Operator を選択します。コミュニティー Operator の場合、Red Hat がそれらの Operator を認定していないことについての警告が出されます。作業を継続する前に、この警告を確認してください。Operator についての情報が表示されます。
- Operator についての情報を確認してから、**Install** をクリックします。
- Create Operator Subscription** ページで以下を実行します。
  - 以下のいずれかを選択します。
    - All namespaces on the cluster (default)**は、デフォルトの **openshift-operators** namespace で Operator をインストールし、クラスターのすべての namespace を監視し、Operator をこれらの namespace に対して利用可能にします。このオプションは常に選択可能です。
    - A specific namespace on the cluster**では、Operator をインストールする特定の単一 namespace を選択できます。Operator は監視のみを実行し、この単一 namespace で使用される場合のみ利用可能になります。Cluster Logging Operator をインストールし

ている場合、このオプションを選択し、**openshift-logging** namespace を選択します。

- b. **Update Channel** を選択します (複数を選択できる場合)。
  - c. 前述のように、**自動 (Automatic)** または **手動 (Manual)** の承認ストラテジーを選択します。
6. **Subscribe** をクリックし、Operator をこの OpenShift Dedicated クラスターの選択した namespace で利用可能にします。
- a. 手動の承認ストラテジーを選択している場合、Subscription のアップグレードステータスは、その Install Plan を確認し、承認するまで **Upgrading** のままになります。

## 図2.2 Install Plan ページからの手動による承認

The screenshot shows the OpenShift console interface for an Install Plan. At the top, the project is set to 'openshift-operators'. The breadcrumb navigation shows 'couchbase-enterprise-certified > Install Plan Details'. The main title is 'install-bqbms' with an 'Actions' dropdown menu. Below this are tabs for 'Overview', 'YAML', and 'Components', with 'Overview' selected. A prominent blue box contains the heading 'Review Manual Install Plan' and the instruction 'Inspect the requirements for the components specified in this install plan before approving.' A 'Preview Install Plan' button is located below this box. The 'Install Plan Overview' section displays the following details:

<b>NAME</b> install-bqbms	<b>STATUS</b> RequiresApproval
<b>NAMESPACE</b> NS <b>openshift-operators</b>	<b>COMPONENTS</b> CSV <b>couchbase-operator.v1.1.0</b>
<b>LABELS</b> No labels	<b>CATALOG SOURCES</b> CS <b>installed-certified-openshift-operators</b>
<b>CREATED AT</b> 🕒 a few seconds ago	
<b>OWNER</b> SUB <b>couchbase-enterprise-certified</b>	

**Install Plan** ページでの承認後に、Subscription のアップグレードステータスは **Up to date** に移行します。

- b. 自動承認ストラテジーを選択している場合、アップグレードステータスは、介入なしに **Up to date** に解決するはずですが。

図2.3 Subscription のアップグレードステータス「Up to date」

Project: openshift-operators ▾

---

**SUB** couchbase-enterprise-certified

---

Overview **YAML**

---

Subscription Overview

CHANNEL preview ✎	APPROVAL Automatic ✎	<b>UPGRADE STATUS</b> ✔ Up to date	1 installed 0 installing
----------------------	-------------------------	---------------------------------------	-----------------------------

NAME  
couchbase-enterprise-certified

INSTALLED VERSION  
**CSV** couchbase-operator.v1.1.0

NAMESPACE  
**NS** openshift-operators

STARTING VERSION  
couchbase-operator.v1.1.0

LABELS  
csc-owner-name=installed-certified-openshift-operators  
csc-owner-namespace=openshift-marketplace

CATALOG SOURCE  
**CS** installed-certified-openshift-operators

7. Subscription のアップグレードステータスが **Up to date** になった後に、**Operators → Installed Operators** を選択して、**Couchbase ClusterServiceVersion (CSV)** が表示され、そのステータスが最終的に関連する namespace で **InstallSucceeded** に解決することを確認します。



### 注記

**All namespaces...** インストールモードの場合、ステータスは **openshift-operators** namespace で **InstallSucceeded** になりますが、他の namespace でチェックする場合、ステータスは **Copied** になります。

上記通りにならない場合:

- a. Pod のログを **openshift-operators** プロジェクトで確認します (または、**A specific namespace...** インストールモードが選択されている場合は別の関連する namespace) で確認します。これは、さらにトラブルシューティングする問題を報告する **Workloads → Pods** ページから実行します。

## 第3章 クラスターからの OPERATOR の削除

以下では、Web コンソールまたは CLI のいずれかを使用してクラスターから Operator を削除する方法について説明します。

### 3.1. WEB コンソールの使用によるクラスターからの OPERATOR の削除

クラスター管理者は Web コンソールを使用して、選択した namespace からインストールされた Operator を削除できます。

#### 前提条件

- **dedicated-admins-cluster** パーミッションを持つアカウントを使用して OpenShift Dedicated クラスター Web コンソールにアクセスできること。

#### 手順

1. **Operators** → **Installed Operators** ページからスクロールするか、または **Filter by name** にキーワードを入力して必要な Operator を見つけます。次に、それをクリックします。
2. **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。
3. **Remove Operator Subscription** ウィンドウでプロンプトが表示されたら、Operator に関連するすべてのコンポーネントを削除する場合は、**Also completely remove the Operator from the selected namespace** チェックボックスを選択できます。これにより CSV が削除され、次に Operator に関連付けられた Pod、Deployment、CRD および CR が削除されます。このチェックはそのままにしておくと、サブスクリプションのみが削除されます。
4. **Remove** を選択します。この Operator は実行を停止し、更新を受信しなくなります。

### 3.2. CLI の使用によるクラスターからの OPERATOR の削除

クラスター管理者は CLI を使用して、選択した namespace からインストールされた Operator を削除できます。

#### 前提条件

- **dedicated-admins-cluster** パーミッションを持つアカウントを使用して OpenShift Dedicated クラスターにアクセスできること。
- **oc** コマンドがワークステーションにインストールされていること。

#### 手順

1. サブスクリプションされた Operator (例: **jaeger**) の現行バージョンを **currentCSV** フィールドで確認します。

```
$ oc get subscription jaeger -n openshift-operators -o yaml | grep currentCSV
currentCSV: jaeger-operator.v1.8.2
```

2. Operator の Subscription (例: **jaeger**) を削除します。

```
$ oc delete subscription jaeger -n openshift-operators  
subscription.operators.coreos.com "jaeger" deleted
```

3. 直前の手順で **currentCSV** 値を使用し、ターゲット namespace の Operator の CSV を削除します。

```
$ oc delete clusterserviceversion jaeger-operator.v1.8.2 -n openshift-operators  
clusterserviceversion.operators.coreos.com "jaeger-operator.v1.8.2" deleted
```

## 第4章 インストールされた OPERATOR からのアプリケーションの作成

以下では、開発者を対象に、OpenShift Dedicated Web コンソールを使用して、インストールされた Operator からアプリケーションを作成する例を示します。

### 4.1. OPERATOR を使用した ETCD クラスターの作成

この手順では、Operator Lifecycle Manager (OLM) で管理される etcd Operator を使用した新規 etcd クラスターの作成について説明します。

#### 前提条件

- OpenShift Dedicated 4.3 クラスターへのアクセス
- 管理者によってクラスターにすでにインストールされている etcd Operator

#### 手順

1. この手順を実行するために OpenShift Dedicated Web コンソールで新規プロジェクトを作成します。この例では、**my-etcd** というプロジェクトを使用します。
2. **Operators** → **Installed Operators** ページに移動します。クラスター管理者によってクラスターにインストールされ、使用可能にされた Operator が ClusterServiceVersion (CSV) の一覧としてここに表示されます。CSV は Operator によって提供されるソフトウェアを起動し、管理するために使用されます。

#### ヒント

以下を使用して、CLI でこの一覧を取得できます。

```
$ oc get csv
```

3. **Installed Operators** ページで、**Copied** をクリックしてから、etcd Operator をクリックして詳細情報および選択可能なアクションを表示します。



## 図4.1 etcd Operator の概要

The screenshot displays the etcd Operator overview page. At the top, there is a header with the etcd logo, version '0.9.2 provided by CoreOS, Inc', and an 'Actions' dropdown menu. Below the header is a navigation bar with tabs: 'Overview' (selected), 'YAML', 'Events', 'All Instances', 'etcd Cluster', 'etcd Backup', and 'etcd Restore'. The main content area is divided into two columns. The left column contains metadata: 'PROVIDER: CoreOS, Inc', 'CREATED AT: Feb 4, 3:10 pm', 'LINKS: Blog, https://coreos.com/etcd', 'Documentation: https://coreos.com/operator/s/etcd/docs/latest', 'etcd Operator Source Code: https://github.com/coreos/etcd-operator', and 'MAINTAINERS: CoreOS, Inc, support@coreos.com'. The right column is titled 'Provided APIs' and contains three API cards: 'etcd Cluster' (described as 'Represents a cluster of etcd nodes'), 'etcd Backup' (described as 'Represents the intent to backup an etcd cluster'), and 'etcd Restore' (described as 'Represents the intent to restore an etcd cluster from a backup'). Each card has a 'Create New' button. Below the APIs is a 'Description' section stating: 'etcd is a distributed key value store that provides a reliable way to store data across a cluster of machines. It's open-source and available on GitHub. etcd gracefully handles leader elections during'.

Provided APIs に表示されているように、この Operator は 3 つの新規リソースタイプを利用可能にします。これには、**etcd クラスター (EtcdCluster リソース)** のタイプが含まれます。これらのオブジェクトは、**Deployments** または **ReplicaSets** などの組み込み済みのネイティブ Kubernetes オブジェクトと同様に機能しますが、これらには etcd を管理するための固有のロジックが含まれます。

4. 新規 etcd クラスターを作成します。
  - a. **etcd Cluster** API ボックスで、**Create New** をクリックします。
  - b. 次の画面では、クラスターのサイズなど **EtcdCluster** オブジェクトのテンプレートを起動する最小条件への変更を加えることができます。ここでは **Create** をクリックして確定します。これにより、Operator がトリガーされ、Pod、サービス、および新規 etcd クラスターの他のコンポーネントが起動します。
5. **Resources** タブをクリックして、プロジェクトに Operator によって自動的に作成され、設定された数多くのリソースが含まれることを確認します。

## 図4.2 etcd Operator リソース

etcdoperator.v0.9.2 > EtcdCluster Details

EC example

Actions ▾

Overview YAML **Resources**

Filter Resources by name...

2 Service

3 Pod

Select All Filters

5 Items

NAME ↑	TYPE	STATUS	CREATED
<span>S</span> example	Service	Created	🕒 3 minutes ago
<span>S</span> example-client	Service	Created	🕒 3 minutes ago
<span>P</span> example-dccdn267hl	Pod	Running	🕒 2 minutes ago
<span>P</span> example-g2shm4cz4l	Pod	Running	🕒 2 minutes ago
<span>P</span> example-sgm2hcktcn	Pod	Running	🕒 3 minutes ago

Kubernetes サービスが作成され、プロジェクトの他の Pod からデータベースにアクセスできることを確認します。

6. 所定プロジェクトで **edit** ロールを持つすべてのユーザーは、クラウドサービスのようにセルフサービス方式でプロジェクトにすでに作成されている Operator によって管理されるアプリケーションのインスタンス (この例では etcd クラスター) を作成し、管理し、削除することができます。この機能を持つ追加のユーザーを有効にする必要がある場合、プロジェクト管理者は以下のコマンドを使用してこのロールを追加できます。

```
$ oc policy add-role-to-user edit <user> -n <target_project>
```

これで、etcd クラスターは Pod が正常でなくなったり、クラスターのノード間で移行する際の障害に対応し、データのリバランスを行います。最も重要な点として、適切なアクセスを持つクラスター管理者または開発者は独自のアプリケーションでデータベースを簡単に使用できるようになります。

## 第5章 OPERATOR ステータスの表示

Operator Lifecycle Manager (OLM) のシステムの状態を理解することは、インストールされた Operator についての問題について意思決定を行い、デバッグを行う上で重要です。OLM は、Subscription およびそれに関連するカタログソースリソースの状態および実行されたアクションに関する知見を提供します。これは、それぞれの Operator の正常性を把握するのに役立ちます。

### 5.1. 条件のタイプ

Subscription は状態についての以下のタイプを報告します。

表5.1 Subscription の状態のタイプ

状態	説明
<b>CatalogSourcesUnhealthy</b>	解決に使用される一部のまたはすべてのカタログソースは正常ではありません。
<b>InstallPlanMissing</b>	Subscription の InstallPlan がありません。
<b>InstallPlanPending</b>	Subscription の InstallPlan のインストールが保留中です。
<b>InstallPlanFailed</b>	Subscription の InstallPlan が失敗しました。

### 5.2. CLI を使用した OPERATOR ステータスの表示

CLI を使用して Operator ステータスを表示できます。

#### 手順

1. **oc describe** コマンドを使用して、Subscription のリソースを検査します。

```
$ oc describe sub <subscription_name>
```

2. コマンド出力で **Conditions** セクションを見つけます。

```
Conditions:
  Last Transition Time: 2019-07-29T13:42:57Z
  Message:             all available catalogsources are healthy
  Reason:              AllCatalogSourcesHealthy
  Status:              False
  Type:                CatalogSourcesUnhealthy
```

## 第6章 CRD