

# IBM MQ V9.3 アップデート・セミナー

---

## 第2章：新機能・変更点

### 1. コンテナ関連



# コンテナ・サポート概要

# MQのコンテナ・サポート概要

## ■ コンテナ環境でのキューマネージャー及びクライアント・アプリケーションの稼働をサポート

- ◆ Docker、Podman、Kubernetes Cluster、OpenShift(OCP)など様々なコンテナ環境で稼働可能



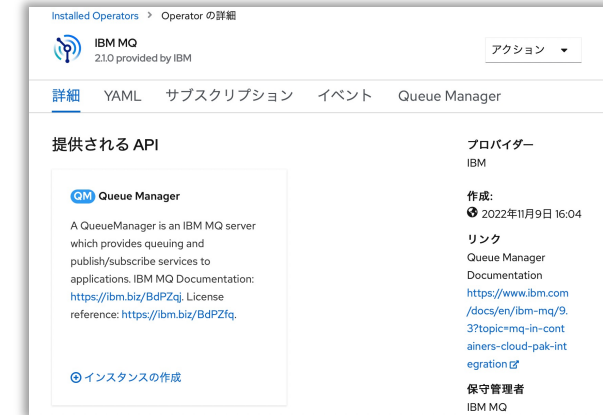
## ■ MQ Operatorの提供(V9.1.5～)

- ◆ OpenShift環境へのデプロイ、運用管理を容易にするためのOperatorを提供
  - Pre-buildコンテナ・イメージやOCP用デプロイメント・コードを含む

## ■ ユーザー独自のコンテナ・イメージを利用するためのサンプル提供

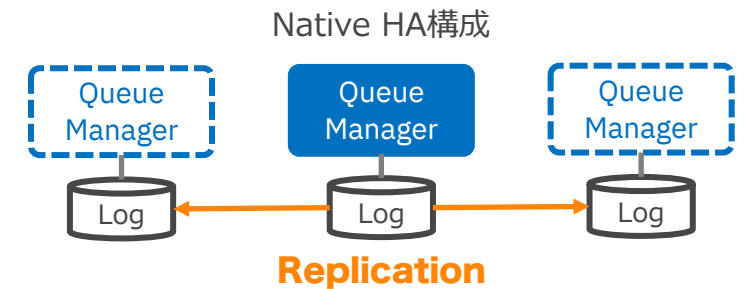
- ◆ サンプルのビルド/デプロイメント・コードをGitHubで公開
  - mq-container repository : <https://github.com/ibm-messaging/mq-container>
- ◆ Kubernetes環境へデプロイするためのサンプルのHelmチャートを公開
  - mq-helm repository : <https://github.com/ibm-messaging/mq-helm>

### MQ Operator



## ■ Native HA構成(V9.2.3 CD/V9.3 LTS～)

- ◆ MQ独自のログ・レプリケーション機能による高可用性構成



## ■ MQ Scaler for KEDA



- ◆ KEDA(Kubernetes-based Event Driven Autoscaler)によるクライアント・アプリケーションのオート・スケーリングの実現

## ■ コンテナ環境におけるデプロイメントのオプション

デプロイメント・コード	利用するコンテナ・イメージ	デプロイ可能なコンテナ環境	制約	サポート
MQ Operator	Pre-buildコンテナ・イメージ もしくはPre-buildコンテナ・ イメージをベースにしたイメージ	OpenShift環境	利用不可のフィーチャーあり (次節参照)	デプロイメント含め、MQの 基本的なサポートあり(※2)
ユーザー独自のコード (サンプル提供)	ユーザーが独自にビルドした イメージ	任意のコンテナ環境 (Docker、Podman、Kubernetes、 OpenShiftなど)	基本的には利用不可のフィー チャーなし(※1)	MQの基本的なサポートあり (※2) ※ビルド/デプロイはユー ザーの責任で実施

※1 IBM MQ Bridge to blockchain はコンテナ環境での利用はサポートされない

※2 仮想化技術に対するMQのステートメントは下記参照

IBM MQ's support position on virtualization, low-level hardware, file systems on networks and high availability

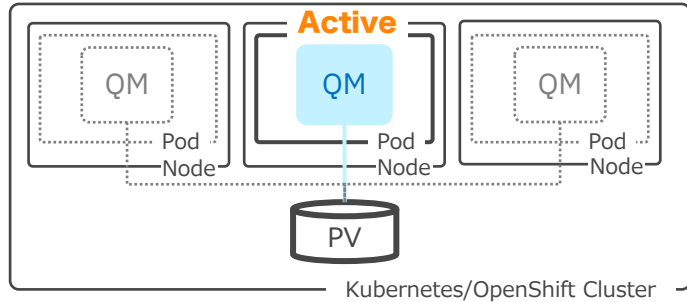
<https://www.ibm.com/support/pages/node/391335>

### ◆ 基本方針

- OpenShift環境にデプロイする場合は、MQ Operatorを利用
  - Pre-buildコンテナ・イメージでは利用できないフィーチャーを利用する場合は、ユーザー独自のデプロイメント・コードを利用
- OpenShift以外のコンテナ環境にデプロイする場合は、ユーザー独自のデプロイメント・コードを利用

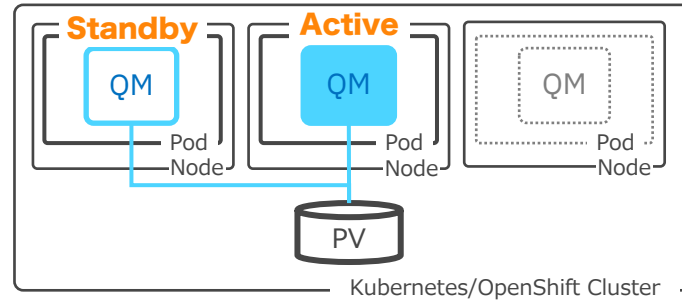
## ■ コンテナ環境におけるキューマネージャーの高可用性構成

### シングル・レジリエント構成



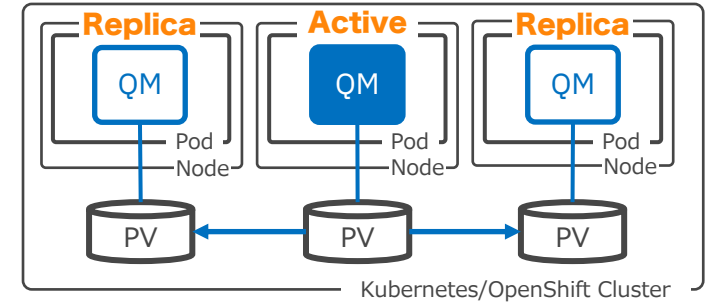
- キューマネージャーは、1つのPodで稼働  
キューマネージャー専用のPV(Persistent Volume)を持つ  
(アクセスモード:RWO、Block Storage)  
PVを持たない構成も可能
- QMgr/Pod/Node障害時はコンテナ・オーケストレーションのセルフ・ヒーリング機能で同一もしくは別ノードでQMgr/Podが再起動する
- PVは再起動したQMgrが引き継ぐ
  
- マルチAZ対応：△  
ゾーンを跨いだPVの冗長構成が難しい  
PVを持たない構成の場合は、○

### マルチ・インスタンス構成



- キューマネージャーは、2つのPodで稼働  
1つがActiveインスタンスとして稼働し、  
もう一方がStandbyインスタンスとして待機  
PVはインスタンス間で共有(アクセスモード: RWX、File Storage)
- QMgr/Pod/Node障害時はStandbyインスタンスがActiveインスタンスに昇格
- PVは昇格したQMgrが引き継ぐ
- 障害QMgr/Podはセルフ・ヒーリング機能で同一もしくは別ノードで再起動  
Standbyインスタンスとして待機
  
- マルチAZ対応：△  
ゾーンを跨いだPVの冗長構成が難しい

### Native HA構成



- キューマネージャーは、3つのPodで稼働  
1つがActiveインスタンスとして稼働し、残り2つが  
Replicaインスタンスとして待機  
インスタンス毎にPVを持つ(アクセスモード:RWO、  
Block Storage)
- QMgr/Pod/Node/Storage障害時はReplicaインスタンスの1つがActiveインスタンスに昇格
- PVはインスタンス自身のPVを利用  
キューマネージャー・ログはレプリケートされているため、  
構成やメッセージは引き継がれる
- 障害QMgr/Podはセルフ・ヒーリング機能で同一もしくは別ノードで再起動  
Replicaインスタンスとして待機
  
- マルチAZ対応：○  
インスタンス毎にPVを持つため、ゾーン跨ぎで構成可能



# MQ Operator

## ■ OpenShift環境へのデプロイ、運用管理を容易にするためのOperatorを提供

### ◆ OpenShift Webコンソールにてパラメータ設定のみでキューマネージャのデプロイが可能

- yamlファイルによるコマンド・ベースのデプロイも可能

### ◆ カスタム・リソース”QueueManager”を利用

- ユーザーは”QueueManager”の各種フィールドを設定するだけでよい
  - ライセンス、Availabilityタイプ、ストレージ・タイプ、リソース割当など
- “QueueManager”配下に必要なOpenShiftリソースが作成される
  - StatefulSet、Service、Routeなど

### ◆ Pre-buildコンテナ・イメージを利用

- IBM Cloud Container Registryにて提供
  - Entitlement keyが必要
- Pre-buildコンテナ・イメージをベースにしたイメージも利用可能

### ◆ Native HA構成にも対応

- Availabilityタイプとして以下を選択可能
  - SingleInstance(シングル・レジリエント)、MultiInstance、NativeHA

OpenShift WebコンソールのMQ Operator設定画面

The screenshot displays the OpenShift Web Console interface for creating a QueueManager. The left sidebar shows the navigation menu with categories like Home, Operator, Workload, Network, and Storage. The main content area is titled 'QueueManagerの作成' and includes a warning message: '注:一部のフィールドはこのフォームビューで表示されない場合があります。完全にコントロールするには、「YAML view」を選択してください。' Below this, there are input fields for '名前' (Name) with the value 'quickstart-cp4i' and 'ラベル' (Label) with the value 'app=frontend'. There are also sections for 'License', 'Version' (set to 9.3.1.0-r1), and 'Queue Manager' settings. At the bottom, there are '作成' (Create) and 'キャンセル' (Cancel) buttons.

## ■ 前提

- ◆ IBM MQ CDリリース、もしくはCP4I-LTSリリースで利用可能
  - MQ CDリリースは1年、CP4I-LTSは2年+1年延長のサポート
- ◆ 以下のアーキテクチャーをサポート
  - amd64、s390x z/Linux、ppc64le Power Systems

## ■ MQ Operatorがサポートする各種バージョン

Operator channel	Operator version	OCP version	MQ version	CP4I version
V2.0	2.0	4.10 up : CDおよびCP4I-LTSをサポート	9.3.0 : CDおよびCP4I-LTSをサポート 9.2.3 – 9.2.5 : CDのみサポート 9.2.0 EUS : CP4I-LTSからCDへの移行時のみサポート	2022.2.1

### ◆ MQ V9.2以前のMQ Operatorがサポートするバージョンは以下を参照

- Release history for IBM MQ Operator

[https://www.ibm.com/docs/en/ibm-mq/9.2?topic=openshift-release-history-mq-operator#ctr\\_release\\_notes\\_\\_available\\_mq\\_versions\\_old](https://www.ibm.com/docs/en/ibm-mq/9.2?topic=openshift-release-history-mq-operator#ctr_release_notes__available_mq_versions_old)



### ■ 制約

#### ◆ 以下のフィーチャーは利用不可

- 管理REST API、メッセージングREST API (IBM Cloud Pak for Integrationライセンスの場合のみ)
- MFT(Managed File Transfer) エージェント
- AMQP
- IBM MQ Bridge to Salesforce
- IBM MQ Bridge to blockchain (MQ Operatorを利用しない場合でも、コンテナ環境での利用が不可)
- IBM MQ Telemetry Transport (MQTT)
- 上記フィーチャーを利用する場合(Bridge to blockchain以外)は、ユーザー自身でコンテナ・イメージとデプロイメント・コードを用意する必要あり

#### ◆ OpenShift以外のコンテナ環境でMQ Operatorを利用することはサポートされない

#### ◆ MQ Operatorを利用して、ユーザー独自のコンテナ・イメージをデプロイすることは不可

# 画面イメージ

## QueueManagerの作成

フォームを完了して作成します。デフォルト値は Operator の作成者によって指定される

次を使用して設定:  フォームビュー  YAML ビュー

**注:**一部のフィールドはこのフォームビューで表示されない場合があります。ロールするには、「YAML view」を選択してください。

名前 \*

qmgr-single

ラベル

app=frontend

License \*

Settings that control your acceptance of the license, and which license metrics to use

License acceptance \*

accept

Whether or not you accept the license associated with this software (required)

License \*

L-RJON-CD3JKX

ライセンス

The identifier of the license you are accepting. This must be the correct license identifier for the are using. See <http://ibm.biz/BdqvCF> for valid values.

License use

NonProduction

Setting that controls how the software will be used, where the license supports multiple uses. <http://ibm.biz/BdqvCF> for valid values.

> 詳細設定

Version \*

9.3.1.0-r1

Version

## Queue Manager

Settings for the Queue Manager container and underlying Queue Manager

### Availability

Availability settings for the Queue Manager, such as whether or not to use an active-standby pair or native availability

Type of availability

Type of availability の選択

MultInstance

NativeHA

SingleInstance

高可用性構成

### Storage

Storage settings to control the Queue Manager's use of persistent volumes and storage classes

Default class

StorageClass の選択

Storage class to apply to all persistent volumes of this Queue Manager by default. Specific persistent volumes can define their own storage class which will override this default storage class setting. If "type of availability" is "SingleInstance" or "NativeHA", storage class can be of type ReadWriteOnce or ReadWriteMany. If "type of availability" is "MultInstance", storage class must be of type ReadWriteMany.

### Queue Manager

Default PersistentVolume for any data normally under `/var/mqm``. Will contain all persisted data and recovery logs, if no other volumes are specified.

Type of volume

persistent-claim

ストレージ・タイプ

Type of volume to use. Choose ``ephemeral`` to use non-persistent storage, or ``persistent-claim`` to use a persistent volume. Defaults to ``persistent-claim``.

> 詳細設定

Class

StorageClass の選択

### Persisted data

PersistentVolume details for MQ persisted data, including configuration, queues and messages. Required when using multi-instance Queue Manager.

### Recovery logs

Persistent volume details for MQ recovery logs. Required when using multi-instance Queue Manager.

> 詳細設定

### INI

Settings for supplying INI for the Queue Manager. Requires MQ Operator 1.1.0 or higher.

### MQSC

Settings for supplying MQSC for the Queue Manager. Requires MQ Operator 1.1.0 or higher.

Name

QM1

Name of the underlying MQ Queue Manager, if different from metadata.name. Use this field if you want a Queue Manager name which does not conform to the Kubernetes rules for names (for example, a name which includes capital letters).

### Resources

Settings that control resource requirements

### Tracing

Settings for tracing integration with the Cloud Pak for Integration Operations Dashboard

### Web

Settings for the MQ web server

> 詳細設定

作成

キャンセル

qm.ini設定

MQオブジェクト設定(MQSCコマンド)

キューマネージャー名

リソース設定(CPU/Mem/Storage)

Webサーバー設定

## ■ MQ Operatorで使用できるストレージのタイプ

ストレージ・タイプ	説明
Ephemeral（一時）	<p>ステートレスな処理の場合に利用 ノン・パーシステント・メッセージのみを使用、チャンネル接続はNPMSPEED=FAST、MQクラスターに属していない場合には利用可能 PV(Persistent Volume)は構成しないため、Podが再作成されるタイミングで全ての構成データは初期化される エラー・ログやFDCファイルも削除される点に注意</p>
Persistent（永続）	<p>ステートフルな処理の場合に利用 マルチインスタンス構成やNative HA構成の場合は必須 構成データはPV上に保持されるため、Pod再作成を跨って引き継がれる PVは以下3つの単位で構成可能</p> <ul style="list-style-type: none"><li>・ /var/mqm以下（MQの構成ディレクトリ全体）</li><li>・ /var/mqm/qmgrs以下（キューマネージャーの構成データ）</li><li>・ /var/mqm/log以下（ログ・ファイル）</li></ul> <p>※構成ディレクトリ全体を1PVで構成するか、キューマネージャーの構成データ、ログ・ファイルそれぞれにPVを構成するか選択できる</p> <p>AvailabilityタイプによってPVに求められるアクセス・モードが異なる</p> <ul style="list-style-type: none"><li>・ SingleInstance/NativeHAの場合、ReadWriteOnce(RWO)</li><li>・ MultiInstanceの場合、ReadWriteMany(RWX)</li></ul> <p>それぞれ対応するストレージ・クラスを指定</p>

# MQ構成のカスタマイズ

## ■ qm.iniやMQオブジェクトの設定は、ConfigMap、Secretを利用

### ◆ 事前にqm.iniやMQSCファイルをConfigMap/Secretとして用意

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: base-ini
data:
  base-ini: |-
  Channels:
    MaxChannels=1000
    MaxActiveChannels=500
```

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: base-mqsc
data:
  base-mqsc: |-
  DEF QL(QL01) MAXDEPTH(999999999)
  DEF CHL(APL.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP) SHARECNV(1)
```

### ◆ MQ Operatorにて、ConfigMap/Secretを指定

INI

Settings for supplying INI for the Queue Manager. Requires MQ Operator 1.1.0 or higher.

INIの削除

Config Map

ConfigMap represents a Kubernetes ConfigMap that contains INI information

▼ 詳細設定

Items \*

The keys inside the Kubernetes config map which should be applied

Name \*

CM base-ini

The name of the Kubernetes config map

MQSC

Settings for supplying MQSC for the Queue Manager. Requires MQ Operator 1.1.0 or higher.

MQSCの削除

Config Map

ConfigMap represents a Kubernetes ConfigMap that contains MQSC information

▼ 詳細設定

Items \*

The keys inside the Kubernetes config map which should be applied

Name \*

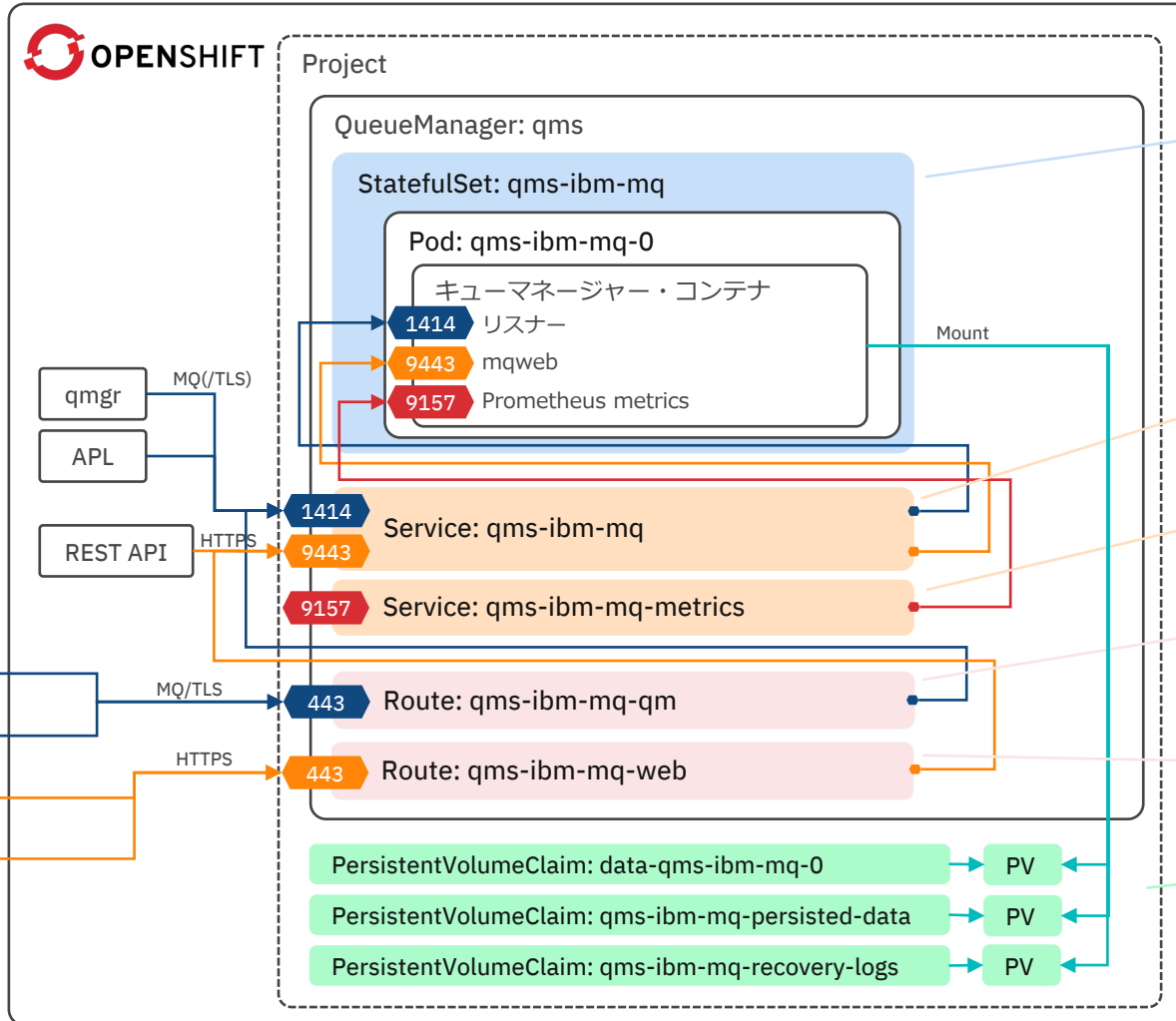
CM base-mqsc

The name of the Kubernetes config map

### ◆ qm.iniやMQSCファイルは、Pod再作成の度にキューマネージャー起動時の自動構成機能によって適用される

- 設定変更、構成変更する場合は、ConfigMap/Secretを更新して、Pod再作成で適用する

## MQ Operatorによって作成されるOpenShiftリソース



キューマネージャー・コンテナが稼働するPodを管理  
Podのレプリカ数(replicas)はAvailabilityタイプによって異なる  
(SingleInstance=1、MultiInstance=2、NativeHA=3)  
Pod障害時は、StatefulSetによってPodを再起動

OpenShiftクラスター内からキューマネージャーのPodにアクセスする際に利用  
チャンネル接続用のポート1414とWebコンソール/REST API用のポート9443を公開

OpenShiftクラスター内からPrometheusのメトリクスを取得する際に利用  
ポート9157を公開

OpenShiftクラスターの外からキューマネージャーのチャンネル接続用ポートに接続する  
際に利用 (passthroughタイプ)  
TLS前提となるため、MQチャンネル接続もTLS構成にする必要がある

OpenShiftクラスターの外からWebコンソール/REST APIで接続する際に利用  
(passthroughタイプ)

キューマネージャーの構成データを格納するためのPVC、PVが作成される  
(ストレージタイプPersistentを選択した場合のみ)  
PVはキューマネージャー・コンテナからマウントされる

## ■ ocコマンドによる構成、稼働状況確認

```
% oc get all
NAME                READY   STATUS    RESTARTS      AGE
pod/qms-ibm-mq-0    1/1    Running   34 (3h38m ago) 10h

NAME                TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/qms-ibm-mq  ClusterIP    172.21.53.105 <none>         9443/TCP,1414/TCP 10h
service/qms-ibm-mq-metrics ClusterIP    172.21.19.122 <none>         9157/TCP          10h

NAME                READY   AGE
statefulset.apps/qms-ibm-mq 1/1    10h

NAME                HOST/PORT                                                                 PATH   SERVICES   PORT   TERMINATION  WILDCARD
route.route.openshift.io/qms-ibm-mq-qm qms-ibm-mq-qm-shoheim.isemqgcloup-3e2d74f-0000.jp-tok.containers.appdomain.cloud qms-ibm-mq 1414 passthrough None
route.route.openshift.io/qms-ibm-mq-web qms-ibm-mq-web-shoheim.isemqgcloup-3e2d74f-0000.jp-tok.containers.appdomain.cloud qms-ibm-mq 9443 passthrough None

NAME                PHASE
queuemanager.mq.ibm.com/qms Running

% oc get pvc
NAME                STATUS    VOLUME                                     CAPACITY   ACCESS MODES   STORAGECLASS          AGE
data-qms-ibm-mq-0   Bound    pvc-91ffe865-e9af-44da-a11e-ea45d4eb9eba  20Gi       RWO             ibmc-block-gold-wait-for-1st-consumer 34h
qms-ibm-mq-persisted-data Bound    pvc-29a0e7a5-68ee-4890-aec1-419a1ddfa47b  20Gi       RWO             ibmc-block-gold-wait-for-1st-consumer 34h
qms-ibm-mq-recovery-logs Bound    pvc-884fd5c5-c730-480c-9575-9cd4e312f14b  20Gi       RWO             ibmc-block-gold-wait-for-1st-consumer 34h

% oc get pv
NAME                CAPACITY   ACCESS MODES   RECLAIM POLICY   STATUS   CLAIM                                                                 STORAGECLASS          REASON   AGE
pvc-91ffe865-e9af-44da-a11e-ea45d4eb9eba  20Gi       RWO             Delete           Bound   shoheim/data-qms-ibm-mq-0                                             ibmc-block-gold-wait-for-1st-consumer          35h
pvc-29a0e7a5-68ee-4890-aec1-419a1ddfa47b  20Gi       RWO             Delete           Bound   shoheim/qms-ibm-mq-persisted-data                                       ibmc-block-gold-wait-for-1st-consumer          35h
pvc-884fd5c5-c730-480c-9575-9cd4e312f14b  20Gi       RWO             Delete           Bound   shoheim/qms-ibm-mq-recovery-logs                                         ibmc-block-gold-wait-for-1st-consumer          35h
```

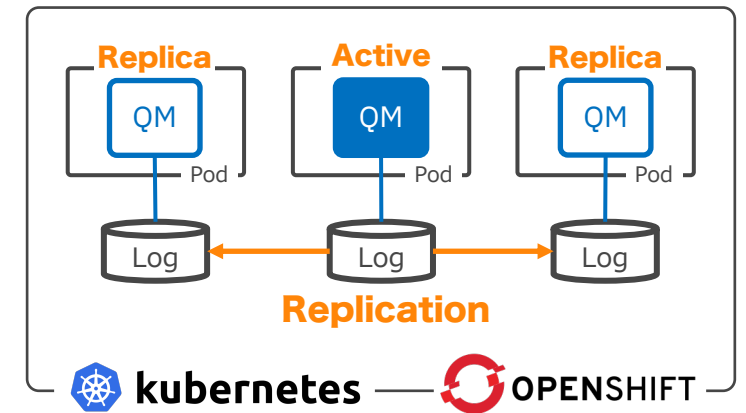
- 具体的な構成手順はSIL「IBM MQ on OpenShift構成ガイド」参照
  - ◆ <https://www.ibm.com/support/pages/node/6523336>



**Native HA**

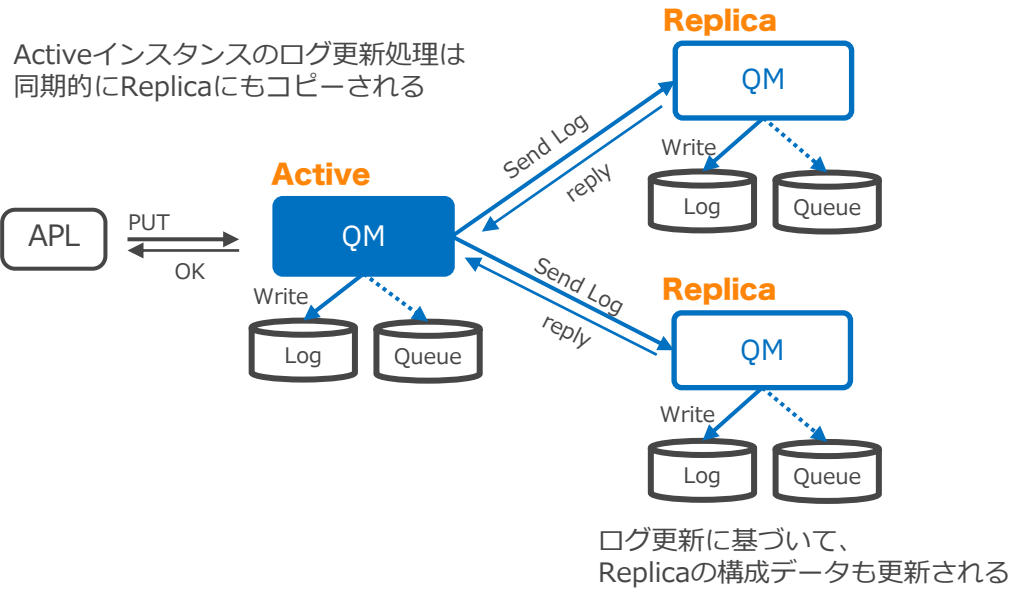


- MQ独自のログ・レプリケーション機能による高可用性構成
  - ◆ Kubernetes/OpenShift環境で構成可能
  - ◆ キューマネージャは3つのPodで稼働
    - 1つがActiveインスタンスとして稼働、残りはReplicaインスタンスとして待機
  - ◆ 各PodがPersistent Volume上にログのコピーを保有
    - MQプロセスにてActiveからReplicaへログがコピーされる
  - ◆ 障害時にFailoverしたキューマネージャへの接続切り替えはKubernetes/OpenShiftのServiceで解決
    - 接続元のチャンネルやクライアントは1つの宛先に対して再接続するだけでよい
  - ◆ 追加のクラスタ製品やストレージの冗長構成が不要
- Cloud Pak for Integrationライセンスに含まれるMQ Advancedでサポート
  - ◆ MQ V9.2.3、IBM Cloud Pak for Integration 2021.2.1からサポート
- MQ V9.3からMQ Operatorを利用せずに構成することも可能
  - ◆ OpenShift以外のKubernetes環境に構成することが可能に
    - サンプルのHelmチャートを公開
    - mq-helm repository : <https://github.com/ibm-messaging/mq-helm>
  - ◆ MQ Operatorを利用しない場合は、ユーザー実装のコンテナ・イメージを利用



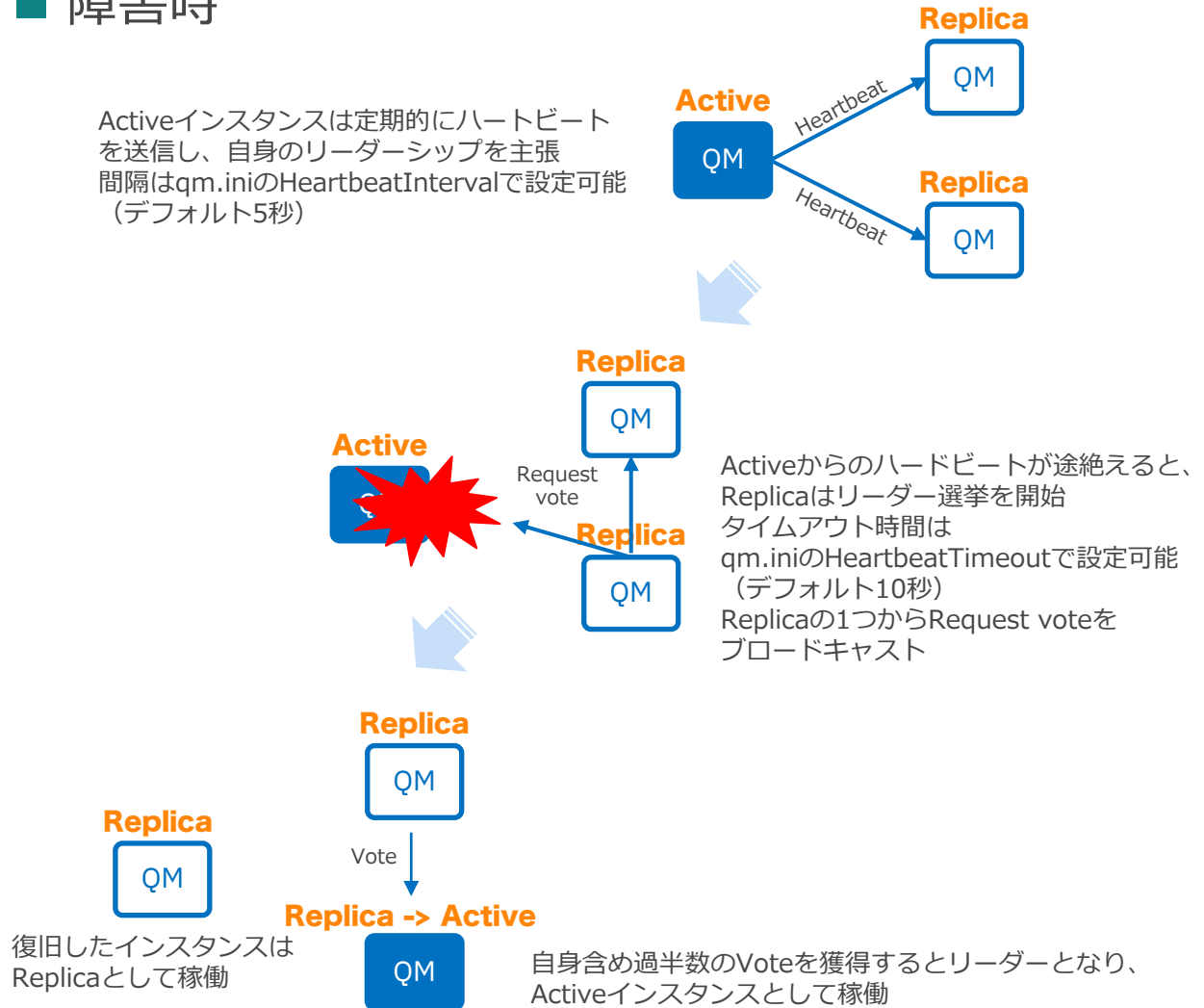
# Native HAの動き

## ■ ログ・レプリケーション



## ■ 障害時

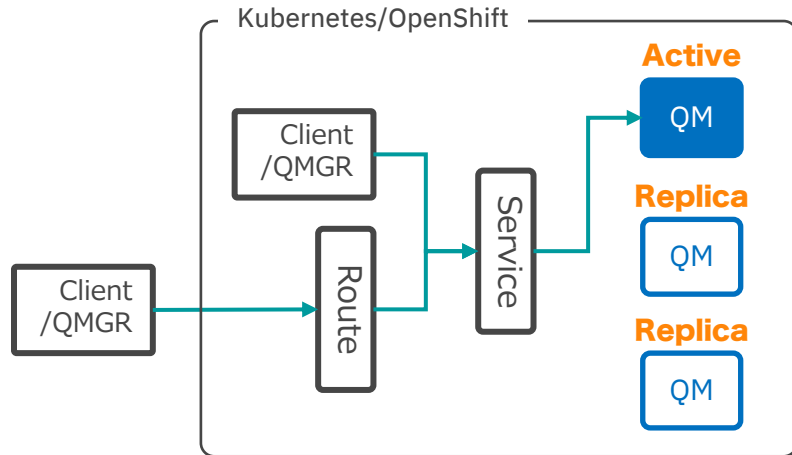
Activeインスタンスは定期的にハートビートを送信し、自身のリーダーシップを主張  
間隔はqm.iniのHeartbeatIntervalで設定可能  
(デフォルト5秒)



# Native HAキューマネージャーへの接続

## ■ 通常時

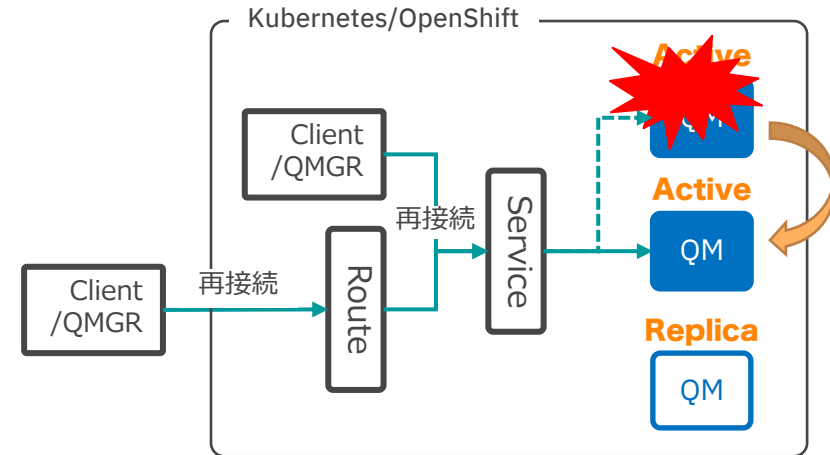
Serviceは3つのPodの内、Active(Readiness Probeが成功)なPodにのみトラフィックを転送  
Readiness Probeは、MQリスナーのポート(1414)がListenされているかをチェック  
一時点では1つのキューマネージャーしかアクティブに稼働しないため、  
Replicaインスタンスにトラフィックが流れることはない



Native HAキューマネージャーへ接続するクライアントアプリや送信チャンネルは  
Serviceの宛先を指定するだけでよい  
Kubernetes/OpenShift Cluster外から接続する場合はRouteの宛先を指定  
Route->Service->Activeインスタンスへトラフィックが流れる

## ■ 障害時のFailover

Activeインスタンスの障害によって別インスタンスがActiveとして稼働し、  
Readyとなると（ポート1414をListen開始）、ServiceはReadyとなった  
Activeインスタンスにトラフィックを転送



クライアントアプリや送信チャンネルはActiveインスタンスの障害によって  
接続は一度切断されるが、同じ宛先(Service or Route)に再接続を試みることで  
F/OしたActiveインスタンスに接続できる

## ■ dspmqコマンドでの構成確認

```
# -o nativeha: ROLEやINSTANCE、INSYNC、QUORUMを出力  
# -s: STATUSを出力  
# -x: NativeHAの他のノード上のインスタンスの情報を出力
```

```
% oc rsh qm1-ibm-mq-0 dspmq -o nativeha -s -x
```

```
QMNAME(NHQM01TLS) STATUS(Running) ROLE(Active) INSTANCE(qm1-ibm-mq-0) INSYNC(yes) QUORUM(3/3)  
INSTANCE(qm1-ibm-mq-0) ROLE(Active) REPLADDR(qm1-ibm-mq-replica-0) CONNACTV(yes) INSYNC(yes) BACKLOG(0) CONNINST(yes) ALTDATE(2022-11-11) ALTTIME(23.24.16)  
INSTANCE(qm1-ibm-mq-1) ROLE(Replica) REPLADDR(qm1-ibm-mq-replica-1) CONNACTV(yes) INSYNC(yes) BACKLOG(0) CONNINST(yes) ALTDATE(2022-11-11) ALTTIME(23.24.16)  
INSTANCE(qm1-ibm-mq-2) ROLE(Replica) REPLADDR(qm1-ibm-mq-replica-2) CONNACTV(yes) INSYNC(yes) BACKLOG(0) CONNINST(yes) ALTDATE(2022-11-11) ALTTIME(23.24.16)
```

フィールド名	説明
QMNAME	キューマネージャー名
STATUS	キューマネージャーのステータス
ROLE	インスタンスの現在の役割を示す Active/Replica/Unknownのいずれか
INSTANCE	インスタンス名(crtmqm -lrオプションで付与された名前)
INSYNC	このインスタンスがActiveインスタンスとしてtakeoverできるか(Yes/No)
QUORUM	クォーラムの状態を示す insyncのインスタンス数/インスタンス数で示される
REPLADDR	レプリケーションで使用するIPアドレス(ホスト名)
CONNACTV	ノードがActiveインスタンスに接続されているか(Yes/No)
BACKLOG	このインスタンスのレプリケーションがどれだけ遅れているかを示す(KB単位)
CONNINST	そのインスタンスがdspmqを実行したインスタンスに接続しているか(Yes/No) ActiveでもReplicaでもないインスタンスではNoが出力される

- Native HAキューマネージャーのステータスの詳細は下記参照

- <https://www.ibm.com/docs/en/ibm-mq/9.3?topic=nh-viewing-status-native-ha-queue-managers-mq-certified-containers>

## ■ qm.ini

MQ Operatorを利用して作成したNative HAキューマネージャーのqm.ini  
(Native HA関連のみ抜粋)

```
Log:
  LogPrimaryFiles=10
  LogSecondaryFiles=5
  LogFilePages=4096
  LogType=REPLICATED
  LogBufferPages=0
  LogPath=/mnt/mqm/data/log/NHQM01TLS/
  LogWriteIntegrity=TripleWrite
  LogManagement=Automatic

NativeHALocalInstance:
  Name=qm1-ibm-mq-0
  CertificateLabel=ha
  KeyRepository=/run/runmqserver/ha/tls/key

NativeHAInstance:
  Name=qm1-ibm-mq-0
  ReplicationAddress=qm1-ibm-mq-replica-0(9414)
NativeHAInstance:
  Name=qm1-ibm-mq-1
  ReplicationAddress=qm1-ibm-mq-replica-1(9414)
NativeHAInstance:
  Name=qm1-ibm-mq-2
  ReplicationAddress=qm1-ibm-mq-replica-2(9414)
```

LogType=REPLICATED  
Native HA専用のログ・タイプ  
crtmqmコマンドで -lr を指定してキューマネージャーを作成すると、この設定となる  
ログ・ファイルの管理としては、自動管理を有効にしたリニア・ログと同等  
(LogManagement=Automaticも設定される)  
メディア・イメージの自動作成も有効  
以下のキューマネージャー属性が設定される  
IMGINTVL(60)  
IMGLOGLN(OFF)      IMGRCOVO(YES)  
IMGRCOVQ(YES)      IMGSCHEM(AUTO)

Native HA構成用のスタンザ  
Nameにはインスタンス名がセットされる (各インスタンスでユニークな名前となる)  
チューニング項目として以下の属性を設定可能  
(設定する場合は事前にConfigMap /Secretを用意して適用)  
HeartbeatInterval  
ActiveインスタンスからReplicaインスタンスに送信するハートビートの間隔 (ミリ秒)  
デフォルトは5000 (5秒)  
HeartbeatTimeout  
ReplicaインスタンスがActiveインスタンスが無反応と判断する時間 (ミリ秒)  
HeartbeatIntervalと同じか大きい値を指定  
デフォルトは、HeartbeatInterval×2 (10秒)  
RetryInterval  
切断されたレプリケーション接続を再接続する間隔 (ミリ秒)  
デフォルトは、HeartbeatInterval×2 (10秒)

Native HAキューマネージャーのインスタンス間の接続情報を設定するスタンザ  
自分自身と他のインスタンスへの接続情報 (ホスト名とポート番号) が設定される

## ■ ocコマンド

```
% oc get all
```

NAME	READY	STATUS	RESTARTS	AGE
pod/qm1-ibm-mq-0	1/1	Running	0	13h
pod/qm1-ibm-mq-1	0/1	Running	0	13h
pod/qm1-ibm-mq-2	0/1	Running	0	13h

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/qm1-ibm-mq	ClusterIP	172.21.70.129	<none>	9443/TCP,1414/TCP	13h
service/qm1-ibm-mq-metrics	ClusterIP	172.21.51.38	<none>	9157/TCP	13h
service/qm1-ibm-mq-replica-0	ClusterIP	172.21.194.40	<none>	9414/TCP	13h
service/qm1-ibm-mq-replica-1	ClusterIP	172.21.233.74	<none>	9414/TCP	13h
service/qm1-ibm-mq-replica-2	ClusterIP	172.21.53.220	<none>	9414/TCP	13h

NAME	READY	AGE
statefulset.apps/qm1-ibm-mq	1/3	13h

NAME	HOST/PORT	PATH	SERVICES	PORT	TERMINATION	WILDCARD
route.route.openshift.io/qm1-ibm-mq-qm	qm1-ibm-mq-qm-shoheim.isemqgcloup-83e2d74f-0000.jp-tok.containers.appdomain.cloud		qm1-ibm-mq	1414	passthrough	None
route.route.openshift.io/qm1-ibm-mq-web	qm1-ibm-mq-web-shoheim.isemqgcloup-83e2d74f-0000.jp-tok.containers.appdomain.cloud		qm1-ibm-mq	9443	passthrough	None

NAME	PHASE
queuemanager.mq.ibm.com/qm1	Running

- 具体的な構成手順はSIL「IBM MQ Native HA構成ガイド」参照
  - ◆ <https://www.ibm.com/support/pages/node/6840043>



**KEDA**





## ■ KEDA : Kubernetes-based Event Driven Autoscaler

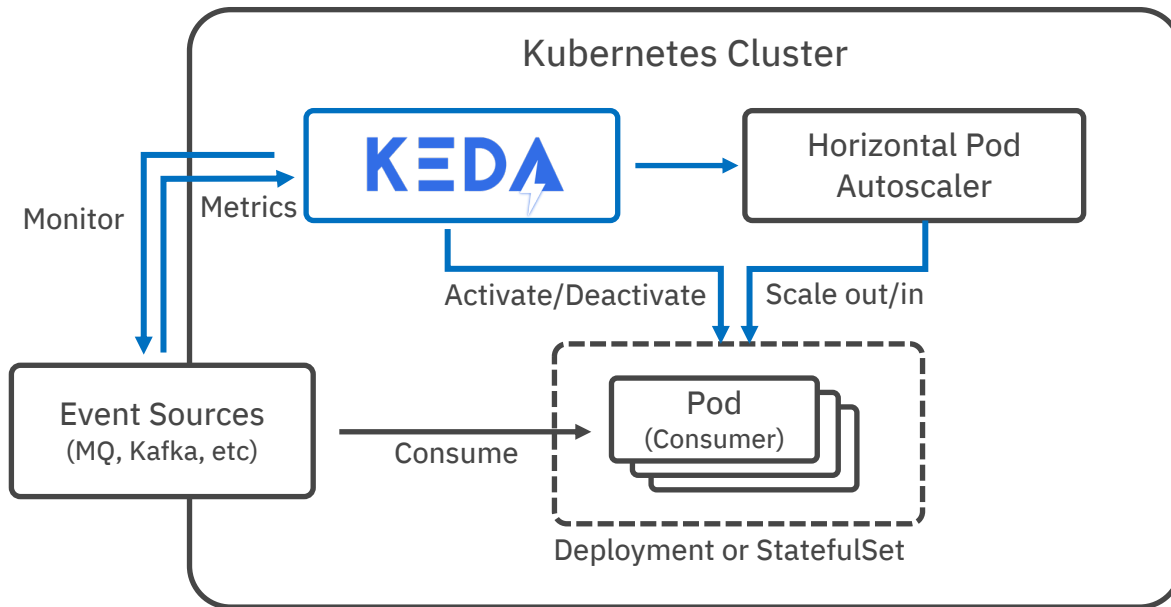
◆ <https://keda.sh>

◆ CNCF(Cloud Native Computing Foundation)のIncubationプロジェクト

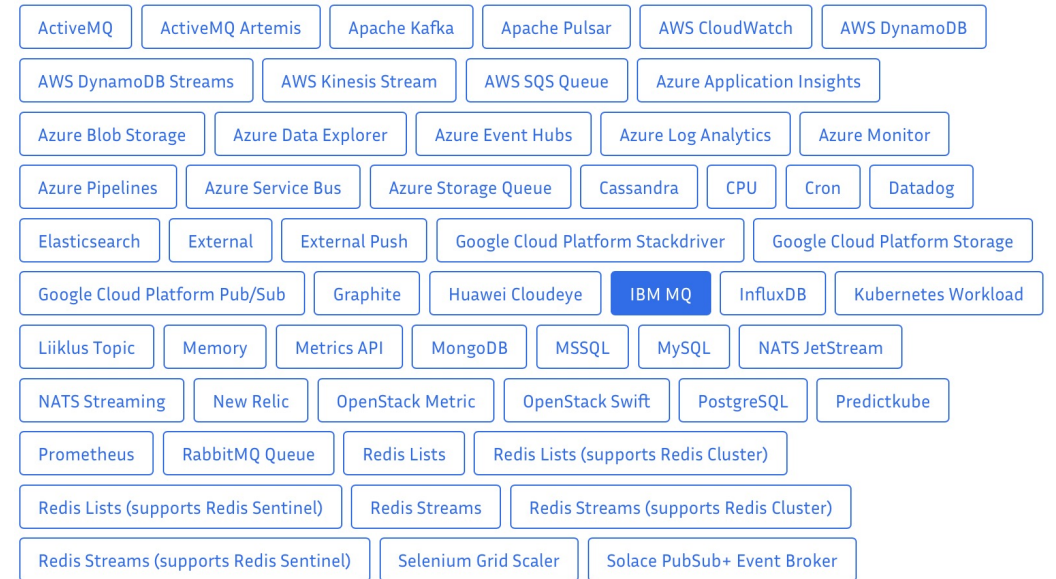
## ■ Kubernetes/OpenShift環境でイベント・ベースのオート・スケーリングを実現する仕組み

◆ HPA(Horizontal Pod Autoscaler)と連携し、アプリケーションPodをスケールアウト/スケールイン

## ■ アーキテクチャー



## ■ 様々なEvent Sourceに対応したScalerを提供



出典: Event sources and scalers

(<https://keda.sh/docs/2.8/concepts/#event-sources-and-scalers>)

## ■ 構成の流れ

### 1. KEDAのデプロイ

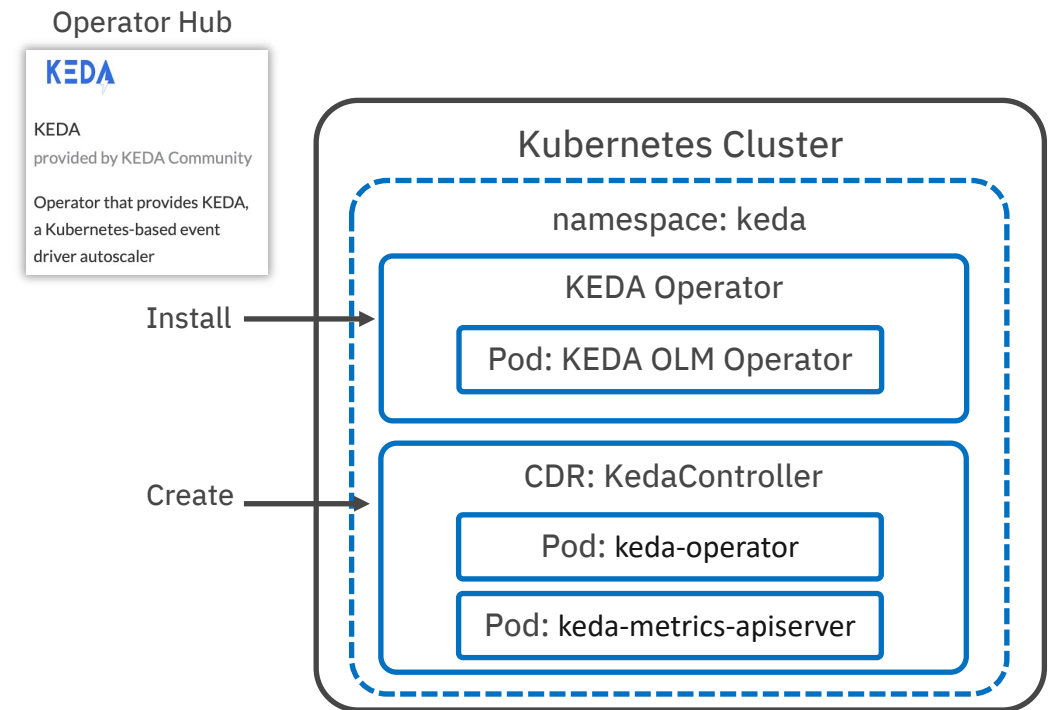
- Helm chart、Operator Hub、YAML等を利用して、KEDAのランタイムをKubernetes Clusterにデプロイ
- デプロイ方法は下記リンク参照
  - <https://keda.sh/docs/2.8/deploy/>

- Operator Hubの場合

1. Operator Hub MarketplaceからKEDA Operatorをインストール
2. KedaController(KEDAのCustom Resource)を作成

※ いずれも名前スペース”keda”にインストール/作成する

- KedaControllerを作成することでKEDAのランタイムがデプロイされる
- 以下の2つのDeploymentが作成され、Podが稼働
- keda-operator
  - HPAの作成、Deployments/StatefulSetのActivate/Deactivateを担う
- keda-metrics-apiserver
  - Event SourceからのイベントをHPAに連携し、スケールアウト/スケールインを促す制御を担う



## 2. Deployment/StatefulSetのデプロイ

- スケールの対象とする任意のアプリケーションをDeploymentやStatefulSetとしてデプロイ
  - Custom Resource(/scaleサブルーチンを持つ場合に限る)やJobもスケール対象にすることが可能
  - 任意のネームスペースにデプロイ

## 3. ScaledObjectのデプロイ

- KEDAのCustom Resource
  - Deployment/StatefulSet/Custom Resourceを対象とする場合に利用
  - Jobを対象する場合は、ScaledJobを利用
  - アプリケーションと同じネームスペースにデプロイ
- イベント・ソースとスケール対象とのマッピングとスケール条件を指定
  - .spec.ScaleTargetRef にスケール対象のDeployment等の名前を指定
  - ScaledObject共通のプロパティ
    - pollingInterval
    - cooldownPeriod
    - maxReplicaCount
    - etc
  - ScaledObjectの詳細は下記リンク参照
    - <https://keda.sh/docs/2.8/concepts/scaling-deployments/#scaledobject-spec>
- (次ページに続く)

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: ibmmq-scaledobject
  namespace: mqapp
  labels:
    deploymentName: ibmmq-deployment
spec:
  scaleTargetRef:
    name: ibmmq-deployment
  pollingInterval: 5 # Default: 30 seconds
  cooldownPeriod: 30 # Default: 300 seconds
  maxReplicaCount: 18 # Default: 100
  triggers:
    - type: ibmmq
      metadata:
        host: <ibm-host> # Admin REST API Endpoint
        queueManager: <queue-manager>
        queueName: <queue-name>
        tlsDisabled: <TLS enabled/disabled>
        queueDepth: <queue-depth>
      authenticationRef:
        name: keda-ibmmq-trigger-auth
```

## 3. ScaledObjectのデプロイ（続き）

- .spec.triggers でScalerの名前、Scaler固有のプロパティを指定
- 個々のScalerの詳細は下記リンク参照
  - <https://keda.sh/docs/2.8/scalers/>
- IBM MQ Scalerの場合
  - <https://keda.sh/docs/2.8/scalers/ibm-mq/>

プロパティ	設定値
type	ibmmq (固定)
host	キューマネージャーの管理REST APIのURL http(s)://<host:port>/ibmmq/rest/v2/admin/action/qmgr/<QMgr name>/mqsc
queueManager	キューマネージャー名
queueName	スケール対象のアプリケーションがGETするキュー名
tlsDisabled	TLS通信するかどうか(true/false)
queueDepth	スケール・アウト/インの閾値とするメッセージ滞留数 (デフォルト5)
activationQueueDepth	Activate/Deactivateの閾値とするメッセージ滞留数 (デフォルト1)

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
metadata:
  name: ibmmq-scaledobject
  namespace: mqapp
  labels:
    deploymentName: ibmmq-deployment
spec:
  scaleTargetRef:
    name: ibmmq-deployment
  pollingInterval: 5 # Default: 30 seconds
  cooldownPeriod: 30 # Default: 300 seconds
  maxReplicaCount: 18 # Default: 100
  triggers:
    - type: ibmmq
      metadata:
        host: <ibm-host> # Admin REST API Endpoint
        queueManager: <queue-manager>
        queueName: <queue-name>
        tlsDisabled: <TLS enabled/disabled>
        queueDepth: <queue-depth>
  authenticationRef:
    name: keda-ibmmq-trigger-auth
```

## 4. TriggerAuthenticationのデプロイ

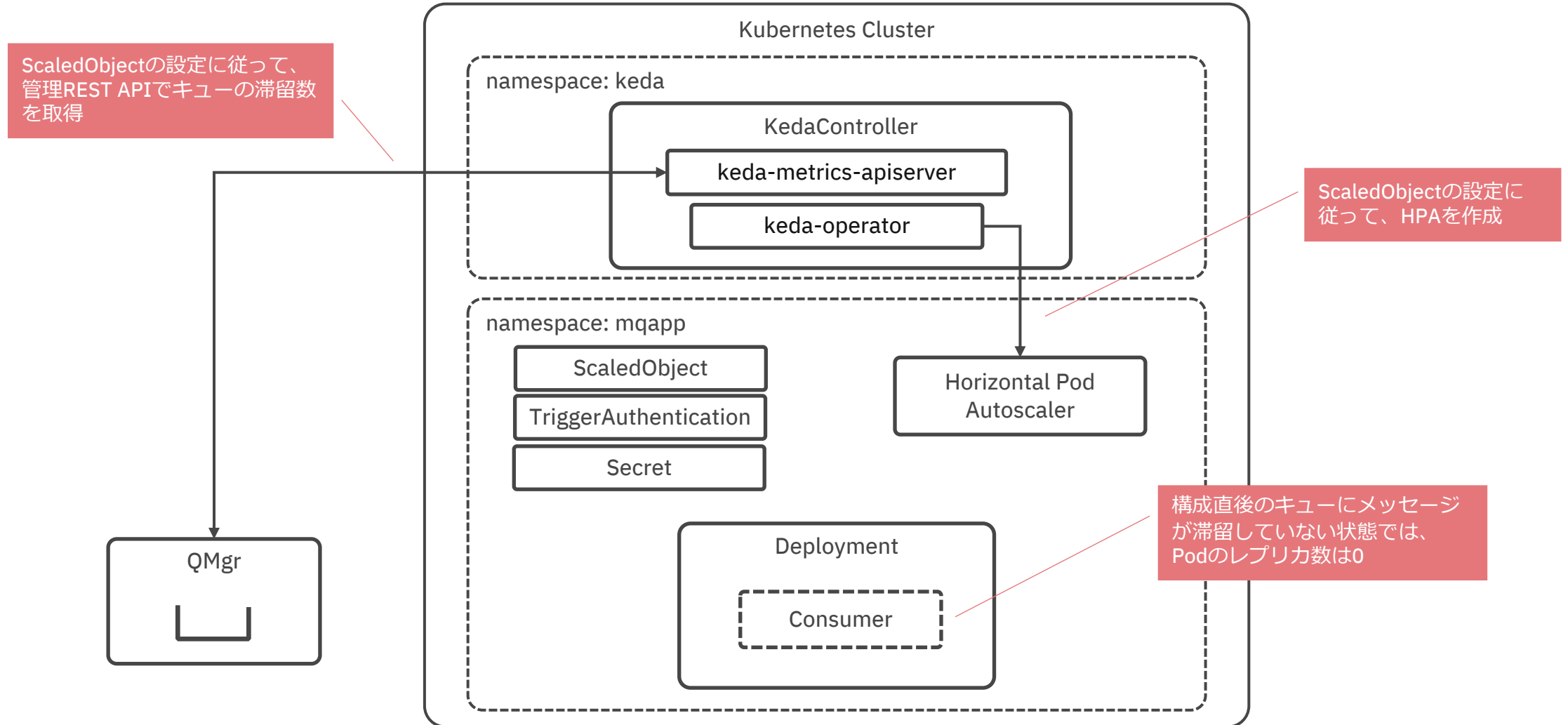
- KEDAのCustom Resource
  - ScaledObjectと同じネームスペースにデプロイ
- イベント・ソースに対する認証情報を設定
  - TriggerAuthenticationを利用する以外の方法もあり
  - 詳細は下記リンク参照
    - <https://keda.sh/docs/2.8/concepts/authentication/>
- TriggerAuthenticationを利用してMQへの認証情報を設定する場合
  - <https://keda.sh/docs/2.8/concepts/authentication/#re-use-credentials-and-delegate-auth-with-triggerauthentication>
  - <https://keda.sh/docs/2.8/scalers/ibm-mq/#authentication-parameters>
  - 1. Secretに管理REST APIでキューマネージャーにアクセスする際のユーザー、パスワードを指定
    - ADMIN\_USER: ユーザー
    - ADMIN\_PASSWORD: パスワード
  - 2. TriggerAuthenticationからSecretを参照し、usernameにADMIN\_USER、passwordにADMIN\_PASSWORDをマッピング
  - 3. ScaledObjectのauthenticationRefでTriggerAuthenticationの名前を指定

```
apiVersion: keda.sh/v1alpha1
kind: ScaledObject
(省略)
triggers:
(省略)
authenticationRef:
  name: keda-ibmmq-trigger-auth
```

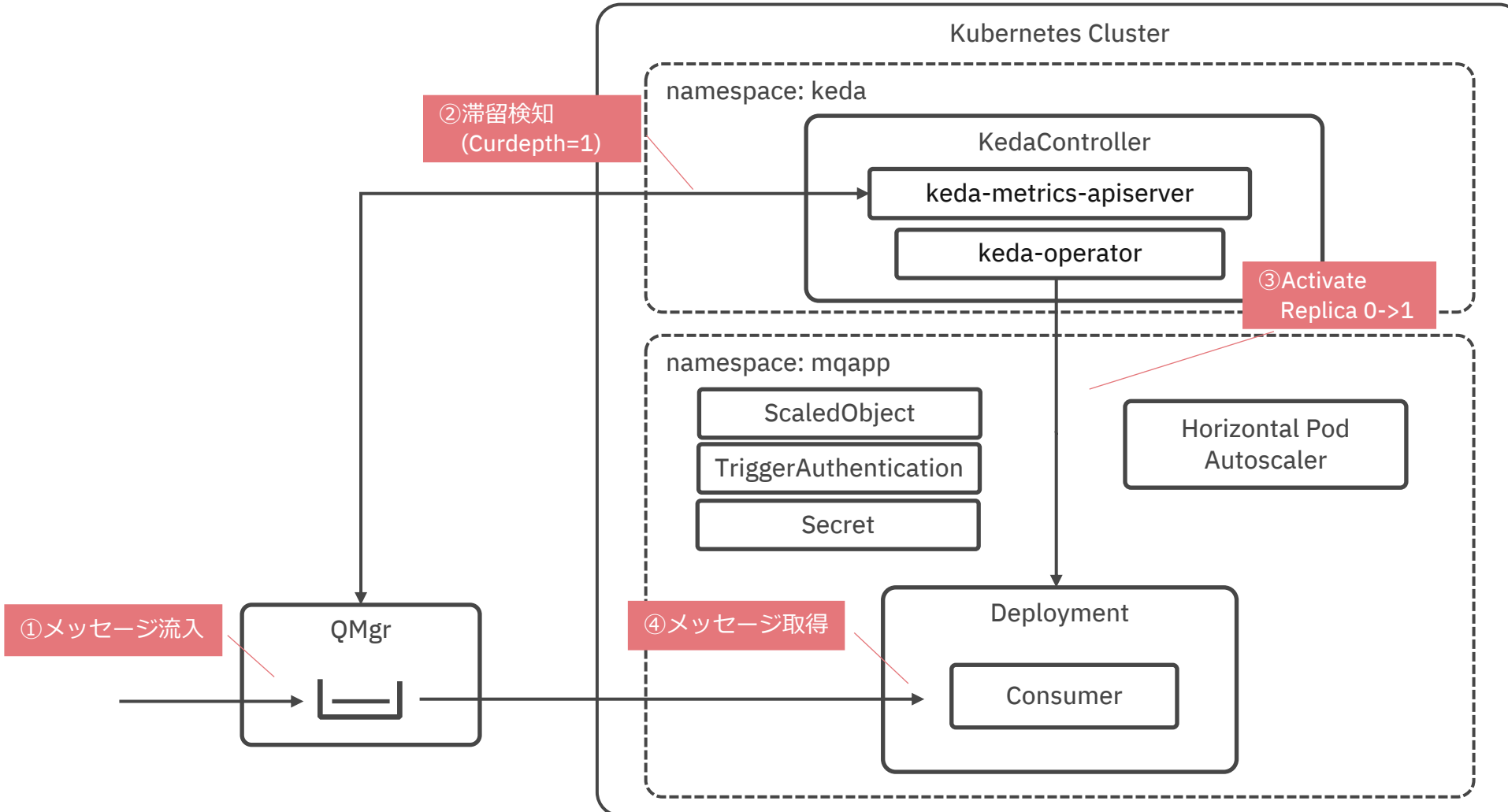
```
apiVersion: keda.sh/v1alpha1
kind: TriggerAuthentication
metadata:
  name: keda-ibmmq-trigger-auth
  namespace: mqapp
spec:
  secretTargetRef:
    - parameter: username
      name: keda-ibmmq-secret
      key: ADMIN_USER
    - parameter: password
      name: keda-ibmmq-secret
      key: ADMIN_PASSWORD
```

```
apiVersion: v1
kind: Secret
metadata:
  name: keda-ibmmq-secret
type: Opaque
data:
  ADMIN_USER: <user>
  ADMIN_PASSWORD: <password>
```

## ■ 構成直後



## ■ Activate



## ■ スケールアウト

